SIMATIC S5

S5-115U Programmable Controller CPU 945-7UA1./-7UA2.

Manual

This manual has the order number:

6ES5998-3UF21

EWA 4NEB 811 6150-02e

Edition 06

Summary

Page

Page

Page

Page

Page

Page

Page

Page

Page

Appendices

Page

Preface

The S5-115U is a programmable controller for the lower and mid performance ranges. It meets all the demands made of a modern programmable controller.
The performance capability of the S5-115U has been subject to constant enhancement.
You now have at your disposal the most powerful CPU for the S5-115U in the shape of the CPU 945.

To make optimum use of the CPU 945 in the S5-115U, you require a certain amount of detailed information. This manual presents all this information in an organized manner.

A proforma for corrections and improvement suggestions is included at the end of the manual. You can use these to help us improve the next edition.

Important differences between the CPU 945 and the CPUs 941 to 944

- New CPU architecture
- Extremely fast operation execution times
- Expanded STEP 5 operation set
- Floating-point arithmetic
- 32-bit accumulators and expanded register set
- Larger address area
- Additional flag area
- Additional system data area
- Improved execution level system
- Additional organization blocks
- Additional integral FBs for analog value processing
- Integral OB 250 for invoking and parameterizing operating system services
- Additional blocks (FX, DX)
- New memory submodules
- Various interface modules can be used as a 2nd interface
- Expanded DB 1
  - -Parameterization of ASCII driver
  - -Parameterization of computer interface
- You require a PG 7xx programmer with S5-DOS from Stage V6.1 for programming the CPU 945.

Compatibility of the CPU 945 with the CPUs 941 to 944

Programs written for the CPUs 941 to 944 must be adapted in the following cases:

- If waiting times have been implemented over operation run time loops
  or
- If the programs contain absolute address accesses
  Please also note that
- Assembler blocks cannot execute on the CPU 945
- For programs containing data handling block calls (SEND, RECEIVE) in quick succession, additional wait times must be programmed for slow communications partners, such as the CP 524 and CP 525.

For these reasons, new standard function blocks are required when using most intelligent I/O modules and some CPs.

You require operating system services for the CPU 945 for assigning some operating system parameters. These are the operating system services for:

- Interrupt response after timeout (OB6)
- Time-controlled program processing (OB10 to 13)
- Manipulation of the process I/O image transfer
- Regeneration of the block address list
- Generating a data block (DB/DX) without TRAF
- I/O accesses and page accesses without QVZ
- Disabling and enabling the output modules (setting and resetting BASP)

FBs 240 to 243, which were previously integrated in the CPUs 941 to 944 are no longer required in the CPU 945 due to the larger operation set. They have been replaced by the following additional FBs for analog value processing:

- FB241 Reading in analog value from the 463 analog input module
- FB242 Reading in analog value from the 464 analog input module
- FB243 Reading in analog value from the 466 analog input module

Modification to parameterization of FB250/251:

- Analog values represented in floating-point format
- Account is taken of the P, O, IM3 and IM4 peripheral areas

Changes in the new manual compared to the manual for the CPUs 941 to 944

The 460-7LA13 and 463-4UA13 analog input modules have been included in the chapter on "Analog Value Processing".

From CPU 945 firmware version Z 02

This manual describes the CPU 945 from firmware version Z 02.
From firmware version Z 02, the CPU 945 has the following new functions:

- Organization Blocks of the CPU 945 for Special Functions
- Copying Data Area: OB 182
- Duplicating DX or DB Blocks: OB 183 and OB 184
- Transferring Flags to Data Blocks: OB 190 and OB 192
- Transferring Data Blocks to Flag Area: OB 191 and OB 193
- Extension for Sign: OB 220
- PID Control Algorithm: OB 251
- New Operating System Services of the OB 250
  - Access to DBS and DBL Registers
  - Indexed Access to DX / FX
  - Cancelling a Block in the Block Address List
  - Changing the Block Identifier
- Generating STEP 5 Blocks: OB 125

Not all problems that might occur in the many and varied applications can be handled in detail in a manual. If you have a problem that is not discussed in the manual, contact your nearest SIEMENS office or representative.

Introduction

The following pages contain information to help you familiarize yourself with the manual.

Description of contents

The contents of the manual can be broken down subject-wise into a number of blocks:

- Description
  (System overview, technical description)
- Installation and operation
  (Installation guidelines, program test and startup, fault diagnostics, addressing)
- Programming instructions
  (Introduction to STEP 5, STEP 5 operations)
- Special capabilities
  (Interrupt processing, analog value processing, parameterizing the CPU with DB1, communications)
- Technical specifications overview

You will find additional information in tabular form in the appendices.
A pocket guide for the STEP 5 operation set of the CPU 945 ist an integral part of the manual.

Please use the forms at the back of the manual for any suggestions or corrections you may have and return the forms to us. This will help us to make the necessary improvements in the next edition.

Training courses

Siemens offer comprehensive training facilities for users of SIMATIC S5.

Details can be obtained from your nearest Siemens office or representative.

Reference literature

The manual contains a comprehensive description of the S5-115U. Subjects that are not specially related to the S5-115U have only been treated in brief, however. More detailed information is available in the following literature:

- **Programmable controls**
  Volume 1: Logic and sequence controls; from the control problem to the control program

  Günter Wellenreuther, Dieter Zastrow
  Brunswick 1987

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

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

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

  Hans Berger
  Siemens AG, Berlin and Munich 1989

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

  Order No.: ISBN 3-89578-022-7

Information on the programmable controller hardware is to be found in the following catalogues:

●	ST 52.3	"S5-115U Programmable Controller"
●	ST 57	"Standard Function Blocks and Driver Software for Programmable Controllers of the U Range"
●	ST 59	"Programmers"
●	ET 1.1	"ES 902 C Modular 19 in. Packaging System"
●	MP 11	"Thermocouples; compensating boxes"

The relevant manuals are available for other components and modules (e.g. CPs and SINEC L1). Reference is made to these sources of information at various points in the manual.

The S5-115U programmable controller is designed to VDE 0160. The corresponding IEC and VDE (Association of German Electrical Engineers) standards are referred to in the text.

Conventions

Structure of the manual

In order to improve readability of the manual, a menu-styled breakdown was used, i.e.:

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

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

- A number of abbreviations have been used.
  Example: Programmer (PG)
- Footnotes are identified by superscripts consisting of a small digit (e.g. "1") or "*". The actual footnote is generally at the bottom left of the page or below the relevant table or figure.
- Lists are indicated by a black dot (●), as in this list for example, or with a dash (-).
  Instructions for operator actions are indicated by black triangles (▸).
- Cross references are shown as follows:
  "(→ Section 7.3.2)" refers to Section 7.3.2.
  No references are made to individual pages.
- All dimensions in drawings etc. are given in millimetres followed by inches in brackets. Example: 187 (7.29).
- Values may be represented as binary, decimal or hexadecimal numbers. The hexadecimal number system is indicated with a subscript (example F000$_{H}$)
- Information of special importance is enclosed in black-edged boxes:

⚠ **Warning**

See the "Safety-Related Guidelines" for definitions of the terms "Warning", "Danger", "Caution" and "Note".

Manuals can only describe the current version of the programmer. Should modifications or supplements become necessary in the course of time, a supplement will be prepared and included in the manual the next time it is revised.

Safety-Related Guidelines for the User

This document provides the information required for the intended use of the particular product. The documentation is written for technically qualified personnel.
Qualified personnel as referred to in the safety guidelines in this document as well as on the product itself are defined as follows.

- System planning and design engineers who are familiar with the safety concepts of automation equipment.
- Operating personnel who have been trained to work with automation equipment and are conversant with the contents of the document in as far as it is connected with the actual operation of the plant.
- Commissioning and service personnel who are trained to repair such automation equipment and who are authorized to energize, de-energize, clear, ground, and tag circuits, equipment, and systems in accordance with established safety practice.

Danger Notices

The notices and guidelines that follow are intended to ensure personal safety, as well as protect the products and connected equipment against damage.

The safety notices and warnings for protection against loss of life (the users or service personnel) or for protection against damage to property are highlighted in this document by the terms and pictograms defined here. The terms used in this document and marked on the equipment itself have the following significance.

Danger

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

Warning

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

Caution

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

Note

contains important information about the product, its operation or a part of the document to which special attention is drawn.

Proper Usage

Warning

- The equipment/system or the system components may only be used for the applications described in the catalog or the technical description, and only in combination with the equipment, components, and devices of other manufacturers as far as this is recommended or permitted by Siemens.
- The product will function correctly and safely only if it is transported, stored, set up, and installed as intended, and operated and maintained with care.

1 System Overview

Figures

1 System Overview

The SIMATIC® S5-115U programmable controller is used worldwide in almost all fields in a wide range of applications. Each of its modular components handles a specific task. Therefore, you can expand the system according to your needs.
The most powerful S5-115U configuration is implemented with the CPU 945. The S5-115U system provides operator panels, monitoring devices, and various programmers to suit your needs. The STEP 5 programming language and an extensive software catalog make programming easy.

1.1 Application

Many different industries use the S5-115U. Even though each automation task is different, the S5-115U adapts optimally to the most varied jobs, whether they involve simple open-loop control or complex closed-loop control.

Present areas of application include the following:

- Automobile Industry
  Automatic drill, assembly and test equipment, painting facilities, shock absorber test bays

- Plastics Industry
  Blow, injection, and thermal molding machines, synthetics production systems

- Heavy Industry
  Molding equipment, industrial furnaces, rolling mills, automatic pit shaft temperature control systems

- Chemical Industry
  Proportioning and mixing systems

- Food and Beverages Industry
  Brewery systems, centrifuges

- Machinery
  Packing, woodworking, and custom-made machines, machine controls, machine tools, drilling mills, fault alarm centers, welding technology

- Building Services
  Elevator technology, climate control, ventilation, lighting

- Transport Systems
  Transport and sorting equipment, high-bay warehouses, conveyor and crane systems

- Energy, Gas, Water, Air
  Pressure booster stations, standby power supply, pump control, water and air treatment, filtering and gas recovery systems

1.2 System Components

The S5-115U system is made up of various modular components, as pictured in Figure 1-1. These components include the following:

- Power supply module (PS)
- Central processing unit (CPU)
- Input and output modules (I/Os)
- Intelligent input/output modules (IPs, WFs)
- Communications processors (CPs)

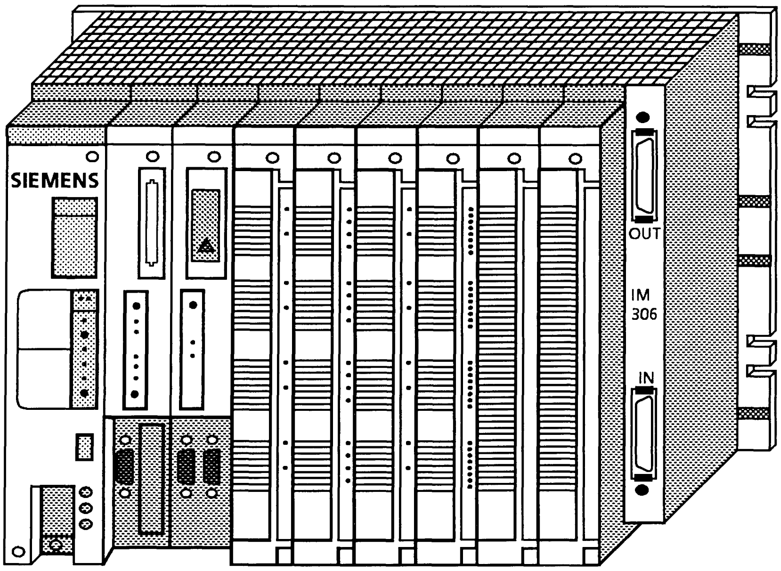


Figure 1-1 S5-115U Components

1.2.1 Power Supply

The power supply module (PS) converts the external power supply to the internal operating voltage. Supply voltages for the S5-115U include 24 V DC, 115 V AC, and 230 V AC.

Screw-type terminals connect the power supply lines to the bottom of the PS. Three maximum output currents are available. Choose 3 A, 7 A, or 15 A according to the number of modules you have or according to their power consumption. A fan is not necessary for output currents up to 7 A.

A lithium battery backs up the program memory and the internal retentive flags, timers and counters in the event of a power failure. An LED signals battery failure. If you change the battery when the power is shut off, connect a back-up voltage from an outside source to the sockets provided for this purpose on the power supply module.

1.2.2 Central Processing Units

The central processing unit (CPU) is the "brain" of the programmable controller. It executes the control program.
As well as the CPU 945, the most powerful of the CPUs, you can also use the following four CPUs in the S5-115U:
CPU 941, CPU 942, CPU 943, CPU 944.
These CPUs offer a varied performance range and are described in a separate manual (Order No.: 6ES5 998-OUF.3).

The CPU 945 allows you to implement the fastest execution times of your control programs. The CPU 945 has the largest STEP 5 operation set and the largest program memory.
A new feature of the CPU 945 is its ability to implement FUZZY control (see Catalog ST 57). The new FUZZY control block replaces the OB251 (PID control algorithm) integrated in the CPUs 941 to 944.

1.2.3 Input and Output Modules

Input and output modules are the interfaces to the sensors and actuators of a machine or controlled system.
The following features make S5-115U modules easy to handle:

- Fast installation
- Mechanical coding
- Large labeling areas

Digital Modules

Digital modules conform to the voltage and current levels of your machine. You do not have to adapt the existing level to the programmable controller. The S5-115U adapts itself to your machine.
Digital modules have the following convenient features:

- connection of signal lines via front connectors
- a choice of screw-type or crimp snap-in connections

Analog Modules

As a programmable controller's degree of performance increases, so does the significance of its analog value processing. The significance of the analog input and output modules increases accordingly.
Analog modules handle mainly closed-loop control tasks, such as automatic level, temperature, or speed control.

The S5-115U offers floating and non-floating analog input modules. They use one range card for every four channels to adapt the desired signal level.

This feature allows you to do the following:

- Have up to four different measuring ranges on one module, depending on the number of channels a module has
- Change the measuring ranges simply by exchanging range cards

Three analog output modules cover the various voltage or current ranges of analog actuators.

1.2.4 Intelligent Input/Output Modules

Counting rapid pulse trains, detecting and processing position increments, measuring time and speed, closed-loop control, and positioning are just a few of many time-critical jobs. The central processor of a programmable controller usually cannot execute such jobs fast enough in addition to its actual control task. The S5-115U provides intelligent input/output modules (IPs) to handle these time-critical jobs. Use these modules to handle measuring, closed-loop control, and open-loop control tasks rapidly in parallel to the program.
Most of the modules have their own processor to handle tasks independently. All these modules have a high processing speed and are easy to handle.
Standard software puts them into operation.

1.2.5 Communications Processors

The S5-115U offers a number of special communications processors (CPs) to make communication easier between man and machine or machine and machine.

These are divided into the following three main groups:

- CPs for bus systems (e.g. CP 530, 143, 5430)
- CPs for linking, reporting and listing (e.g. CP 523, 524, 525)
- Operator control and monitoring (e.g. 526, 527, 528, 552)

1.3 Expansion Capability

If the connection capability of one central controller (CC) is no longer sufficient for your machine or system, increase the capacity with expansion units (EUs).

Interface modules connect a CC to EUs and connect EUs to each other. Choose an interface module suitable to the controller configuration you need.

1.3.1 Centralized Configuration

A centralized configuration allows you to connect up to three expansion units to one CC. The interface modules for this purpose connect bus lines and supply voltage to the expansion units. The expansion units in such configurations therefore need no power supplies of their own. The cables between the individual controllers have a total maximum length of 2.5 m (8.2 ft.).

1.3.2 Distributed Configuration

A distributed configuration allows you to relocate expansion units nearer to the sensors and actuators of your machine.
Distributed configurations reduce cabling costs for these devices.

1.4 Communications Systems for the S5-115U

Controller flexibility is critical to manufacturing productivity. Complex control tasks can be divided and distributed over several controllers to achieve the greatest flexibility possible.

Distribution offers the following advantages:

- Small units that are easier to manage. You can plan, start up, diagnose, modify, and operate your system more easily, and observe the entire process more easily
- Enhanced system availability because, if one unit fails, the rest of the system continues to function

Information must flow between distributed controllers to ensure the following:

- Data exchange between programmable controllers
- Central monitoring, operation, and control of manufacturing systems
- Collection of management information such as production and warehousing data

For this reason, we offer the following communications facilities for the S5-115U programmable controller:

- Point-to-point connection with the CP 524 and CP 525 communications processors
- Local area network communications via the SINEC L1 network
- SINEC H1
- SINEC L2
- Point-to-point connection with the CPUs 943, 944 and 945
- ASCII interface for connecting printer, keyboard, etc. in the case of the CPU 943, 944 and 945
- Computer connection with 3964/3964R protocol in the case of the CPU 944 and CPU 945

The CPU 945 provides a wide variety of communications possibilities thanks to different interface modules which can be connected to SI2.

1.5 Operator Control and Monitoring and Programming

SIMATIC offers you a range of operator control and monitoring devices graded according to price and performance. These devices provide you with a user-friendly means of selectively following and, if necessary, intervening in processes.

There is a meaningfully graded and compatible range of programmers available:

- PG 710
- PG 730
- PG 750
- PG 770

All the programmers feature high performance, simple handling, user-friendly operator prompting, and the standard, easily learned STEP 5 programming language.

2 Technical Description

2 Technical Description (continued)

Figures

2

Tables

2 Technical Description

This chapter describes the design of an S5-115U with the CPU 945, giving an overview of the following:

- The modular design of the PLC
- The power supply modules
- The features of the CPU 945 and its various operating modes
- Ways in which you can influence the scan time of your PLC
- The new operating system services of the CPU 945
- Further functions provided by the CPU 945 over integral blocks.

2.1 Modular Design

The S5-115U consists of various functional units that can be combined to suit the particular problem.

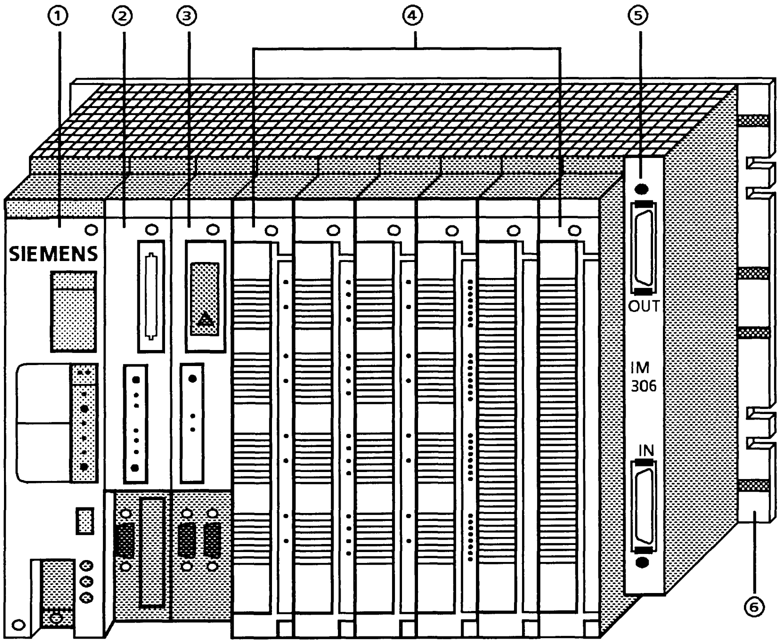


Figure 2-1 The S5-115U PLC with the CPU 945 (Central Controller)

The numbered information below briefly describes the most important components of the S5-115U.

① **Power Supply Module (PS 951)**
The PS 951 power supply module generates the operating voltage for the PLC from the 115 V AC/230 V AC or 24 V DC power system voltages. This module uses a battery or an external power supply to back up the RAM.
The PS 951 power supply module also performs monitoring and signalling functions.

② Central Processing Unit (CPU)

The central processing unit reads in input signal states, processes the control program, and controls outputs. In addition to program scanning functions, the CPU provides internal flags, timers and counters. You can preset the restart procedure and diagnose errors using the CPU's LEDs. Use the Overall Reset switch on the CPU to delete the RAM contents.
Use a programmer or a memory submodule to transfer the control program to the CPU.

③ Communications Processors (CP)

Communications processors can be used in the S5-115U for communication between man and machine and between machines. Communications processors perform the following functions:

- Operator monitoring and control of machine functions or process sequences
- Reporting and listing of machine and process states

You can connect various peripheral devices to these processors, e.g. printers, keyboards, CRTs and monitors as well as other controllers and computers.

④ Input/Output Modules (I/Os)

- Digital input modules adapt digital signals, e.g. from pressure switches or BERO® proximity switches, to the internal signal level of the S5-115U.
- Digital output modules convert the internal signal level of the S5-115U into digital process signals, e.g. for relays or solenoid valves.
- Analog input modules adapt analog process signals, e.g. from transducers or resistance thermometers, to the S5-115U, which functions digitally.
- Analog output modules convert internal digital values of the S5-115U to analog process signals, e.g. for speed controllers.

⑤ Interface Modules (IM)

The S5-115U is installed on mounting racks with a specific number of mounting locations (slots). A configuration comprising power supply, CPU, and input/output modules is called a central controller. If the slots on the central controller's mounting rack are insufficient, you can install expansion units (systems without CPUs) on additional mounting racks. Interface modules connect an expansion unit to a central controller.

⑥ Mounting Racks

A mounting rack consists of an aluminium rail to which all the modules are fastened mechanically. It has one or two backplanes that connect the modules to each other electrically.

Not represented in Figure 1-1:

Intelligent Input/Output Modules (IPs)

Intelligent input/output modules are available for handling the following special tasks:

- Counting rapid pulse trains
- Measuring and processing positioning increments
- Measuring speed and time
- Controlling temperatures and drives, and so on.

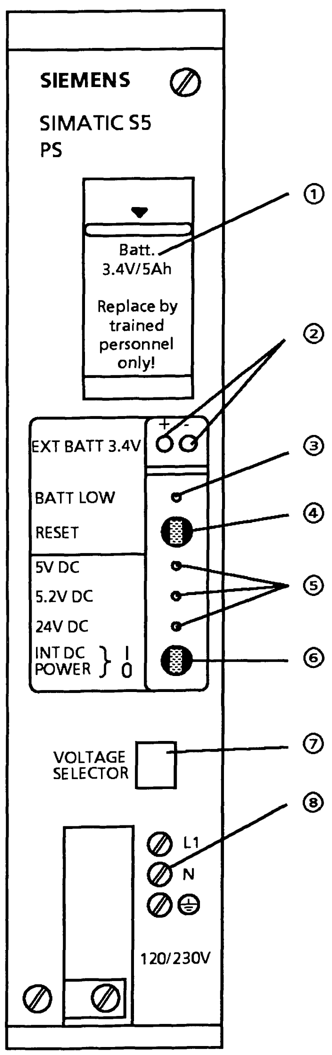
Intelligent input/output modules generally have their own processor and thus off-load the CPU. Consequently, they can process measuring and open- and closed-loop control tasks quickly while the CPU handles other jobs.

2.2 Power Supply Modules

Power supply modules generate the operational voltages for the PLC from the 120/230 V AC or 24 V DC mains supply and they provide backup for the RAM with a battery or an external power supply.
Power supply modules also execute monitoring and signalling functions.

You can set the following switches on the PS 951 power supply module:

- The Voltage Selector switch sets the line voltage at either 120 V AC or 230 V AC for AC modules. The PS 951 can also be operated with a 24 V DC power supply.
- The ON/OFF switch turns the operating voltages on or off.
- The RESET switch acknowledges a battery failure indication.



① Battery compartment

② Sockets for external 3, 4 to 9 V DC for backup (when battery is changed and power supply is shut off)

③ Battery failure indicator
The LED lights up under the following conditions:
- There is no battery.
- The battery has been installed incorrectly.
- The battery voltage has dropped below 2.8 V.

If the LED lights up, the "BAU" signal is sent to the CPU.

④ RESET switch
Use this switch to acknowledge a battery failure signal after you have installed a new battery. If you are operating the PS 951 power supply module without a battery, activate this switch to suppress the "BAU" signal.

⑤ Operating voltage displays
- +5 V supply voltage for the input/output modules
- +5.2 V supply voltage for PG 605U or PG 615, OPs, BT 777 bus terminal
- +24 V for serial interface (20 mA current loop interface).

⑥ ON/OFF switch (I = ON, 0 = OFF)
When the switch is in the "OFF" position, the operating voltages are disabled without interrupting the connected line voltage.

⑦ 120 V AC/230 V AC voltage selector switch with transparent cover

⑧ Screw-type terminals for connecting the line voltage

Figure 2-2 Power Supply Module Control Panel

2.3 CPU 945

This chapter describes the CPU 945. The following are the most important differences between the CPU 945 and the CPUs 941 to 944:

- New CPU architecture
- Extremely fast operation execution times
- Expanded STEP 5 operation set; e.g. floating-point arithmetic
- 32-bit accumulators and expanded register set
- Larger program memory
- Expanded address area; O area
- Additional flag area (S flags)
- Additional system data area (RT), which is not deleted at Overall Reset
- Improved program execution on the different program execution levels
- Call interval of the timer OBs (1 ms)
- Additional organization blocks (OBs)
- Firmware update onsite using programmer or memory submodule
- New memory submodules
- 2nd interface in the form of an interface module

2.3.1 Functional Units of the CPU 945

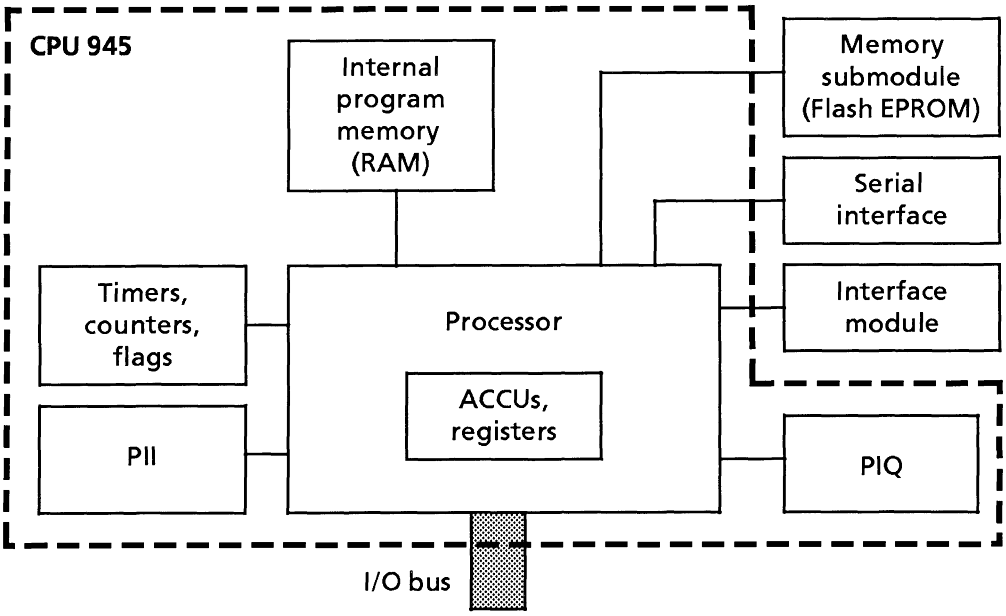


Figure 2-3 Schematic of the S5-115U

Program memory (internal program memory, memory submodule)

The control program can be stored on a memory submodule or it is stored by the programmer in RAM. The CPU 945 always copies the entire control program from the memory submodule into internal program memory (RAM) from where it is processed.
As security against powerfails, the control program is also stored on the flash EPROM external to the PLC. The internal RAM has the following characteristics in contrast to the external flash EPROM:

- The memory contents can be changed quickly.
- User data can be stored and changed.
- If memory contents are to be retained on powerfail, the RAM must be provided with battery backup.

Note: The memory submodule may be plugged in and removed in the POWER OFF state only!

Process Images (PII, PIQ)

Signal states of the digital input and output modules are stored in the CPU in "process images." Process images are reserved areas in CPU RAM.
Input and output modules have separate images as follows:

- Process input image (PII)
  and
- Process output image (PIQ)

Interfaces

The CPU 945 is equipped with the following:

- A serial interface
- A slot for interface modules

You can connect the following to serial interface 1:

- A programmer
- An OP 393 operator panel
- The SINEC L1 local area network

There are various interface modules available for different connections:

- Programmer interface module (15-pin)
- TTY interface module (25-pin)
- RS 232C (V.24) interface module (25-pin)
- RS 422-A/485 interface module (15-pin)
- SINEC L1 interface module (15-pin)

Timers, Counters and Flags

Each CPU provides the control program with internal timers, counters and flags. Flags are memory locations for storing signal states. Timers, counters and flags can each be set as "retentive" (by area), i.e. their contents are not lost at POWER OFF. Memory areas whose contents are reset at POWER OFF are "non-retentive".

Processor

The processor calls statements in the program memory in sequence and executes them in accordance with the control program. It processes the information from the PII and takes into consideration the values of internal timers and counters as well as the signal states of internal flags.
The processor contains the accumulators (shortened to "ACCUs") and further registers.
The ACCUs are arithmetic registers, over which the values of, for example, the internal timers and counters are loaded. In addition, comparison operations, arithmetic operations and conversion operations are executed in the ACCUs.
The CPU 945 also provides a range of additional registers, e.g. the BR (basic address register) and DBS (data block start register) (see Section 7.1).

I/O Bus

The I/O bus establishes the electrical connection for all signals that are exchanged between the CPU and the other modules in a central controller or an expansion unit.

Memory submodules

Memory submodules (flash EPROMs) are available for storing the control program or for transferring programs to the PLC.
These memory submodules are SIMATIC S5 memory cards with a capacity of 128, 256 and 512 Kbytes.

Memory Capacity	Order No.	Organization	Submodule ID
128 Kbytes	6ES5 374-1FG11	Byte	500
256 Kbytes	6ES5 374-1FH11	Byte	
512 Kbytes*	6ES5 374-1FJ21	Word	
128 Kbytes	6ES5 374-1KG11**	Byte	500
256 Kbytes	6ES5 374-1KH21**	Word	
512 Kbytes	6ES5 374-1KJ11**	Byte	
1 MB	6ES5 374-1KK21**	Word	

* *Only 256/384 Kbytes can be used*

** *Programmer Software level 6.6 and higher*

Overview of the CPU 945

The CPU 945 has a new CPU architecture compared to the CPUs 941 to 944. The CPU 945 contains the following:

- A STEP 5 processor (SP 90)
- A floating-point coprocessor (CP 90)
- A bus controller (AMBUS)
- A microcontroller (Motorola 68302)

The new CPU architecture results in the following advantages:

- Coprocessor execution of STEP 5 operations with SP 90 and CP 90
- Control program less burdened by communications thanks to internal multiprocessor architecture.
- The operating system is located on a flash EPROM. This makes it possible to update the operating system onsite using a programmer or via the memory submodule.

The microcontroller manages all functions implemented over both serial interfaces. This includes all programmer functions and SINEC L1 connection at serial interface 1. Various interface modules can be plugged in to constitute interface 2 (see Section 12.7).
The bus controller controls data interchange between the individual processors and the S5 bus. The CPU has an internal RAM from which the control program can be executed. The control program is always copied from the memory submodule into internal RAM.

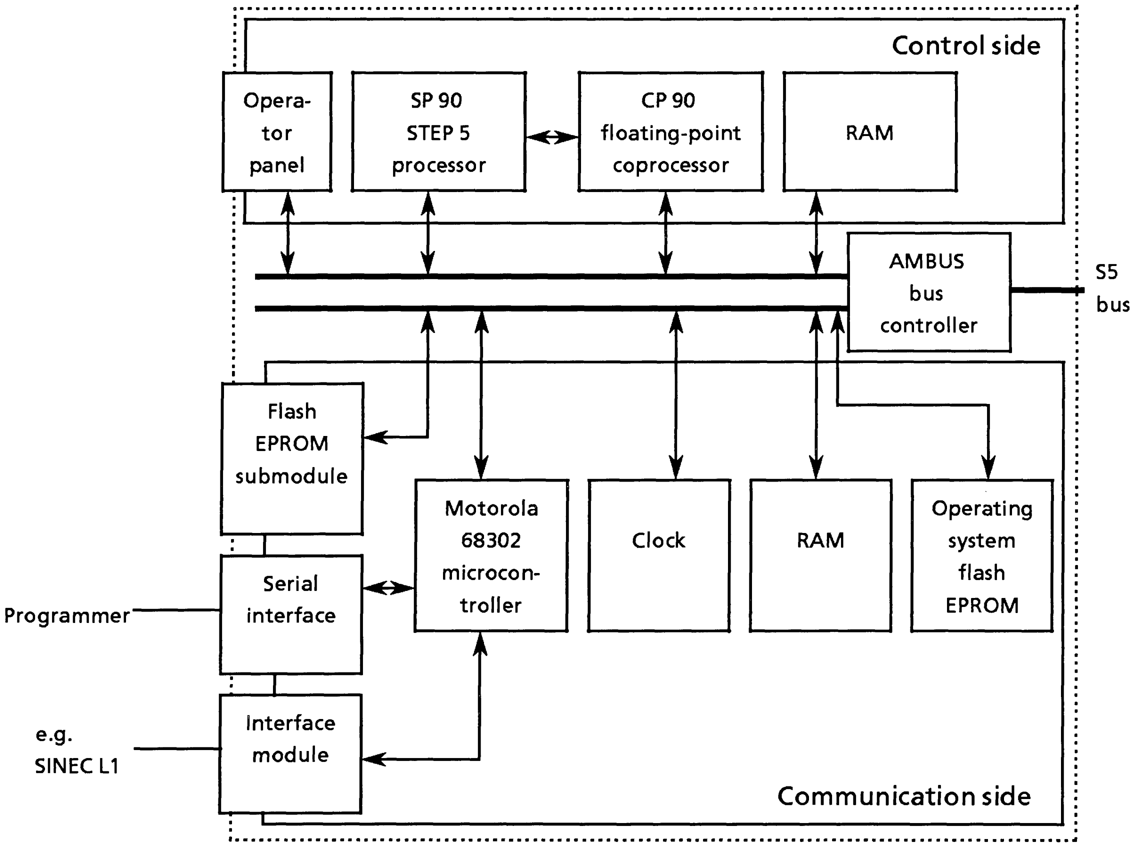


Figure 2-4 Schematic of the CPU 945

2.3.2 Features of the CPU 945

The following table shows the most important features of the CPU 945.

Table 2-1 Features of the CPU 945

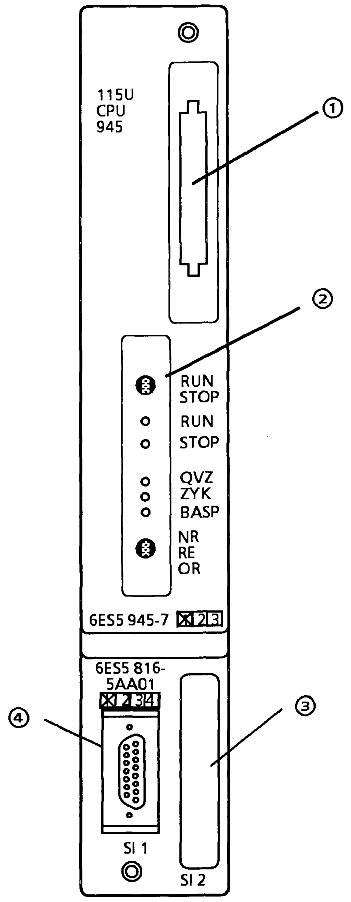
	CPU 945
Execution time per - 1000 statements (see Pocket Guide for specific information)	approx. 100 µs
Internal program memory (RAM)	256 Kbytes (945-7UA1.) 384 Kbytes (945-7UA2.)
Scan time monitoring	Default approx. 500 ms, programmable
Program scanning	Cyclic, time-controlled, interrupt-driven, timed-interrupt-driven
Address range, max. (digital inputs, analog inputs)	P area 256 bytes 0 to 127 in process I/O image 128 to 255 without process I/O image
	O area 256 bytes
	IM 3 area 256 bytes
	IM 4 area 256 bytes
Flags	2048 bits, optionally ● all retentive ● half retentive ● all non-retentive
S flags	32768 bits, optionally ● all retentive ● half retentive ● all non-retentive
Timers	256 bits, optionally ● all retentive ● T 0 to 63 retentive ● all non-retentive
Counters	256 bits, optionally ● all retentive ● C 0 to 63 retentive ● all non-retentive
Time range	0.01 to 9990 s
Counting range	0 to 999
Operation set	Approx. 250 operations

2.3.3 Operator Functions of the CPU 945

The following operator functions are possible on the front panel of the CPUs:

- Plug in a memory submodule
- Connect a programmer (PG) or an operator panel (OP)
- Connect SINEC L1
- Connect PLCs or devices of other manufacture
  - Connection with ASCII driver
  - Point-to-point connection (master function)
  - Computer interface (3964(R) procedure)
- Set the operating mode
- Preset retentive feature
- Perform Overall Reset

LEDs indicate the current CPU status.

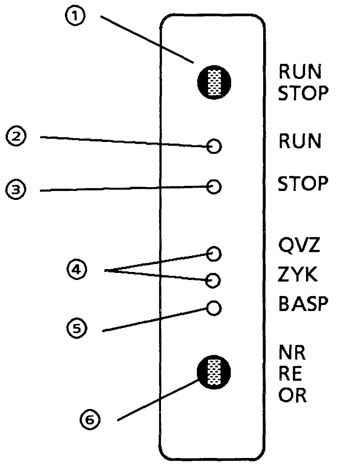


View of CPU 945

① Receptacle for memory submodule
② Control panel
③ Slot for interface module (see Section 12.7)
④ Connection sockets for PG, OP or SINEC L1 LAN

Figure 2-5 Front View of the CPU 945

The CPU controls are arranged in a panel. Figure 2-8 shows the control panel of the different CPUs.



① Mode selector STOP/RUN
② RUN LED
③ STOP LED

④ Error LEDs (QVZ, ZYK)
⑤ BASP (output disable); outputs of the output modules are not enabled
⑥ Switch for the following RESTART settings:
- nonretentive presetting (NR)
- retentive presetting (RE)
- overall reset (OR)

Figure 2-6 Control Panel of the Different CPUs

Meaning of the LEDs

Two LEDs on the control panel of the CPU indicate the operating status of the CPU (② and ③ in Figure 2-8). Table 2-3 lists the possible indications.
A flashing or flickering red LED indicates PLC malfunctions (see Chapter 5).

Table 2-2 Operating Mode LEDs

Red BASP - LED	Red STOP LED	Green RUN - LED	Meaning
☀	☀	☀	CPU is in cold restart routine or in RESTART mode
☀	☀	○	STOP mode
○	○	☀	RUN mode
☀	○	○	- Program check running or CPU is in RESET - Overall Reset active
☀	☀ Flashing	○	Request Overall Reset
☀	☀ Flickering	○	Hardware fault

2.4 Overview of the CPU 945 Operating Modes

Use the mode selector to set the STOP (ST) or RUN (RN) mode.

Everything that takes place between

- a STOP→ RUN transition (manual cold restart)
- a POWER UP→ RUN transition (automatic cold restart after power up) or
- a POWER UP→ STOP transition

is referred to as "restart characteristics".

Two phases can be distinguished during restart:

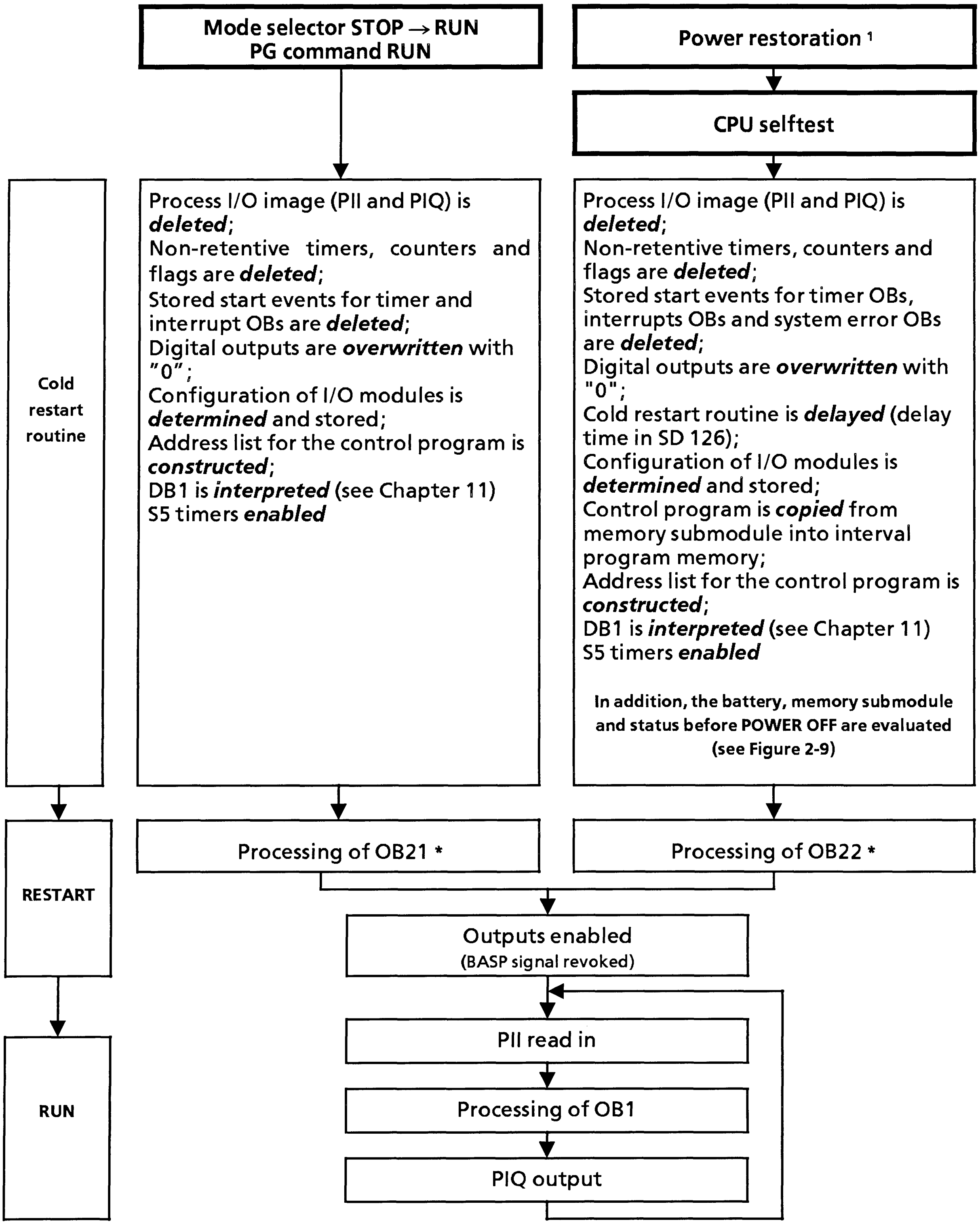
- The ***cold restart routine*** (PLC cannot be directly influenced)
- The ***RESTART*** control program (PLC characteristics can be controlled in RESTART OBs (OB21 and OB22)).

The following figure gives a clear breakdown of the restart characteristics of the CPU and cyclic operation (see Section 2.7).

Figure 2-8 shows the possible causes of operating status change.

Figure 2-9 shows how restart characteristics depend on the following:

- The status of the backup battery
- Errors
- The operating status before POWER OFF
  and
- The position of the mode selector.



Process I/O image (PII and PIQ) is ***deleted***;
Non-retentive timers, counters and flags are ***deleted***;
Stored start events for timer and interrupt OBs are ***deleted***;
Digital outputs are ***overwritten*** with "0";
Configuration of I/O modules is ***determined*** and stored;
Address list for the control program is ***constructed***;
DB1 is ***interpreted*** (see Chapter 11)
S5 timers ***enabled***

Process I/O image (PII and PIQ) is ***deleted***;
Non-retentive timers, counters and flags are ***deleted***;
Stored start events for timer OBs, interrupts OBs and system error OBs are ***deleted***;
Digital outputs are ***overwritten*** with "0";
Cold restart routine is ***delayed*** (delay time in SD 126);
Configuration of I/O modules is ***determined*** and stored;
Control program is ***copied*** from memory submodule into interval program memory;
Address list for the control program is ***constructed***;
DB1 is ***interpreted*** (see Chapter 11)
S5 timers ***enabled***

In addition, the battery, memory submodule and status before POWER OFF are evaluated (see Figure 2-9)

[1] If the PLC was in RUN at POWER OFF.

* If OB21 or OB22 contains the RA (enable interrupt) operation, a central process interrupt is possible from this point. If this operation has not been used in the RESTART OB, interrupt and timed-interrupt OBs can only become effective after the RESTART OB has been processed.

Figure 2-7 Restart Characteristics of the CPU

Changing the Operating Mode

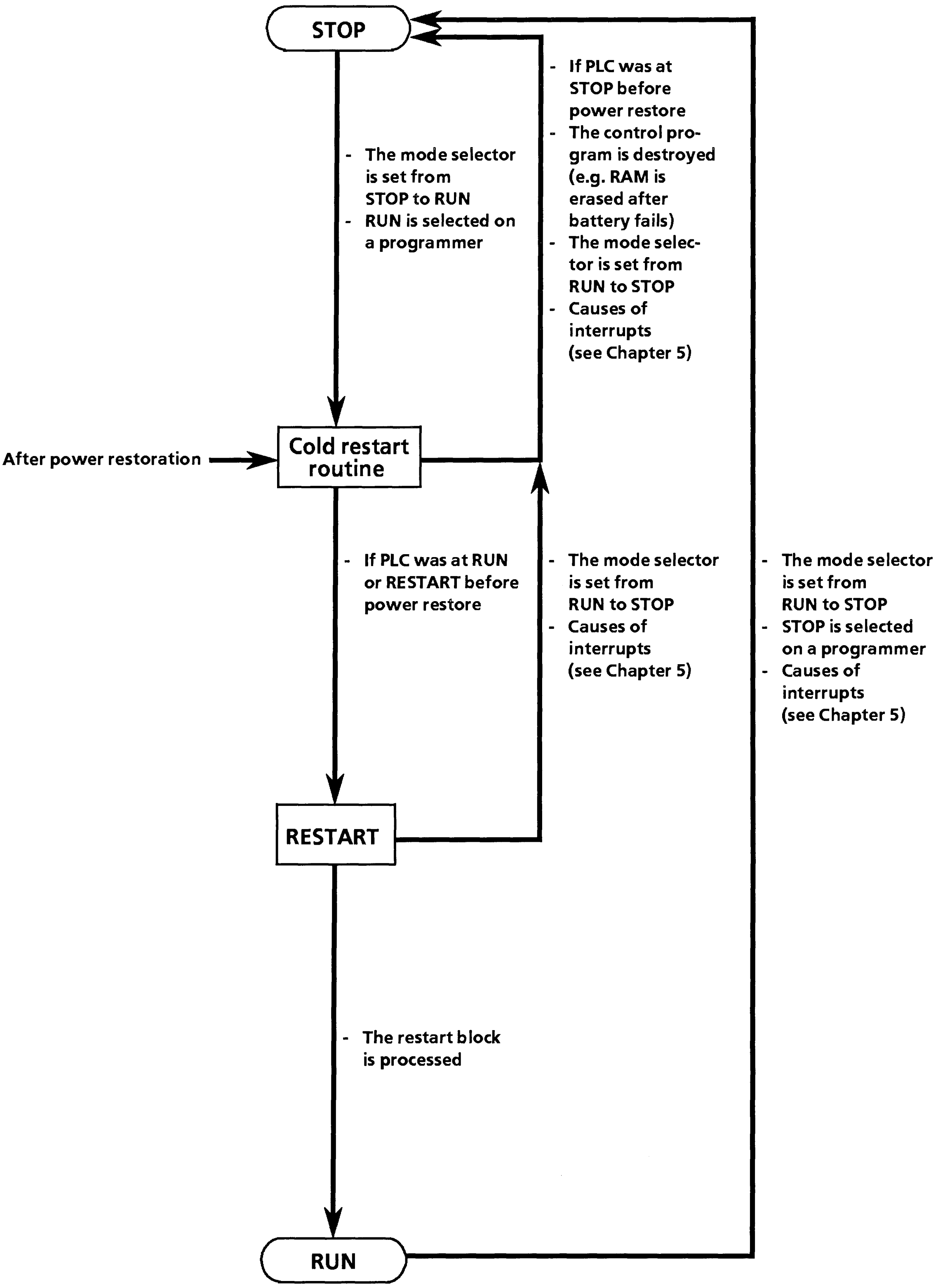
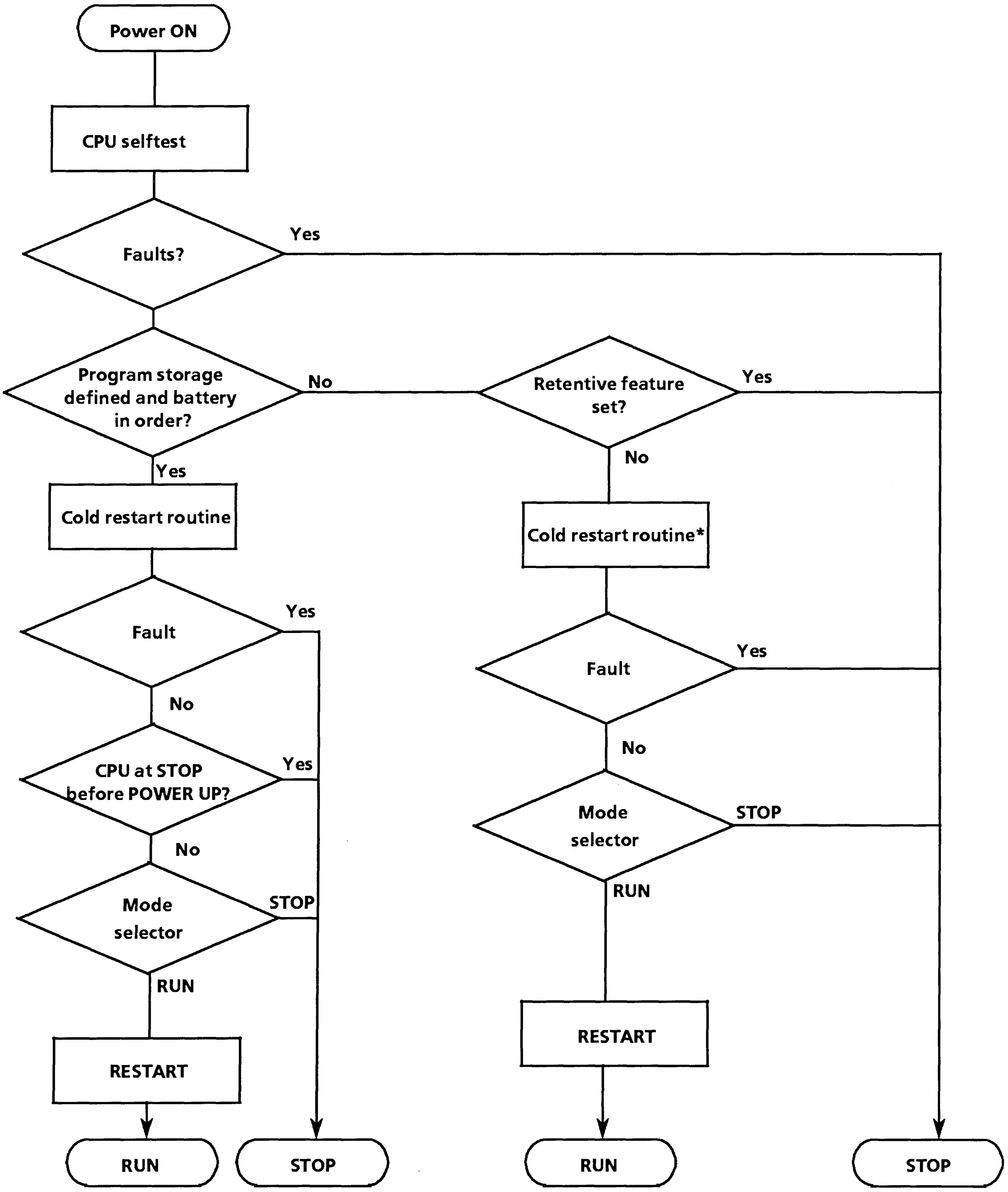

STOP
- The mode selector is set from STOP to RUN
- RUN is selected on a programmer
- If PLC was at STOP before power restore
- The control program is destroyed (e.g. RAM is erased after battery fails)
- The mode selector is set from RUN to STOP
- Causes of interrupts (see Chapter 5)
After power restoration
Cold restart routine
- If PLC was at RUN or RESTART before power restore
- The mode selector is set from RUN to STOP
- Causes of interrupts (see Chapter 5)
- The mode selector is set from RUN to STOP
- STOP is selected on a programmer
- Causes of interrupts (see Chapter 5)
RESTART
- The restart block is processed
RUN

Figure 2-8 Conditions for Changing the Operating Mode

Cold restart characteristics after power restore

The following figure shows the cold restart characteristics after power restore depending on battery status, possible faults, the status before POWER OFF and the mode selector.



* Restart delay set to default of approx. 1 second

Figure 2-9 Cold Restart Characteristics After Power Restore

2.5 STOP Mode

- BASP (command output disable) is active
- The control program is not executed
- The process I/O image and the interprocessor communication flags are not processed
- The values of timers, counters, flags and process I/O images current at the point of transition to STOP are retained
- S5 timers are not processed
- Interrupts are not stored

2.6 Restart Characteristics of the CPU 945

Everything that takes place between

- a STOP→ RUN transition (manual cold restart)
- a POWER UP→ RUN transition (automatic cold restart after power up)
  or
- a POWER UP→ STOP transition
  is referred to as "restart characteristics".

Two phases can be distinguished during restart:

- The ***cold restart routine*** (PLC cannot be directly influenced)
- The ***RESTART*** control program in which PLC characteristics can be controlled in restart OBs (OB21 and OB22).

2.6.1 Cold Restart Routine

The following takes place during the cold restart routine

- The "BASP" LED lights up and the "BASP" signal is active
  - Error flags are deleted in the case of manual cold restart
  - All fault LEDs light up briefly in the case of automatic cold restart after power restore
- The control program is copied from the memory submodule into internal program memory (RAM) (***not*** in the case of STOP → RUN)
- The address list of the blocks is re-constructed
- The start events stored for timer, interrupt and system error OBs are deleted
- Non-retentive timers, counters, flags and S flags are deleted
- Digital outputs take signal "0" if all output modules are disabled
- All inputs and outputs take signal "0" in the process I/O image
- Scan time monitoring is inactive
- DB1 is interpreted
- The configuration of I/O modules is determined and stored.
  This procedure is described on the next page.

To establish the configuration of the I/O modules, the control processor checks the full address area of the input/output modules word by word.

If it addresses a module over an I/O byte, the processor "notes" this byte by setting a bit allocated to it in a special memory area called the system data area.

This bit is only set by the processor if both I/O bytes of an I/O word are addressable.
The processor uses one system data word (RS) to check 16 I/O bytes.

Using this method, the processor determines the bytes of the process I/O image to be updated during process I/O image transfer. Table 2-3 lists all relevant system data words in the system data area.
If, for instance, I/O byte 13

- ***can be read***, bit 5 is set in ***system data word (RS) 128;***
- ***can be written to***, bit 5 is set in ***system data word (RS) 136;***

Table 2-3 System Data Area; List of All Addressable I/O Words (DI=Digital Input Byte, DQ=Digital Output Byte, AI=Analog Input Byte, AQ=Analog Output Byte)

RS	Bit								Abs. Addr. of the RS
	15	14	13	12	11	10	9	8	
	7	6	5	4	3	2	1	0	
128	DI 7 DI 15	DI 6 DI 14	DI 5 DI 13	DI 4 DI 12	DI 3 DI 11	DI 2 DI 10	DI 1 DI 9	DI 0 DI 8	E1100$_H$
129	DI 23 DI 31	DI 22 DI 30	DI 21 DI 29	DI 20 DI 28	DI 19 DI 27	DI 18 DI 26	DI 17 DI 25	DI 16 DI 24	E1102$_H$
⋮									⋮
135	DI 119 DI 127	DI 118 DI 126	DI 117 DI 125	DI 116 DI 124	DI 115 DI 123	DI 114 DI 122	DI 113 DI 121	DI 112 DI 120	E110E$_H$
136	DQ 7 DQ 15	DQ 6 DQ 14	DQ 5 DQ 13	DQ 4 DQ 12	DQ 3 DQ 11	DQ 2 DQ 10	DQ 1 DQ 9	DQ 0 DQ 8	E1110$_H$
⋮									⋮
143	DQ 119 DQ 127	DQ 118 DQ 126	DQ 117 DQ 125	DQ 116 DQ 124	DQ 115 DQ 123	DQ 114 DQ 122	DQ 113 DQ 121	DQ 112 DQ 120	E111E$_H$
144	AI 7 AI 15	AI 6 AI 14	AI 5 AI 13	AI 4 AI 12	AI 3 AI 11	AI 2 AI 10	AI 1 AI 9	AI 0 AI 8	E1120$_H$
⋮									⋮
151	AI 119 AI 127	AI 118 AI 126	AI 117 AI 125	AI 116 AI 124	AI 115 AI 123	AI 114 AI 122	AI 113 AI 121	AI 112 AI 120	E112E$_H$
152	AQ 7 AQ 15	AQ 6 AQ 14	AQ 5 AQ 13	AQ 4 AQ 12	AQ 3 AQ 11	AQ 2 AQ 10	AQ 1 AQ 9	AQ 0 AQ 8	E1130$_H$
⋮									⋮
158	AQ 103 AQ 111	AQ 102 AQ 110	AQ 101 AQ 109	AQ 100 AQ 108	AQ 99 AQ 107	AQ 98 AQ 106	AQ 97 AQ 105	AQ 96 AQ 104	E113C$_H$
159	AQ 119 AQ 127	AQ 118 AQ 126	AQ 117 AQ 125	AQ 116 AQ 124	AQ 115 AQ 123	AQ 114 AQ 122	AQ 113 AQ 121	AQ 112 AQ 120	E113E$_H$

Note

These bits are flag bits, i.e. manipulation of these bits by the control program has no effect on update of the process I/O image. Changes to the determined module configuration affecting updating of the process I/O image can only be made over operating system service No. 6 (see Section 2.10).

2.6.2 RESTART

While the CPU is in RESTART, the following applies:

- The fault LEDs are dark; the RUN, STOP and BASP LEDs light up
- All output modules are disabled (outputs show signal "0")
- The PII is not yet updated; evaluation of the inputs is only possible with direct I/O access (L PY../L PW..)

  Example:
  ```
  L    PW 0
  T    IW 0
  A    I  0.0
  ```
- Scan time monitoring is inactive
- The relevant RESTART OB is processed (in the case of manual cold restart of OB21, in the case of automatic cold restart of OB22 - if the mode selector is at RUN)
- Timers are processed
- Interrupt OBs (OB2 to OB6) and timed-interrupt OBs (OB10 to OB13) are only processed if the interrupts are explicitly enabled (RA operation).

RESTART program execution

During RESTART, i.e.

- after a STOP → RUN transition (manual cold restart)
  and
- after a POWER OFF → POWER UP transition (automatic cold restart after power restore if the CPU was previously in the RUN mode), the CPU operating system automatically invokes a RESTART OB, provided one has been programmed:
- OB21 (in the case of manual cold restart)
  or
- OB22 (in the case of automatic cold restart after power restore if the CPU was previously in RUN).

If you have programmed these blocks, this program will be executed before cyclic program execution; this makes it suitable for, e.g.

- (one-off) setting of specific system data
  or
- carrying out parameterization for the process to be automated.

If the relevant RESTART OB has not been programmed, the CPU branches direct to RUN (cyclic program execution, OB 1).
At this point, we have an example of how a RESTART OB can be programmed.

Example 1: Programming OB 22

A check is to be made after power restore to ensure that all input/output modules are ready for operation. If one or more modules cannot be accessed (not plugged in or faulty), the PLC must branch to STOP.

OB22 STL	Explanation
:L KB 0	Output words 0, 2 and 4 are set to "0"
:T PW 0	
:T PW 2	
:T PW 4	
:L PW 6	The information in input words 6, 8 and 10 is loaded consecutively into ACCU 1.
:L PW 8	
:L PW 10	
:BE	

If an input or output module cannot be accessed with the statement L PW or T PW, the CPU enters STOP at this statement and the QVZ (timeout) interrupt bit is set in the ISTACK (see Chapter 5) if OB23 has not been programmed to respond to a QVZ (see Section 2.8.7).

Example 2: Programming OB 21 and FB 1

After cold restart using the mode selector, flag bytes 0 to 99 are to be preset with "0", flag bytes 100 to 127 are to be retained since they contain important machine data.

Prerequisite: Retentive switch at retentive position (RE).

OB21 STL	Explanations
:JU FB 1	Unconditional FB1 call
NAME :DELETE M	
:BE	

FB1 STL	Explanations
NAME :DELETE M	
:L KB 0	Flag word 200 is preset with "0"
:T FW 200	The value "0" is stored in ACCU 1
M10 :L KB 0	The contents of FW 200 specify the
:B FW 200	current flag word.
:T FW 0	The current flag word is set to "0"
:L FW 200	
:I 2	The contents of FW 200 are incremented by 2.
:T FW 200	
:L KF +100	The comparison value "100" is loaded into ACCU 1
:<F	As long as contents of FW 200 < 100,
:JC =M10	jump to marker 10.
:BE	The bytes FY 0 to 99 are set to "0".

Programmable Restart Delay at Cold Restart and After Power Restore

If you want to delay checking the module configuration because, for example, switching the voltage to a remotely connected expansion unit is delayed, you must modify system data word 126 (E $10FC_H$) in one of the following ways:

- With the "Display memory contents" programmer function (only permissible when the CPU is in the STOP mode!)
- With STEP 5 operations in the control program (only in FBs).

In any event, the restart delay will only become effective after the next POWER OFF → POWER ON transition and remains effective until the next modification to this system data word. After Overall Reset, the default applies (0000_H, i.e. no delay). One unit in system data word 126 corresponds to a restart delay of 1 ms; the longest possible delay is 65535 ms ($FFFF_H$).

Example: Programming a restart delay of approximately one minute

STL FB 98	Explanation

`:L KH EA60`
`:T RS 126`
`:BE`
	Restart delay of 60,000 ms (1 minute)

Note

If no backup battery has been inserted in the power supply module (or if the inserted battery is defective) and the control program is stored on a flash EPROM submodule, the restart will be delayed by approximately one second.

2.7 RUN Mode

After the CPU operating system has run the restart program, it starts cyclic program execution (see Section 2.8.2).

Cyclic program execution means the following:

- The process I/O image and the interprocessor communication flags are processed cyclically (see Section 2.8.2)
- The control program is processed cyclically (OB1) (see Section 2.8.2)
- The time-controlled, interrupt-driven and timed-interrupt-driven program execution levels and the system fault level are enabled for processing.

In addition, scan time monitoring is active (see Section 2.9.5).

2.8 Program Execution Levels of the CPU 945

2.8.1 Overview of the Program Execution Levels of the CPU 945

In RUN mode, we distinguish between 5 types of program execution:

- Cyclic program execution
- Time-controlled program execution
- Interrupt-driven program execution
- Timed-interrupt-driven program execution
- System error handling

Programs are executed on different program execution levels. The program execution levels are invoked by the operating system over organization blocks (OBs).

The various program execution levels have different priorities for program processing. The interrupt characteristics of each program execution level depend on the priority of that level.

It is a basic rule of interrupt behaviour that:
A higher-priority level can interrupt a lower-priority level after every STEP 5 statement.

You must note the following with regard to interrupts:

- The operations DO DW/DO FW ***and*** the next operation count as ***one*** operation.
- The operations TNB/TNW cannot be interrupted.
- There are restrictions as to the STEP 5 statements after which integral blocks can be interrupted.

If a level is interrupted by a higher-priority level, program execution is continued at the interrupt point after the higher-priority level has been executed.

The following table gives an overview of the various program execution levels and their interrupt characteristics.

Table 2-4 Program Execution Levels of the CPU 945

Program Execution	Level	OB	Interrupt Characteristics	Priority
Cyclic	Cyclic level	OB1	The cyclic level can be interrupted by ● all time levels ● all interrupt levels ● the timed-interrupt level ● the system error level The cyclic level cannot interrupt any other level	Low
Time-controlled	Time level 3 Time level 2 Time level 1 Time level 0	OB10 OB11 OB12 OB13	A time level can be interrupted by ● a higher-priority time level ● all interrupt levels ● the timed-interrupt level ● the system error level A time level can interrupt the following ● the cyclic level - in the control program - in the operating system ● lower-priority time levels	
Interrupt-driven	Interr. level D Interr. level C Interr. level B Interr. level A	OB5 OB4 OB3 OB2	An interrupt level can be interrupted by ● a higher-priority interrupt level ● the timed-interrupt level ● the system error level An interrupt level can interrupt the following ● the cyclic level - in the control program - in the operating system ● all time levels ● lower-priority interrupt levels	
Timed-interrupt driven	Timed-interrupt level	OB6	The timed-interrupt level can only be interrupted by ● the system error level The timed-interrupt level can interrupt the following ● the cyclic level - in the control program - in the operating system ● all time levels ● all interrupt levels	
System error handling	System error level - Scan time monitoring - Collision of two timed interrupts - I/O error	 OB26 OB33 OB35	The system error level cannot be interrupted by any other level and cannot even interrupt itself. The system error level can interrupt all other levels.	High

2.8.2 Cyclic Program Execution

After the operating system of the CPU has run the restart program, it starts cyclic program execution.

The maximum duration of the cyclic program is defined by the scan monitoring time (see Section 2.9.5).

Cyclic program execution means the following
① Scanning the input signals at the input modules and representing them in the PII
② Updating the input interprocessor communication flags (see Section 12.2.1)
③ Excuting OB1
(running the control program; results are written into the PIQ or, e.g. stored in flags or DBs)
④ Transferring data from the PIQ to the output modules
⑤ Transferring output interprocessor communication flags to the CPs.

Transfer of the process I/O image

In contrast to the CPUs 941 to 944, process I/O image is transferred in two steps in the CPU 945:

- The bus controller (AMBUS) updates the PII' in accordance with the determined module configuration.
- The control processor fetches the updated PII' from the bus controller and transfers it into the PII for further processing.

Output of the process I/O image to the digital output modules is executed in the same two steps. Figure 2-10 shows the two steps of process I/O image transfer.

Note

Unused inputs (I x.y, IB x, IW x, ID x) cannot be used as additional flag areas since they are overwritten with "0" every time the process input image is transferred (I' transfer).

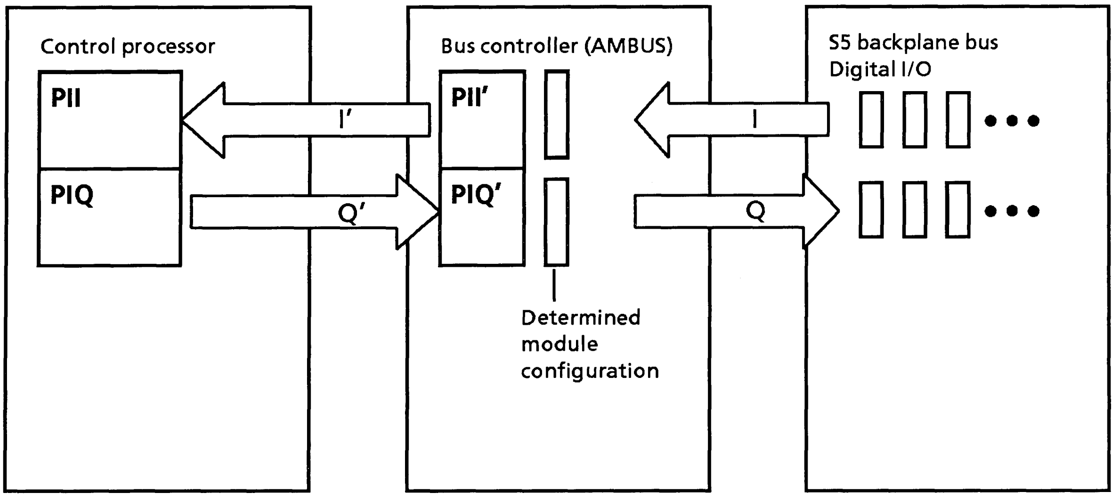


Figure 2-10 Data Flow in the Case of Process I/O Image Transfer

There are two different types of process I/O image transfer:

- Sequential
- Parallel

There are two methods of setting the type of process I/O image transfer:

- In DB1 with the PPIT parameter (see Chapter 11)
- With system data bit 120 (E $10F0_H$); parallel transfer of the process I/O image is set by setting bit No. 7 = "1".

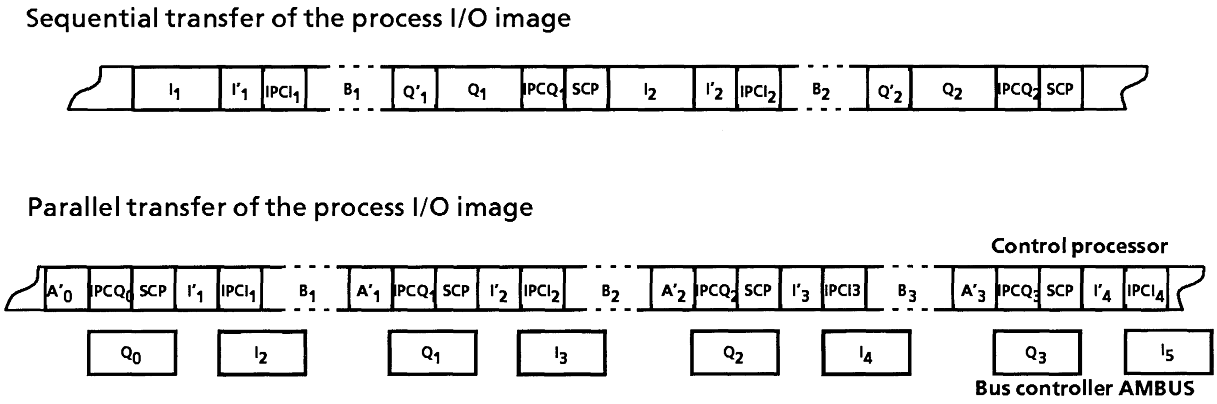
Sequential process I/O image transfer

In sequential process I/O image transfer, the PII is transferred before the control program. The new signal states resulting from the control program are transferred from the PIQ to the output modules.
All sequential process I/O image transfer times are accommodated fully in the scan time (see Figure 2-11).

Parallel transfer of the process I/O image

Here, the process I/O image is transferred in two independent procedures (see Figure 2-11)

- At the beginning and end of the control program between the control processor and the bus controller
- Parallel to the control program between the bus controller and the I/O.


Sequential transfer of the process I/O image

Parallel transfer of the process I/O image

Legend: I ... Reading the digital input modules into the PII of the bus controller
I' ... Transfer of the PII from the bus controller into the PII of the control processor
IPCI ... Reading in the interprocessor communication flags
B ... Control program
Q' ... Transfer of the PIQ from the control processor into the PIQ of the bus controller
Q ... Output of the PIQ from the bus controller to the digital output modules
IPCQ ... Outputting the interprocessor communication flags
SCP... Scan control point

Figure 2-11 Sequential and Parallel Transfer of the Process I/O Image

Reducing PIQ transfer

The CPU 945 offers the possibility of executing PIQ transfer in a reduced form.

For this purpose there is a memory area for the digital outputs in addition to the memory area containing the determined module configuration (see Section 2.6.1). This additional memory area contains the output bytes which have changed in the current scan compared to the previous scan.

If the reduced PIQ transfer is active, only those output bytes which have actually changed are transferred to the output modules.

Failure of an output module is not detected during reduced PIQ transfer if the modules output bytes only change infrequently. To prevent this, all available output bytes mut be output after a specific number of scans.

You can program the number of scans with reduced PIQ transfer as follows:

- With the DB1 RPIC parameter (Reduced Process Image Output Counter) (see Chapter 11) or
- With operating system service No. 7 "Reduction of PIQ transfer" (see Section 2.10).

The programmed number of scans with reduced PIQ transfer is followed by one scan with a complete PIQ transfer.

The RPIC parameter can be set in the range 1 to 255. If a default value of RPIC = 0 is set, reduced PIQ transfer is switched off, i.e. a complete PIQ transfer takes place after each scan.

Note

Defective modules are only detected after one or several scans, since usually not all connected output modules are addressed within one scan.

Scope for fast responses to signal changes

Even during cyclic program execution there is scope for fast responses to signal changes by, for example

- programming organization blocks for interrupt processing and using interrupt modules
- using operations with direct I/O access (e.g. L PW, T PY)
- multiple programming of direct I/O scans in the control program.

2.8.3 Time-Controlled Program Execution

OBs 10 to 13 are available for time-controlled program execution. The time OBs are called by the operating system at fixed intervals.

You can set the call intervals for the time OBs in different ways:

- In DB1 (see Chapter 11)
- Over the operating system services in OB250 (see Section 2.10).

The call interval can be set in the range from 1 ms to 1 min (range 0 to FFFF_H; see Table 2-5). You can also change the call interval using the operating system services. A call interval of 100 ms is the default for OB13.

Time OBs interrupt the cyclic program after each STEP5 operation. Time OBs ***cannot*** interrupt the following:

- Higher-priority time OBs
- OB6
- Process interrupts (OB2 to OB5)
- The system error OBs 26, 33, 35

Time OBs can be interrupted by the following:

- A higher-priority time OBs
- OB6
- Process interrupts (OB2 to OB5)
- System error OBs.

The order of priority for the time OBs is as follows:

Highest priority:	OB13
	OB12
	OB11
Lowest priority:	OB10.

Please note also:

- You can disable invocation of all time OBs with the "IA" operation. You can then enable invocation of all time OBs agian with the "RA" operation.
- You can use the "SIM" operation to disable one or several time OBs (see section 8.2.8).
- Disabling invocation of time OBs is needed if integral handling blocks are used in the cyclic and time-controlled program and if they access the same page number in doing so. The interrupts must be disabled and then re-enabled before every invocation of an integral data handling block in the lower-priority program.

- During an interrupt disable or while a time OB is executing, one call request per time OB is stored. If a second call request occurs for the same time OB, the CPU signals "collision of two timed interrupts".

- If time OBs are to be processed in the RESTART OB (OB21, OB22), you must enable interrupts in the RESTART OB with "RA".

- The block nesting level of 50 levels applies also when processing a time-controlled OB and must not be exceeded.

- If a time-controlled OB uses "scratch flags" which are also being used in the cyclic control program or in a lower-priority time-controlled program, these flags must be saved in a data block during processing of the timed-interrupt OB.

- Invocation of the timed-interrupt OBs can be delayed in the following cases (see Table 2-6)
  - If integral blocks are used
  - If the clock has been parameterized
  - If programmer/OP functions are active
  - If SINEC L1 is connected
  - If the computer interface or ASCII driver are active
  - If the S5 bus is accessed direct by the IP 252
  - If higher-priority execution levels are active or if interrupts are disabled

- The values in system data words RS 97 to RS 100 are only flags for the set intervals. Changing system data words RS 97 to RS 100 does not effect any change in the call intervals. The call intervals can only be changed over operating system services No. 2 to 5 (see Section 2.10).

Table 2-5 Parameter Block for Timed-Interrupt OBs

System Data Word	Absolute Address	High Byte	Low Byte	Default	Service No.
RS 97	E 10C2	Interval for OB13 (0 to FFFF$_H$)		100 (=100 ms)	5
RS 98	E 10C4	Interval for OB12 (0 to FFFF$_H$)		0 (=no call)	4
RS 99	E 10C6	Interval for OB11 (0 to FFFF$_H$)		0 (=no call)	3
RS 100	E 10C8	Interval for OB10 (0 to FFFF$_H$)		0 (=no call)	2

Example
Setting an interval for 1 s for OB13 with operating system service No. 5

OB21/22 STL	Explanation
	Set OB13 interval
:L KF+1000	Load interval (1 s)
:L KB 5	Load operating system service function number for OB13
:JU OB 250	Call operating system service
:BE	

2.8.4 Interrupt-Driven Program Processing

OBs 2 to 5 are invoked automatically by the operating system if a (process) interrupt (interrupt A, B, C or D) occurs.

Programming interrupt blocks

You can use interrupt-initiating modules in the S5-115U (e.g. intelligent I/O modules, the 434-7 digitial input module or the 485-7 digital input/output module (see Chapter 9)). These modules activate the CPU over an interrupt line in the I/O bus (S5 backplane bus). The CPU distinguishes between A, B, C or D interrupts depending on which interrupt line has been activated.

Each of these interrupts causes the operating system of the CPU to interrupt the cyclic or time-controlled program and to call an interrupt OB:

OB2 in the case of interrupt A (interrupt A is triggered by the 434-7 DI module (see Chapter 9), 485-7 DI/DQ, by some CPs or by IPs)
OB3 in the case of interrupt B (interrupt B is triggered by some CPs or by IPs)
OB4 in the case of interrupt C (interrupt C is triggered by some CPs or by IPs)
OB5 in the case of interrupt D (interrupt D is triggered by some CPs or by IPs)

What interrupts what and where?

The following execution levels can be interrupted:

- The cyclic program (OB1)
- The time-controlled program
- A lower-priority interrupt

The priority of the interrupts is set as follows:

Highest priority:	Interrupt A
	Interrupt B
	Interrupt C
Lowest priority:	Interrupt D

The timed-interrupt level (OB6) and the system error level (OB33, 35, 36) have higher priority and cannot be interrupted.
Lower-priority program levels can be interrupted after each operation.
Integral function blocks and operating system routines can only be interrupted at predefined locations (cannot be changed!).

If you have not programmed an interrupt OB, the interrupted program will be continued immediately at the point of interruption.

If a second interrupt request (edge) occurs during processing of the same interrupt OB, the request can be stored. One request can be stored for every interrupt line.

Example: An interrupt B and, shortly afterwards, an interrupt A occur while the CPU is processing OB2.

After the CPU has processed OB2, it re-invokes OB2 (with interrupt A), and only then does it invoke OB3.
If another interrupt A (OB2) occurs during processing of OB3, OB3 is immediately interrupted in order to process OB2. Processing of OB3 is continued after OB2 has been processed.

If any section of your cyclic or time-controlled program is not to be interrupted, you must protect this section from interruption using the "IA" (disable interrupt) operation. At the end of the "protected" section, interrupts must be enabled again using the "RA" operation. During interrupt disable, one interrupt per interrupt line can be stored!
You can also disable each interrupt level individually using the "SIM" operation if the interrupts are not disabled by the "IA" operation (see Section 8.2.8).

Disabling interrupts is necessary in the following cases:

- If you are using integral data handling blocks both in the cyclic/time-controlled program and in the interrupt program
- If the data handling blocks access the same page number.

You must disable the interrupts before every integral data handling block call in the cyclic porogram, time-controlled program and in a lower-priority program!

> ⚠ **CAUTION**
>
> Many standard function blocks for IPs cancel the interrupt disable because they operate with the IA and RA operations!
> If you use these standard function blocks during restart or in an "interrupt-protected" program section, the relevant interrupt OBs may be invoked inadvertantly!

Enabling interrupts in the restart program (OB21, OB22)

If you want interrupt responses already at restart, you must enable the interrupts at the beginning of the restart OB with the "RA" operation (not possible with the 434-7 digital input module and the 485-7 digital input/output module). Otherwise, the interrupts will only come into effect after the restart OB has been processed.

Notes on avoiding programming errors

- Note that the block nesting depth of 32 levels must not be exceeded even when calling interrupt OBs!
- If you use the same flags in the interrupt service routine as in the cyclic program, you must save the contents of these flags at the beginning of the interrupt service routine (e.g. in a data block); at the end of an interrupt service routine, transfer the saved contents of the flags back to the relevant flag bytes (words).

Example of interrupt OB (OB2, OB3, OB4, OB5)

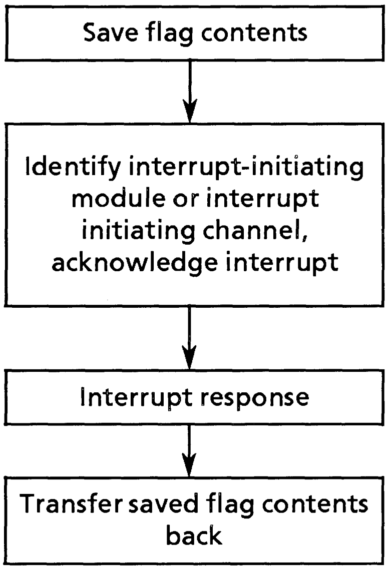

Save flag contents
Identify interrupt-initiating module or interrupt initiating channel, acknowledge interrupt
Interrupt response
Transfer saved flag contents back

Figure 2-12 Program for Interrupt OB (Principle)

Calculating interrupt response times

The total response time is the sum of the following:

- Signal delay of the interrupt-initiating module (=time from interrupt-initiating input signal change until activation of the interrupt line)
- Interrupt response time of the CPU
- Execution time of the interrupt program (=sum of all STEP 5 operations in the interrupt-evaluating program).

Calculate the interrupt response time of the CPU as follows:

Interrupt response time of the CPU = basic response time + additional response times + processing time of a higher-priority program execution level

The basic response time is 23 to 85 µs.
Additional response times result from:

- Integral FBs
- The parameterized clock
- Programmer/OP functions
- Activated computer interface (3964(R) procedure) or activated ASCII driver
- SINEC L1
- S5 bus accesses (worst case ready delays see Table 2-10)
- Direct S5 bus access by IP 252

See Table 2-6 for the additional response times which can vary.

Table 2-6 Additional Response Times

Additional CPU Functions Used	Delay of the Interrupt Response Time
Data handling blocks and other integral FBs*	≤0.1 ms
Clock parameterized	≤0.1 ms
SINEC L1-Bus connected to SI1 or SI2	≤0.2 ms
Computer link (3964(R)), ASCII driver	≤0.35 ms
S5 bus accesses** Greatest ready delay times of the plugged-in modules	→ Tab. 2.10
Operating system services (OB250)	OB250 cannot be interrupted; additional response time corresponds to execution time of OB250, see Pocket Guide
Direct S5 bus access via IP 252	0.32 ms
OP functions	≤0.1 ms
Programmer functions at one interface	
FORCE VAR/STATUS VAR	≤0.2 ms
Status Block/Transfer Block	≤0.2 ms
Display address	≤0.2 ms
Compress block with programmer or FB COMPR	
- if no blocks are being shifted	≤3.3 ms
- if blocks are being shifted	≤3.3 ms + 2 ms per 1K statement of the block to be shifted

* Cf. "Programmer functions, compress block" for FB238 (COMPR)

** Are primarily relevant if TNB/TNW are used to access the I/O area (see the Pocket Guide for the execution times of TNB/TNW). These operations cannot be interrupted.

2.8.5 Timed-Interrupt-Driven Program Processing

OB6 is called by the operating system when the time started by operating system service no. 1 (see Section 2.10) has run (provided interrupts are not disabled by the "IA" or "SIM" operations).
In OB6, you program the response after the set time has run (timed interrupt).

> **Note**
>
> A running clock prompt can be "restarted" by invoking operating system service no. 1 "Activation of OB6" again. The operating system then restarts the clock prompt specified by operating system service no. 1. A running clock prompt can be stopped by transferring the value "0" to the operating system service (prevents invocation of OB6!).

After the clock prompt has been started, system data word 101 (E 10CA$_H$) contains the selected time. When the programmed time has run, the operating system enters the value "0" in system data word 101 and calls OB6.

The following applies for OB6:

- To start the clock prompt, you must activate operating system service no. 1 "Activation of OB6". The set time is flagged in system data word 101 (E 10CAH) while the time is running.
- You can set the clock prompt in steps of 1 ms. The programmable clock prompt is therefore in the range 1 to 65535 ms (possible deviation: -1 ms).
- OB6 can be interrupted by system error OBs 26, 33, 35.
- OB6 can interrupt the cyclic and time-controlled program and can also interrupt a running interrupt program (OB 2 to 5) (save scratchflags, if necessary)!
- Invocation of OB6 can be delayed if
  - Integral blocks are in use
  - The clock has been parameterized
  - Programmer/OP functions are active
  - SINEC L1 is connected
  - Computer interface or ASCII driver are active
  - The IP 252 accesses the S5 bus direct
  - The system error level is active or interrupts have been disabled.

  See Table 2-6 for the time by which invocation of OB6 is delayed.
- You can disable processing of OB6 selectively with the "SIM" operation or generally with the "IA" operation.
- OB6 requests received while interrupts are disabled, or while OB6 is running, are stored. If a further OB6 request occurs, the CPU signals "collision of two timed interrupts".

Example: Starting the clock prompt

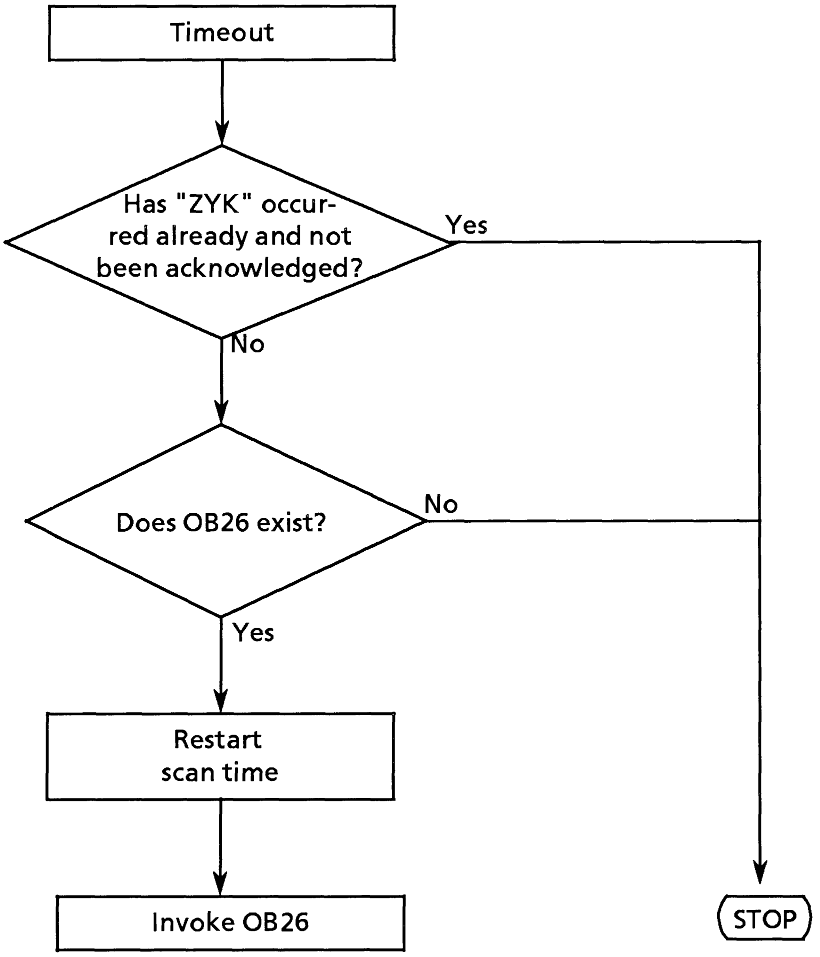
PB1 STL	Explanation
:L KF+1000	Load time value 1 s
:L KB 1	Load function number of the operating system service
:JU OB 250	Invoke operating system service

2.8.6 System Error Level

The CPU 945 allows you to respond to system errors with OBs 26, and 35. System errors are errors which can occur at any time in the control program and are not tied to any one program execution level (see Section 2.8.1). The error OBs 26, 33 and 35 have the highest priority. These OBs can only be selectively disabled with the SIM operation. They cannot be disabled with IA.
If several system errors occur, the CPU 945 stores them and processes them consecutively.

OB26 Response to timeout

If the scan time is exceeded, the scan time can be restarted over OB26.
The following sequence takes place after the "ZYK" signal:



If the error is not acknowledged by restarting the scan time with OB31 (e.g. in OB26) and the scan time runs out again, the CPU goes to STOP with ZYK.

If there is no OB26, the CPU goes to STOP with ZYK.

OB33 Response to collision of two timed interrupts

If any of the timed interrupt collisions listed in Table 2-7 occur, the operating system interrupts program processing and branches to OB33. An error code has been assigned to each of the collision events. This error code is in ACCU 1 when entering OB33 and can be evaluated. If you have not programmed an OB33, no response is made to the collision. Until a collision event with a specific error code has been processed no further collision event with the same error code will be stored.

Table 2-7 Errors Handled in OB33

Collision of Two Timed Interrupts	Error Code	Cause of Error
Timed-interrupt level (OB6)	1	● A timed-interrupt OB or time OB is running and a second request arrives for this OB or ● Interrupt is disabled and a second request arrives for this OB
Time level (OB13)	2	
Time level (OB12)	3	
Time level (OB11)	4	
Time level (OB10)	5	

OB35 Response to I/O errors

If one of the I/O error events listed in Table 2-8 occurs, the operating system interrupts program processing and branches to OB35. An error code has been assigned to each of the I/O error events. This error code is in ACCU 1 when entering OB35 and can be evaluated. If you have not programmed an OB35, the CPU goes to STOP when these errors occur. If one of the I/O errors is detected during restart, the CPU will not start, even with a programmed OB35. Until an I/O error event with a specific error code has been processed, no further I/O error event with the same error code will be stored.

Table 2-8 Errors Handled in OB35

I/O Errors	Error Code	Cause of Error
PEU	1	Powerfail in expansion unit
HOLD	2	Bus fault - Module defect - Short-circuit on backplane bus
RDY	3	
ASF	4	● Termination connector missing in central controller ● Connection to expansion unit broken

If PEU occurs during transfer of a process I/O image, both PEU and QVZ will be detected and stored at process I/O image transfer. This QVZ is responded to when no more OB35 requests are pending, i.e. you may have to program an OB24.

If one of these I/O errors persists, OB35 is called again every time it is processed. This makes any other program processing impossible. However, scan time monitoring continues although it may have to be restarted.

Example

In a system with an expansion unit connected in distributed configuration, the PLC is to restart after POWER UP if the POWER OFF for the expansion unit was up to 2 s before the POWER OFF of the central controller.

OB35 STL	Explanation
:JU FB 35	Invocation of FB35
NAME:WAIT	
:BE	

FB35 STL		Explanation
:A	F 0.0	Force RLO = 0
:AN	MF0.0	
:L	KT 200.0	Load time value 2 s
:SE	T0	Start timer and run with RLO = 0
:O	F 0.0	Force RLO = 1
:ON	F 0.0	
:SE	T0	Start time with RLO = 1
WAIT:JU	OB 31	Wait loop
:A	T0	
:JC=	WAIT	
:STS		STOP if no POWER OFF in central controller
:BE		2 s after detection of PEU

2.8.7 Handling Programming Errors and PLC Malfunctions

It is possible to determine the response of the CPU to programming errors and PLC malfunctions using error response OBs.
The operation which triggers the timeout, substitution or transfer errors can be replaced by calling the relevant error response OB. Appropriate responses to errors can be programmed in these OBs. If only "BE" is programmed there, no response follows, i.e. the PLC does ***not*** go to STOP. If no relevant OB exists, the CPU jumps to the STOP mode.

OB19 Response when an unloaded block is called

In OB19, you can program the response of the CPU when an unloaded block is called. In the event of a fault, RS 102 (E 10CC$_H$) contains the machine code of the last block call. Only the second word of the machine code is stored in the case of DOU FX and DOC FX.

Example: The CPU is to go to STOP when an unloaded block is called:

OB19	Explanation
:JU FB 19	
NAME:STS	Set CPU to STOP
:BE	

FB19 STL	Explanation
:L RS 102	Evaluate error code
:T FW 0	Store in FW
:STS	Stop statement
:BE	

If OB19 has not been programmed, the control program continues execution (no response) immediately after the jump statement (the destination of the jump does not exist!)

OB23 Response to timeout in the case of direct I/O access

The following operations can result in a timeout: L PY; L PW; T PY; T PW; L OY; L OW; T OY; T OW and accesses via indirect addressing LIR, TIR, TNB, TNW, TRW, TRD, TRW, LRD, ...
The timeout (QVZ) error occurs if a module fails to acknowledge within 160 µs of being accessed. The cause may be a program error, a defect in the module or the removal of the module during the RUN mode. The operating system stores the absolute module address at which the QVZ occurred in system data word 103 and 104 (E 10CE$_H$ to E 10D1$_H$) and calls OB23. If OB23 does not exist, the CPU goes to STOP with QVZ (ISTACK error code: "QVZ").

OB24 Response to timeout when updating the process I/O image or the interprocessor communication flags

If a timeout occurs when updating the process I/O image and the interprocessor communication flags, the absolute address at which the timeout occurred is stored in system data words 103 and 104 (E 10CEH to I 10D1H) and OB24 is called. If OB24 does not exist, the CPU goes to STOP with "QVZ" (ISTACK error code: "QVZ").

OB27 Response to substitution error

A substitution error (SUF) can occur in the following cases

- If the formal parameters of a function block are changed after the block is called ("HU FBx", "JC FBx", DOU FXa", "DOC FXa").
- If the parameter is greater than the current DB length in the case of the DO DW operation.

The operating system interrupts the control program when a substitution error has been detected and then executes OB27 instead of the substitution operation. If OB27 does not exist, the CPU goes to STOP with the ISTACK error code "SUF".

OB32 Response to transfer errors

A transfer error (TRAF) occurs under the following circumstances:

- When data words are accessed without previously calling a data block (C DB, CX DX).
- If the parameter is longer than the data block opened in the case of the operations L DW; T DW; TBD; TBN D; SU D; RU D; etc.
- If there is not sufficient program memory available to generate the specified data block or a length parameter greater than FFF9$_H$ in the case of the G DB or GX DX (Generate Data Block).

Response to transfer error: The operating system interrupts processing of the operation where the transfer error occurred and processes OB32 instead. If OB32 does not exist, the CPU goes to STOP with the ISTACK error code "TRAF".

OB34 Response to the BAU (battery failure) signal

The PLC continuously checks the status of the battery in the power supply. If battery failure (BAU) occurs, OB34 is processed before each cycle until the battery has been replaced and the battery failure indicator on the power supply has been acknowledged (RESET key). You program the repsonse to battery failure in OB34. If OB34 has not been programmed, no response results.

2.9 Scan times of the CPU 945

This chapter shows you how to estimate and measure the execution time and scan time of a program.

The scan time is the sum of the run times of all operations executed by the CPU or the PLC. For example,

- Execution of a control program
- I/O accesses, e.g. process I/O image and interprocessor communication flags
- Operating system operations, e.g. updating the clock
- "Interventions in a normal program scan", e.g. by the programmer/OP functions

You can influenc the scan time by manipulation of the process I/O image transfer.
This process I/O image transfer, newly organized compared to the CPU 941 to 944, is described in Section 2.8.2.
Section 2.9.3 contains an overview of the times relevant for estimating the scan time. In the case of purely cyclic execution with interrupt processing and without direct I/O access, the scan time constitutes the main proportion of the response time.

2.9.1 Response Time in the Case of Exclusively Cyclic Program Execution

The response time is the time between change of the input signal and change of the output signal. The response time depends on the delay of the input module and the scan time.
The delay of the output modules can be ignored.
The response time in the case of exclusively cyclic program execution is being considered here, i.e. without interrupt processing or direct I/O accesses.

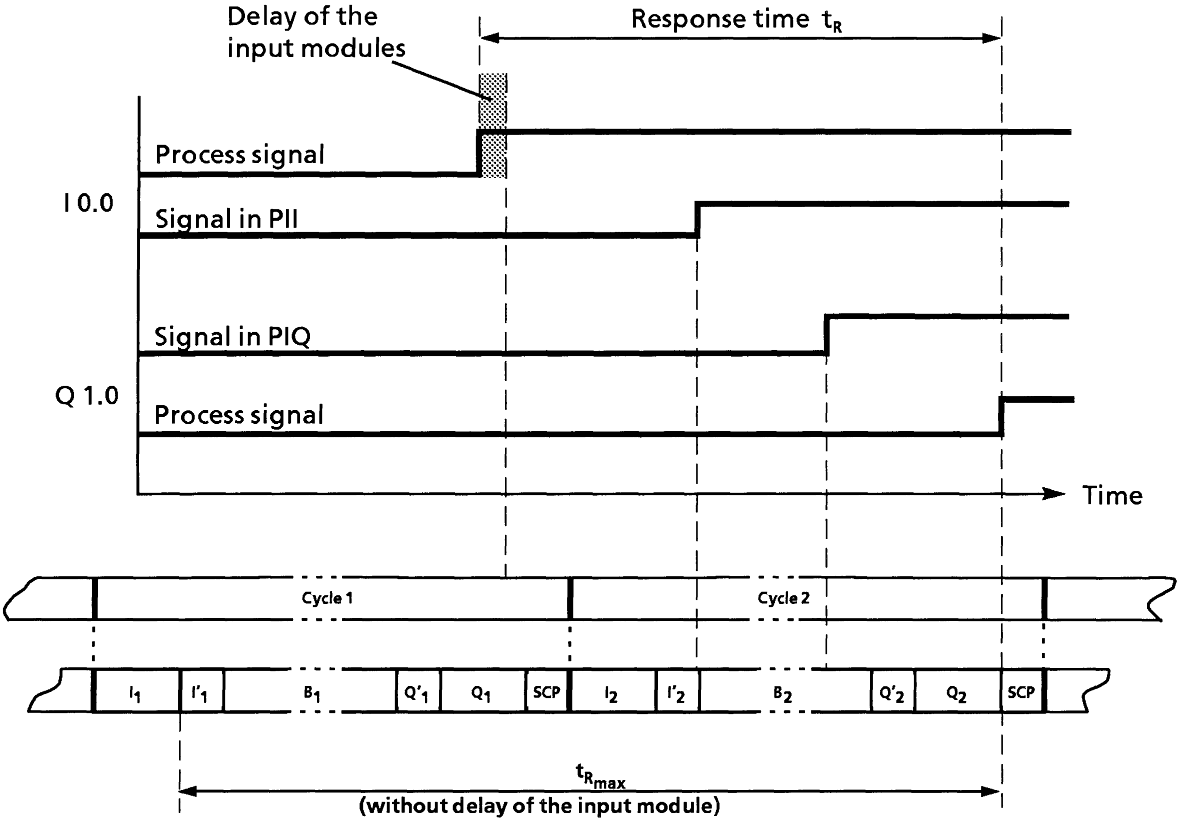
The next two figures are intended to illustrate how the response time differs between

- Sequential process I/O image transfer (see Figure 2-13)

and

- Parallel process I/O image transfer (see Figure 2-14).

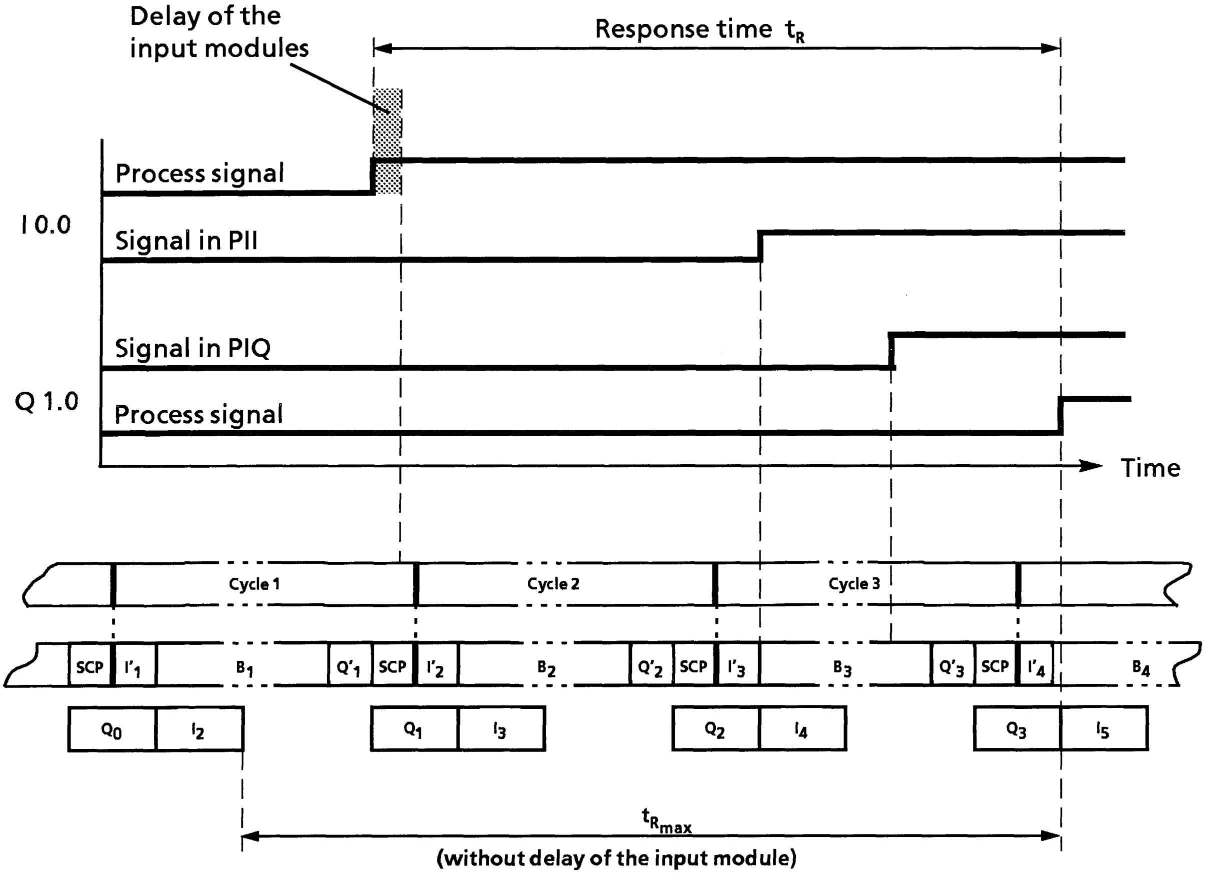
The following figure shows an example of the reponse time in the case of sequential process I/O image transfer. As well as the response time tR, the figure also shows the worst case $t_{R_{max}}$.



Legend: I ... Reading the digital input modules into the PII of the bus controller
I' ... Transferring the PII from the bus controller into the PII of the control processor
B ... Control program
Q' ... Transferring the PIQ from the control processor into the PIQ of the bus controller
Q ... Output of the PIQ from the bus controller to the digital output modules
SCP ... Scan control point
t_R ... Response time

Figure 2-13 Response Time in the Case of Sequential Process I/O Image Transfer

The following figure shows an example of the reponse time in the case of parallel process I/O image transfer. As well as the response time $_tR$, the figure also shows the worst case $t_{R_{max}}$.



Legend: I ... Reading the digital input modules into the PII of the bus controller
I' ... Transferring the PII from the bus controller into the PII of the control processor
B ... Control program
Q' ... Transferring the PIQ from the control processor into the PIQ of the bus controller
Q ... Output of the PIQ from the bus controller to the digital output modules
SCP ... Scan control point
t_R ... Response time

Figure 2-14 Response Time in the Case of Parallel Process I/O Image Transfer

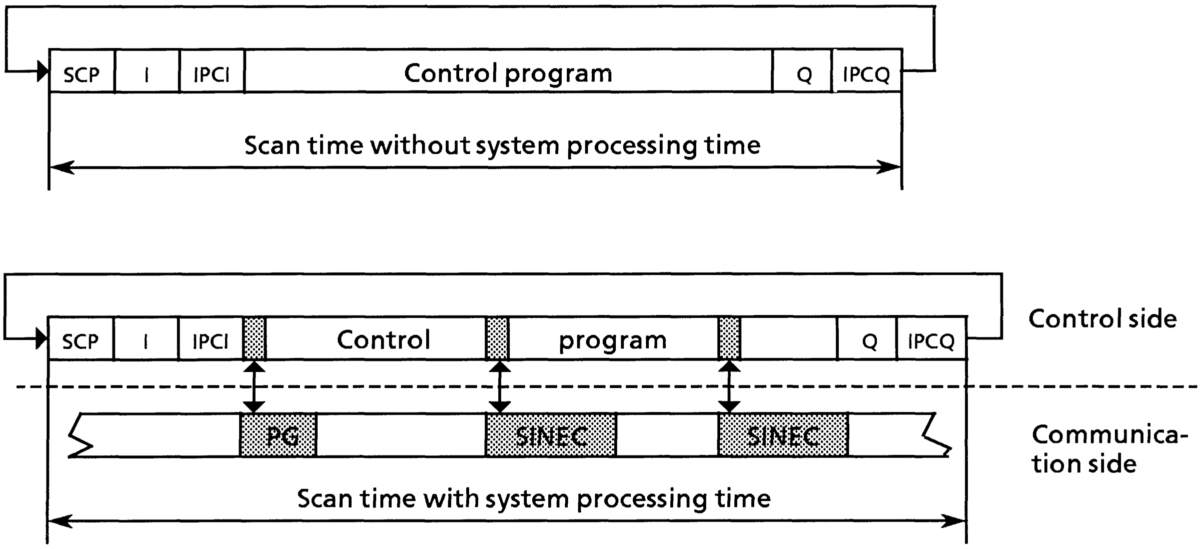
2.9.2 Estimating the Scan Time

The scan time has been divided into various units below to help you estimate the program runtime and thus the amount of time needed to scan the program.

The scan time is determined by
- The control program
  and
- All operating system operations required for executing the control program.

The scan time increases if system processing times are additionally necessary. System processing times result from
- Programmer/OP operations
  or
- Demands arising from SINEC L1, ASCII driver or computer interface.

System processing times can vary are depend on the operations executed.
System processing times can almost be ignored in the case of the CPU 945 since the actual communication functions (programmer/OP functions, etc.) are handled by the microcontroller.



SCP ...	Scan control point
I ...	Process input image
IPCI ...	Reading in the input interprocessor communication flags
Q ...	Outputting the process output image
IPCQ ...	Outputting the output interprocessor communication flags
PG ...	Programmer/OP operations
SINEC ...	SINEC L1 demands

Figure 2-15 Breakdown of the Scan Time

2.9.3 Basis for Calculating the Scan Time

The following table contains the basis for calculating the individual times of a cycle.

Table 2-9 Breakdown of the Scan Times

Service	Time in µs
Basic scan time (scan time control point)	< 50
Reading in the process input image	
I' transfer	$t_{I'} < 25$
I transfer	$t_I = 54 + n$ (1.5 + ready delay of the module)
Control program (incl. runtime of higher-priority levels, e.g. OB2)	Sum of the runtimes of all processed STEP 5 statements
Outputting the process output image	
A' transfer	$t_{A'} < 25$
A transfer	$t_A = 54 + m$ (1.5 + ready delay of the module)
A transfer (reduced)	$t_A = 54 + p$ (1.5 + ready delay of the module)
Reading in and outputting the interprocessor communication flags	Basic runtime (interprocessor communication f)lag (IPC) list valid, no entry: 50µs Additional runtime per IPC flag list word with at least one entry: 35µs Additional runtime per entry: 10µs + Ready delay of the module Runtime in the case of invalid IPC flag list: 5µs
System processing times	
Programmer/OP operations	Dependent on the functions to be executed

I'... Transfer of the PII from the PII of the bus controller into the PII of the control processor

I ... Reading the digital input modules into the PII of the bus controller

Q'... Transfer of thePIQ of the control processor to the PIQ of the bus controller

Q ... Output of the PIQ of the bus controller to the digital output modules

n ... Number of connected input bytes

m ... Number of connected output byte

p ... Number of changed output bytes

The Ready delay time is the time that elapses between the arrival of the Request signal in the module and the module's Ready signal.
The delay time depends on

- The Ready delay time of the module itself
- The interface module used and
- The length of the cable

In a distributed configuration, the communications link delay must also be taken into account.

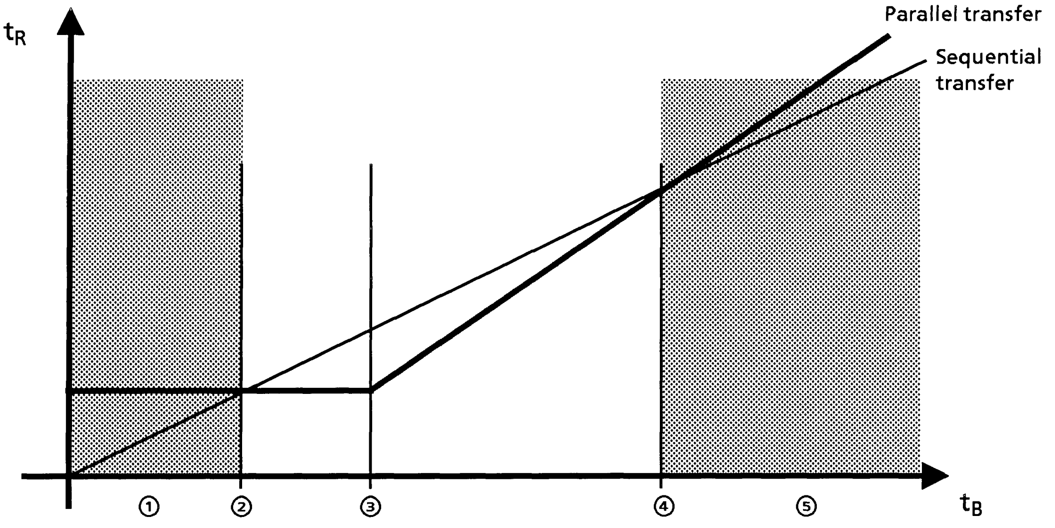
If the CPU does not receive the Ready signal within 160 µsec., it stops and outputs the "QVZ" (time-out) error message if OB23/OB24 has not been programmed (see Section 2.8.7).

Table 2-10 Ready Delays of the Different I/O Modules

I/O Modules	Ready Delays in µs
Digital modules	2
Analog modules	16
313 watchdog modules	1
IP 240	1
IP 241/241USW	1
IP 242 B	50
IP 243 (Analog module)	35
IP 244	150
IP 246	1.5
IP 247	1.5
IP 252	10
IP 281	0.5
IP 288	1.5
CP 516	1
CP 523	3 ... 100
CP 524/544	1
CP 525	3
CP 526	3
CP 527/528	3
CP 530	3 ... 130
CP 535	3
CP 551	3
CP 552	3
CP 5430/5431	1
CP 143	3
WF 705/706/707/721/723	2

Different response times in the case of sequential and parallel process I/O image transfer

The following figure shows a comparison of the response times in the case of sequential and parallel process I/O image transfer depending on the control program processing time.



① $t_B < \frac{t_I + t_Q}{2} - 40\ \mu s$

③ $t_B = t_I + t_Q - 80\ \mu s$

⑤ $t_B > 2 \times (t_I + t_Q) - 120\ \mu s$

t_R ... Response time
t_B ... Control program processing time
$t_E + t_A$ ... Process I/O image transfer

Figure 2-16 Response Times in Case of Sequential and Parallel Process I/O Image Transfer

Example:

The process I/O image transfer requires 500 µs for reading in the process I/O image and 300 µs for outputting the process I/O image to the I/O modules. These transfer times result in the following control program processing times:

- At point ②: $t_B = 360\ \mu s$
- At point ④: $t_B = 1480\ \mu s$

If the control program processing time t_B lies within the interval $360\ \mu s < t_B < 1480\ \mu s$, the response time in the case of parallel process I/O image transfer is less than that of sequential process I/O image transfer.

However, if t_B lies outside this interval, the response time in the case of parallel process I/O image transfer is greater than that of sequential process I/O image transfer.

2.9.4 Measuring the Scan Time

The scan time is measured by the CPU and stored in the system data area. You can access the current, the minimum and the maximum scan time in the control program at any time. The resolution of the scan time is one millisecond, and the range of scan time values extends from 0 to 32,767 (=7FFF$_H$) milliseconds. At the end of a scan cycle, after it has updated the process output image (PIQ) and the interprocessor communication flags, the operating system stores the scan time, i.e.:

- Current scan time in SD 121
- Maximum scan time in SD 122
- Minimum scan time in SD 123

If the scan time exceeds 32,767 milliseconds, bit 15 (which is the overflow bit) of the current scan time is set and entered in system data word SD 123 (maximum scan time). Scan time measurement begins anew in the next scan cycle.

Note

The contents of the watchdog timer are also entered in system data words 121 to 123 when the PLC stops and outputs the "ZYK" (scan time exceeded) message.

Example: Function block for measuring the scan time

FB99 STL	Description
`NAME :ZYKLUS-Z`	
`DECL :MINI      I/Q/D/B/T/C: A  BI/BY/W/D: W`	
`DECL :AKTU      I/Q/D/B/T/C: A  BI/BY/W/D: W`	
`DECL :MAXI      I/Q/D/B/T/C: A  BI/BY/W/D: W`	
`DECL :LOES      I/Q/D/B/T/C: E  BI/BY/W/D: BI`	
`: L   RS 121`	
`: T   =AKTU`	
`: L   RS 122`	
`: T   =MAXI`	
`: L   RS 123`	
`: T   =MINI`	
`: AN  =LOES`	The LOES operand is used to reset system
`: BEC`	data words 121, 122 and 123
`: L   KF +0`	(if LOES = 1).
`: T   RS 121`	
`: T   RS 122`	
`: T   RS 123`	
`: BE`	

2.9.5 Setting the Scan Monitoring Time

The scan time comprises the duration of the cyclic program. At the beginning of each program scan, the processor starts a monitoring time (cycle trigger). This monitoring time is preset to approximately 500 ms. If the scan trigger is not initiated again within this time - for example, as a result of programming an endless loop in the control program or as a result of a fault in the CPU - the PLC goes to STOP and disables all output modules. If the control program is very complex, and the monitoring time can be exceeded, you should change the monitoring time in the control program.

There are two possible methods for changing the default scan monitoring time:

- Initialization in DB1 (see Chapter 11)
  or
- STEP 5 operations.

You can set the scan monitoring time up to 2.55 s (KF= +255) without having to restart the monitoring time.

If you want to change the default scan monitoring time (approximately 500 msec.) via STEP 5 operations, you must transfer a factor for this purpose to system data word 96. The CPU operating system interprets this factor as a multiple of 10 msec.

OB31 scan time triggering

A scan time monitor monitors the program scan time. If program execution takes longer than the specified scan monitoring time (e.g., 500 msec.), the CPU enters the STOP mode

- CPU goes to STOP with "ZYK"
  or
- CPU calls the "Error OB"26 (scan time exceeded) if you have programmed an OB26.
  Calling OB26 restarts the scan time once. However, the timeout must be acknowledged with an OB31 call otherwise the CPU will go to STOP with ZYK in the event of a repeated timeout even if OB26 is programmed (see Section 2.8.6).

This situation can occur, for instance, when

- The control program is too long
- The program enters a continuous loop.

You can restart the scan watchdog at any time by calling OB31 at any point in the control program (JU OB31). The scan monitoring time is restarted by calling OB31.

The scan monitoring time can be set in the following ways:
- In system data word 96 (E 10C4)
  or
- In DB1 with the WD parameter

Please note that the ACCUs and the BR register are changed by OB31!

⚠ **Important**

Do not call OB31 in an endless loop since important operating system functions are processed at the scan control point.

2.10 Operating System Services in OB250

The operating system services allow you to:

- Activate different operating system functions or
- Change specific parameters during cyclic operation.

These functions are:

- Invocation of OB6 (collision of two timed interrupts)
- Setting of the OB10 to 13 intervals (time-controlled program execution)
- Manipulation of the process I/O image transfer
- Regeneration of the block address list
- Generating a data block (DB/DX) without TRAF
- I/O accesses and page accesses without QVZ
- Disabling and enabling the output modules (setting and resetting BASP)
- Access to DBS and DBL registers
- Indexed access to DX
- Indexed access to FX
- Cancelling a block in the block address list
- Changing the block identifier

The system data can be overwritten direct in the case of the CPUs 941 to 944.
In the case of the CPU 945, direct overwriting of the system data has no effect on the above-listed functions. For this reason, you must call these operating system services over the integral OB250.

A function number has been assigned to the operating system services. You must load this function number into ACCU 1-L. You must load service-specific parameters into ACCU 2 or ACCU 1-H.

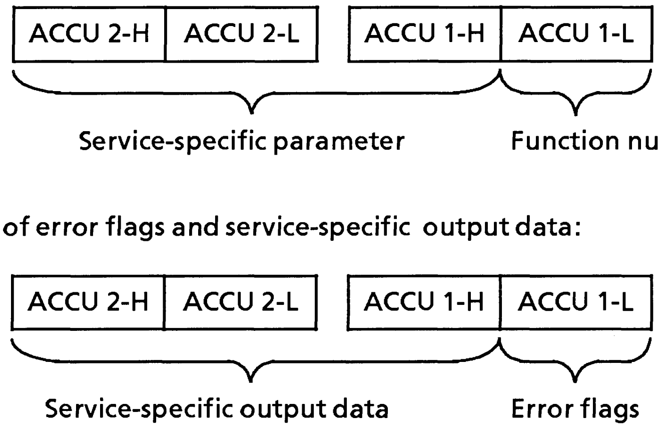
The operating system transfers error flags to ACCU 1-L.
The following are global error flags:

- 0→no error
- 1→invalid function number
- 2 to 4→reserved

Service specific error flags: 5 to FFFF$_H$

Please note that the ACCUs and the BR register are changed by OB250!

ACCU 1-H and ACCU 2 contain output data of the operating system service.



Input of the function numbers and the service-specific parameters:

Output of error flags and service-specific output data:

Figure 2-17 Assignment of the ACCUs when Inputting and Outputting Operating System Service Parameters

Operating system services

Interrupt response after timeout

The following happens when you call operating system service 1:

- The clock prompt transferred to ACCU 2-L is started
  or
- A started clock prompt is aborted.

You program the response of the program after a preset time ("timed interrupt") in OB6 (see Section 2.8.5).

Table 2-11 Operating System Service for Activating OB6

Operating System Service	Function Number	Parameter	Error Flag
Activation of OB6	1	ACCU 2-L: 0: Abort activation 1 to 65535: Time in ms	ACCU 1-L 0: No error

Time-controlled program processing

OBs 10 to 13 are available for time-controlled program execution (see Section 2.8.3).
The time OBs are called by the operating system at fixed intervals. You can assign new intervals to these time OBs over operating system services 2 to 5.

Table 2-12 Operating System Services for Time-Controlled Program Execution

Operating System Service	Function Number	Parameter	Error Flag
New interval for OB10	2	ACCU 2-L: 0: No OB10 call 1 to 65535: Interval in ms	ACCU 1-L 0: No error
New interval for OB11	3	ACCU 2-L: 0: No OB11 call 1 to 65535: Interval in ms	ACCU 1-L 0: No error
New interval for OB12	4	ACCU 2-L: 0: No OB12 call 1 to 65535: Interval in ms	ACCU 1-L 0: No error
New interval for OB13	5	ACCU 2-L: 0: No OB13 call 1 to 65535: Interval in ms	ACCU 1-L 0: No error

Changing the process I/O image transfer

During the cold restart routine, the CPU determines the I/O module complement (see Section 2.6.1) and writes this into system data words 128 to 159.
You can selectively change the entries for the module complement of the digital I/O modules (RS 128 to 143) using operating system service 6.
When transferring the process image of the digital outputs, the CPU has the option of outputting to the modules only those PIQ bytes in which the PIQ has changed (see Section 2.8.2). You can set this "reduction of PIQ transfer" in the control program with operating system service 7 "Reduction of PIQ transfer" or with the DB1 parameter RPIC (see Chapter 11).

Table 2-13 Operating System Services for Changing the Process I/O Image Transfer

Operating System Service	Function Number	Parameter	Error Flag
Changing the entries in RS 128 to 143 (digital I/O)	6	ACCU 2: Bits 0 ... 6: Number of the I/O byte (0 to 127) Bit 7: 0: Input module 1: Output module Bit 8: 0: Delete entry 1: Insert entry Bit 9 ... 31: Not evaluated	ACCU 1-L 0: No error 5: Function not currently possible (parallel PI transfer active)
Reduction of PIQ transfer	7	ACCU 2-LL: 0: "Mechanism" inactive 1 ... 255: Number of program scans with reduced PIQ transfer after one complete PIQ transfer	ACCU 1-L 0: No error
Generation of the list of all addressable I/O bytes (entries in RS 128 to 143 and RS 144 to 159)	8	None	ACCU 1-L 0: No error 5: Function not currently possible (parallel PI transfer active)

⚠ **Important**

Please note the following when using operating system service no. 8:

- The outputs are reset when operating system service no. 8 is called.
- Operating system service no. 8 must not be used when you are using the 434-7 digital input module and the 485-7 digital input/output module.

Generating a data block (DB/DX) without TRAF

Generating data blocks (DB/DX) over operating system services 10 and 11 offers the following advantages:

- If a DB/DX cannot be generated, the CPU 945 does not go to STOP with TRAF
- Error evaluation is possible immediately over the error flag in ACCU 1-L
- It is not necessary to program the error-response OB32 (response to TRAF)

Table 2-14 Operating System Services for Generating a DB/DX Without TRAF

Operating System Service	Function Number	Parameter	Error Flag
Generating a DB without TRAF	10	ACCU 2-LL: 0 to 255: DB number ACCU 2-H: 0: Delete DB 1 to FFF9$_H$: DB length up to and including DW	ACCU 1-L 0: No error 5: TRAF (as in G DB)
Generating a DX without TRAF	11	ACCU 2-LL: 0 to 255: DX number ACCU 2-H: 0: Delete DX 1 to FFF9$_H$: DX length up to and including DW	ACCU 1-L 0: No error 5: TRAF (as in GX DX)

Regenerating the block address list

The block address list is regenerated after calling operating system service 12.

Note

Use of operating system service 12 leads to an increase in the scan time of up to several 100 ms (see Pocket Guide). If necessary, the scan time must be restarted with OB31.
If an error is detected during generation of the address list, the CPU goes to STOP with an Overall Reset request.

Table 2-15 Operating System Services for Regenerating the Block Address List

Operating System Service	Function Number	Parameter	Error Flag
Regenerating the block address list	12	None	ACCU 1-L 0: No error

I/O accesses and page accesses without QVZ

I/O accesses and page accesses over operating system services 13 to 18 offer the following advantages:

- If a timeout (QVZ) occurs, the CPU 945 does not go to STOP
- Error evaluation is possible immediately over the error flag in ACCU 1-L
- It is not necessary to program the error-response OB23 (response to QVZ)

In the event of a QVZ, operating system services 13 to 18 enter the QVZ addresses in system data words RS 103 and 104.

Table 2-16 Operating System Services for I/O Accesses and Page Accesses Without QVZ

Operating System Service	Function Number	Parameter	Error Flag
Read address (byte) from S5 bus	13	ACCU 2-L: 0 to FFFF$_H$: Address	ACCU 1-L: 0: No error 5: QVZ ACCU 2-LL: Byte read
Write address (byte) to S5 bus	14	ACCU 2-L: 0 to FFFF$_H$: Address ACCU 2-HL: 0 to FF$_H$: Byte to be written	ACCU 1-L: 0: No error 5: QVZ
Read address (word) from S5 bus	15	ACCU 2-L: 0 to FFFE$_H$: Address	ACCU 1-L: 0: No error 5: QVZ 7: Invalid address ACCU 2-L: Word read
Write address (word) to S5 bus	16	ACCU 2-L: 0 to FFFE$_H$: Address ACCU 2-H: 0 to FFFF$_H$: Word to be written	ACCU 1-L: 0: No error 5: QVZ 7: Invalid address

Table 2-16 Operating System Services for I/O Accesses and Page Accesses Without QVZ

Operating System Service	Function Number	Parameter	Error Flag
Read byte from page	17	ACCU 2-L: 0 to 07FF$_H$: Offset within the page ACCU 2-HL: 0 to FF$_H$: Page number	ACCU 1-L: 0: No error 5: QVZ when selecting the page 7: QVZ when accessing the page 9: Invalid offset in the page ACCU 2-LL: Byte read
Byte to be written	18	ACCU 2-L: 0 to 07FF$_H$: Offset within the page ACCU 2-HL: 0 to FF$_H$: Page number ACCU 2-HH: 0 to FF$_H$: Byte to be written	ACCU 1-L: 0: No error 5: QVZ when selecting the page 7: QVZ when accessing the page 9: Invalid offset in the page

Disabling and enabling digital outputs

Table 2-17 Operating System Service for Disabling Digital Outputs

Operating System Service	Function Number	Parameter	Error Flag
Disabling and enabling digital outputs (setting and resetting BASP)	19	ACCU 2-L: Bit 0: 0: Reset BASP 1: Set BASP	ACCU 1-L 0: No error

Important

Service no. 19 must not be used if the 434-7 digital input module and the 485-7 digital input/output module are being used.

Access to DBS and DBL Registers

Operating system services 20 to 23 can be used to read/write in the DBS and DBL registers. Note the following, however:

- Basically, the DBS register contains only even addresses.
- The DBS register has a width of 24 bits. When reading the DBS register, the register contents are stored in bits 0 to 23 of ACCU 2; bits 24 to 31 have the value zero. For writing, only bits 0 to 23 of ACCU 2 are relevant.
- If the DBS register is set to addresses outside the range of the user RAM, subsequent access operations via the DBS register (TDW, LDW, ...) might result in errors such as QVZ or FAD. No plausibility check is carried out for the address by the operating system service.

Table 2-18 Operating System Services for Access to DBS and DBL Registers

Operating System Utility	Function Number	Parameter	Error Message
Read DBS register Exec. time: 25 µs	20	None	ACCU 1-L 0: No error ACCU 2 Contents of DBS reg.
Write DBS register* Exec. time: 26 µs	21	ACCU 2: 0 ... FFFFFE_H: address	ACCU 1-L 0: No error 5: Invalid address (Bit 0 ≠ 0)
Read DBL register Exec. time: 25 µs	22	None	ACCU 1-L 0: No error ACCU 2-L Contents of DBL reg.
Write DBL register Exec. time: 25 µs	23	ACCU 2-L: 0 ... FFFF_H	ACCU 1-L 0: No error

* If you set the DBS register to an address between 020000_H and 0205FF_H, subsequent execution of an "MBA" command may result in data falsification in this range. In order to avoid this effect, use the command sequence "ABR+0, MBA" instead of the "MBA" command.

Indexed Access to DX

Operating system service 24 can be used for indexed access when opening a DX block (replacement for DO DW/DO FW, CX DX). Note the following, however:

- If the DX to be opened does not exist at all, the DB/DX opened before is closed (DBS and DBL registers are initialized with 0).

Table 2-19 Operating System Utilities for Indexed Access to DX

Operating System Utility	Function Number	Parameter	Error Message
Indexed access to DX Exec. time: 29 µs	24	ACCU 2-LL: 0 to 255: DX number	ACCU 1-L 0: No error 5: DX not available

Indexed Access to FX

Operating system service 25 can be used for indexed access when opening an FX block (replacement for DO DW/DO FW, DOU DX). Note the following, however:

- If the FX to be called does not exist at all, the FX call is suppressed.
- The FX block is not accessed until OB 250 is completed. The ACCUs, the STATUS register and the BR register have been modified. When the FX is called, however, the same DB/DX as in the block calling the operating system service is opened.
- Evaluation of the feedback from the operating system service is possible only with certain restrictions as the FX called might have modified the ACCU contents.

Table 2-20 Operating System Utilities for Indexed Access to FX

Operating System Utility	Function Number	Parameter	Error Message
Indexed access to FX Exec. time: 30 µs	25	ACCU 2-LL: 0 to 255: FX number	ACCU 1-L 0: No error 5: FX not available

Cancelling a Block in the Block Address List

Operating system service 26 can be used to cancel a block in the block address list. Note the following, however:

- A user block is retained as valid block in the RAM, i.e. it is re-entered in the address list the next time the block address list is regenerated, and for compressing (only when the cancelled block is shifted in the memory).
- The operating system service can also be used for cancelling integrated blocks in the block address list. Integrated blocks are only re-entered in the address list when regenerating the block address list but **not** when compressing.
- The operating system considers a block as being non-existent while it is cancelled in the block address list.

Table 2-21 Operating System Services for Cancelling a Block in the Block Address List

Operating System Utility	Function Number	Parameter	Error Message
Cancel block in the block address list Exec. time: 28 µs	26	ACCU 2-LL: 0 to 255: Block number ACCU 2-LH: Block type: 1: PB 2: SB 3: FB 4: FX 5: DB 6: DX 7: OB	ACCU 1-L 0: No error 5: Invalid block type 7: Block does not exist

Changing the Block Identifier

Operating system services 27 and 28 can be used to change the block identifiers in the RAM to "valid in EPROM" or "valid in RAM".
Note the following, however:

- The block identifier is changed only in blocks entered in the block address list.
- When changing the identifier to "valid in EPROM", the identifiers of DBs and DXs are **not changed.**
- **Changing of the identifier to "valid in RAM" changes the identifiers of all user blocks.**
- **The identifiers of integrated blocks are not** changed.

Table 2-22 Operating System Services for Changing Block Identifiers

Operating System Services	Function Number	Parameter	Error Message
Change block identifier to "valid in EPROM" Execution time: 4.8 ms + n · 19 µs n = Number of blocks to be modified	27	None	ACCU 1-L 0: No error
Change block identifier to "valid in RAM" Execution time: 6 ms + n · 19 µs n = Number of blocks to be modified	28	None	ACCU 1-L 0: No error

Please refer to Section 4.3.2 in the manual for further information on transferring the user program to the CPU.

Operating system service call example

The following example shows the principle of operation with OB250.

A reduced PIQ transfer is to be executed.
The PIQ is only to be output every fifth program scan. For the four program scans between, only the output bytes which have changed will be updated.

2

STL	Explanation
:L KB 4	4 program scans with reduced PIQ transfer
:L KB 7	Load the function number of the operating system service
:	
:	ACCU1: 0000 0007_H
:JU OB 250	ACCU2: 0000 0004_H
:L KB 0	Call operating system service
:!= F	Load code for "No error"
:BEC	Checkback signal "No error"?
:	YES: Program end
:STS	NO: Error evaluation (e.g. CPU to STOP)
:BE	

2.11 Further CPU 945 Functions in Integral Blocks

This chapter is concerned with integral blocks that you call in the control program for special functions.

These functions are:

- Compressing the program memory (FB238 "COMPR")
- Deleting a block (FB239 "DELETE")
- Generate a block (OB125)
- Variable time loop (OB160)
- Copying data areas (OB182)
- Duplicate DX or DB blocks (OB183 and OB184)
- Transferring flags to a data block (OB190 and OB192)
- Transferring data blocks to flag areas (OB191 and OB193)
- Extension for sign (OB220)
- Reading the digital inputs into the PII (OB254)
- Outputting the PIQ to the digital outputs (OB255)
- PID Control Algorithm. (OB251)

More integrated CPU 945 blocks can be found in:

- Setting the Scan Monitoring Time (OB31), see Chapter 2.9.5
- Operating System Services (OB259), see Chapter 2.10
- Analog Value Matching Blocks (FB241, FB242, FB243, FB250), see Chapter 10.10
- Outputting an Analog Value (FB251), see Chapter 10.10.5
- Data Interchange over Data Handling Blocks FB244 to FB249), see Chapter 12.2.3

Note

The integral modules change the ACCUs in the BR register.

2.11.1 Compressing the Program Memory with FB238 "COMPR"

The integral "COMPR" block (no. 238) compresses the internal program memory.

Calling the function block

Calling FB238 in the control program has the effect of executing the "Compress PLC" function. This function block signals back with the "AKT" bit whether this function is still active. When compressing is finished, the "AKT" bit signals back independently of FB238. This signal takes place in the "AKT" bit of the first FB238 call.
The "ERR" bit signals that the "Compress PLC" function cannot be executed.
Compressing is restarted at every FB238 call whenever compressing is not activated.

Compressing is terminated
- If there are no further blocks to be shifted
  or
- If an invalid block synchronization pattern is detected.

In the case of an invalid block synchronization pattern, the CPU goes to STOP with an "Overall reset request".

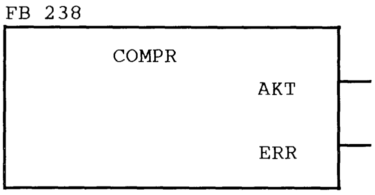
<table>
<tr><th>STL</th><th>CSF/LAD</th></tr>
<tr><td><pre>
      :A   I   0.0
      :AN F   0.0
      :=   F   0.1
      :A   I   0.0
      :=   F   0.0
      :
      :A   F   0.1   Edge flag
      :JC FB 238
NAME :COMPR
AKT   :F   1.0
ERR   :F   1.1
</pre></td><td>
</td></tr>
</table>

Table 2-23 FB238 Flags

Bit	Meaning
AKT	"Compress PLC" function is active
ERR	"Compress PLC" function cannot be executed

> **Note**
>
> The FB COMPR has the same effect as the programmer "Compress" function, i.e. if FB COMPR is active, other programmer/OP functions will be rejected, e.g. STATUS or block input/output.

2.11.2 Deleting a Block with FB239 "DELETE"

The integral FB "DELETE" (No. 239) deletes blocks.

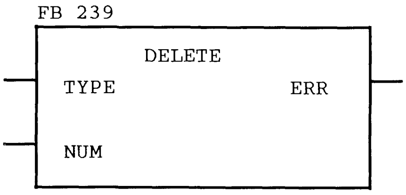
Parameterizing the Integral FB DELETE

Parameterize the integral FB239 as follows:

- Store the type of the block to be deleted in an input word, flag word or data word as an ASCII character (KS). The characters OB. PB, FB, SB and DB are permissible as block identifiers.
- Store the block number in an input byte or a flag byte.

You must also specify a flag byte or an output byte which the operating system can use to flag errors (see Table 2-24).

Calling the Function Block (Example)

STL	CSF/LAD
:JU FB 239 NAME :DELETE TYPE :FW 5 NUM :FY 7 ERR :FY 8 :***	

Contents of FW 5: Block type in ASCII code (e.g. PB for program block)
Contents of FY 7: Block number (e.g. KF+7)
Contents of FY 8: No entry is made in FY 8 until the function block has been invoked
(see Table 11-18)

Table 2-24 Error Bits Set by FB239 (ERR Parameter)

Hexadecimal Value of the ERR Parameter	Description
00	No error
F0	No such block
F1	Invalid block type specified in the TYPE parameter
F2	Block exists, but has an EPROM identifier
F4	DELETE function cannot execute because another function is in progress (e.g. a programmer function)

2.11.3 Generating STEP 5 Blocks: OB 125

Function

The OB 125 can be used to generate any STEP 5 blocks (code and data blocks) in the user memory. The generation of code blocks, however, should be left to specialists.

The specified block is generated in the internal RAM with block header and block body and then entered in the block list. The block body contains the BEU statement in the first word and the BE statement in the last word. In between the two words, any data can be stored. For this reason, data must first be written in a newly generated block before it can be employed for processing any useful statements.

Execution time: 41 µs

Parameters

1. ACCU 1-LH:	Type of the block to be generated
2. ACCU 1-LL:	Number of the block to be generated

Permissible types and numbers of blocks

Table 2-25 Permissible Types and Numbers of Blocks

ACCU 1-LH (Block Type)	ACCU 1-LL (Block Number)
1 = PB	0 to 255
2 = SB	0 to 255
3 = FB	0 to 255
4 = FX	0 to 255
5 = DB	0 to 255
6 = DX	0 to 255
7 = OB	0 to 255

3. ACCU 2-L:	Number of words (desired block length without block header). Block lengths of 1 to 65530 (1 to FFFA$_H$) words can be parameterized.

Result:

If a block has been processed correctly and without any errors, the system program sets the RLO to "0" and deletes condition codes CC 1 and CC 0.

Note

During generation of a block, the user interrupts are blocked. Neither timed interrupts nor process interrupts are registered.

Errors and Warnings

In the case of an error, the system program aborts processing of OB 125 and continues program execution at the next STEP 5 operation. It also sets the RLO to "1" and enters an identifier in ACCU 1-LL (see Table 2-26).

When a function is aborted with a warning, a repeated call of the special function (if necessary, several times) may possibly achieve a correct execution of OB 125.

Condition Codes

After calling OB 125, you can read the result of the logic operation and condition codes CC 1 and CC 0 to check whether the special function has been processed properly or aborted with an "error" or "warning". The result can be evaluated by means of a conditional jump.

Evaluation of Condition Codes

Table 2-26 Condition Codes of OB 125

RLO	CC 1	CC 0	Description of Error	Check
0	0	0	Special function has been processed correctly	JB JZ
1	1	0	Special function has been aborted with a "warning"	JB JP JN
1	0	1	Special function has been aborted with an "error"	JB JM JN

Identifiers in ACCU 1-LL

The system program uses ACCU 1-LL for storing descriptions of the causes of a warning or error in identifiers for the result of the logic operation.

Table 2-27 Condition Codes of the OB 125 in ACCU 1-LL

Identifier	Meaning
01_H	Function has not been processed correctly
 42_H 43_H 44_H 45_H	Error: Block already exists Insufficient memory space Block length = 0 or > $FFFA_H$ Block type not permissible
 $8D_H$	Warning: Conflict with online function

Example:

STL	Explanation
:L KF +2000 :L KY 5,24 :JU OB 125 : :	THIS SEQUENCE OF OPERATIONS GENERATES THE DB 24 WITH A LENGTH OF 2000 DATA WORDS (TOTAL LENGTH INCLUDING HEADER: 2005 WORDS)

2.11.4 Variable Time Loop with OB160

OB160 simulates operation runtimes. You can use this to program waiting times of up to 65.5 ms.

Proceed as follows:

The waiting time must be loaded into the ACCU in µsec. (range: 3 to 65535 or 3_H to $FFFF_H$).

Example: A waiting time of one millisecond is to be programmed.

```
L   KF +1000
JU OB 160.
```

Please note the following when programming OB160:

A process interrupt (OB2 to OB5) and the timed interrupt (OB6) can interrupt the waiting time (provided no interrupt disable (IA) has been programmed). The waiting time stops during the interrupt! Similarly, the waiting time is increased by running PG/OP operations. The times set are therefore minimum times!

2.11.5 Copying Data Area: OB 182

OB 182 copies a data frame of variable length from one data block to another. Both DB and DX blocks can be used as source and destination blocks. The beginnning of the frame can be programmed in the source and destination blocks. OB 182 may be up to 65530 ($FFFA_H$) data words long.

> **Note**
>
> The source and destination blocks can be identical and the data frames of source and destination may overlap. The **original data** of the source area are copied to the destination area in their original form even if the areas overlap. After the copy function, the **area of the source that overlaps with the destination area is overwritten**. You may use this feature for shifting a data area within a block.

Before calling OB 182, you have to assign the parameters necessary for the copy function to a data field. The data field can be created in the F or S flag area of either a DB or DX block. However, you still have to specify the location of the data field in ACCU 1-L and ACCU 2-L before calling OB 182.

Data Field with Copy Parameters

The data field designates the source and destination blocks, the start address of the frames in both blocks and the number of data words to be transferred. It consists of five words.

Bit No.	**15 ... 8**	**7 ... 0**
1st word	Source DB type	Source DB No.
2nd word	No. of 1st source DB word to be transferred	
3rd word	Destination DB type	Destination DB No.
4th word	No. of 1st word to be written in destination DB	
5th word	Number of data words	

The parameters have the following meanings and permissible value ranges:

Data block type (source and destination)	1 = DB 2 = DX
Data block number (source and destination)	0 to 255
No. of 1st data word (source and destination)	0 to 65529 (0 ... $FFF9_H$)
Number of data words:	1 to 65530 (1 ... $FFFA_H$)

If you create the data field in the flag area, enter the number of the first flag word in ACCU 1-L when calling OB 182.

Execution time: 35 µs + n . 2.65 µs (n: number of data words to be transferred)

Defining the Location of the Data Field and Calling OB 182

Parameters:

1. Data field: Before calling OB 182, you have to enter the relevant copy parameters in the data field as described above.

Bit No.	15 8	7 0
2. ACCU 2-L:	Address area type	Data block No.

Address area type, permissible values:	1 = DB block 2 = DX block 3 = F flag area 4 = S flag area
Data block No., permissible values:	DB: 0 to 255; DX: 0 to 255 irrelevant for F and S flags

3. ACCU 1-L: Number of first data field word; permissible values (depending on the type of address range):

DB, DX:	0 to 65525
F flag:	0 to 246 (= No. of flag word)
S flag:	0 to 4086 (= No. of flag word)

Possible Reactions in the Case of Errors
In both cases, error identifiers are entered in ACCU 1-L and ACCU 2-L (see Table 2-28).

Table 2-28 Error Identifiers

ACCU 1-L	ACCU 2-L	Causes of Errors
$1A06_H$	-	Data block not loaded (source or destination not existing)
$1A34_H$	0001_H	Description of data field invalid
	0100_H	Illegal address area type
	0102_H	"Number of first word in data field" illegal
	0200_H	Illegal "source data block type"
	0203_H	Length of source data block in block header < 5 words
	0210_H	Illegal "destination data block type"
	0213_H	Length of destination data block in block header < 5 words
	0220_H	Illegal "number of data words to be transferred" (=0 or >65530)
	0221_H	Source data block too short
	0222_H	Destination data block too short

If the block is processed correctly, ACCU 1 has the value 0.

2.11.6 Duplicating DX or DB Blocks: OB 183 and OB 184

The OB 183 and OB 184 organization blocks for special functions are used for duplicating a data block in the memory of the CPU. OB 183 handles DX blocks whereas OB 184 is used to copy DB blocks.
The source DB is not manipulated by the copy function and remains valid.
Execution time: 52 µs + n · 2.35 µs (n: number of data words in DB/DX).

Parameters:

ACCU 1-LL:	Number of source block; permissible range: 0 to 255
ACCU 1-HL:	Number of destination block; permissible range: 0 to 255

If the block is processed correctly, the RLO register is set to 0 and ACCU 1 has the value 0.

In the case of an error, however, the RLO = 1 and an error code is entered in ACCU 1.

Possible Errors:

Table 2-29 Possible Errors

Error Code	Description of Error
1	Destination block already existing
2	Source block does not exist
3	Destination block cannot be generated (free memory space insufficient!)

2.11.7 Transferring Flags to Data Blocks: OB 190 and OB 192

The OB 190 and OB 192 organization blocks can be used for transferring a certain number of flag bytes specified by the user to a particular data block.
This can be useful prior to a block call, in error organization blocks or when cyclic program execution is interrupted by time or interrupt-controlled program execution.
The OB 191 and OB 193 organization blocks can be used to transfer the flag bytes back to the data block.

Note

- Use OB 192 and OB 193 for saving and reloading flag bytes to achieve considerably shorter execution times.
- Prior to calling OB190/192, the respective data block (DB/DX) must be opened!
- OBs 190/192 can be used for transfer to a data block **from the F flag area only** but **not** from the **S flag area**.

After calling OBs190/192, the flag bytes are entered starting from the data word address specified for the data block opened. OBs 190/192 read the flag area to be saved in ACCU 2.
OB 190 and OB 192 are identical apart from the manner in which the flag bytes are transferred:

- OB 190 transfers flags byte by byte.
- OB 192 transfers flags word by word.

This is relevant for cases in which data transferred to a data block is to be processed subsequently and the data block is not merely used for intermediate storage.

The diagram below illustrates the difference:

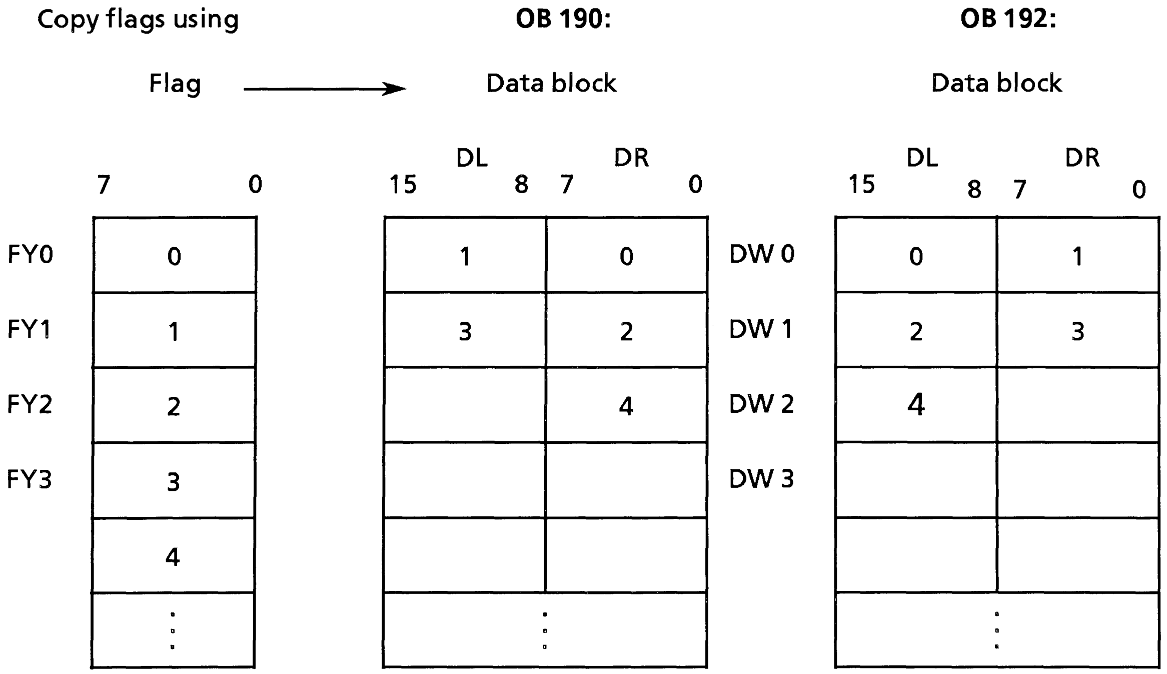


Figure 2-18 Byte-By-Byte (OB 190) and Word-By-Word (OB 192) Transfer

Note

If an **odd** number of flag bytes is transferred, only **one half** of the **last** data word accessed in the data block is used. In the case of OB 190, the left data byte remains unchanged in the destination DB, whereas, in the case of OB 192, the right data byte remains unchanged.

Execution times:

OB 190: 13 µs + n · 1.85 µs	(n: number of bytes)
OB 192: 12 µs + n · 1.3 µs	(n: number of bytes)

Parameters:

Source specifications:

1. ACCU 2-LH: First flag byte to be transferred; permissible range: 0 to 255

2. ACCU 2-LL: Last flag byte to be transferred; permissible range: 0 to 255

(Last flag byte ≥ First flag byte)

Destination specifications:

3. ACCU 1-L: Number of the first data word to be written in the open data block. The permissible values depend on the length of the data block in the memory.The values may thus also be > 255
0 to 65529 (0 to FFF9$_H$)

If the special function OBs 190/192 are processed **correctly**, the RLO is deleted (RLO = 0).

The RLO is set (RLO = 1) in the case of an error.

Possible Errors:

- No DB or DX block open
- Wrong flag area (last flag byte < first flag byte)
- No data word number available
- Length of DB or DX block insufficient

2.11.8 Transferring Data Blocks to Flag Area: OB 191 and OB 193

Organization blocks OB 191 and OB 193 can be used to transfer data from a data block to a flag area. In this way, for instance, flag bytes previously "saved" into a data block can be written back into the flag area.

OBs 191/193 differ from OBs 190/192 in as far as source and destination are exchanged:

OB 190/192: Flag area $\xrightarrow{\text{Flags}}$ Data block

OB 191/193: Flag area $\xleftarrow{\text{Data}}$ Data block

Note

A **sufficiently long** data block (DB/DX) must be opened before calling OB 191/193! OBs 191/193 transfer flags from the data block **only into the F flag area** but not into the **S flag area.**

OB 191 and OB 193 are identical apart from the manner in which the data is transferred:

- OB 191 transfers data words byte by byte.
- OB 193 transfers data words word by word.

The diagram below illustrates the difference.

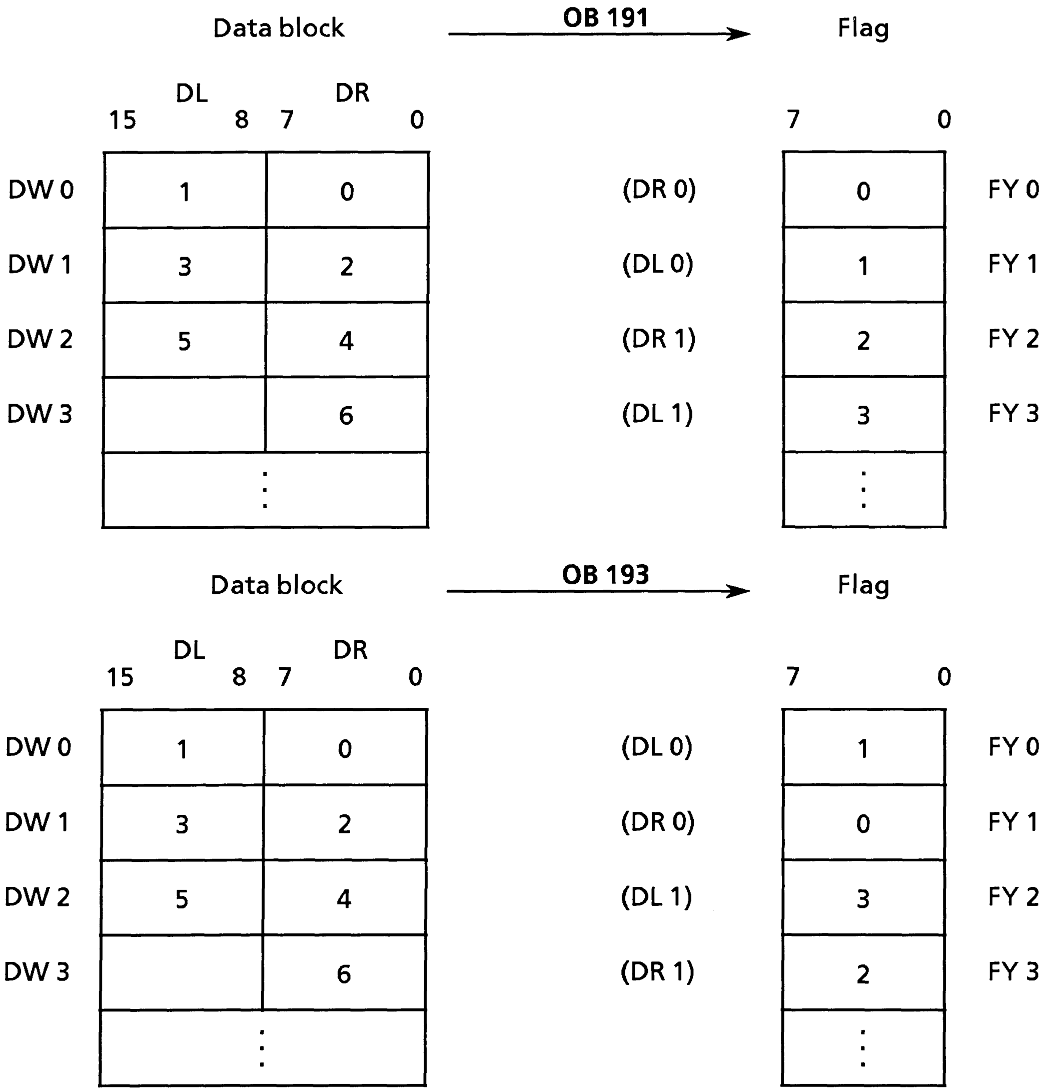


Figure 2-19 Byte-By-Byte (OB 191) and Word-By-Word (OB 193) Transfer

Parameters:

Source specifications:

1. ACCU 2-L: Number of the first data word to be transferred in the open data block: 0 to 65529 (0 to FFF9_H)

Destination specifications:

2. ACCU 1-LH: First flag byte to be written; permissible range: 0 to 255

3. ACCU 1-LL: Last flag byte to be written; permissible range: 0 to 255

(Last flag byte ≥ First flag byte)

If the special function OBs 191/193 are processed **correctly**, the RLO is deleted (RLO = 0).

The RLO (RLO = 1) is set in the case of an error.

Possible Errors:

- No DB or DX block open
- Wrong flag area (last flag byte < first flag byte)
- No data word number available
- Length of DB or DX block insufficient

Example 1:
Prior to calling program block PB 12, all flags (FY 0 to FY 255) are to be saved into data block DX 37 from address 100 onwards and then written back to the flag area.

STL	Explanation
	SAVE:
:CX DX 37	CALL DATA BLOCK
:L KY 0,255	FLAG AREA FROM FY 0 TO FY 255
:L KB 100	NUMBER OF FIRST DATA WORD IN DESTINATION
:JU OB 190	SAVE FLAGS
	CHANGE TO OTHER BLOCK:
:JU PB 12	
	WRITE BACK TO ORIGIN:
:	(DATA BLOCK ALREADY CALLED UP)
:L KB 100	NUMBER OF FIRST DATA WORD IN SOURCE
:L KY 0,255	FLAG AREA FROM FY 0 TO FY 255
:JU OB 191	WRITE FLAGS BACK TO ORIGIN

Execution times:

OB 191: 13 µs + n · 1.85 µs	(n: number of bytes)
OB 193: 12 µs + n · 1.3 µs	(n: number of bytes)

Example 2 for OB 190/OB191:
Flags used for cyclic execution of the user program cannot be used by a time or interrupt-controlled user program at the same time. To each program execution level, a certain section of the flag area must be assigned.

Example:

● Cyclic user program:	FY 0	....	FY 99
● Time-controlled user program:	FY 100	....	FY 199
● Interrupt-controlled user program:	FY 200	....	FY 255

If, however, the cyclically processed user program uses all of the 256 flag bytes and, for instance, the time-controlled user program also requires all of the 256 flag bytes assigned to it, all flags must be changed and buffered when changing to the next execution level.

The fastest way of saving and loading flags is by means of the special functions provided by OB 192 and OB 193. Figure 2-20 shows how the flag area FY x to FY y, which is commonly used by OB 1 and OB 13 (100 ms timed interrupt), is buffered in a data block DB z.

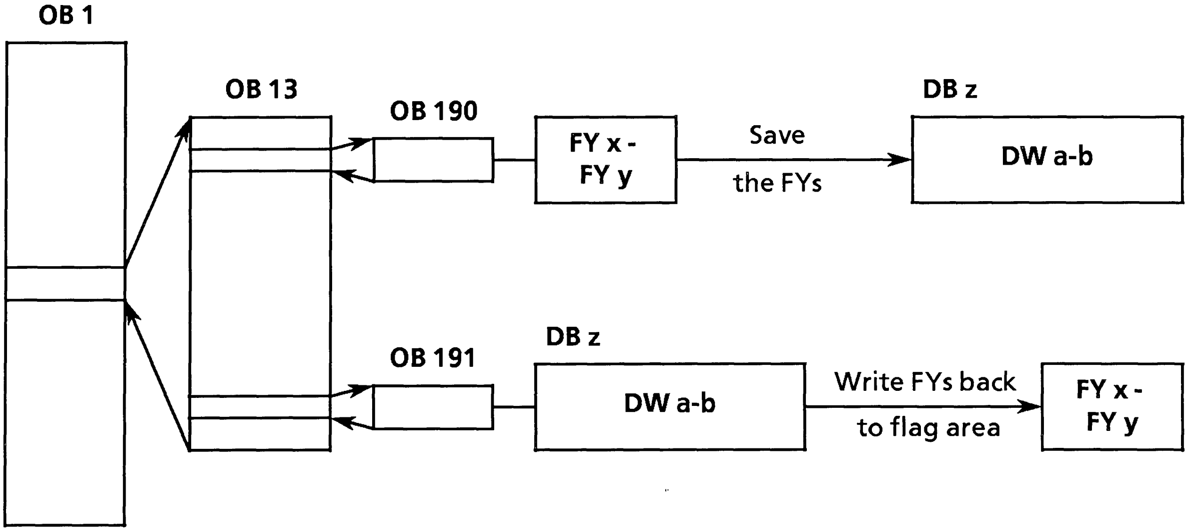


Figure 2-20 Saving Flag Areas when Changing the Program Execution Level

Caution

Coordination bytes CBS/CBR (flag bytes or DL) defined for SINEC L1, ASCII driver or computer link may not be saved or reloaded since CBS/CBR are controlled by the operating system independent of the user program cycle.

STEP 5 Program in OB 13

STL	Explanation
:C DB 100	OPEN DATA SAVE DB
:L KY 0,255	DEFINE FLAG AREA FROM 0 TO 255
:L KB 0	FLAG BYTES ARE STORED IN DB FROM DW 0 ONWARDS
:JU OB 190	TRANSFER FLAGS BYTE BY BYTE TO DB
:L KB 128	LOAD NEW DATA FROM DB FROM DW 128 ONWARDS
:L KY 0,255	FLAG AREA FROM 0 TO 255
:JU OB 191	TRANSFER DATA BYTE BY BYTE TO FLAG AREA
:	
:	Program
:	
:C DB 100	OPEN DATA SAVE DB
:L KY 0,255	SAVE FLAG BYTES FROM 0 TO 255 BYTE BY BYTE
:L KB 128	IN DB FROM DW 128 ONWARDS
:JU OB 190	
:L KB 0	RELOAD DATA BYTE BY BYTE FROM DW 0 OF DB
:L KY 0,255	INTO FLAG AREA FROM 0 TO 255
:JU OB 191	
:BE	BLOCK END

Other Applications of OBs 190 to 193

- High-order byte and low-order byte can be exchanged in the data block without great effort by transferring data words to the flag area and reloading them in the DB, as shown in Figure 2-21.
- Data frames can be "shifted" within a data block if you specify the same DB No. but a different data word when reloading the data from the flag area.

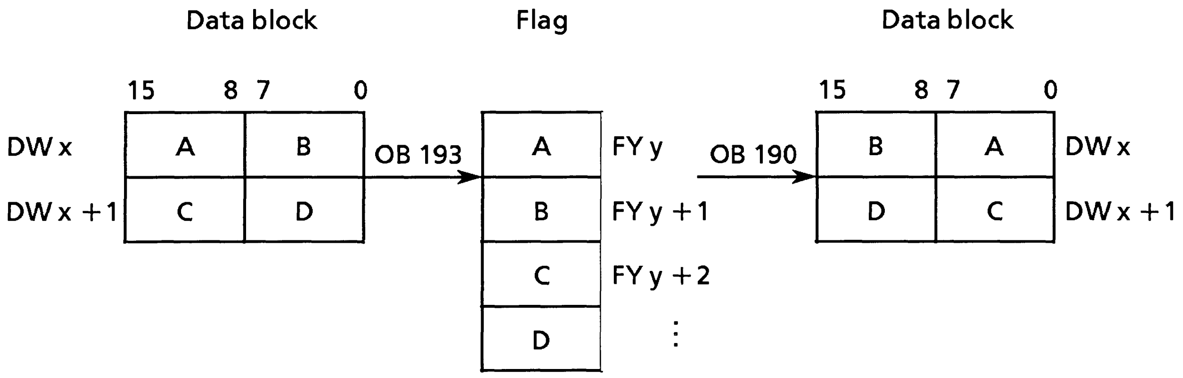


Figure 2-21 Exchanging High-Order Byte and Low-Order Byte in a DB Using OB 193/OB190

2.11.9 Extension for Sign: OB 220

This special function extends the sign of a 16-bit fixed-point number in ACCU 1-L to the higher-order word (ACCU 1-H):

- If bit $2^{15} = 0$ (positive number), KH 0000 is loaded in the higher-order word.
- If bit $2^{15} = 1$ (negative number), KH FFFF is loaded in the higher-order word.

This extension is necessary for converting a negative 16-bit fixed-point number into a 32-bit fixed-point number prior to a fixed-point/floating-point conversion (32-bit, FDG statement).

Parameter:

1. ACCU 1-L:	16-bit fixed-point number

Possible Errors: None

Execution time: 3 µs

2.11.10 Reading the Digital Inputs Into the Process Image of the Inputs with OB254

Calling OB254 (JU OB254 or JC OB254) causes the digital inputs to be written new into the process image of the inputs (PII).

When OB254 is called, the digital inputs are read in new at the call point. This takes place regardless of whether bits 1 to 7 in system data word 120 are set. These bits are responsible for enabling cyclic reading in or parallel process I/O image transfer (see Section 2.8.2, 6.3.1, 6.3.2).

Please note the increase in the runtime of OB254 when parallel process I/O image transfer is set.

If OB254 is called

- while a parallel PII transfer is running, the system waits until this is terminated and the result of the PII transfer has been entered in the PII.
- while a parallel PIQ transfer is running, the system waits until this is terminated and until the PII transfer has subsequently been executed.

2.11.11 Sending the Process Image of the Outputs to the Digital Outputs with OB255

Calling OB255 causes the process image of the outputs (PIQ) to be sent to the digital outputs.

In OB255, the PIQ is alwys sent to the I/Os at the call point. This takes place regardless of whether bits 2 to 7 in system data word 120 are set. These bits are responsible for enabling cyclic output of the process image of the outputs (see Section 2.8.2, 6.3.1, 6.3.2).

Please note the increase in the runtime of OB255 when parallel process I/O image transfer is set.

If OB255 is called

- while a parallel PIQ transfer is running, the system waits until this is terminated and until the PIQ has been output again
- while a parallel PII transfer is running, the system waits until this is terminated and until the PIQ transfer has subsequently been executed.

2.11.12 PID Control Algorithm: OB 251

The operating systems of the central processing units have an integral PID control algorithm which you can use for your own purposes with the help of organization block OB251.

Before calling OB251, a data block (PID controller DB) containing the controller parameters and other controller-specific data must be opened. The PID control algorithm is called periodically (sampling interval) and generates the manipulated variable. The more closely the sampling interval is observed, the more accurately can the controller fulfill its appointed task. The control parameters specified in the controller DB must be matched to the sampling interval. Typically, timed interrupts are serviced by a timed interrupt OB (OB10 to OB13).

Timed interrupt OBs can be called at intervals between 1 ms and 1 min. The maximum execution time of the PID control algorithm is 110 µs.

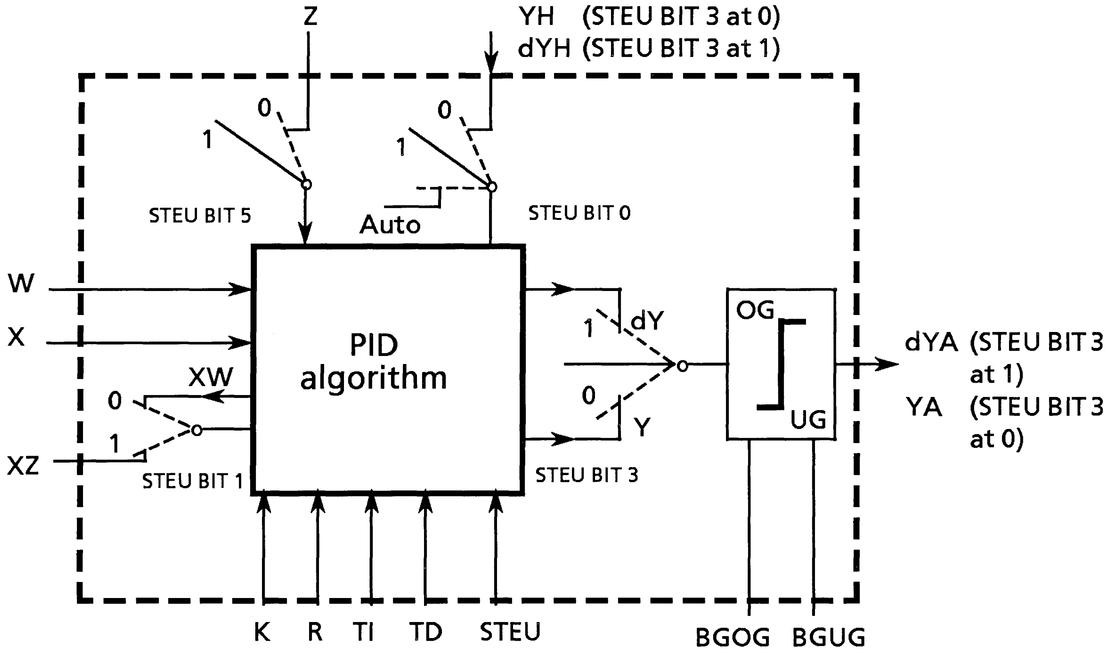


Figure 2-22 Block Diagram of the PID Controller

Legend:

K	=	Proportional coefficient K>0 positive control direction K<0 negative control direction	Y dY YH	=	Manipulated variable Correcting increment Value for manual input
R	=	R parameter (usually 1000)	dYH	=	Correcting increment for manual input
TA	=	Sampling interval	BGOG	=	Upper limiting value
TN	=	Integral-action time	BGUG	=	Lower limiting value
TV	=	Derivative-action time	X	=	Actual value
TI	=	TA/TN	Z	=	Disturbance variable
TD	=	TV/TA	XZ	=	Substitute variable for system deviation
STEU	=	Control word	YA	=	Controller output: manipulated variable limited
W	=	Setpoint	dYA	=	Controller output: correcting increment limited
XW	=	System deviation			

The continuous action controller is designed for controlled systems such as those used in pressure, temperature, or flow rate control.

The "R" parameter sets the proportional component of the PID controller.
If proportional action is required, most controller designs use the value R=1.

The individual proportional-action, integral-action, and derivative-action components can be deactivated via their parameters (R, TI, and TD) by presetting the pertinent data words with zero. This enables you to implement all required controller structures without difficulty, e.g., PI, PD, or PID controllers.

You can forward the system deviation XW or, using the XZ input, any disturbance variable or the inverted actual value X to the differentiator. Specify a negative K value for an inverted control direction.
When the correction information (dY or Y) is at a limit, the integral-action component is automatically deactivated in order not to impair the dynamic response of the controller.

The switch settings in the block diagram are implemented by setting the associated bits in control word STEU when the D controller is initialized.

Table 2-30 Description of the Control Bits in Control Word STEU

Control Bit	Name	Signal State	Description
0	AUTO	0	Manual mode The following variables are updated in Manual mode: 1) X_K, XW_{K-1} and PW_{K-1} 2) XZ_K, XZ_{K-1} and PZ_{K-1}, when STEU bit 1 = 1 3) Z_K and Z_{K-1}, when STEU bit 5 = 0 Variable dD_{K-1} is set to 0. The algorithm is not computed.
		1	Automatic mode
1	XZ EIN	0	XW_k is forwarded to the differentiator. The XZ input is ignored.
		1	A variable other than XW_k is forwarded to the differentiator.
2	REG AUS	0	Normal controller processing
		1	When the controller is invoked (OB251), all variables (DW 18 to DW 48) with the exception of K, R, TI, TD, BGOG, BGUG, YH_k and W_k are reset in the controller DB. The controller is deactivated.
3	GESCHW	0	Correction algorithm
		1	Velocity algorithm
4	HANDART	0	When GESCHW = 0: Following the transfer to Manual mode, the specified manipulated variable value YA is adjusted exponentially to the manual value in four sampling steps. Additional manual values are then forwarded immediately to the controller output. When GESCHW = 1: The manual values are forwarded immediatley to the controller output. The limiting values are in force in Manual mode.
		1	When GESCHW = 0: The manipulated variable last output is retained. When GESCHW = 1: Correction increment dY_K is set to zero.
5	NO Z	0	With feedforward control
		1	No feedforward control
6 to 15	-		The PID algorithm uses these bits as auxiliary flags.

The control program can be supplied with fixed values or parameters. Parameters are input via the assigned data words. The controller is based on a PID algorithm. Its output signal can be either a manipulated variable (correction algorithm) or a manipulated variable modification (correction rate algorithm).

Correction Rate Algorithm

The relevant correction increment dY_k is computed at instant $t = k \bullet TA$ according to the following formula:

- Without feedforward control (D11.5 = 1); XW is forwarded to the differentiator (D11.1 = 0)

$$dY_k = K[(XW_k - XW_{k-1}) R + TI \bullet XW_k + \tfrac{1}{2}(TD(XW_k - 2XW_{k-1} + XW_{k-2}) + dD_{k-1})]$$
$$= K(dPW_k + dI_k + dD_k)$$

- With feedforward control (D11.5 = 0); XW is forwarded to the differentiator (D11.1 = 0)

$$dY_k = K[(XW_k - XW_{k-1}) R + TI \bullet XW_k + \tfrac{1}{2}(TD(XW_k - 2XW_{k-1} + XW_{k-2}) + dD_{k-1})] + (Z_k - Z_{k-1})$$
$$= K(dPW_k + dI_k + dD_k) + dZ_k$$

- Without feedforward control (D11.5 = 1); XZ is forwarded to the differentiator (D11.1 = 1)

$$dY_k = K[(XW_k - XW_{k-1}) R + TI \bullet XW_k + \tfrac{1}{2}(TD(XZ_k - 2XZ_{k-1} + XZ_{k-2}) + dD_{k-1})]$$
$$= K(dPW_k + dI_k + dD_k)$$

- With feedforward control (D11.5 = 0); XZ is forwarded to the differentiator (D11.1 = 1)

$$dY_k = K[(XW_k - XW_{k-1}) R + TI \bullet XW_k + \tfrac{1}{2}(TD(XZ_k - 2XZ_{k-1} + XZ_{k-2}) + dD_{k-1})] + (Z_k - Z_{k-1})$$
$$= K(\underbrace{dPW_k}_{\text{P component}} + \underbrace{dI_k}_{\text{I component}} + \underbrace{dD_k}_{\text{D component}}) + \underbrace{dZ_k}_{\text{Z component}}$$

k: kth sample

When XW_k is applied:	XW_k	=	$W_k - X_k$
	PW_k	=	$XW_k - XW_{k-1}$
	QW_k	=	$PW_k - PW_{k-1}$
		=	$XW_k - 2XW_{k-1} + XW_{k-2}$
When XZ is applied:	PZ_k	=	$XZ_k - XZ_{k-1}$
	QZ_k	=	$PZ_k - PZ_{k-1}$
		=	$XZ_k - 2XZ_{k-1} + XZ_{k-2}$
The result is:	dPW_k	=	$(XW_k - XW_{k-1})R$
	dI_k	=	$TI \bullet XW_k$
	dD_k	=	$\tfrac{1}{2}(TD \bullet QW_k + dD_{k-1})$ when XW is applied
		=	$\tfrac{1}{2}(TD \bullet QZ_k + dD_{k-1})$ when XZ is applied
	dZ_k	=	$Z_k - Z_{k-1}$

Correction Algorithm

The formula used to compute the correction rate algorithm is also used to compute the correction algorithm.

In contrast to the correction rate algorithm, however, the sum of all correction increments computed (in DW 48), rather than the correction increment dY_k is output at sampling instant t_k.

At instant t_k, manipulated variable Y_k is computed as follows:

$$Y_k = \sum_{m=0}^{m=k} dY_m$$

Initializing the PID Algorithm

OB251's interface to its environment is the controller DB.
All data needed to compute the next manipulated variable value is stored in this DB. Each controller must have its own controller data block.
The controller-specific data are initialized in a data block that must comprise at least 49 data words.
An error code is entered in ACCU 1 and the CPU goes to STOP with a transfer error (TRAF) if no DB has been opened or if the DB is too short.

Possible Errors:

Table 2-31 Possible Errors

Error Code	Description of Error
1	No DB/DX opened
2	DB/DX not long enough

Caution

Make sure that the right controller DB has been invoked before calling control algorithm OB251.

Table 2-32 Format of the Controller DB

Data Word	Name	Comments
1	K	Proportional coefficient (- 32 768 to + 32 767) for controllers without D component Proportional coefficient (- 1500 to + 1500) for controllers with D component* K is greater than zero when the control direction is positive, and less than zero when the control direction is negative; the specified value is multiplied with the factor 0.001**.
3	R	R parameter (- 32 768 to + 32 767) for controllers without D component R parameter (-1500 to + 1500) for controllers with D component* Normally 1 for controllers with P component: the specified value is multiplied with the factor 0.001**.
5	TI	Constant TI (0 to 9999) $TI = \frac{\text{Sampling interval TA}}{\text{Integral-action time TN}}$ The specified value is multiplied with a factor of 0.001**.
7	TD	Constant TD (0 to 999) $TD = \frac{\text{Derivative-action timeTV}}{\text{Sampling interval TA}}$

* Larger gains are possible if abrupt changes to the system deviation are sufficiently small. Large changes of the system deviation should therefore be divided up into small changes; e.g. by feeding the setpoint via a ramp function.

** The factor 0.001 is an approximate value. The precise value for the factor is 1/1024 or 0.000976.

Table 2-32 Format of the Controller DB (Continued)

Data Word	Name	Comments
9	W	Setpoint (- 2047 to +2047)
11	STEU	Control word (bit pattern)
12	YH	Value for manual operation (- 2047 to +2047)
14	BGOG	Upper limiting value (- 2047 to +2047)
16	BGUG	Lower limit ing value (- 2047 to +2047)
22	X	Actual value (- 2047 to +2047)
24	Z	Disturbance value (- 2047 to +2047)
29	XZ	Derivative time (- 2047 to +2047)
48	YA	Output variable (- 2047 to +2047)

All parameters (with the exception of the control word STEU) must be specified as 16-bit fixed point numbers.

Note

The PID algorithm uses the data words that are not listed in the table as auxiliary flags.

Initializing and Invoking the PID Controller in the STEP 5 Program

A number of different PID controllers can be implemented by calling OB251 repeatedly, so make sure that the relevant controller DB has been invoked before calling OB251.

Note

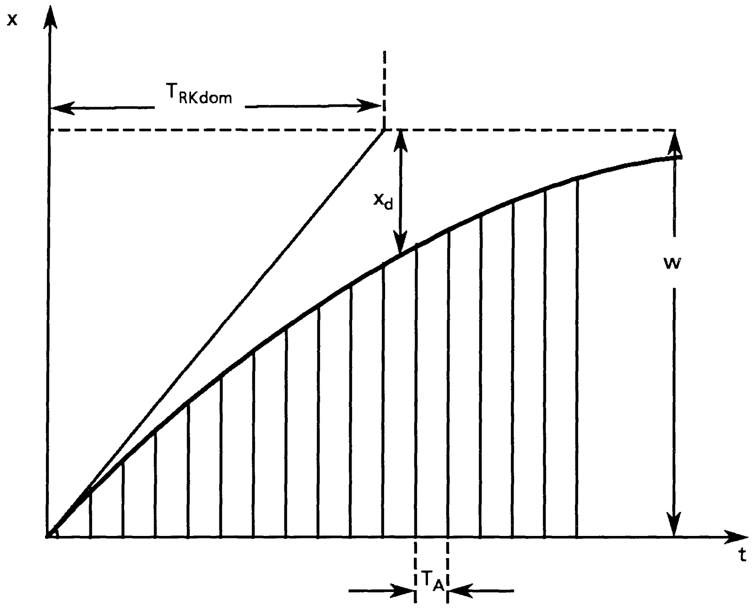
Important controller data are stored in the high-order byte of control word DW 11 (DL 11). Make sure that only write access via T DR 11/SU D 11.0 to D 11.7 or RU D 11.0 to D 11.7 operations is used to modify user-specific bits in the control word.

Selecting the Sampling Interval

The value selected as sampling interval must not be excessiveley high in order to be able to use the well-known analog method in the case of digital control loops.
Experience has shown that a sampling interval of approximately 1/10 of the counter constant T_{RKdom}* produces a control result comparable to the equivalent analog result. Dominant system time constantT_{RKdom} determines the step response of the closed control loop.

$$T_A = 1/10 \bullet T_{RKdom}$$

In order to ensure constant sampling intervals, OB 251 must always be called up in the timed interrupt OB (OB 10 to OB 13).



x	=	Controlled variable
t	=	Time
T_A	=	Sampling time
T_{RKdom}	=	Dominant system time constant
w	=	Reference variable/setpoint
x_d	=	System error

Figure 2-23 Estimating the Dominant System Time Constant (T_{RKdom})

Example for the Use of the PID Control Algorithm

Using a PID controller to keep an annealing furnace at a constant temperature.
The temperature setpoint is entered via a potentiometer.
The setpoints and actual values are acquired using an analog input module and forwarded to the controller. The computed manipulated variable is then output via an analog output module.
The controller mode is set in input byte 0 (see control word DW 11 in the controller DB).
You must use the well-known controller design procedure to determine how to tune the controller for each controlled system.

* T_{RKdom} = Dominant system time constant

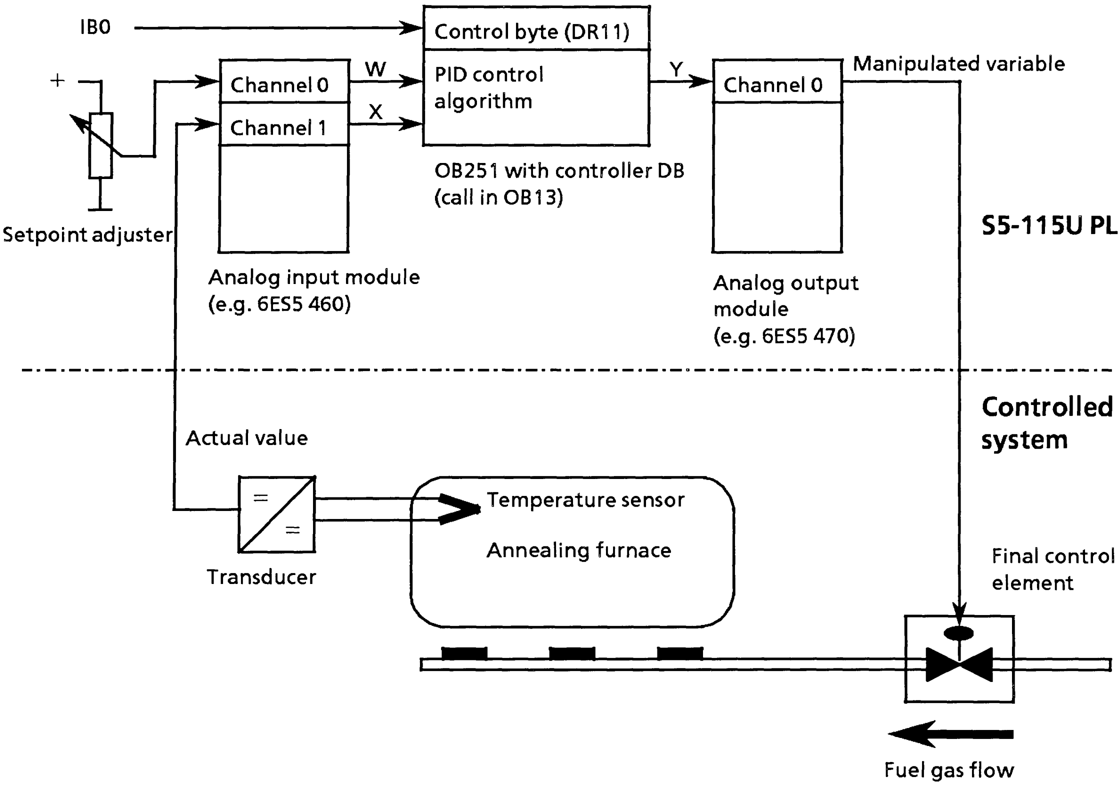


Figure 2-24 Process Schematic

The analog signals of the setpoint and actual values are converted into corresponding digital values at each sampling instant. OB251 uses these values to compute the new digital manipulated variable, from which, in turn, the analog output module generates a corresponding analog signal. This signal is then forwarded to the controlled system.

Invoking the controller in the program:

OB13 STL	Description
:	
:JU FB 10	PROCESS CONTROLLER
NAME :REGLER 1	
:	THE CONTROLLER'S SAMPLING INTERVAL
:	DEPENDS ON THE TIME BASE USED
:	TO CALL OB 13 (SET IN SD 97).
:	THE DECODING TIME OF THE
:	ANALOG INPUT MODULES MUST BE TAKEN
:	INTO ACCOUNT WHEN SELECTING
:	THE SAMPLING INTERVAL.
:	
:BE	

```
FB 10        STL                          Description

     NAME :REGLER 1
          :
          :C    DB   30                   OPEN CONTROLLER DB
          :
          :                               ********************************
          :                               READ CONTROLLER'S CONTROL BITS
          :                               ********************************
          :
          :L    PY    0                   READ CONTROLLER'S CONTROL
          :T    FY   10                   INPUTS
          :T    DR   11                   AND STORE IN DR11
          :                               CAUTION:
          :                               DL11 CONTAINS IMPORTANT CONTROL
          :                               DATA FOR OB251
          :                               THE CONTROL BIT MUST
          :                               THEREFORE BE TRANSFERRED WITH
          :                               T DR11 TO PREVENT CORRUPTING
          :                               OF DL11
          :
          :                               ********************************
          :                               READ ACTUAL VALUE AND SETPOINT
          :                               ********************************
          :
          :A    F    12.0                 FLAG 0 (FOR UNUSED FUNCTIONS IN
          :R    F    12.0                 FB 250)
          :AN   F    12.1                 FLAG 1
          :S    F    12.1
          :
          :JU   FB 250                    READ ACTUAL VALUE
     NAME :RLG:AE
     BG   :     KY 0,128                  MODULE ADDRESS IN P AREA
     KNKD :     KY 0,6                    CHANNEL NUMBER 0, FIXED-POINT BIPOLAR
     OGR  :     KD +2047+04               UPPER LIMIT FOR ACTUAL VALUE
     UGR  :     KD -2047+04               LOWER LIMIT FOR ACTUAL VALUE
     EINZ :     F    12.0                 NO SELECTIVE SAMPLING
     XA   :     FD   22                   BUFFER SCALED ACTUAL VALUE
     FB   :     F    12.2                 ERROR BIT
     BU   :     F    12.3                 RANGE VIOLATION
     TBIT :     F    12.4                 ACTIVITY BIT
          :
          :                               ********************************
          :                               ADAPT ACTUAL VALUE TO KF FORMAT
          :                               ********************************
          :
          :L    FD   22                   LOAD ACTUAL VALUE IN ACCU 1
          :GFD                            CONVERT INTO FIXED-POINT VALUE
          :T    DW   22                   TRANSFER VALUE TO CONTROLLER DB
          :
```

FB 10 (Continued) STL	Description
:JU FB 250	READ SETPOINT
NAME :RLG:AE	
BG : KF 0,128	MODULE ADDRESS
KNKD : KY 1,6	CHANNEL NO. 1, FIXED-POINT BIPOLAR
OGR : KD +2047+04	UPPER LIMIT FOR SETPOINT
UGR : KD -2047+04	LOWER LIMIT FOR SETPOINT
EINZ : F 12.0	NO SELECTIVE SAMPLING
XA : FD 6	BUFFER SCALED SETPOINT
FB : F 13.1	ERROR BIT
BU : F 13.2	RANGE VIOLATION
TBIT : F 13.3	ACTIVITY BIT
:	
:	********************************
:	ADAPT SETPOINT TO KF FORMAT
:	********************************
:	
:L FD 6	LOAD SETPOINT IN ACCU 1
:GFD	CONVERT TO FIXED-POINT VALUE
:T DW 9	TRANSFER VALUE TO CONTROLLER DB
:	
:A F 10.0	IN MANUAL MODE, THE SETPOINT
:JC =WEIT	IS SET TO THE ACTUAL VALUE
:L DW 22	TO FORCE THE CONTROLLER
:T DW 9	TO REACT TO A SYSTEM DEVIATION,
:	IF ANY, WITH A P STEP
:	ON TRANSFER TO
:	AUTOMATIC MODE
WEIT :	
:	********************************
:JU OB 251	INVOKE CONTROLLER
:	********************************
:	
:	********************************
:	OUTPUT MANIPULATED VARIABLE Y
:	********************************
:	
:L DW 48	LOAD MAN. VAR. Y FROM CONTROLLER DB INTO
:SPA OB 220	EXTENSION FOR SIGN
:FDG	ACCU 1, CONVERT TO FIXED-POINT FORMAT
:T FD 48	BUFFER IN FD 48
:JU FB 251	
NAME :RLG:AA	
XE : FD 48	FORWARD MAN. VAR. Y TO ANALOG OUTPUT MOD.
BG : KY 0,176	MODULE ADDRESS
KNKD : KY 0,1	CHANNEL 0, FIXED-POINT BIPOLAR
OGR : KD +2047+04	UPPER LIMIT FOR ACTUATING SIGNAL
UGR : KD -2047+04	LOWER LIMIT FOR ACTUATING SIGNAL
FB : F 13.5	ERROR BIT WHEN LIMITING VALUES DEFINED
BU : F 13.6	RANGE VIOLATION
:BE	

DB 30	STL	Description
0:	KH = 0000;	
1:	KF = +01000;	K PARAMETER(HERE=1),FACTOR 0.001
2:	KH = 0000;	(VALUE RANGE: -32768 TO 32767)
3:	KF = +01000;	R PARAMETER(HERE=1),FACTOR 0.001
4:	KH = 0000;	(VALUE RANGE: -32768 TO 32767)
5:	KF = +00010;	TI=TA/TN(HERE=0.01),FACTOR 0.001
6:	KH = 0000;	(VALUE RANGE: 0 TO 9999)
7:	KF = +00010;	TD=TV/TA (HERE=10), FACTOR 1
8:	KH = 0000;	(VALUE RANGE: 0 TO 999)
9:	KF = +00000;	SETPOINT W, FACTOR 1
10:	KH = 0000;	(VALUE RANGE: -2047 TO 2047)
11:	KM = 00000000 00100000;	CONTROL WORD
12:	KF = +00500;	MANUAL VALUE YH, FACTOR 1
13:	KH = 0000;	(VALUE RANGE: -2047 TO 2047)
14:	KF = +02000;	UPPER CONT. LIMIT BGOG, FACTOR 1
15:	KH = 0000;	(VALUE RANGE: -2047 TO 2047)
16:	KF = -02000;	LOWER CONT. LIMIT BGUG, FACTOR 1
17:	KH = 0000;	(VALUE RANGE: -2047 TO 2047)
18:	KH = 0000;	
19:	KH = 0000;	
20:	KH = 0000;	
21:	KH = 0000;	
22:	KF = +00000;	ACTUAL VALUE X, FACTOR 1
23:	KH = 0000;	(VALUE RANGE: -2047 TO 2047)
24:	KF = +00000;	DISTURBANCE VALUE Z, FACTOR 1
25:	KH = 0000;	(VALUE RANGE: -2047 TO 2047)
26:	KH = 0000;	
27:	KH = 0000;	
28:	KH = 0000;	
29:	KF = +00000;	FORWARD XZ FOR DIFF.,
30:	KH = 0000;	FACTOR 1, (-2047 TO 2047)
31:	KH = 0000;	
32:	KH = 0000;	
33:	KH = 0000;	
34:	KH = 0000;	
35:	KH = 0000;	
36:	KH = 0000;	
37:	KH = 0000;	
38:	KH = 0000;	
39:	KH = 0000;	
40:	KH = 0000;	
41:	KH = 0000;	
42:	KH = 0000;	
43:	KH = 0000;	
44:	KH = 0000;	
45:	KH = 0000;	
46:	KH = 0000;	
47:	KH = 0000;	
48:	KF = +00000;	CONTROLLER OUTPUT Y, FACTOR 1
49:	KH = 0000;	(VALUE RANGE: -2047 TO 2047)
50:		

3 Installation Guidelines

Figures

Tables

3 Installation Guidelines

Programmable controllers of the S5-115U system consist of a central controller to which you can connect several expansion units if required.
This chapter tells you

- which different mounting racks are available for the different central controllers and expansion units
- how the modules are mounted on the mounting racks
- how you can implement the connection between a central controller and one or more expansion units with the IM304/IM314 interface module
- how you must wire your programmable controller

In addition, this chapter contains all the important rules for the noise-immune configuration of your programmable controller.

3.1 Mounting Racks

Various mounting racks are available to suit the performance or the degree of expansion the control system is to have. Each mounting rack consists of an aluminum mounting rail for fastening all modules mechanically and one or two backplanes for connecting the modules to each other electrically. The module locations (slots) are numbered in ascending order from left to right (see Figure 3-1).

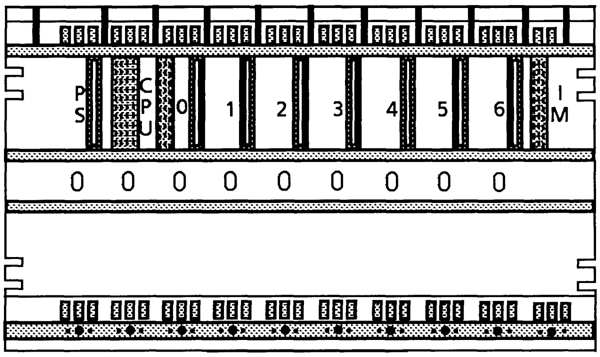


Figure 3.1 Typical Mounting Rack (CR 700-1)

3.1.1 Central Controllers

A central controller has a power supply module (PS), a central processing unit (CPU), and various input/output modules (I/Os). Depending on requirements, digital or analog modules, communications processors (CPs), or intelligent input/output modules can be used. Figure 3-1 shows a basic CC configuration.

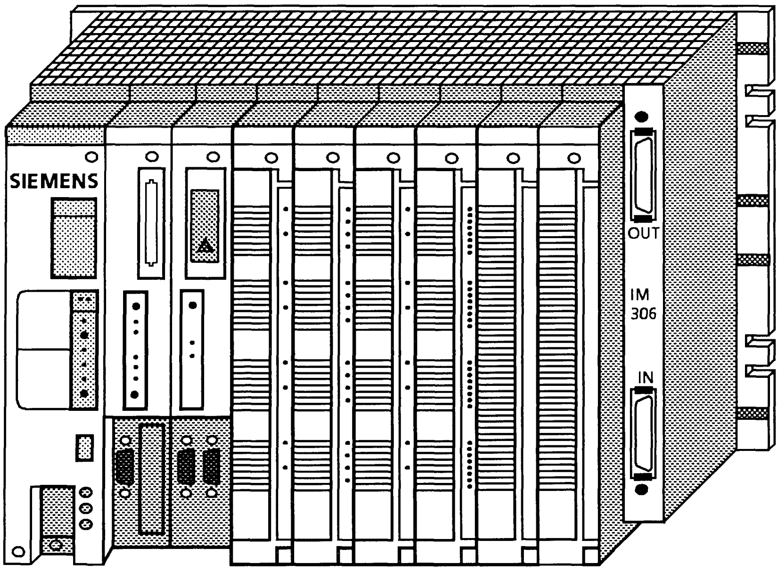


Figure 3-2 Typical Central Controller

We offer you five different mounting racks for mounting your central controller:

- CR 700-0LA12 and CR 700-0LB11
- CR 700-1
- CR 700-2
- CR 700-3

The first two slots of all mounting racks contain a power supply module (PS) and a central processing unit (CPU).
The mounting racks differ in the number of slots (4 or 7 I/O modules) and configuration possibilities.

The following table gives you an overview of the configuration possibilities the various mounting racks of a central controller.

Table 3-1 Possible Configurations on the CR 700-... Mounting Rack

Module	Occupied Slots on the Mounting Rack				
	CR700-0LA12	CR700-0LB11	CR700-1	CR700-2	CR700-3
Power supply	PS	PS[3]	PS[3]	PS[3]	PS[3]
CPU	CPU	CPU	CPU	CPU	CPU
Digital modules (block type)	0 to 3	1, 2	0 to 6	0 to6	3 to 5
Digital modules (PCBs[1] [9])	0	0 to 2	0	0 to 5	0 to 5
Analog modules (block type)	0 to 3	1, 2	0 to 6	0 to 6	3 to 5
Analog modules (PCBs[1])	0	0 to 2	0	0 to 5	0 to 5
IM 305/306[2]	IM	IM	IM	IM	IM
IM 304/308[1]		3		6	6
IM 307		0 to 3[7]		0 to 6[8]	0 to 6
AS 301/302[1]		3		6	6
CP 530-7	0	1, 2	0	0 to 5	3 to 5
CP 523[1]	0	0 to 2	0	0 to 5	0 to 5
CP 516[1]/524[1]/ 525[1]/528[1]/530-3[1]/ 544[1]		0 to 2		0 to 5	0 to 5
CP 526/527 basic board[1]/528		0 to 2		0 to 5	0 to 5
CP 526/527 expansion board[1]		0			0 to 2
CP 5430[1]/5431[1]		0[4], 1		0 to 5	0 to 2[4], 3 to 5
CP 552-2[1]		0[5]			0 to 2[5]
CP 535-3MA12[1]/ CP 143-0AB01[1]		0[5], 1, 2		0 to 5	0 to 2[5], 3 to 5
CP 580/581[1]		0[6]			0 to 2[6]

[1] In adapter casing

[2] If neither the IM 305 nor the IM 306 interface module is plugged in, the termination connector must not be removed

[3] Use of the IP 246/247 and the CP 513/524/525/526/527/535/580/581/143 and WF 705/706/707/721/723 is not permissible with a 3 A power supply module. Order No. 6ES5 951-7LB14/7NB13, (the DSI signal is not generated by the 3 A power supply). CPs 524/525 must not be used with 3 A power supplies because their power consumption is too high

[4] Can only be used in the lefthand slot when operated without fan

[5] Righthand slot cannot be used because of double-width module

[6] Righthand slot cannot be used because of quadruple-width module

[7] Interrupt processing not possible in slot 3

[8] Interrupt processing not possible in slot 6

[9] Not in operating mode with interrupt processing, except for digital input module 432-4UA11

Table 3-1 Possible Configurations on the CR 700-... Mounting Rack (Continued)

Module	Occupied Slots on the Mounting Rack				
	CR700-0LA12	CR700-0LB11	CR700-1	CR700-2	CR700-3
IP 241[1]	0	0[5], 1, 2	0	0 to 5	0 to 2[5], 3 to 5
IP 240[1]/242B[1]/ 243[1]/244[1]	0	0 to 2	0	0 to 5	0 to 5
IP 252[1]		0[5], 1, 2		0 to 5[9]	0 to 2[5], 3 to 5[9]
IP 288[1]		0[5], 1, 2		0 to 5	0 to 5[5], 3 to 5
IP 246[1]/247[1]/281[1]		0 to 2		0 to 5	0 to 5
WF 705[1]/707[1]		0 to 2	0	0 to 5	0 to 5
WF 706-3AA..[1]		0 to 2	0	0 to 5	0 to 5
WF 706-3AB..[1]		0[5], 1, 2	0	0 to 5	0 to 5[5], 3 to 5
WF 721/723		0 to 2	0	0 to 5	0 to 5

[1] In adapter casing

[5] Righthand slot cannot be used because of double-width module

[9] Direct I/O access of IP 252 only in slots 0, 1, 2; if 3 A power supply is used, direct I/O access is generally not possible (HOLD and HOLDA signals are not provided)

3.1.2 Expansion Units

If the central controller does not have sufficient slots for the whole PLC, it is possible to include one or more expansion units. Depending on the type of connection, there are four subracks available for expansion units:

- ER 701-0
- ER 701-1
- ER 701-2
- ER 701-3

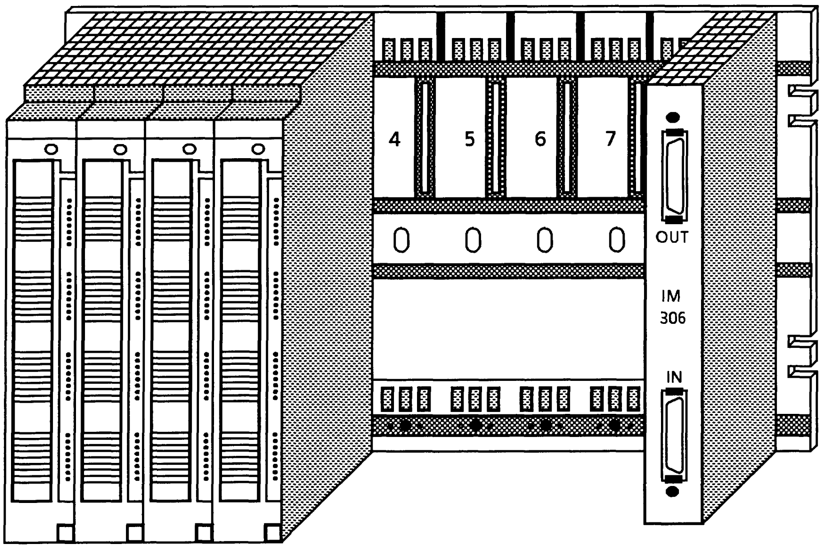


Figure 3-3 Expansion Unit 1 (Example)

The following expansion unit interface modules connect EU 1 expansion units to a central controller in centralized configurations (see Section 3.3.1):

- IM 305
- IM 306

The following interface modules connect expansion units to a central controller in distributed configuration (see Section 3.3.2):

- AS 301/310
- AS 302/311
- IM 304/314
- IM 307/317
- IM 308/318

Areas of application of the ER 701-... mounting rack

ER 701-0/-1 mounting rack

The ER 701-0/-1 mounting rack is suitable for connecting a central controller locally (centralized connection). Interrupt-initiating modules cannot be used. The expansion unit is supplied over the interface module. Up to three expansion units can be connected to one central controller or to one ER 701-2/ER 701-3.

ER 701-2 mounting rack

The ER 701-2 mounting rack is suitable for connecting a CR 700-2/CR 700-3 locally and remote. Up to three ER 701-1s can be connected to one ER 701-2 over an IM 306. Interrupt-initiating modules cannot be used.
The ER 701-2 can also be connected to the S5-135U and S5-155U programmable controllers over the AS 310, AS 311, IM 314, IM 317 and IM 318 interface modules.

ER 701-3 mounting rack

Interrupt-initiating modules can only be used over the IM 307/317. Up to three ER 701-1s can be connected to one ER 701-3 over an IM 306.
The ER 701-3 can also be connected to the S5-135U, S5-150U and S5-155U programmable controllers over the AS 310, AS 311, IM 314, IM 317 and IM 318 central controller interface modules.

The following table gives you an overview of the configuration possibilities of the various mounting racks of an expansion unit.

Table 3-2 Possible Configurations on the ER 700-... Mounting Rack

Module	Occupied Slots on the Mounting Rack			
	ER701-0	ER701-1	ER701-2	ER701-3
Power supply			PS	PS[5]
Digital modules [6]) (block type)	0 to 5	0 to 8	0 to 7	0 to7
Digital modules (PCBs[1] [9])			6	0 to 6
Analog modules (block type)	0 to 5[2]	0 to 8[2]	0 to 7[3]	0 to 7[7]
Analog modules (PCBs[1])			6[7]	0 to 6[7]
IM 305	IM	IM	IM	
IM 306	IM	IM	IM	IM
IM 307[1]				0 to 7
IM 314[1]			7	7
IM 317[1]			7	0 to 7
IM 318[1]			7	
IM 318-3[1]				7
AS 310/311[1]			7	7
ÜBW 313[1]			6[4]	0[8]
CP 523[1]			6[4]	
CP 516[1]/523[1]/524[1]/ 525[1]/526[1]/535[1]/552[1]				0 to 6[8]
CP 530				0 to 6
CP 5430[1]/CP 5431[1]/CP 143-0AB01[1]				0 to 6[8]
IP 241[1]/242[1]/243[1]/244[1]				0 to 6
IP 240[1]/246[1]/247[1]/252[1]/281[1]/288[1]				0 to 6[8]
WF 705[1]/706[1]/707[1]/721[8]/723[8]				0 to 6

[1] In adapter casing

[2] Only when using IM 306

[3] Only when using IM 306; not permissible with connections using AS 302/311

[4] Not permissible with connections using AS 302/311

[5] Use of the IP 246/247 and the CP 513/524/525/526/527/535/580/581/143 and WF 705/706/707/721/723 is not permissible with a 3 A power supply module. Order No. 6ES5 951-7LB14/7NB13, (the DSI signal is not generated by the 3 A power supply). CPs 524/525 must not be used with 3 A power supplies because their power consumption is too high

[6] Except 434-7 input module; 485-7 input/output module not in mode with interrupt processing

[7] Not permissible with connections using AS 302/311

[8] Only with connections using IM 304/314 and 307/317

[9] Not in operating mode with interrupt processing, except for digital input module 432-4UA11

3.2 Mechanical Installation

Fasten all modules on the appropriate mounting racks. The mounting racks must be enclosed in grounded metallic housings (e. g. switchgear cabinets).
Figure 3.4 shows the prescribed method for mounting. You can also fasten the racks to surfaces that are at an angle of up to 15° from a vertical surface. Block-type modules are mounted directly on the rack. Place printed circuit boards in double-height Eurocard format in adapter casings (see Figure 3.6).

3.2.1 Installing the Modules

Any person opening the switchgear cabinet or switchbox should first provide for sufficient discharge of the body in order to protect the modules against electrostatic discharge.
Install block-type modules according to the following procedure:

- Remove the protective caps from the socket connectors on the backplane.
- Hook the top of the module into place between the two guides on the top of the mounting rack①.
- Swing the module back until it engages with the socket connectors on the backplane②.
- Fasten the screws at the top and bottom of the module.

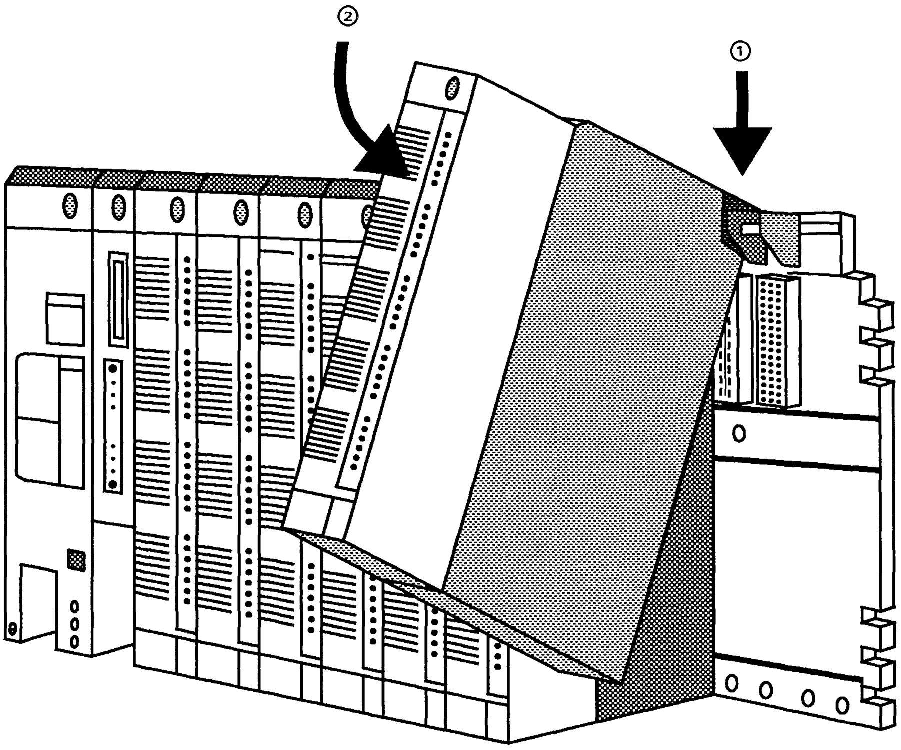


Figure 3-4 Installing the Modules

If the modules are subjected to mechanical vibration, they should be installed as close together as possible.

Warning

Plug in or remove modules only when the power supply has been turned off.

Mechanical slot coding

On the back of each module, with the exception of the power supply and central processing unit, is a slot coding element in the form of a two-part plastic cube. This coding element ensures that when one module is replaced, only another module of the same type will be plugged in in its place.

The coding element consists of two parts, one like a lock and one like a key. The two parts fit together in a defined position. This position is specific to each type of module. When you install the module, the back of the coding element is inserted into the mounting rack. When you swing the module out, the key-shaped part of the element stays in the mounting rack and the lock-shaped part stays on the module.

Now you can install only this particular module or an identical one in this slot. If you want to install a different module, you have to remove the coding element from the mounting rack.

You can also work without slot coding. To do this, you must pull the coding element off the module before you swing the module into place for the first time.

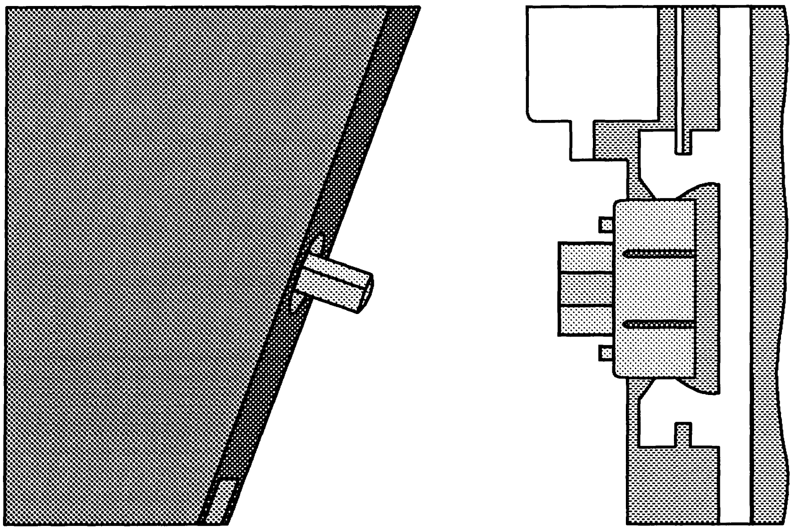


Figure 3-5 Slot Coding Element

Adapter casing

Use an adapter casing (6ES5 491-0LB12, 6ES5 491-0LC11 oder 6ES5 491-OLD11) to fasten printed circuit boards in double-height Eurocard format to a mounting rack as you would fasten block type modules.

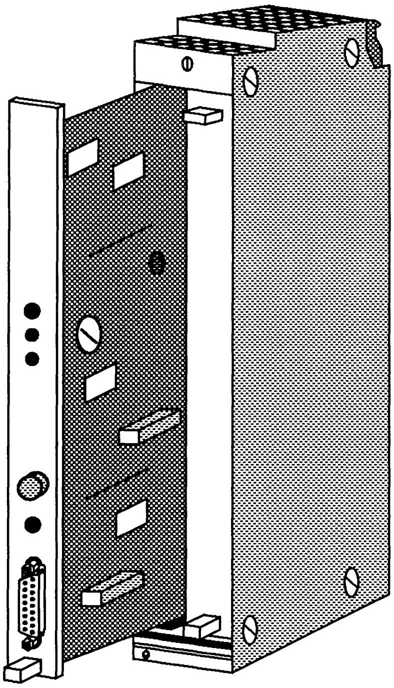


Figure 3-6 Installation of a Printed Circuit Board into an Adapter Casing (6ES5 491-0LB11)

Installing an adapter casing

- Swing the adpater casing onto the mounting rack and screw it.
- Push the printed circuit board into the casing along the guide tracks.
- Lock the module into place with the eccentric locking collar at the top of the casing.

If an opening remains on the front after the module has been inserted, cover it with a blanking plate.

Note

A fan is required for adapter casings with two printed circuit boards.

3.2.2 Installing Fans

Installing fans

Install a fan subassembly under the following conditions:

- If the power supply modules carry a load of more than 7 A
- If the controller uses modules with a high power consumption, e.g., certain communications processors and intelligent input/output modules (see Chapter 15 "Technical Specifications").

The fan subassembly has two fans, a dust filter, and fan monitors with floating changeover contact.
You need a set of installation parts to mount the fan subassemby. The set consists of two installation brackets and a cable duct. The brackets support the fan subassembly and the cable duct. The cable duct enables you to run the field cables off neatly to the side.

Install the fan subassembly as follows:

① Use screws to fasten the installation brackets onto the uprights of the cabinet or on the mounting surface under the mounting rack.
② The guide tracks on the brackets should be at the bottom. Hook the fan subassembly onto the guide tracks of the installation brackets and
③ Push it back.
④ Push the fan subassembly up and
⑤ Latch it into place with the two slides at the top of the installation brackets.
⑥ If the machine is subject to vibration, secure the fan subassembly to the installation brackets with screws (M 4 x 20 screws with washers).
⑦ Hook the cable duct into the installation brackets.

Special features of the fan subassmbly and installation parts enable you to do the following:

- Use the cable duct without the fan subassembly
- Install or remove the fan subassembly even when the cable duct is hooked on
- Screw the fan subassembly to the installation brackets through the cable duct
- Replace the filters while the unit is in operation (see Appendix B).

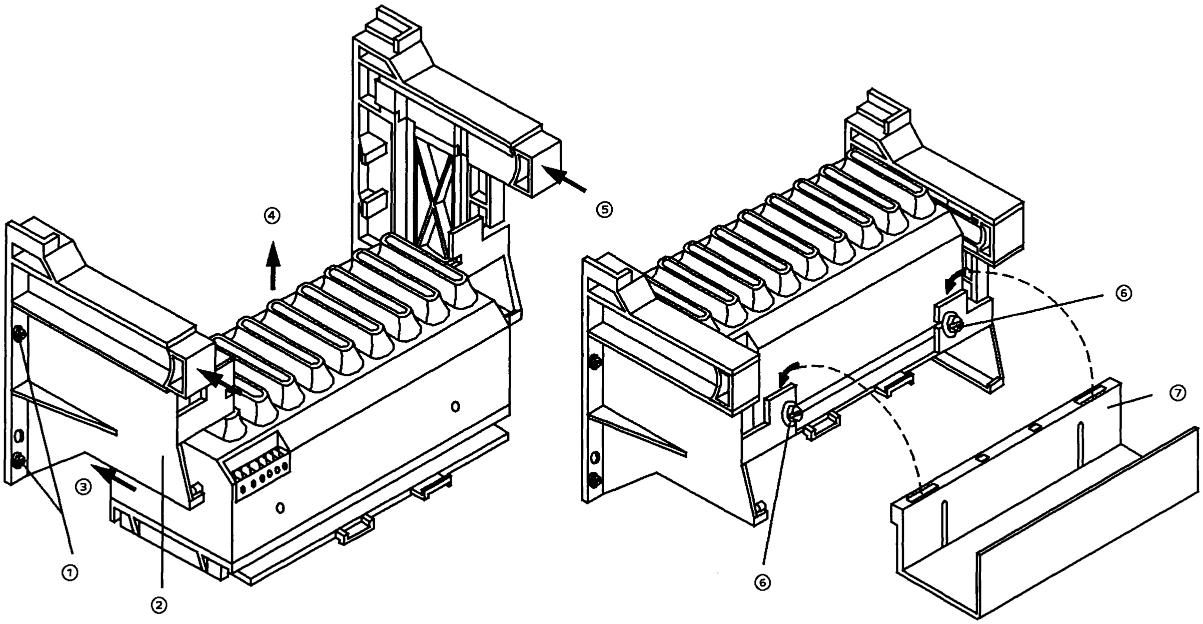


Figure 3-7 Installing the Fan Subassembly

Connecting the fan subassembly

Figure 3.8 shows the wiring necessary to operate a fan subassembly.

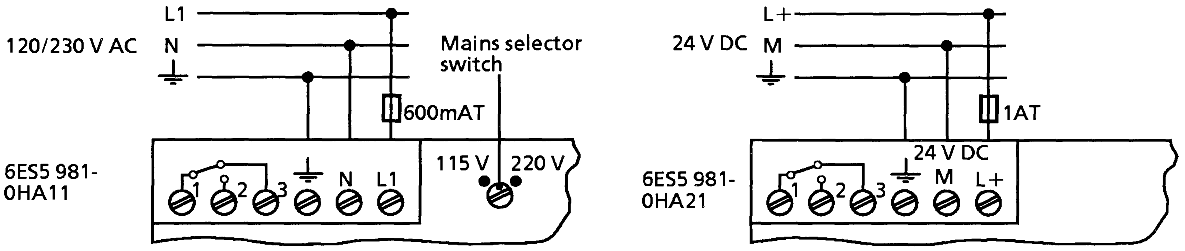


Figure 3-8 Fan Subassembly Terminal Assignment

A floating changeover contact gives a fault signal via terminals 1, 2 and 3 if the fan fails.
The diagram in Figure 3.8 shows the switch positions in the case of a fault! Under normal operating conditions, the contacts 1 - 2 are closed and the contacts 1 - 3 open.

3.3 Configurations

The links between a central controller and expansion units are referred to as configurations.

We distinguish between two types of configuration:
- Centralized configuration
- Distributed configuration

Centralized configuration

Centralized configurations are connections between a central controller and up to three expansion units over a distance of up to 2.5 m.
The expansion units are supplied and connected to the S5 bus over interface modules. The possible distance and the number of expansion units are dictated by the type of interface module in each case.

Distributed configuration

Distributed configurations are connections between one central controller and expansion units over a distance of up to 23.8 km. The possible distance and the number of expansion units are dictated by the type of interface module in each case.

3.3.1 Centralized Configurations

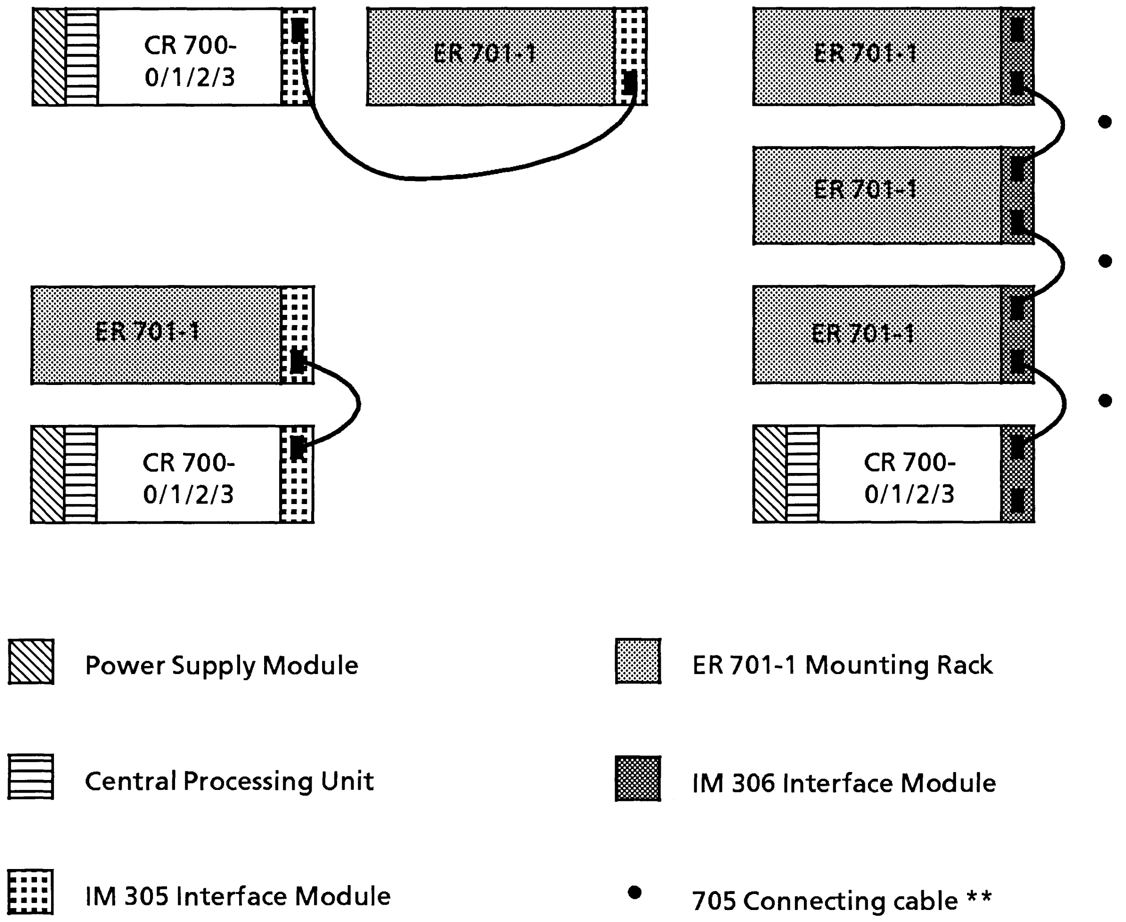
A CR 700-0/1/2 central controller connected via short connecting cables to as many as three expansion units of the type ER 701-1 makes up a centralized configuration. Use only the IM 305 or IM 306 interface modules to connect an ER 701-1 mounting rack.
For centralized configuration with the IM 305, please note the following points:

- You can use fixed slot addressing only (see Chapter 6).
  Addressing on slot 0 of the expansion unit always starts at 28.0 regardless of how many slots are available on the central controller.
- The 0.5 m (1.6 ft.) connecting cable is not long enough to connect the EU under the CC. Use an IM 306 interface module or the IM 305 version with a longer cable for such an arrangement.

Table 3-3 Comparison of the IM 305 and IM 306 Interface Modules

	Configuration with IM 305	Configuration with IM 306
Number of expansion units (maximum)	1	3
Total cable length	0.5 m or 1.5 m	maximum 2.5 m
Slot addressing	fixed	variable
Current supplied to expansion units (maximum)	1 A	2 A *

* The expansion unit with the most current supplied should be as close to the central controller as possible.



** You can also order a 1.25 m (4.1 ft.) 705 connecting cable (Order No. 6ES5 705-0BB20) or a 2.5 m (6.7 ft.) 705 connecting cable (Order No. 6SE5 705-0BC50), and use them to mount two EUs next to each other.

Figure 3-9 Centralized Configuration with the IM 305 and IM 306 Interface Modules

3.3.2 Distributed Configurations

This chapter describes distributed configurations with the IM 304/314 interface modules.

Distributed configurations using the following are not described here:

- AS 301/AS 310
- AS 302/AS 311
- IM 307/IM 317
- IM 308 -3UA.. /IM 318-3UA.. (S5-115U I/O)
  or
  IM 318-8MA.. (ET 100U)
- IM 308-3UB.. /IM 318-8MB.. (ET 200U)
  or
  IM 418-8MB.. (ET 200K)

You can order these interface modules with a separate description.

Please note the following points concerning distributed configuration versions:

- Each ER 701-2 or ER 701-3 expansion rack requires a PS 951 power supply module and an IM 306 interface module for addressing input/output modules (Exception: ET 100/ET200).
- If the expansion units have their own power supply, please note the following:
  - When switching on:
    Switch on the power supplies of the expansion units first and only then the power supply of the central controller.
  - If you switch on the power supplies of the central controllers and the expansion units at the same time, you must program a restart delay.
- See Section 3.5.7 (Shielding)!
- If you use digital input modules on the ER 701-2 or ER 701-3, it is recommended that you use modules with revision level "2" (or higher).

Table 3-4 Technical Specifications of the Interface Modules for Distributed Configurations

	Number of Expansion Units (Max.)	Total Cable Length (Max.)	Power Consumption at 5 V
AS 301	4	200 m	0.8 A
AS 310			0.7 A
AS 302	3	1000 m	2.0 A
AS 311			1.5 A
IM 304	8	600 m	1.2 A
IM 314			0.85 A
IM 307	14*	50 to 3000 m	1.0 A
IM 317			1.0 A
IM 308-3UA	63	3000 m	0.5 A
IM 318-3UA or IM 318-8MA			0.3 A
			0.35 A at 24 V
IM 308-3UB	124	23,8 km	0.6 A
ET 200-Slave Stations			(see ET 200 Manual)

* Number of EUs depends on the length of the fiber optic cable used and the ready delay time of the individual modules

Distributed configurations with the IM 304/IM 314 interface modules

Using these interface modules, you can connect up to 4 distributed expansion units per interface to one central controller.

Prerequisites are

- One IM 304 interface module must be installed on a CR 700-2/-3-0LB mounting rack
- One IM 314 interface module must be installed on each ER 701-2/-3 mounting rack
- The interface modules are linked via a connecting cable.

The following sections show the different switch and jumper settings for

- The IM 304-3UB1.
  and
- The IM 314.

Switch and jumper settings on the IM 304-3UB1. for distributed connection

Figure 3-21 shows the positions of the switches and jumpers on the IM 304-3UB1. module. All switches on switch block S3 must be in the ON position.

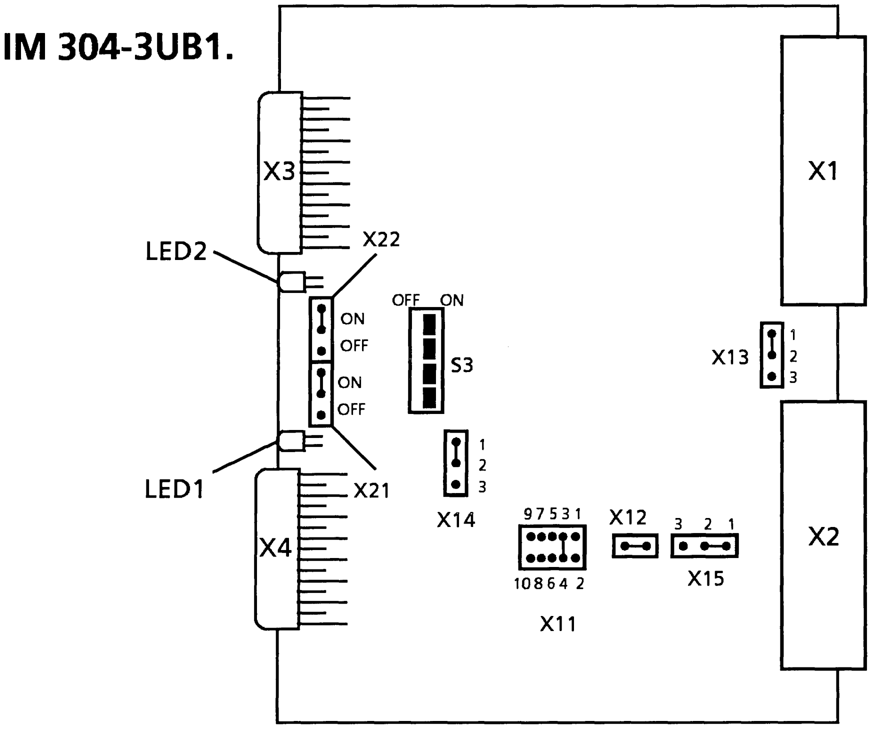


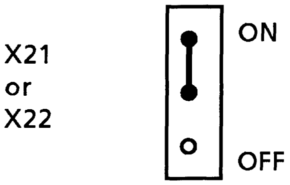
Figure 3-10 Switch and Jumper Settings on the IM 304-3UB1. for Distributed Connection

In Figure 3-21, the IM 304 has been set for distributed connection.

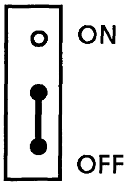
- Permissible cable length up to 100 m (330 ft.) (X11)
- PEU signal (I/Os not ready) is located at Pin b18 of the X2 base connector, (setting at X15)
- The PEU signal is generated by the IM 304 if ***at least one*** interface reports "Not ready" (X14)
- An EU is connected to both interfaces (X21 and X22).

You can change the setting at jumpers X21, X22 as well as at X11, X14 and X15.

- You can switch the interfaces on or off with jumpers X21 and X22.

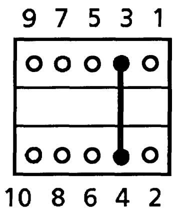
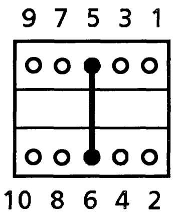
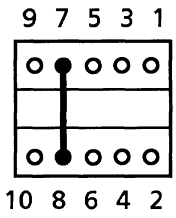
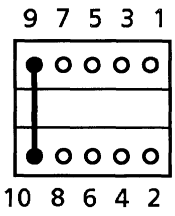


Interface is switched on



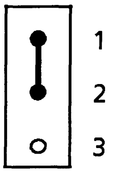
Interface is switched off
(no EU connected)

- Use jumper X11 to set the total length of the 721 connecting cables of one interface up to the last EU. The decisive factor for setting jumper X11 is the interface with the longest connection line.
  If you use IPs and CPs on the EU, you must set the longest cable length!

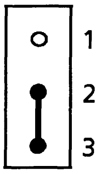
	Jumper X11			
Cable length	Up to 100 m	100 to 250 m	250 to 450 m	450 to 600 m
Jumper location				

- Jumpers X14 and X15 can be set as follows for the IM 304/314 distributed connection:

X14

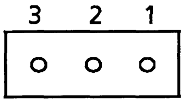


PEU signal is generated if ***at least one*** interface reports "Not ready".

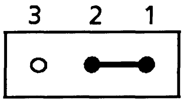


PEU signal is generated only if ***both*** interfaces report "Not ready".

X15



PEU signal is ***not*** evaluated.



PEU signal is evaluated.
Note: When power is turned on in the EU or in the CC, manual cold restart (RN-ST-RN) is also necessary.

Note

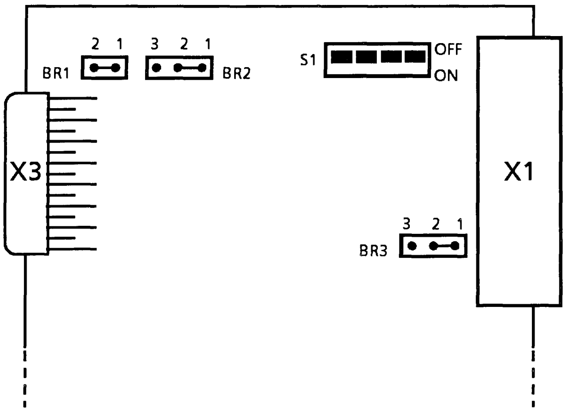
If the PEU signal is not evaluated, the following must be ensured at restart:

- The expansion unit is ready for operation before the central controller
  or
- A relevant waiting time must be programmed.

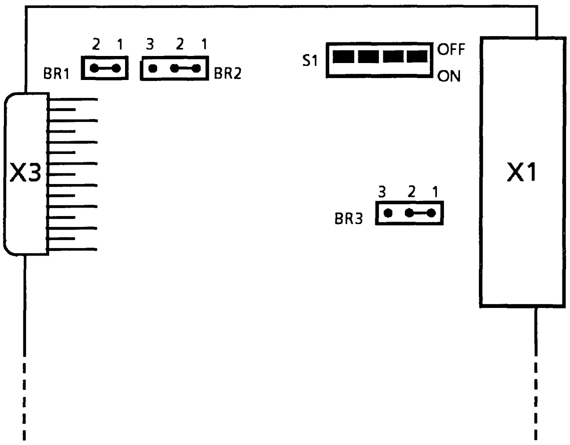
Switch and jumper settings on the IM 314 interface module for distributed connection

Jumpers BR1 to BR3 must be set as follows depending on the EU used:

Using the IM 314 in the ER 701-2. ER 701-3 (S5-115U)



Using the IM 314 in the EU 185U and EU 186U



Using the IM 314 in the EU 183U

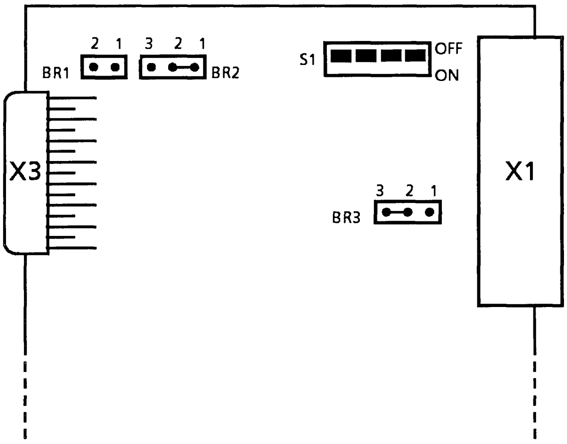


Figure 3-11 Jumper Settings on the IM 314

If you use the 313 watchdog module in the EU, you must switch off the PESP (memory I/O select) monitoring facility on the watchdog module.

The switch positions at S1 are preset on the IM 314 for the P area of module addresses. Figure 3-12 shows the switch position of S1 if you want to address the I/O modules in the O peripheral area on the ER or in the EU.

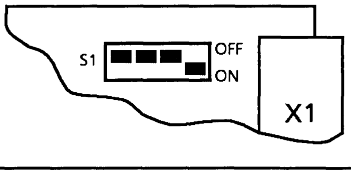


Figure 3-12 Switch Position on the IM 314 for Addressing in the O Peripheral Area

Distributed configuration with IM 304/IM 314 interface modules

After setting all the necessary switches and jumpers on the IM 404/IM 314 interface modules, you can set up the configuration in the PLC.

- Install the IM 304 on a CR 700-2/-3/-0LB mounting rack
- Install the IM 314 on an ER 701-2/-3 or EU 183/185/186 mounting rack
- Link the modules with the 6ES5 721-... connecting cable
- Plug a 6ES5-760-1AA11 termination connector into the lower front socket (X4) in the last IM 314 in each case.

Figure 3-13 shows a connection example.

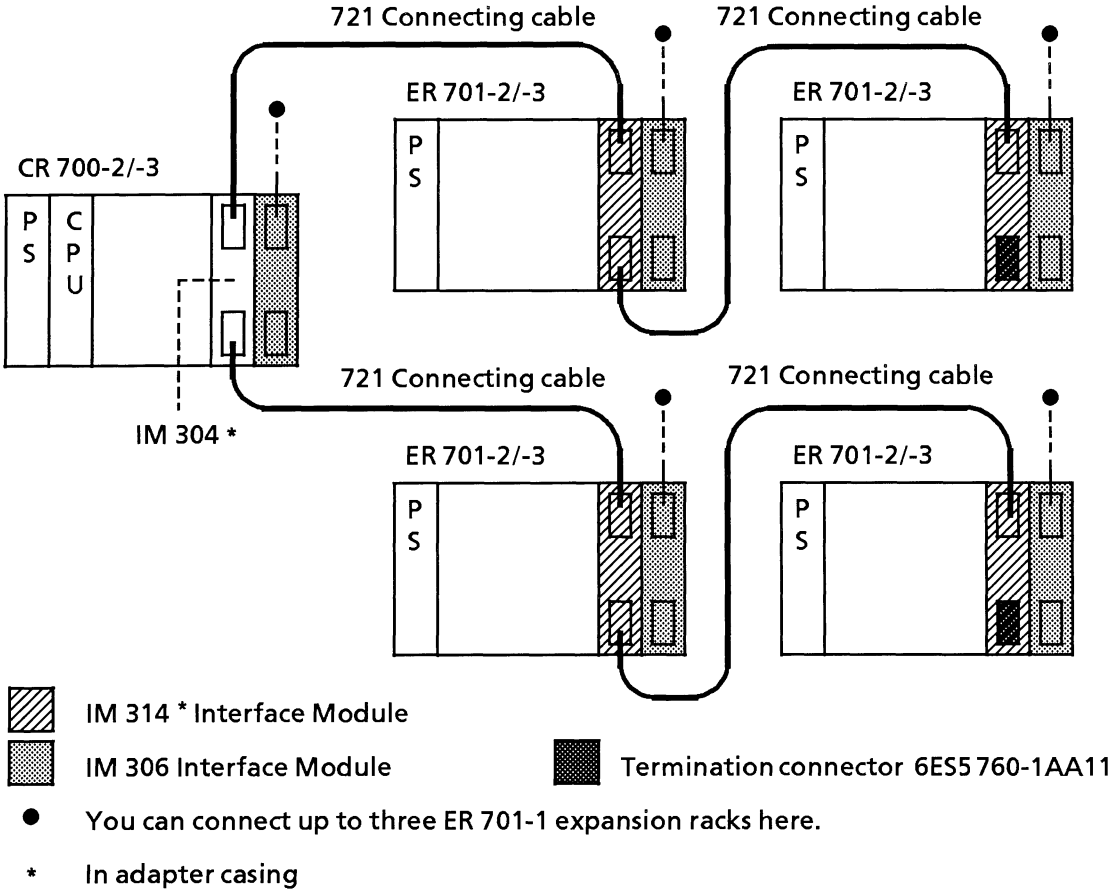


Figure 3-13 Distributed Configuration with IM 304/IM 314

Connections with the IM 304/IM 314 interface modules have the following special features:

- Using the IM 304/IM 314 symmetrical interface modules, you can connect EUs on ER 701-2 or ER 701-3 expansion racks with full address bus to CCs of the S5-115U, S5-135U, S5-150U, and S5-150S and S5-155U systems.
- Connection to EU 183, EU 185 and EU 186 is possible.
- The potential difference between CC and EU must not exceed 7 V. An equipotential bonding conductor should therefore be provided (see Section 3.5.8)!

3.3.3 Connection Possibilities with Other SIMATIC S5 Systems

Central controllers and expansion units of the S5-115U system can also be connected to CCs and EUs of other SIMATIC S5 systems. Table 3-5 shows the possible configurations.

Table 3-5 Connection of the S5-115U System to other SIMATIC S5 Systems

Configuration	Central Controller	Central Controller Interface Module	Expansion Unit	Expansion Unit Interface Module	Connecting Cable
Centralized up to 2.5 m (8 ft.)	110S 130A, 150A	6ES5 300-5LA11	EG1 (ER 701-1) or EG2 (ER 701-2 without PS)	6ES5 306-7LA11	705
	130K, 130W 135U, 150K 150S, 150U, 155U*	6ES5 300-5LB11			
Distributed up to 200 m (650 ft.)	130A, 150A	6ES5 301-5AA13	EG2 (ER 701-2) EG3 (ER 701-3)	6ES5 310-3AB11	721
	115U 130K, 130W 135U, 150K 150S, 150U,155U	6ES5 301-3AB13			
Distributed up to 600 m (2000 ft.) serial	135U 150S, 150U,155U	6ES5 304-3UA11		6ES5 314-3UA11	721
Distributed up to 1000 m (3800 ft.) serial	130A, 150A	6ES5 302-5AA11		6ES5 311-3KA11	723
	110S/B 130K/W 135U 150K/S/U ** 155U	6ES5 302-3KA11			

* No word-orientated I/O access (L PW; T PW) possible

** This connection is possible only if a cold restart is prevented by the "STP" statement in OB22

3.4 Wiring the Modules

The backplane on the mounting rack establishes the electrical connection between all modules.

Make the following additional wiring connections:

- The PS 951 power supply module to the power line
- The sensors and actuators to the digital or analog modules. Connect the sensors and actuators to a front connector that plugs into the contact pins on the front of each module. You can connect the signal lines to the front connector before or after you plug it into the module. The connection diagram of each module is on the inside of the front door. Perforated label strips are included with each input and output module. Use these strips to note the addresses of the individual channels on the module. Slip the strips along with their protective transparent covers into the guides on the front door.
  Chapter 10 "Analog Value Processing" describes how transducers are connected up to analog input modules and the feedback modules of the analog output modules.

The following sections explain how to connect individual modules.

Please consult the appropriate operator's guide or manual for information on wiring the intelligent input/output modules and communications processors.

3.4.1 Connecting the PS 951 Power Supply Module

Connect the PS 951 power supply module as follows:

① Set the voltage selector switch ① to the appropriate voltage (only in the case of AC modules).

② Connect the power cable to terminals L1, N and ⏚ .

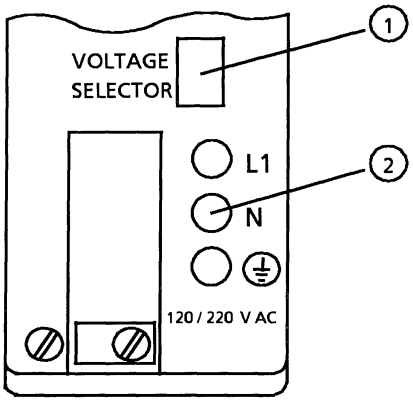


Figure 3-14 Power Supply Module PS 951

3.4.2 Connecting Digital Modules

Digital modules are available in nonfloating and floating versions. For the nonfloating modules, the reference voltage of the external process signals (M_{ext}) has to be connected to the internal reference voltage (M_{int}, i.e., PE) (see Figure 3.15). For floating modules, an optocoupler separates the external voltages from the internal ones.

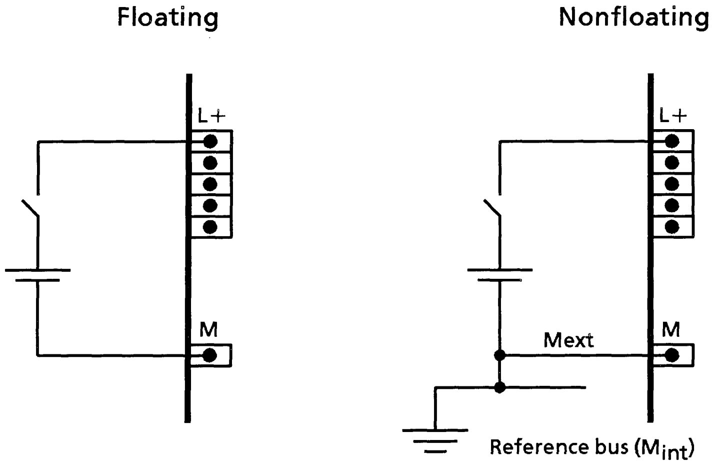


Figure 3-15 Connection to Floating and Nonfloating Modules

Note

See Chapter 6 "Addressing/Address Assignment" for information on address assignment in the case of digital modules.

3.4.3 Connecting Analog Modules

The connection of analog modules in described in Chapter 10 "Analog Value Processing".

3.4.4 Front Connectors

Various front connectors are available for wiring:

Table 3-6 Front Connector Overview

Order No.	Terminals per Front Connector	Connection Method	Wire Cross Section per Terminal[1]
6ES5 490-7LB11	24	Screw connection (SIGUT)	1 x (1.0 ... 2.5) mm^2 or 2 x (0.5 ... 1.5) mm^2 *
6ES5 490-7LC11	46	Spring-loaded connection	1 x (0.25 ... 1.5) mm^2 ** or max. 1.5 mm^2 in the case of combinations of conductors in one end sleeve
6ES5 490-7LB21	46	Screw connection*** (box terminal)	
6ES5 490-7LA11 (with crimp snap-in contacts)	46	Crimp snap-in (mini-spring contact)	1 x (0.5 ... 2.5) mm^2 or 2 x (0.5 ... 0.75) mm^2
6ES5 490-7LA12 (without crimp snap-in contacts)[2]			

[1] When plug-in jumpers are used, the conductor cross sections are reduced.

[2] Use crimp snap-in contacts with the order no. 6XX5 070 (Qty: 250)

* Flexible cable with end sleeves: 0.75 to 1.5 mm^2

** With end sleeves: 0.5 to 1.5 mm^2

*** 1.5 mm^2 with jumer comb

In general, we recommend use of end sleeves, especially where corrosion is to be expected.

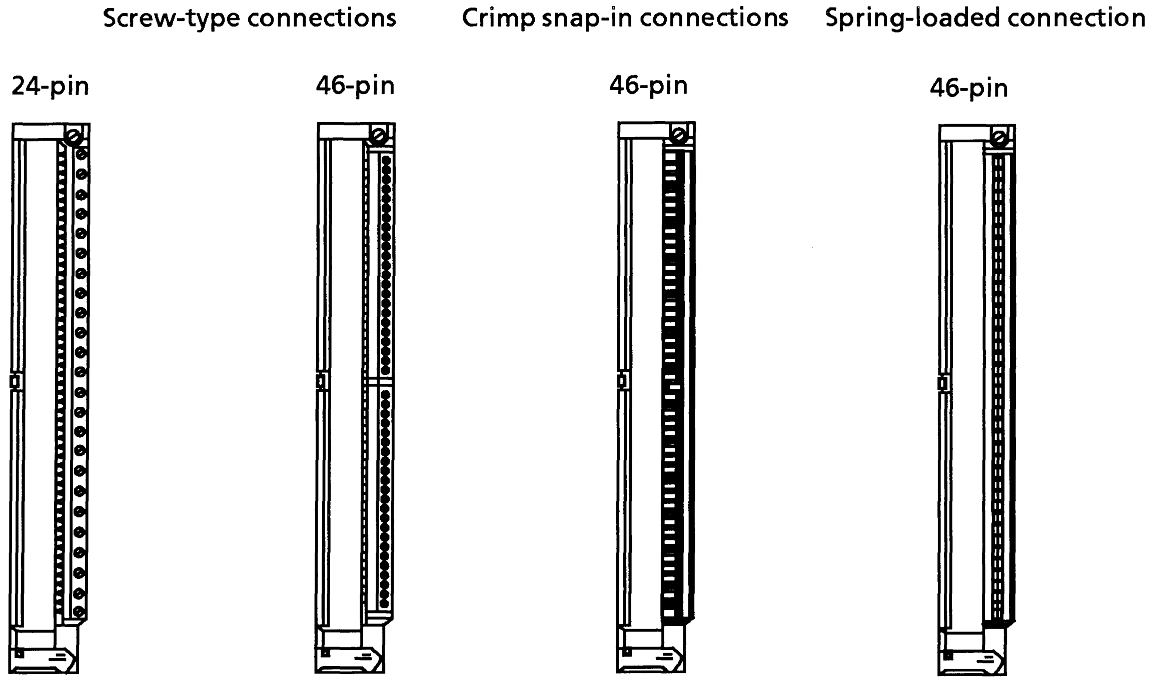


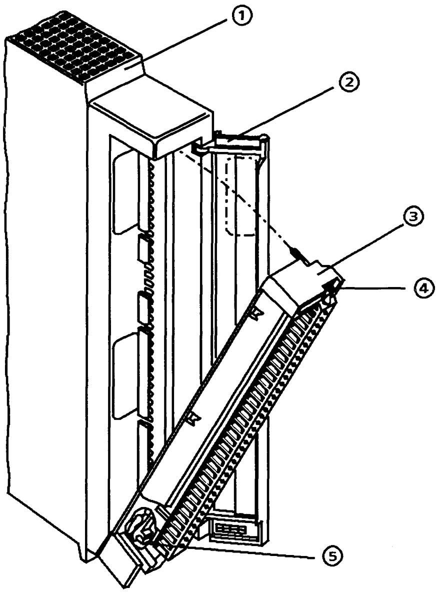
Figure 3-16 Front Connectors - Front View

The connectors have openings at the bottom for standard strain-relief clamps.

Installing the front connector

Install the front connector as follows:

1. Open the front door of the module.
2. Hook the front connector in the pivot at the bottom of the module.
3. Swing the front connector up and in until it engages with the module.
4. Tighten the screw at the top of the front connector to secure it.



① Module

② Front door is open

③ Front connector in pushed back

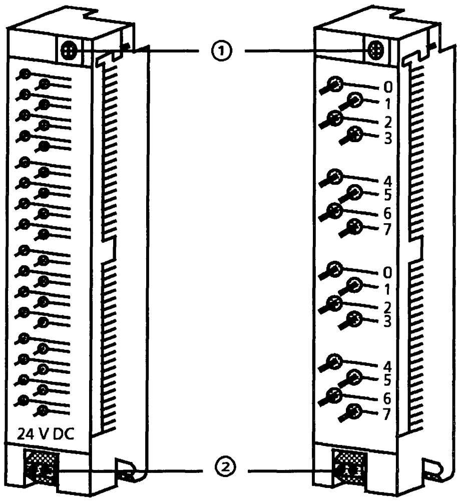
④ Fastening screw

⑤ Pivot

Figure 3-17 Installing the Front Connector

3.4.5 Simulator

You can use an appropriate simulator instead of a front connector. Use the toggle switches on the front of this device to simulate input signals (see Figure 3.18). A simulator needs an external power supply.
The simulators cannot be used in the case of mixed digital input/output modules or in the case of output modules.



① Fastening screw

② Screw-type terminals for supply voltage

Figure 3-18 Simulators

3.5 Guidelines for Interference-Free Design of the PLC

This chapter describes

- The circuits controller you must distinguish and the requirements made of the power supply module
- Connection and grounding concepts in the case of higher-level supply from grounded, centrally grounded and nongrounded mains
- Connection of power supply modules to nonfloating and floating modules
- Routing cables to ensure electromagnetic compatibility (EMC) and measures against interference voltage

3.5.1 Power Supply

You require the following for a completely configured SIMATIC S5 controller

- Power supply with internal PLC circuits (control power supply module)
  and
- Load power supply modules for the input and output circuits (load power supply module).

PS 951 control power supply module

The control power supply module supplies the following:

- CPU
- Programmer interface
- Control circuits of the I/O modules.

The following table gives you an overview of the power supply modules for an S5-115U.

Table 3-7 Overview of the Power Supply Modules

Input Voltage	Output Current	Galvanic Isolation	Power Supply Module	Output Voltages
120/230 V AC	3 A	Yes	6ES5 951-7LB21	5 V DC 5.2 V DC 24 V DC
120/230 V AC	7 A (15 A with fan)	Yes	6ES5 951-7LD21	
24 V DC	3 A	No	6ES5 951-7NB21	
24 V DC	7 A (15 A with fan)	No	6ES5 951-7ND51	
24 V DC	7 A (15 A with fan)	Yes	6ES5 951-7ND41	

Note

Please ensure that the control power supply module is not overloaded. Estimate the power consumption of all modules.

When using the different PS 951 power supply modules, you must note the following:

- For the 6ES5 951-7ND41 floating module, the input voltage must be a functional extra-low voltage in accordance with VDE 0100/5.73 § 8c or a comparable standard. Otherwise, the PE terminal must be conncted to the protective ground wire.
- For the 6ES5 951 7NB21/7ND41/7ND51 power supply modules, there is no galvanic isolation between the 24 V side and the 5 V side, whose reference potential is permanently connected to the mounting rack.
- The use of the following modules is ***not*** permissible due to the missing DSI signal in the case of the 3 A power supplies with the order numbers 6ES5 951-7LB14/7NB13:
  - IP 246/247
  - CP 513/526/527/535/580/581/143.
- The CP 524/524 must not be used with 3 A power supply modules since their power consumption is too high.
- Magnetic voltage stabilizers must ***not*** be connected direct on the input side of power supply modules!

  If you use magnetic voltage stabilizers in parallel network branches, you must expect overvoltages to occur as a result of mutual interference. These voltage peaks can destroy the power supply module! If such a case arises, please consult the department responsible.
- The power supply modules 6ES5 951-7LD21/7ND41/7ND51 have 2 backup batteries. If one of the batteries is discharged, the corresponding singal lamp lights up and the 2nd battery automatically takes over the backup function.
- You must observe the following for external backup in the case of the power supply modules with 2 batteries:
  - If you connect an external backup battery without inserting a new battery in the power supply module, the "BATT LOW" LEDs continue to flash.
  - Execute a RESET on the power supply module after connection of the external backup battery. You thus reset the battery low signal. The "BATT LOW" LEDs, however, continue to flash after the reset.
- You must observe the following for external backup in the case of the power supply modules with 1 battery:
  - Execute a RESET on the power supply module after connection of the external backup battery. You thus reset the battery low signal.

The load power supply

The load power supply supplies the following:

- Input/output circuits (load circuits)

  and
- Sensors and actuators.

⚠ **Warning**

For SIMATIC modules supplied with functional extra-low voltages (V ≤ 120 V DC, V ≤ 50 V AC), you require load power supply units with safe (electrical) isolation to DIN VDE 0106, Part 101. All Siemens power supply units of the 6EV1 range meet this requirement.

Dimensioning the load power supplies

The electronic short-circuit protection of DQ modules activates only when the triple nominal current has been exceeded. For this reason, dimension the load power supply units in such a way that the power supply can deliver the current required for switching off in the case of a short-circuit at an output.

If the load power supply unit has not been sufficiently dimensioned, this can result in a current higher than the nominal current flowing for an extended period in the case of a short-circuit at digital outputs, without the short-circuit protection of the DQ module activating.

Load power supply for nonfloating modules

If you use nonfloating modules, you must create a common reference potential for the internal control circuits of the PLC and for the load circuits. For this reason, connect the reference potential of the load power supply unit with the ground connection of the PLC (PE terminal or ⏚). The ground connection is permanently connected to the internal reference potential of the controller.

Load power supply for floating modules

> **Note**
>
> If you use switched-mode power supply units to supply floating analog modules and BEROs, you must first run this supply over a mains filter.

3.5.2 Electrical Installation with Field Devices

The following figures each show an example circuit for connecting control power supply and load power supply. They also show the grounding concept for operation from the following:

- Grounded supplies
- Centrally grounded supplies
- Nongrounded supplies.

Please note the following when installing your controller. The text contains reference numbers which you can find in Figures 3.19 to 3.21.

Master switch and short-circuit protection

- You must provide a master switch ① to DIN VDE 0113, Part 1, or a disonnecting device to DIN VDE 0100, Part 460, for the programmable controller, sensors and actuators.
  These devices are not required in the case of subsystems where the relevant device has been provided at a higher level.
- You can provide the circuits for sensors and actuators with short-circuit protection and/or overload protection ② in groups. According to DIN VDE 0100, Part 725, single-pole short-circuit protection is required in the case of grounded secondary side and all-pole protection is required in all other cases.
- For nonfloating input and output modules, connect terminal M of the load power supply unit with the PE ground conductor of the control circuit's PS 951 power supply module.

Load power supply

- For 24 V DC load circuits, you require a load power supply unit ③ with safe electrical isolation.
- You require a back-up capacitor ④ (rating: 200µF per 1 A load current) for nonstabilized load power supply units.
- For controllers with more than five electromagnetic operating coils, galvanic isolation by a transformer is required by DIN VDE 0113, Part 1; it is recommended by DIN VDE 0100, Part 725 ⑤.
- For nonfloating input and output modules, connect terminal M of the load power supply unit with the PE ground conductor of the control circuit's PS 951 power supply module.

Grounding

- You should ground load circuits where possible ⑥. Provide a removable connection to the protective conductor on the load power supply unit (terminal L- or M) or at the isolating transformer in secondary circuit.
- To protect against stray noise, use copper conductors of at least 10 mm^2 cross section to ground the mounting racks by the shortest possible route.

> ⚠ **Warning**
>
> You must provide insulation monitoring devices for nongrounded power supply modules
>
> - If hazardous plant conditions could arise from double-line-to-ground faults or double fault to frame faults
> - If no safe (electrical) isolation is provided
> - If circuits are operated with voltages > 120 V DC
> - If circuits are operated with voltages > 50 V AC.

- The mounting racks of the S5-115U must be connected to the protective conductor. This grounds the reference potential of the controller.
  Nongrounded operation of S5-115U controllers is only permissible if all the circuits are operated with functional extra-low voltage. In this case, connect the mounting rack or DIN rail over an RC network with the protective conductor.

Operating a programmable controller with field devices on grounded supply

Operation from grounded power supplies offers the best protection against interference.

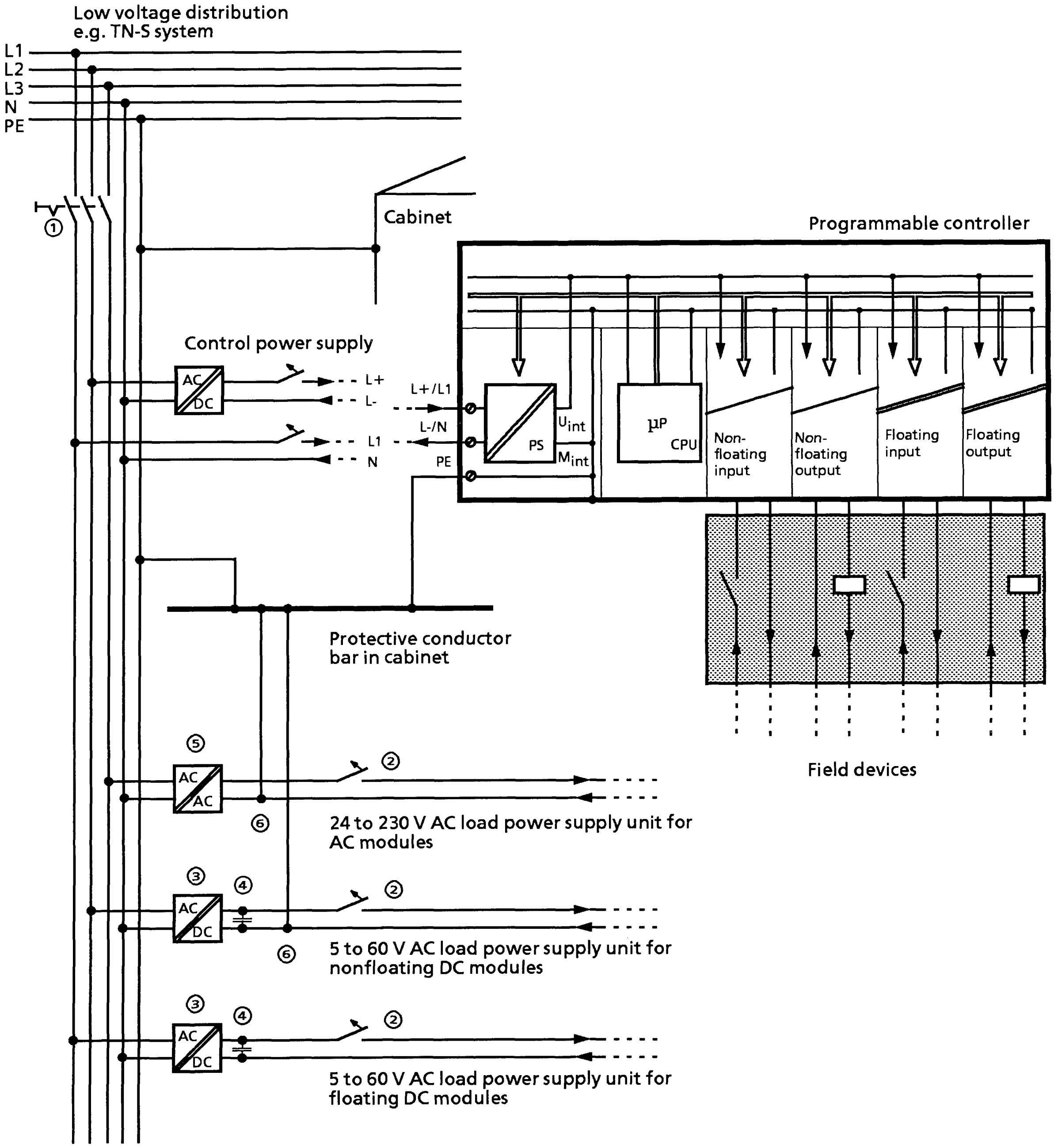


Figure 3-19 Operating a Programmable Controller with Field Devices on Grounded Supply

Operating a programmable controller with field devices on a centrally grounded supply

In plants with their own transformers or generators, the PLC is connected to the central grounding point. A removable connection must be provided for measuring ground faults.

Installation of the PLC must be such that there is insulation between the cabinet potential and the protective conductor potential. In order to maintain the insulation, all connected devices must be ***grounded capacitively or they must be nongrounded***. For this reason, programmers must be supplied only over an isolating transformer.

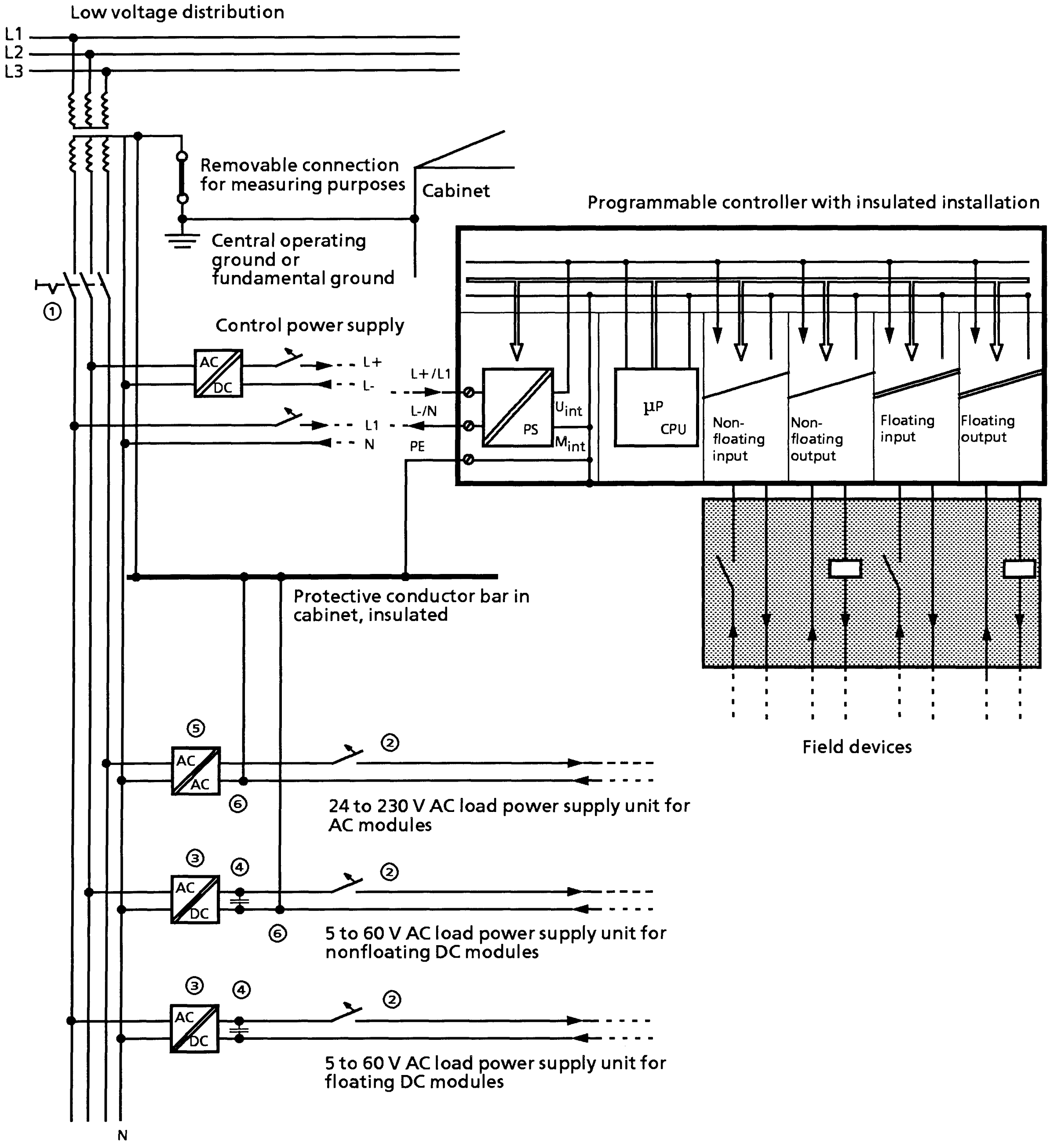

Low voltage distribution
L1
L2
L3
Removable connection for measuring purposes
Cabinet
Central operating ground or fundamental ground
①
Control power supply
AC
DC
L+
L-
L+/L1
L1
N
L-/N
PE
Programmable controller with insulated installation
PS
Uint
Mint
µP
CPU
Non-floating input
Non-floating output
Floating input
Floating output
Field devices
Protective conductor bar in cabinet, insulated
⑤
AC
AC
⑥
②
24 to 230 V AC load power supply unit for AC modules
③
④
AC
DC
②
⑥
5 to 60 V AC load power supply unit for nonfloating DC modules
③
④
AC
DC
②
5 to 60 V AC load power supply unit for floating DC modules
N

Figure 3-20 Operating a Programmable Controller with Field Devices on Centrally Grounded Supply

Operating a Programmable Controller with Field Devices on Ungrounded Supply

Neither the outer conductor nor the neutral are connected to the protective conductor in the case of nongrounded supplies. Operation of the PLC with nonfloating power supply modules is ***not permissible.***

Please note the following when connecting power supply modules:

In networks with 3 x 230 V, you can connect the power supply module direct to two outer conductors (see Figure 3.21).

In networks with 3 x 400 V, connection between the outer conductor and the neutral conductor is not permissible (unacceptably high voltage in the case of ground fault). Use intermediate transformers in these networks.

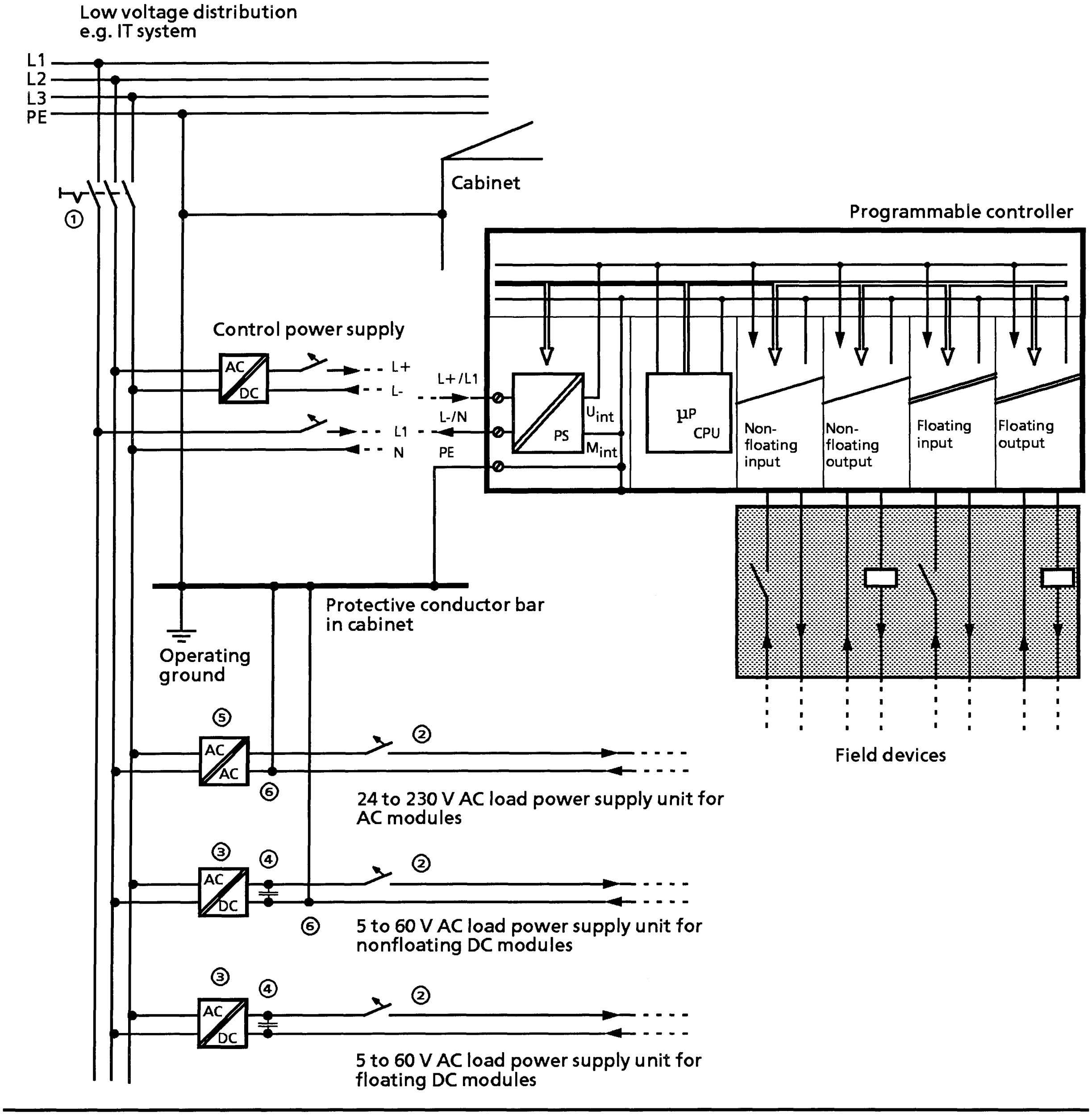


Figure 3-21 Operating a Programmable Controller with Field Devices on Nongrounded Supply

3.5.3 Connecting Nonfloating and Floating Modules

The following sections show the special features involved in installations with nonfloating and floating modules.

Installation with nonfloating modules

In installations with nonfloating modules, the reference potential of the control circuit ($M_{internal}$) and the load circuits ($M_{external}$) are not galvanically isolated.

The reference potential of the control circuit ($M_{internal}$) is at the PE terminal or ⏚ and must be connected to the reference potential of the load circuit via a line to be run externally.

Figure 3.22 shows a simplified representation of an installation with nonfloating modules. The installation is independent of the grounding concept. The connections for the grounding measures are therefore ***not*** shown:

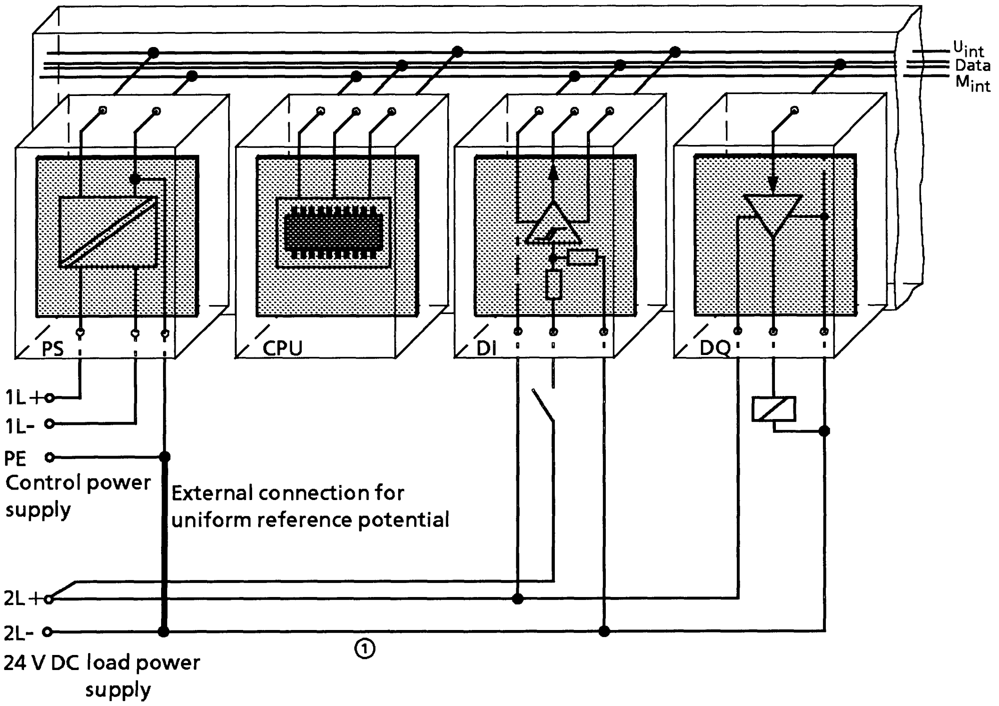

Uint
Data
Mint
PS
CPU
DI
DQ
1L+
1L-
PE
Control power supply
External connection for uniform reference potential
2L+
2L-
24 V DC load power supply
①

Figure 3-22 Simplified Representation of an Installation with Nonfloating Modules

Voltage drop on line ① must not exceed 1 V, otherwise the reference potentials will shift and result in failures of the module.

Note

It is imperative that you connect the reference potential of the load power supply unit with the L- terminal of the module in the case of 24 V DC DQ modules. If this connection is missing (e.g. wirebreak), a current of typically 15 mA can flow at the outputs. This output current can be sufficient to ensure that

- Energized contactors do not drop out
  and
- High-resistance loads (e.g. miniature relays) can be driven.

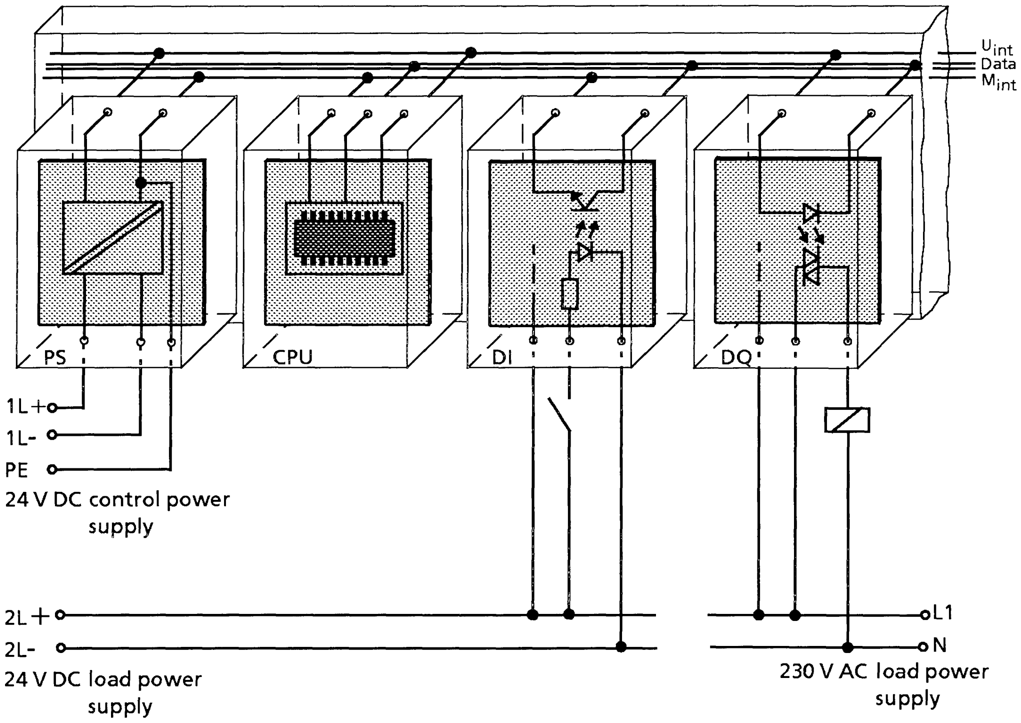
Installation with floating modules

Control circuit and load circuit are galvanically isolated in the case of floating modules.

Installation with floating modules is necessary in the following cases:

- All AC load circuits
  and
- Non-connectable DC load circuits.
  The reasons for this are, e.g. different reference potentials of the sensors or the grounding of the plus poles of a battery, ...

Figure 3-23 shows the simplified representation of an installation with floating modules. The installation is independent of the grounding concept. The connections for grounding measures are therefore ***not*** shown.



U_{int}
Data
M_{int}
PS
CPU
DI
DQ
1L+
1L-
PE
24 V DC control power supply
2L+
2L-
24 V DC load power supply
L1
N
230 V AC load power supply

Figure 3-23 Simplified Representation for Installation with Floating Modules

3.5.4 Running Cables Inside a Cabinet

Dividing the lines into the following groups and running the groups separately will help you to achieve electromagnetic compatibility (EMC).

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

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

Group C: Unshielded lines for AC voltage > 400 V

Group D: Lines for SINEC H1

You can use the following table to see the conditions which apply to the running of the various combinations of line groups.

Table 3-8 Rules for Common Running of Lines

	Group A	Group B	Group C	Group D
Group A	①	②	③	④
Group B	②	①	③	④
Group C	③	③	①	④
Group D	④	④	④	①

Legend for table:

① Lines can be run in common bundles or cable ducts

② Lines must be run in separate bundles or cable ducts (without minimum distance)

③ Inside cabinets, lines must be run in separate bundles or cable ducts and outside cabinets but inside buildings, lines must be run on separate cable trays with a gap of a least of 10 cm between lines.

④ Lines must be run in separate bundles or cable ducts with at least 50 cm between lines.

3.5.5 Running Lines Outside Buildings

Run lines outside buildings where possible in metal cable supports. Connect the abutting surfaces of the cable supports galvanically with each other and ground the cable supports.

When you run cables outdoors, you must observe the regulations governing lightning protection and grounding.

Lightning protection

If cables and lines for SIMATIC S5 devices are to be run outside buildings, you must take measures to ensure internal and external lightning protection.

Outside buildings run your cables
either

- In metal conduit grounded at both ends
  or
- In steel-reinforced concrete cable channels

Protect signal lines from overvoltage by using:

- Varistors
  or
- Lightning arresters filled with inert gas

Install these protective elements at the point where the cable enters the building.

> **Note**
>
> Lightning protection measures always require an individual assessment of the entire system. If you have any questions, please consult your local Siemens branch office or any company specializing in lightning protection.

3.5.6 Taking Measures Against Interference Voltage

Frequently, measures to suppress interference voltage are taken only after the control system is in operation and problems develop with the reception of an information signal. You can reduce the effort involved in such measures (e.g. using special contactors) significantly if you note the following points when you are configuring your control system:

- Arrange devices and wiring spaciously
- Ground all inactive metal parts to chassis
- Filter power lines and signal lines
- Shield devices and wiring
- Take special measures to suppress interference

Arranging devices and wiring spaciously

Sufficiently attenuating direct-current or alternating magnetic fields of low frequency (e.g. 50 Hz) is expensive. You can often solve this problem by maintaining the largest possible clearance between the source of the interference and any potentially susceptible device.

Chassis grounding all inactive metal parts

Proper chassis grounding is an important factor in ensuring that your installation is immune to interference. Chassis grounding refers to the conductive connection of all inactive metal parts (VDE 0160). Always use surface-contact grounding. Ground all conductive inactive metal parts to chassis.

When chassis grounding, please note the following:

- Connect inactive metal parts with as much care as active parts
- Ensure low-resistance metal to metal connections, e.g. large-surface, good-conductive contacts
- If painted or anodized metal parts are involved in grounding, these protective insulating layers must be penetrated. Use special contact washers for this purpose or remove the insulating layers.
- Protect the connected surfaces from corrosion, e.g. by using grease
- Movable grounded parts (e.g. cabinet doors) are to be connected over flexible grounding strips. The grounding strips should be short and they should have a large surface area since the surface area is a decisive factor in discharging high-frequency interference.

3.5.7 Shielding Devices and Cables

Shielding is a measure to weaken (attenuate) magnetic, electric or electromagnetic interference fields. Shielding can be divided into the following two categories:

- Device shielding
- Line shielding

Shielding devices

- You must connect inactive metal parts (e.g. cabinet doors and supporting plates) with grounding strips. The grounding strips should be short and they should have a large surface area.
- There must be a large surface metal to metal connection between the supporting bar (of the cabinet) and the fixing bracket of the mounting rack.
- In the case of shielded signal lines, the shield must be secured with cable clamps to the protective conductor bar or an additionally secured shield bar.
- Cable clamps must grip and connect with the shield braiding over a large surface area
- The line to the protective conductor system (ground point) must have a large area of contact with the protective conductor system.

Shielding cables

Shielding is a measure to weaken (attenuate) magnetic, electric or electromagnetic interference fields.

Interference currents on cable shields are discharged to ground over the shield bar which has a conductive connection to the housing. So that these interference currents do not become a source of noise in themselves, a low-resistance connection to the protective conductor is of special importance.

Use only cables with shield braiding if possible. The effectiveness of the shield should be more than 80%. Avoid cables with foil shielding since the foil can easily be damaged by tension and pressure; this leads to a reduction in the shielding effect.

As a rule, you should always shield cables at both ends. Only shielding at both ends provides good suppression in the high frequency range.

As an exception only, you can connect the shielding at one end. However, this attenuates only the lower frequencies. Shielding at one end can be of advantage in the following cases:

- If you cannot run an equipotential bonding conductor
- If you are transmitting analog signals (e.g. a few microvolts or microamps)
- If you are using foil shields (static shields).

Always use metallic or metalized connectors for data lines for serial connections. Secure the shield of the data line at the connector housing. Do ***not*** connect the shield to the PIN1 of the connector strip!
In the case of stationary operation, you are recommended to insulate the shielded cable without interrupt and to connect it to the shield/protective ground bar.

Note

If there are fluctuations in potential to ground, a compensating current can flow over the shielding that is connected at both ends. For this reason, connect an additional equipotential bonding conductor.

3.5.8 Equipotential Bonding in the Case of Distributed Configurations

For distributed configurations, differentiate between the following cases:

- Separate arrangement of central controllers and expansion units when connected by the 301/310 interface modules (up to 200 m/656.2 ft) or the 304/314 interface modules (600 m/1986 ft)
  The 301/310 and 304/314 interface modules are ***not*** floating.
  If the potential difference between the devices can be more than 7 V, you must provide an equipotential bonding line.
  Use the following dimensions for the cross-section of the equipotential bonding line:
  - 16 mm^2 copper wire for equipotential bonding line up to 200 m/656.2 ft
  - 25 mm^2 copper wire for equipotential bonding line over 200 m/656.2 ft

  The equipotential bonding line should be run in such a way as to achieve the smallest possible surfaces between equipotential bonding line and signal lines
- Separate arrangement of central controllers and expansion units when connected serially by the 302/311, IM 308/318 interface modules or IM 307/317 fiber optic link.
  These interface modules are floating. In this case, equipotential bonding line are not required.
- Signal transfer between separate systems via input and output modules.
  Floating input and output modules must be used for signal transfer. Equipotential bonding lines are not required here either.

3.5.9 Special Measures for Interference-Free Operation

Arc suppression elements for inductive circuits

Normally, inductive circuits (e.g. contactor or relay coils) energized by SIMATIC S5 do not require to be provided with external arc suppressing elements since the necessary suppressing elements are already integrated on the modules.

It only becomes necessary to provide supressing elements for inductive circuits in the following cases:

- If SIMATIC S5 output circuits can be switched off by additionaly inserted contactors (e.g. relay contactors). In such a case, the integral suppressing elements on the modules become ineffective.
- If the inductive circuits are ***not*** energized by SIMATIC S5.

You can use free-wheeling diodes, varistors or RC elements for wiring inductive circuits.

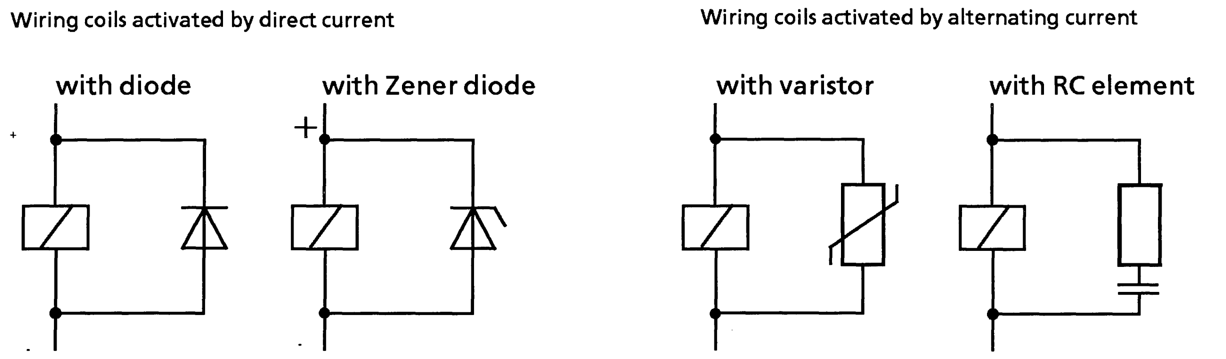

Wiring coils activated by direct current
Wiring coils activated by alternating current
with diode
with Zener diode
with varistor
with RC element
+
-
+
-

Figure 3-24 Wiring Coils

Mains connection for programmers

Provide a power connection for a programmer in each cabinet. The plug must be supplied from the distribution line to which the protective ground for the cabinet is connected.

Cabinet lighting

Use, for example, LINESTRA® lamps for cabinet lighting. Avoid the use of fluorescent lamps since these generate interference fields. If you cannot do without fluorescent lamps, you must take the measures shown in Figure 3.25.

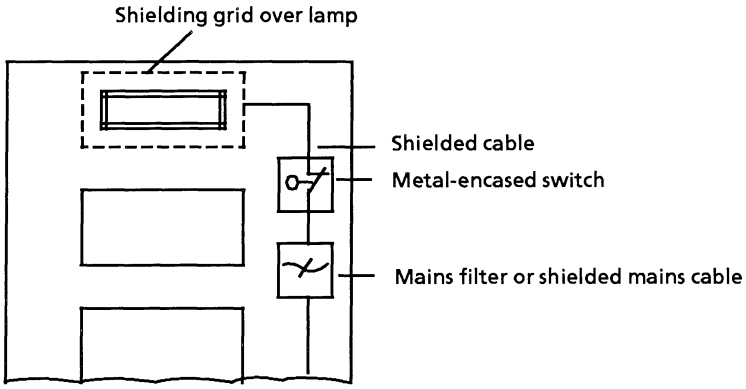


Figure 3-25 Measures for Suppressing Interference from Fluorescent Lamps in the Cabinet

3.6 Safety Measures and Monitoring Facilities

When configuring programmable controllers - and also contactor controllers - follow the relevant VDE regulations (e.g. DIN VDE 0100, DIN VDE 0113 Part 1 (corresponds to IEC 204-1)). Pay special attention to the following measures for avoiding dangers:

- Prevent conditions that can endanger people or property.
- When power is restored after a power failure or after EMERGENCY OFF devices are released, machines must not be able to restart automatically.
- When a PLC malfunctions, commands for EMERGENCY OFF devices and safety limit switches must remain effective under all circumstances. These safety measures must affect the actuators in the power circuit directly.
- When EMERGENCY OFF devices are activated, safety must be guaranteed for personnel and the controlled system as follows:
  - Actuators and drives that could cause dangerous situations (e.g. main spindle drives for machine tools) must be shut off.
  - On the other hand, actuators and drives that could endanger persons or the controlled system by being shut off (e.g. clamping devices) must not be shut off by EMERGENCY OFF devices.
- The programmable controller must be able to record the activation of EMERGENCY OFF equipment and the user program must be able to evaluate it.

Protection in case of indirect contact

Accessible parts must not be dangerous to touch even in the event of a fault. Such parts must be included in measures for protection against electrical shock.

This requirement is met if all accessible metal parts, such as standard sectional rail, supporting bar and the cabinet itself, which could all be dangerous to touch in the event of a fault, are made electrically safe and connected to protective ground (PE). Maximum permissible resistance between the protective conductor connection and the accessible part to be protected is 0.5 ohms.

4 Testing and Loading the Control Program and Starting Up a System

Figures

Tables

4 Testing and Loading the Control Program and Starting Up a System

This chapter contains notes on starting up an S5-115U with the CPU 945.
It describes how to

- Test your STEP 5 control program
- Load it into the PLC
- Start it.

Knowledge of the principle of operation of the PLC is a prerequisite (see Chapter 2).
There are notes on configuring and starting up a system at the end of the chapter. There is also a short description of active and passive faults of a programmable control system.

4.1 Prerequisites for Starting Up the PLC

Check the following before starting to test and load your control program:

- All required I/O modules must be plugged into suitable slots (see Chapter 3)
- The address assignment of the inputs and outputs must be in order (see Chapter 6)
- Your system must have interference-free wiring (see Chapter 3)
- The control program to be tested must be available on the programmer.

4.2 Testing the Control Program

Before loading your control program into the PLC, you must test it.

In this Chapter, we will show you the procedure for testing the control program.
This is followed by a description of the test functions for finding logical errors in program execution.

4.2.1 Testing the Control Program

Note

Always test your control program with the load circuits switched off. You can check the function of the outputs by the LEDs on the output modules.

- ▸ Overall reset of the PLC before transferring the control program (see Section 4.3.1)
- ▸ Transfer of the control program **without** the OBs (see Section 4.3.2)
- ▸ Transfer of OB21 and OB22 and testing of the restart characteristics of the PLC at cold restart
- ▸ If you have programmed the function blocks yourself, you should call them up individually in OB1, parameterize them and test them.
- ▸ Finally, call each block from the control program individually in OB1 and test it. Always start with the last block of a block chain. Please note that the relevant DBs will also be called.

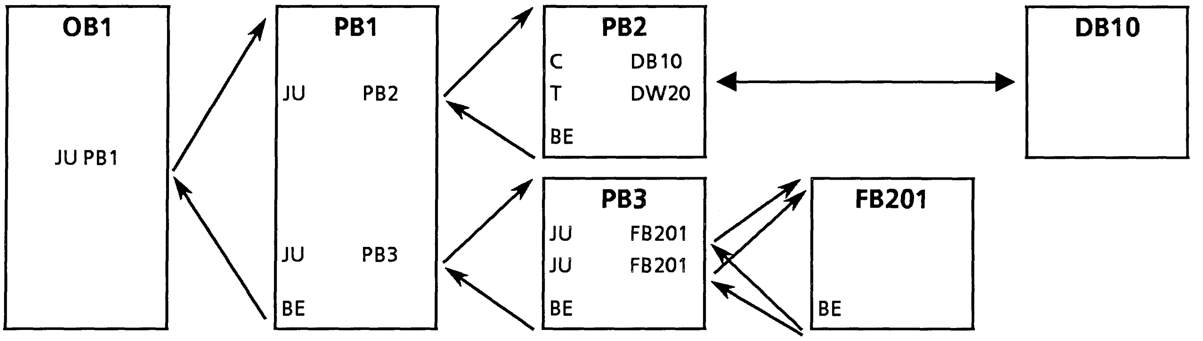


Figure 4.1 Typical Program Structure

You can proceed as follows with the program structure represented in Figure 4.1:

- Call and test FB201 in OB1
- Call PB2 in OB1 and test
- Call PB3 in OB1 and test
- If PB2 and PB3 are free of errors, call PB1 in OB1 and test it. This means that all blocks of the PB1, PB2 and PB3 chain have been tested.

You can test the entire control program using this procedure.

4.2.2 "Program Test" Function

This programmer function causes the CPU to process a control program step by step.
You can set a breakpoint in the control program. This breakpoint is a statement in the program which you specify with the cursor. The control program will then be processed up to this breakpoint by the CPU.

The CPU scans the program up to the selected statement. The current signal states and the RLO are displayed up to the selected statement.

You can scan the program in sections by shifting the breakpoint as you require.

Program scanning takes place as follows:

- All jumps in the block called are traced.
- Blocks called are executed without delay.
- Program scanning is terminated automatically when block end (BE) is reached.

The following applies during Program Test:

- Both operating mode LEDs are off
- All outputs are switched off. The "BASP" LED lights up.
- Inputs and outputs are not scanned. The program writes to the PIQ and reads the PII.

Corrections are not possible during Program Test. However, the following test and PLC functions can be executed:

- Input and output (program modifications are possible)
- Direct signal status display (STATUS VAR)
- Forcing outputs and variables (FORCE, FORCE VAR)
- Information functions (ISTACK, BSTACK)

If the Program Test function is interrupted by PLC or program errors, the PLC goes into the STOP mode and the corresponding LED lights up on the CPU control panel.

Consult the relevant manual for information on calling the Program Test function on a programmer.

4.2.3 STATUS/STATUS VAR Test Function

The STATUS and STATUS VAR test functions indicate signal states of operands and the RLO. Depending on when the signal states are observed, a distinction is made between program-dependent signal status display (STATUS) and direct signal status display (STATUS VAR).

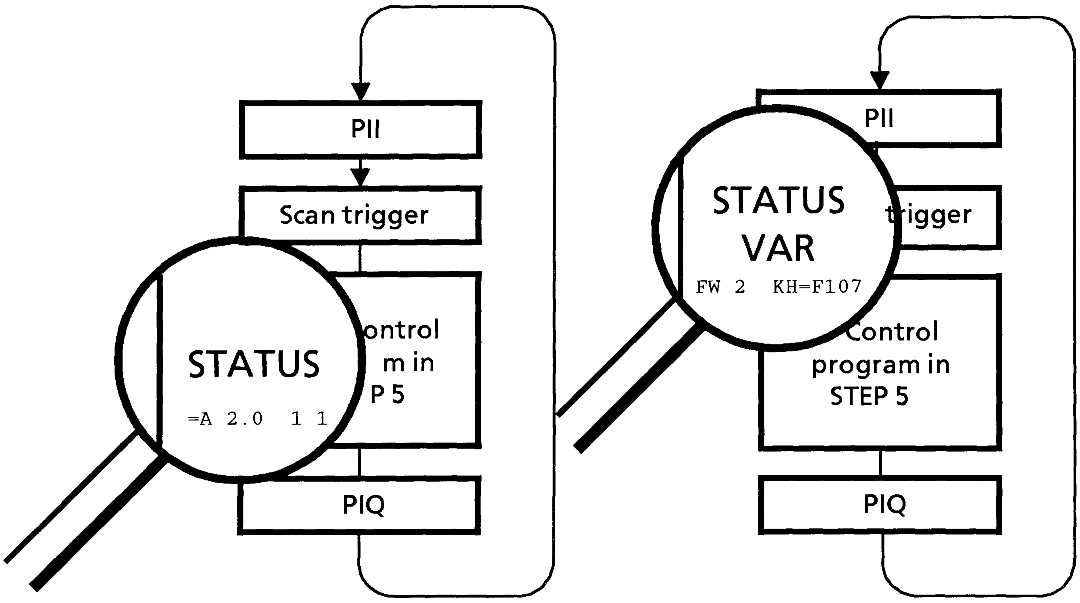


Figure 4.2 Comparison of the "STATUS" and "STATUS VAR" Test Functions

Program-dependent signal status display "STATUS"

This test function shows the current signal states and the RLO of the individual operands during program execution. When FBs are called, the values of the actual operands are shown.
In addition, program corrections can be made.

Direct signal status display "STATUS VAR"

This test function gives the status of any operand (inputs, outputs, flags, S flags, data word, counters or timers) at the end of a program scan, i.e. at the cycle control point. This information is taken from the relevant data area of the operands involved. During "Program Test" or in the STOP mode, the inputs are read direct. In other cases, only the process I/O image of the called operands is shown.

4.2.4 FORCE Outputs and Variables

"FORCE" outputs

All outputs are set to "0" at the start and the end of the FORCE function. You can set outputs to a specific signal state directly without using the control program. Use this direct method to check the wiring and functioning of output modules. This procedure does not change the process image but it does cancel the output disable state.

Note

For the "FORCE" test function, the PLC must be either set to the Program Test function or in the STOP mode. The function must only be executed without the load voltage.

"FORCE" variables

With the "FORCE VAR" test function, the process image of binary and digital operands is modified regardless of the PLC mode.
The following variables can be modified: I, Q, F, S, T, C, and D.
Program scanning with the modified process variables is executed in the RUN mode. However, the variables can be modified again in the remaining program run, without a checkback signal. Process variables are forced asynchronously to the program run.

4.2.5 Points to Note When Using the 2nd Interface as a Programmer Interface

If you use an interface module at the second interface, you connect programmers and operator panels to SI 1 and SI 2.
Connecting a programmer, OP or SINEC L1 to SI 1 or SI 2 has a negligible effect on the program scan time (see Section 2.9.3).

There are restrictions on the simultaneous use of interfaces SI 1 and SI 2 as programmer interfaces. Depending on the status (activity) of an interface, certain requests from the programmer/OP to the other interface are at times not possible.

If this fault occurs, the function of the relevant interface will be aborted by the CPU. The "Interface function disabled: current function" error message appears.

This message draws your attention to the fact that a function is running on the interface and blocking the requested function.

Example: If "STATUS BLOCK" is running on SI 1, "INPUT BLOCK" cannot be performed on the same block on SI 2.

4.3 Loading the Control Program

The following sections describe:

- Overall reset of the PLC
- The two methods of transferring the control program to the PLC
- The retentive characteristics of the timers, counters and flags
- Activation of software protection
- Start of the program.

4.3.1 Overall Reset

You are recommended to perform the Overall Reset function before entering a new program. Overall Reset deletes the following:

- PLC program memory
- All data (flags, S flags, timers and counters)
- All error IDs.

In addition, all system data is automatically assigned default values after Overall Reset so that the system data area assumes a defined "basic status".

The extended system data area (RT) is not deleted.

There are two ways of deleting the internal program memory:

- Offline via the switch for "Default/Overall Reset"
- Online with the "Delete" programmer function.

Overall Reset via the switch for "Default/Overall Reset" on the control panel of the CPU

- Switch on the power supply module
- Set the CPU mode selector to STOP (ST)
- Set the switch for "Default/Overall Reset" to the "OR" position and hold it in this position (if the switch is not held in position it will automatically spring back to the "RE" position).

While you hold down the switch for "Default/Overall Reset" in the "OR" position:

- Switch the CPU mode selector twice from "ST" to "RN".

The STOP LED will momentarily go off.

- Release the switch for "Default/Overall Reset".

The switch automatically springs back to the "RE" position.

Overall reset has now been performed on the internal program memory.

Overall Reset with the "Delete" programmer function

- Link the programmer and the CPU over a suitable connection cable
- Switch on the PLC power supply module
- Set the CPU mode selector to "ST"
  or
  set the CPU to the STOP state using the STOP programmer function.
- Call the "Delete" auxiliary function on the programmer and select "Delete all blocks"

Overall reset has now been performed on the internal program memory.

4.3.2 Transferring the Program

There are two ways of transferring the control program to the CPU:

- Transfer the control program to the memory submodule
  and then insert the memory submodule in the receptacle on the CPU. The CPU copies the contents of the memory submodule into internal program memory after POWER UP or Overall Reset so that they can be processed at very high speed. This method allows you to start up a PLC or a system without using a programmer.
  Note: The memory submodule may be plugged in and removed in the POWER OFF state only!
- Transfer the control program into the internal program memory of the CPU using a programmer.

Transferring the program to a memory submodule

You require the following to program a memory submodule

- A programmer with the S5-DOS "EPROM/EEPROM" PACKAGE
- An adapter (6ES5 985-2MC11) which is plugged into the submodule receptacle of the programmer, if the programmer has no receptacle for S5 memory cards.

Transferring your STEP 5 control program to a memory submodule:

- ▸ Plug the adapter for the memory submodule into the submodule receptacle of the programmer (if required).
- ▸ Plug the memory submodule into the programmer or the adapter and program it with the S5-DOS "EPROM/EEPROM" PACKAGE. This package is explained in detail in the programmer manuals.
- ▸ After programming the memory submodule, plug it into the submodule receptacle of the CPU when the PLC is switched off.
- ▸ Switch on the power supply of the PLC.

  If there are still valid blocks in internal program memory after POWER UP, the following happens:

  - Blocks loaded from flash EPROM have the ID "Block in EPROM" in internal program memory. These blocks are deleted after POWER UP and then loaded again from the flash EPROM.
  - Blocks with the ID "Block valid in RAM" are retained in internal program memory after POWER UP.
  - Before the blocks are loaded into internal program memory from the memory submodule, the CPU compresses the internal program memory (Compress function)!

- ▸ Overall reset

Point to note after Overall Reset:

After Overall Reset, the STEP 5 control program is loaded automatically from the memory submodule into the internal program memory of the CPU.

Transferring the program into the internal program memory of the CPU using the programmer

If you transfer the control program into the internal program memory of the CPU using the programmer, you must do the following:

- Connect the programmer to the CPU with a suitable connecting cable.
  Connection is possible
  - over interface 1
    or
  - over the programmer interface module of SI 2
    Prerequisite for programmer connection to the interface module s that none of the following functions is active:
    - ASCII driver
    - Point-to-point master function
    - Computer interface
- Switch on the power supply of the PLC
- Check that the backup battery has been inserted and is functional

Note

It is possible for an internal passivation coating to develop in new lithium batteries or in lithium batteries left unused for long periods. This coating has the effect of substantially increasing the internal resistance.
Remedy: Depassivate the battery by loading it for approx. 2 hours with 100 ohms.

- Select operating mode "Online" in the Presets (Defaults) screen form
- Transfer the program from the programmer to the CPU with the "Transfer" function

Note

Transfer takes place in the RUN or STOP state of the CPU. If you transfer blocks in the RUN state, you should:

- transfer tested blocks only
- transfer blocks in the correct order so that the CPU does not enter the STOP state (e.g. first the data blocks, then function blocks and lastly blocks which use these data and function blocks).

If blocks of the same name are already in the internal program memory of the CPU, the following message appears in the message line "... already in the PLC, overwrite?"
By pressing the transfer key again, a new block is transferred to the program memory of the CPU and the old block declared invalid. Invalid blocks continue to take up memory space and can only be deleted using "Overall reset" or "Compress" (see Chapter 2).

Points to note when setting up data blocks

- Data blocks generated in the control program with the "G DB" or "GX DX" operation are automatically dumped by the operating system direct in internal program memory.
  The contents of the data blocks can be changed using STEP 5 operations.

- The contents of the memory submodule are copied into the internal program memory after POWER UP and after Overall Reset; this means that the contents of the data blocks can also be changed. However, after POWER UP (and after Overall Reset), the "old" data blcoks are copied from the memory submodule into the internal program memory; the "current" contents are lost.

4.3.3 Activating Software Protection

Activate software protection by setting bit 0 in system data word 120 (E $10F0_H$). You can use the software protection facility in RAM operation, i.e. when the control program is not stored on the memory submodule (flash EPROM).

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		RS 120

0: Software protection not active
1: Software protection active

x = Bits which determine system characteristics (must not be changed when software protection is active!)

Figure 4.3 Activating Software Protection by Setting Bit 0 in System Data Word 120

Software protection prevents the following:
- Input and output of blocks by the programmer
  and
- Deletion of blocks by the programmer.

Software protection can be activated in the following ways:
- In DB1 via the "PROT" parameter (see Chapter 11)
- In the control program (e.g. with the SU RT operation)
  or
- with the programmer function "Display memory contents".

Bit 0 in system data word 120 can be reset only in the following cases:
- In the control program (e.g. with the RU RS operation)
  or
- By Overall Reset.

4.3.4 Determining the Retentive Feature of Timers, Counters, Flags and S Flags

Use the "Presetting the retentive feature/Overall Reset" switch on the operator panel of the CPU to determine the behaviour of timers, counters, flags and S flags at cold restart (both manually and automatically after power restore).

Timers, counters, flags and S flags are "retentive" if they do ***not*** lose their contents at cold restart. Those timers, counters, flags and S flags which are reset at cold restart are "nonretentive".

The following retentive feature is set as default ***after Overall Reset***:
All timers, counters, flags and S flags are nonretentive in the NR switch position. In the RE switch position, half of all timers, counters, flags and S flags are retentive:

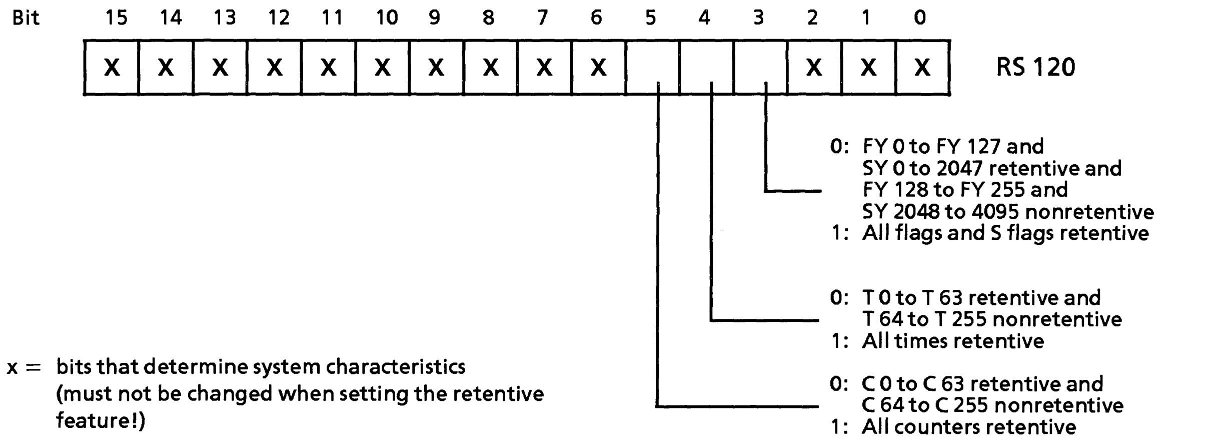
Table 4.1 Preset Retentive Feature of the CPU after Overall Reset

Switch Position	Flags	S Flags	Timers	Zähler
RE (retentive)	F 0.0 to 127.7 retentive F128.0 to 255.7 nonretentive	SY 0.0 to 2047.7 retentive SY 2048.0 to 4096.7 nonretentive	T0 to T63 retentive T64 to T255 nonretentive	C0 to C63 retentive C64 to C255 nonretentive
NR (nonretentive)	No retentive flags	No retentive S flags	No retentive timers	No retentive counters

Note

If the battery fails on a cold restart after POWER UP and the switch is in the RE (retentive) position, the programmable controller goes to STOP with BAU (Overall Reset request).

The retentive feature in the RE switch position is determined by an entry in system data word 120 (EAF0_H):



Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	X	X	X	X	X	X	X	X	X	X				X	X	X	RS 120

0: FY 0 to FY 127 and SY 0 to 2047 retentive and FY 128 to FY 255 and SY 2048 to 4095 nonretentive
1: All flags and S flags retentive

0: T 0 to T 63 retentive and T 64 to T 255 nonretentive
1: All times retentive

0: C 0 to C 63 retentive and C 64 to C 255 nonretentive
1: All counters retentive

x = bits that determine system characteristics (must not be changed when setting the retentive feature!)

Figure 4.4 Relevant Bits for Setting the Retentive Feature in System Data Word 120

Bits 3, 4 and 5 of system data word 120 are set to "0" after Overall Reset of the CPU.
You can influence the retentive feature separately for flags, timers and counters by setting these bits

- in the restart program (OB20, OB21)
  or
- using the "Display memory contents" programmer function (only permissible when the PLC is in the STOP state!).

You can also determine the retentive feature by setting parameters in DB1 (see Chapter 11).

In the case of Overall Reset, all timers, counters, flags and S flags are reset regardless of the switch position or the contents of system data word 120.

4.4 Starting the Control Program

Starting point:

- PS 951 power supply module is switched off
- The CPU mode selector is at STOP
- The control program is stored on the E(E)PROM submodule

▸ Insert the memory submodule in the receptacle of the CPU

▸ Switch on the power supply module

The green LEDs of the PS 951 light up (otherwise: power supply module (PS 951) defective)

▸ (if desired:) Overall Reset of CPU

After switching on the power supply module or after Overall Reset (mode selector at STOP), the STOP LED and the BASP LED light up.

If the STOP LED

- ***flickers***, the CPU is defective;
- ***flashes***, there is a fault in the memory submodule (see Chapter 5.1).

Online functions with the programmer over the serial interface are possible when the CPU is in the STOP state.

▸ Set the mode selector from STOP to RUN

Both operating mode LEDs light up during the entire restart.
After the restart OB has been processed, the RUN LED lights up (cyclic program execution).

In the event of a fault, the CPU remains in the STOP state, i.e. the STOP LED lights up. Analysis of the cause of the interrupt is described in Chapter 5.
If the control program does not work properly, you can debug the program with the "BLOCK STATUS", "STATUS VAR" and "FORCE VAR" test functions.

4.5 System Configuration and Startup

The following section contains:

- Notes on configuring a system with important regulations which must be observed in order to avoid hazardous situations.
- Notes on the use of input and output modules
- The description of the system startup procedure.
- Notes on active and passive faults in automation equipment

4.5.1 Notes on Configuring and Installing a System

Since the product is usually used as a constituent part of a larger sytem or plant, these notes are designed as guidelines for the ***hazard-free integration*** of the product into its larger environment.

When configuring programmable controller systems, you must also take into account the appropriate VDE regulations (e.g. VDE 0100 or VDE 0160).

- The safety regulations and accident avoidance regulations in individual applications must be observed.

- Situations posing a hazard to personnel or material must be prevented.

- In the case of equipment with fixed connections (fixed devices, systems) without all-pole mains disconnector and/or fuses, a mains disconnector or fuse must be installed in the building; the equipment must be connected to a protective conductor.

- In the case of devices operated on mains voltage, check before startup that the rated voltage range set agrees with the local mains voltage.

- In the case of 24 V supply, safe electric isolation of the low voltage must be ensured. Use only power supplies which conform to IEC 364-4-41 or HD 384-04-41 (VDE 0100 Part 410).

- Fluctuations in and deviations from the rated value in the supply voltage must not exceed the tolerance limits specified in the technical specifications, otherwise malfunctions and hazardous states at the electrical modules/equipment cannot be excluded.

- Precautions must be taken to ensure that an interrupted program can be resumed properly after voltage dips and power failures. In doing so, hazardous operating states, even of short duration, must be avoided. If necessary, "emergency stop" must be programmed.

- Emergency stop facilities according to EN 60204/IEC 204 (VDE 0113) must remain effective in all operating modes of the automation system. Deactivation of the emergency off facility must not result in any uncontrolled or undefined warm restart.

- In the event of faults in the PLC, commands from EMERGENCY OFF facilities and from safety limit switches must remain in effect under all circumstances. These protective measures must take effect direct at the actuators in the power section.

- Activation of the EMERGENCY STOP facility must create a hazard-free state for personnel and plant:
  - Actuators and drives which could cause hazardous states (e.g. main spindle drives for machine tools) must be switched off.
  - On the other hand, actuators and drives which could constitute a hazard to personnel or plant when switched off (e.g. clamping devices) must not be switched off by the EMERGENCY STOP facility.
- Activation of the EMERGENCY STOP facility must be detected by the programmable controller and evaluated in the control program.
- Connecting cables and signal cables must be installed in such a way that inductive and capacitive interference does not adversely affect the automation functions.
- Automation equipment and the operator controls for the equipment must be adequately protected against unintentional operation.
- In order that wirebreaks on the signal side cannot lead to undefined states in the automation equipment, relevant hardware and software precautions must be taken when connecting inputs and outputs.

4.5.2 Notes on the Use of Input/Output Modules

Digital Input/Output Modules

We offer floating or nonfloating modules to suit the different signal levels. The wiring of the power supply, signal sensors and actuators is printed on the front flaps of the modules.
LEDs on the front side display the signal statuses of the inputs and outputs. The LEDs are assigned to the terminals of the front connector (see also Chapter 15, "Technical Specifications").

Analog Input/Output Modules

See Chapter 10 ("Analog Value Processing") for information on the use of analog modules.

Note

Input/output modules can only be inserted or removed when the power supply for the central controller and the signal sensors is switched off.

4.5.3 System Startup Procedure

The following is a prerequisite for starting up a system:
The system and the S5-115U must not be live, i.e. the main switch must be off.

▸ **Step 1: Visual check of the installation**; to VDE 0100 and 0113.

- Check mains voltage.
  Protective ground conductor must be connected.
- Make sure that all plugged-in modules are screwed tight to the subrack.
- Compare I/O modules plugged in with the assignment plan (note fixed or variable slot addressing).
- In the case of I/O modules, make sure that high-voltage lines (e.g. 220 V AC) do not terminate at low-voltage connectors (e.g. 24 V DC).
- When using nonfloating I/O modules, make sure that the M (0V reference) potential of the supply voltages for sensors and actuators is connected to the grounding terminal of the mounting rack (M_{Ext}-M_{Int} connection).

▸ **Step 2: Starting up the PLC**

- Disconnect fuses for sensors and actuators.
- Switch off the power circuits to the actuators.
- Turn on the main switch.
- Turn on the power supply.
- Switch the PLC without memory submodule to STOP.
- Connect the programmer to the CPU.

After the power switch is turned on, the green LEDs light up on the power supply and the red STOP LED lights up on the CPU.

- OVERALL RESET of the PLC.
- Transfer the program in the case of RAM operation.
- Switch the PLC to RUN.

The red STOP LED goes out and the green RUN LED lights up.

▸ **Step 3: Testing the signal inputs (peripheral)**

- Insert the fuse for the signal sensors. Leave the fuses for the actuators and the power circuits disconnected.
- Activate all sensors in sequence.
- You can scan all inputs using the "STATUS VAR" programmer function.

If the sensors function properly and their signals are received, the appropriate LEDs must light up on the I/O module.

- **Step 4: Testing the signal outputs (peripheral)**

  - Insert the fuse for the actuators. Leave the power circuits of the actuators disconnected.
  - You can force each output using the "FORCE VAR" programmer function.

The LEDs of the forced outputs must light up and the circuit states of the corresponding actuators must change.

- **Step 5: Entering, testing and starting the program**

Leave the power circuits for the actuators disconnected.

- Enter the program using the "INPUT" programmer function. You can enter the program in the STOP or RUN mode.

The red STOP LED or the green RUN LED lights up. A battery must be installed if a RAM submodule is used.

- Test the program block by block and make any necessary corrections.
- Dump the program in a memory submodule (if desired).
- Switch the PLC to STOP
- Switch on the power circuits for the actuators.
- Switch the PLC to RUN.

The green RUN LED lights up and the PLC scans the program.

4.5.4 Active and Passive Faults in Automation Equipment

- Depending on the particular task for which the electronic automation equipment is used, both **active** as well as **passive** faults can result in a **dangerous** situation. For example, in drive control, an active fault is generally dangerous because it can result in an unauthorized startup of the drive. On the other hand, a passive fault in a signalling function can result in a dangerous operating state not being reported to the operator.
- The differentiation of the possible faults and their classification into dangerous and non-dangerous faults, depending on the particular task, is important for all safety considerations in respect to the product supplied.

⚠ **Warning**

In all cases where a fault in automation equipment can result in severe personal injury or substantial property damage, i.e., where a dangerous fault can occur, additional external measures must be taken or equipment provided to ensure or force safety operating conditions even in the event of a fault (e.g., by means of independent limit monitors, mechanical interlocks, etc.).

Procedures for maintenance and repair

If you are carrying out measurement or testing work on an active unit, you must adhere to the rules and regulations contained in the "VBG 4.0 Acccident Prevention Regulations" of the German employers liability assurance association ("Berufsgenossenschaften"). Pay particular attention to paragraph 8, "Permissible exceptions when working on live parts."

Do not attempt to repair an item of automation equipment. Such repairs may only be carried out by **Siemens service personnel or repair shops Siemens has authorized to carry out such repairs.**

5 Error Diagnostics

Figures

Tables

5 Error Diagnostics

Malfunctions in the S5-115U can have various causes. If the PLC malfunctions, first determine whether the problem is in the CPU 945, the program, or the I/O modules (see Table 5-1).

Table 5.1 General Error Analysis

Fault/Error Condition	Fault/Error Analysis
The CPU is in the STOP mode. The red LED is lit up.	The problem is in the CPU. Perform an interrupt analysis with the programmer (see Section 5.2).
The CPU is in the RUN mode. The green LED is lit up. Operation is faulty.	There is a program error. Determine the error address (see Section 5.4). There is an I/O problem. Perform a malfunction analysis (see Section 5.5).

Note

To make a general distinction between PLC and program errors, program OB1 with "BE" as the first statement. A properly functioning PLC enters the RUN mode on a Cold Restart.

⚠ Caution

There are risks involved in changing the internal program memory direct with the "Display memory contents" programmer function.
For example, if the CPU is in RUN mode, memory areas (e.g. BSTACK) may be overwritten causing the CPU to "crash".
Take the following measures to avoid such risks:

- Change only the system data area documented in this manual
- Use only the control program to change the system data area!

5.1 LED Error Signalling

Certain errors are indicated by LEDs on the CPU depending on its design. Table 5.2 explains these error signals.

Table 5.2 Meaning of the Error LEDs on the CPUs

LED	Bedeutung
QVZ lights up	Timeout (CPU went into STOP mode)
ZYK lights up	Scan time exceeded (CPU went into STOP mode)
BASP lights up	Digital outputs are disabled (CPU is in RESTART or STOP mode)
Red STOP LED is flashing	Memory error (block structure damaged) After CPU COLD RESTART and after POWER UP, a memory error message may result if the user program contains the TNB, TNW, TRW, TRD, TIR or TDI operations. It is possible to inadvertently overwrite the following with these operations: ● Block headers ● Memory areas designated as "free" by the operating system. The operating system writes the erroneous address found when generating the address list into system data word 103 and 104 (E 10CE to E 10D1). You can display the contents of the system data locations using the "Display memory contents" programmer function.
Red STOP LED is flickering	Error in the CPU self-test routine Remedy: Exchange CPU

5.2 Interrupt Analysis with the Programmer

When malfunctions occur, the operating system sets various "analysis bits" that can be scanned with the programmer using the "ISTACK" function. LEDs on the CPU also report some malfunctions.

5.2.1 "ISTACK" Analysis

The interrupt stack (ISTACK) is an internal memory of the CPU where malfunction reports are stored. When a malfunction occurs, the appropriate bit is set.
Use a programmer to read this memory byte by byte.

> **Note**
>
> You can read only part of the ISTACK when the PLC is in the RUN mode.

The following tables show which control bits and which malfunction causes are reported in the ISTACK. The system data words containing the ISTACK bits are also specified.
See the subsequent tables for an explanation of the abbreviations or error codes used here.

ISTACK display in the PG 710/730/750 and 770

The following tables show ISTACK as displayed on programmers. The bits relevant for the CPU 945 are in bold type.

Table 5.3 Display of the Control Bits

Control Bits								Absolute Address	System Data Word
NB	PBSSCH	**BSTSCH**	**SCHTAE**	**ADRBAU**	**SPABBR**	NAUAS	QUITT	E 100A	RS 5
KM-AUS	**KM-EIN**	NB	**REMAN** X	NB	NB	NB	NB	E100B	
STOZUS	**STOANZ** X	**NEUSTA**	NB	**BATPUF** X	NB	**BARB**	**BARBEND**	E100C	RS 6
NB	UAFEHL	MAFEHL	EOVH	NB	**AF** X	NB	NB	E100D	
ASPNEP	ASPNRA	**KOPFNI**	PROEND	**ASPNEEP**	PADRFE	ASPLUE	RAMADFE	E100E	RS 7
KEINAS X	**SYNFEH**	**NINEU**	NB	NB	NB	SUMF	**URLAD**	E100F	

Table 5.4 Display of the Interrupt Stack

Interrupt Stack							
BLOCK:	01						
		SAC (new):	80FB0	DB-ADD:	83EA0		
BLK-STP:	E2820	FB-No.:	91	DB-No.:	81		
PAGE		REL-SAC:	00066	DBL-REG:	0064	RS-REG:	E26DE
NUMBER:	0000						
ACCU1:	0000 7C09	AKKU2:	0000 7C08				
CONDITION CODE:		CC1	CC0	OVFL	OVFLS	ODER	$\overline{\text{ERAB}}$
			X				
		STATUS	RLO				
		X	X				
CAUSE OF INTERR.:		KOLIF	SYSFE	TRAF	SUF	STUEB	BAU
		NAU	QVZ	NNN	ASPFA	ZYK	STOP
		STS	FAD	PEU	HALT		

The ISTACK is located in system data words RS 203 to 228. Table 5-5 shows the assignment of ISTACK entries to system data words.

Table 5.5 ISTACK Entries in System Data Words 203 to 229

System Data Bit	ISTACK Entry	Address
SD 203	ACCU 1-H	E 1196
SD 204	ACCU 1-L	E 1198
SD 205	ACCU 2-H	E 119A
SD 206	ACCU 2-L	E 119C
SD 207	Step address counter H	E 119E
SD 208	Step address counter L	E 11A0
SD 209	DBA register H	E 11A2
SD 210	DBA register L	E 11A4
SD 211	BR register H (RS-REG)	E 11A6
SD 212	BR register L (RS-REG)	E 11A8
SD 213	DBL register	E 11AA
SD 214	STATUS register (CONDITION CODE)	E 11AC

Table 5.5 ISTACK Entries in System Data Words 203 to 229 (Continued)

System Data Word	ISTACK Entry	Address
SD 215	Nesting level 0	E 11AE
SD 216	Nesting level 1	E 11B0
SD 217	Nesting level 2	E 11B2
SD 218	Nesting level 3	E 11B4
SD 219	Nesting level 4	E 11B6
SD 220	Nesting level 5	E 11B8
SD221	Nesting level 6	E 11BA
SD 222	Nesting level 7	E 11BC
SD 223	Nesting level pointer	E 11BE
SD 224	reserved	E 11C0
SD 225	BSTACK pointer H	E 11C2
SD 226	BSTACK pointer L	E 11C4
SD 227	User interrupt mask	E 11C6
SD 228	Interrupt condition code word	E 11C8
SD229	Last set page number	E 11CA

Figure 5.1 shows the assignments in the interrupt condition code word

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	STOPS	FAD	SUF	TRAF	NNN	STS	STUEB	n.a.	n.a.	QVZ	KOLIF	ZYK	SYSFE	PEU	BAU	ASPFA	RS 228

n.a. ... not assigned

Figure 5.1 Assingments in the Interrupt Condition Code Word

5.2.2 Meaning of the ISTACK Displays

Use Table 5-5 to determine the cause of a fault or an error when program execution is interrupted. In each case, the CPU goes into the STOP mode.

Table 5.6 Meaning of the ISTACK Displays

Fault/Error	Fault/Error Cause (ID in ISTACK)	Cause	Remedy
Cold Restart is not possible	NINEU SYNFEH/ KOPFNI	Faulty block: ● Compressing has been interrupted by a power failure ● Block transfer between programmer and PLC was interrupted by a power failure ● Program error (TIR, TNB, TNW, TRW, TRD, TDI)	Perform an Overall Reset. Reload the program.
	KOLIF	Interprocessor communication flags in DB1 programmed incorrectly.	Check the following: ● ID for interprocessor communication flag definitions ("MASK01"); (see Section 12.2.1) ● the end IDs in each case for interprocessor communication flag definitions
	SYSFE	Error in interpreting DB1	See Chapter 11.5
Faulty submodule	ASPFA	The submodule ID is illegal	Plug in the correct submodule
Battery failure	BAU	There is no battery or the battery is low and the retentive feature is required.	Replace the battery. Perform an Overall Reset. Reload the program.
I/Os not ready	PEU	The I/Os are not ready: ● There has been a power failure in the expansion unit. ● The connection to the expansion unit has been interrupted. ● There is no terminator in the central controller.	● Check the power supply in the expansion unit. ● Check the connection. ● Install a terminator in the central controller.
Program scanning interrupted	STOP	The mode selector is on STOP.	Put the mode selector on RUN.

Table 5.6 Meaning of the ISTACK Displays (Continued)

Fault/Error	Fault/Error Cause (ID in ISTACK)	Cause	Remedy
Interruption of program execution	SUF	Substitution error: A function block was called with an incorrect actual parameter	Correct the function block call
	TRAF	Transfer error: - A data block statement has been programmed with data word number greater than the data block length - A data block statement has been programmed without opening a DB/DX first - The DB/DX to be generated is too long for the program memory (G DB/CX DX operation)	Correct the program error
	STS	- Software stop by statement (STP, STS) - STOP request from programmer - STOP request from SINEC L1 master	
	NNN	- A statement cannot be decoded - A parameter has been exceeded	Correct the program error
	STUEB	BLock stack overflow - The maximum block call nesting depth (50) has been exceeded	Correct the program error
	QVZ	Timeout from I/Os: - A I/O byte that was not addressed has been referenced in the program or an I/O module does not acknowledge	Correct program eror or replace the I/O module
	ZYK	Scan time exceeded: The program execution time is greater than the set monitoring time	Check the program for continuous loops. If necessary, restart the scan time with OB31 or change the monitoring time
	FAD	Incorrect addressing - Write-protected areas accessed - Memory gaps accessed - Timers/counters accessed with TNB - Odd addresses accessed with TNW	Correct program error

Table 5.7 Mnemonics for Control Bits and Cause of Error

Control Bit Mnemonics	
BSTSCH	Block shift requested
SCHTAE	Block shift active (function: KOMP:AG)
ADRBAU	Construction of address lists
SPABBR	Compress operation aborted
KM-AUS	Interprocessor communication flag output address list available
KM-EIN	Interprocessor communication flag input address list available
REMAN	0: all timers, counters, and flags are reset on Cold Restart 1: the second half of timers, counters, and flags are reset on Cold Restart
STOZUS	STOP state (external request for example via the programmer)
STOANZ	STOP display
NEUSTA	PC in Cold Restart
BATPUF	Battery backup okay
BARB	Program check
BARBEND	Request for end
BARBEND	STOP state after program check (cold restart required)
AF	Interrupt enable
ASPNEP	Memory submodule is an EPROM
KOPFNI	Block header cannot be interpreted
ASPNEEP	Memory submodule is an EEPROM
KEINAS	No memory submodule
SYNFEH	Synchronization error (blocks are incorrect)
NINEU	Cold Restart not possible
URLAD	Bootstrapping required
Other mnemonics:	
RS	System data (from address E100$_H$)

Mnemonics for Cause of Error/Fault (=Error ID)	
STOP	Interrupt display word
SUF	Substitution error
TRAF	Transfer error for data block statements: data word number >data block length.
NNN	Statement cannot be interpreted in the S5-115U
STS	Operation interrupted by a programmer STOP request or programmed STOP statements.
STUEB	Block stack overflow: The maximum block call nesting depth of 50 has been exceeded.
QVZ	Time-out from I/Os: A nonexistent module has been referenced.
KOLIF	Interprocessor communication flag transfer list is incorrect.
ZYK	Scan time exceeded: The set maximum permissible program scan time has been exceeded.
SYSFE	Error in DB1
PEU	I/Os not ready: power failure in the I/O expansion unit; connection to the I/O expansion unit interrupted No terminator in the central controller
BAU	Battery failure
ASPFA	Illegal memory submodule
FAD	Incorrect addressing
Other mnemonics:	
UAW	Interrupt display word
ANZ1/ANZ0	See Section 8.4 Condition Code Generation
OVFL	Arithmetic overflow (+ or -)
OVFLS	Latching arithmetic overflow
ODER (OR)	OR memory (set by "O" operation)
STATUS	STATUS of the operand of the last binary statement executed
RLO	Result of logic operation
ERAB	First scan
SAC	Step address counter
DB-ADD	Data block address
BLK-STP	Block stack pointer
NO	Block number (OB, PB, FB, FX, SB, DB, DX)
REL-SAC	Relative step address counter
BS-REG	Basic address register (see Section 7.1)

5.3 Error Messages When Using Memory Submodules

A flashing red LED (STOP LED) indicates errors when memory submodule blocks are loaded into the internal RAM. The cause of error is stored in system data word 102.

Table 5.8 Errors when Using Memory Submodules

Cause of Error	Display in SD 102 E 10CC	Display in ISTACK	Remedy
Invalid memory submodule.	F002$_H$	URLAD ASPFA	Use the appropriate submodule.
Faulty contents of memory submodule.	F003$_H$	URLAD	Delete and reprogram the submodule.
All blocks cannot be copied.	F001$_H$	URLAD	The internal memory already contains blocks. Check to see if these blocks are needed. Delete the blocks or optimize the program.
User program operations TNB, TNW, TIR, TRD, TRW or TDI have overwritten block headers or free memory space.	0000$_H$	NINEU SYNFEH	Overall Reset; reload program

5.4 Program Errors

Table 5-9 lists malfunctions caused by program errors.

Table 5-9 Program Errors

Error	Action
All inputs are zero	Check the program
All outputs are not set	
One input is zero. One output is not set	Check program assignments (double assignment, edge formation)
Timer or counter is not running or is incorrect	
Cold Restart is faulty	Check Cold Restart blocks OB21/OB22 or insert them
Sporadic malfunctions occur	Check the program with STATUS

5.4.1 Determining the Error Address

The STEP address counter (SAC) in the ISTACK (bytes 25 to 26) indicates the absolute memory address of the STEP 5 statement in the PLC ***before*** which the CPU went into the "STOP" mode.

The relative STEP address counter (REL SAC) indicates the relative address of the STEP 5 statement within the block ***before*** which the CPU went into the "STOP" mode.

Example:

You have entered a control program consisting of OB1, PB0 and PB7. L DW 10 has been programmed in PB7 without previously opening a DB.

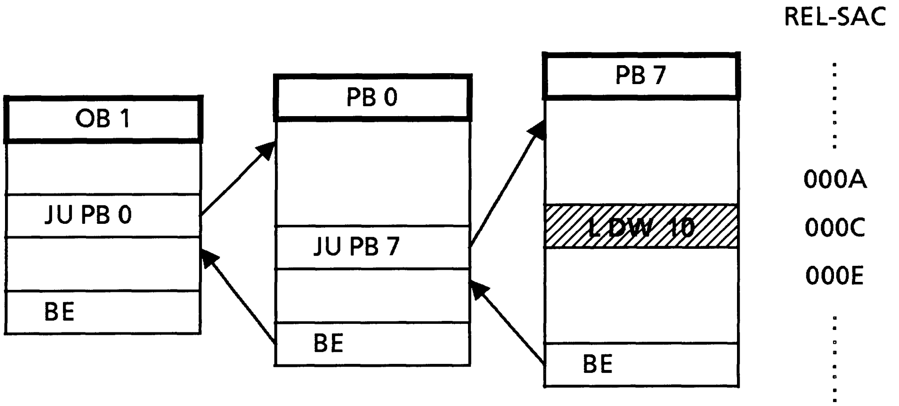


Figure 5.2 Structured Program with Incorrect Programming

The CPU interrupts program execution at the illegal statement and goes to STOP with the TRAF error message.
The STEP address counter is at the absolute address of the next unexecuted statement in program memory.
The REL SAC is at the relative address of the next unexecuted statement within PB7 (000E).

5.4.2 Program Trace with the Block Stack ("BSTACK") Function

During program execution, jump operations enter the following information in the block stack:

- the data block that was valid before program execution exited a block;
- the relative return address. This address indicates the location at which program execution continues after it returns from the block that was called.
- the absolute return address. This address indicates the location in the program memory at which program execution continues after it returns from the block that was called.

You can call the information listed above using the "BSTACK" programmer function in the STOP mode if the CPU has entered this mode as the result of a malfunction. The "BSTACK" reports the status of the block stack at the time the interruption occurred.

Example: Program scanning was interrupted at function block FB2. The CPU went into the STOP mode with the error message "TRAF" (because of incorrect access. DB5 is two words long. DB3 is ten words long).
You can use the "BSTACK" function to determine the path used to reach FB2 and to determine which block has passed the wrong parameter. The "BSTACK" contains the three return addresses (as marked in Figure 5-6).

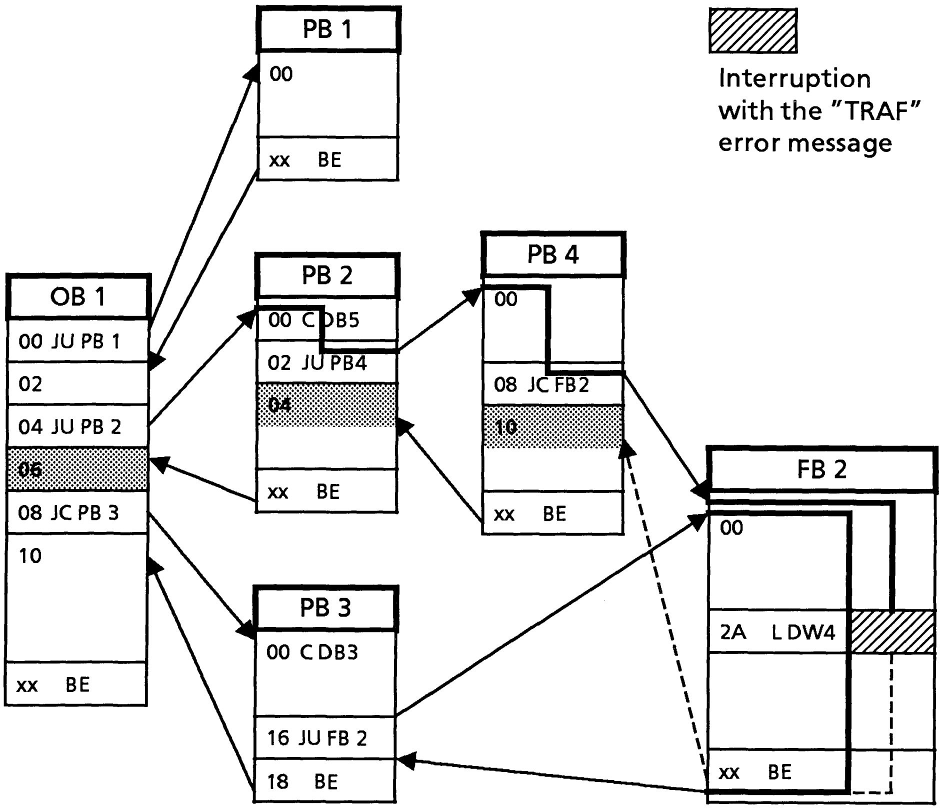


Figure 5.3 Program Trace with the "BSTACK" Function

The display in Figure 5-7 indicates that DB5 was accessed incorrectly on the path OB1 → PB 2 → PB 4.

5.5 Other Causes of Malfunction

Hardware components or improper installation can also cause malfunctions. Table 5-10 summarizes such malfunctions.

Table 5-10. Other Causes of Malfunction

Fault/Error	Action
All inputs are zero.	Check the module and the load voltage.
All outputs are not set.	
One input is zero. One output is not set.	
The green LEDs on the power supply module do not light up.	Check the module and replace it if necessary.
Sporadic malfunctions occur.	Check the memory submodule. Check to see if the controller has been set up according to EMC guidelines.
The PLC will not go into the RUN mode.	Perform an Overall Reset.

Note

If the PLC still does not operate properly after you have taken the appropriate action recommended in Table 5-10, try to determine the faulty component by replacement.

5.6 System Parameters

Use the "SYSPAR" programmer function to read the system parameters (e.g. software release) out of the CPU.

6 Addressing/Address Assignments

Figures

Tables

6 Addressing/Address Assignments

In order to be able to access input/output modules, these modules must be assigned addresses. With the CPU 945, you can address the O extended peripheral area in addition to the address areas for digital and analog modules.

6.1 Address Structure

Digital modules are usually addressed by bit over the process I/O image but also by byte, by word and by doubleword.
Analog modules are addressed either by byte or by word.
Consequently, the addresses have different structures.

6.1.1 Digital Module Addresses

One bit represents a channel on a digital module. You must therefore assign a number to each bit. When numbering, note the following:

- The CPU program memory is divided into different address areas (see Section 6.3).
- Number individual bytes consecutively in relation to the start address of the relevant address area.
- Number the eight bits of each byte consecutively (0 to 7).

Figure 6-1 shows the format of a digital address:

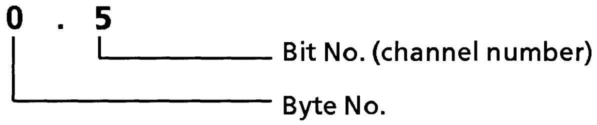


Figure 6-1. Format of a Digital Address

6.1.2 Analog Module Addresses

Each channel of an analog module is represented by two bytes (= one word).
An analog channel address is thus represented by the number of the low address (high-order byte).

6.2 Slot Address Assignments

You can establish addresses for S5-115U modules in the following two ways:

- Fixed Slot Addressing
  Each slot has a fixed address under which you can reference the module plugged into that slot.
- Variable Slot Addressing
  The user can specify an address for each slot.

Fixed slot and variable slot addresses are relevant only for modules of block design. The addresses of intelligent I/O modules and modules of ES 902 design (S5-135U/155U) are set on the modules themselves. In this case, the address need not be set on the IM 306.

6.2.1 Fixed Slot Address Assignments

Fixed slot addressing is only possible in the case of block-type digital and analog modules.
Input/output modules are referenced under permanently assigned slot addresses when the following conditions exist for the S5-115U:

- The PLC is operated without an expansion unit interface module and a terminating resistor is used.
- The PLC is operated with the IM 305 interface module (centralized configuration, see Section 3.3.1

The number of address bytes available for digital and analog modules varies.

Digital modules

Each slot has four bytes, so that 32 binary inputs or outputs can be addressed. If you plug in digital modules with 8 or 16 channels, use the low-order byte numbers for addressing. In this case, the high-order byte numbers are irrelevant.

Analog modules

For fixed slot addressing, analog modules can be plugged into slots 0 to 3 of a central controller only.
Each slot has 32 bytes. You can thus address 16 analog channels. If you plug in 8-channel modules, use the 16 low-order byte numbers for addressing. In this case, the 16 high-order byte numbers are irrelevant.

Note the following:

- Input and output modules cannot have the same address.
- If an analog module has been assigned an address for a particular slot, this address cannot be used for digital modules and vice versa.
- In slot 0 of the expansion unit, addressing always begins with address 28.0, regardless of the number of slots of the central controller.

Figures 6-2 and 6-3 show the exact assignment of fixed addresses (please observe the "Installation Guidelines" in Sections 3.1.1 and 3.1.2).

Slot numbers in the central controller	PS	CPU	0	1	2	3	4	5	6	IM
Digital modules			0.0 . . 3.7	4.0 . . 7.7	8.0 . . 11.7	12.0 . . 15.7	16.0 . . 19.7	20.0 . . 23.7	24.0 . . 27.7	
Analog modules			128 . . 159	160 . . 191	192 . . 223	224 . . 255	Analog modules cannot be plugged into these slots			
Modules	Addresses									

Figure 6-2. Fixed Slot Addressing in the Central Controllers

Slot numbers in the expansion unit	0	1	2	3	4	5	6	7	8	IM
Digital modules	28.0 . . 31.7	32.0 . . 35.7	36.0 . . 39.7	40.0 . . 43.7	44.0 . . 47.7	48.0 . . 51.7	52.0 . . 55.7	56.0 . . 59.7	60.0 . . 63.7	IM 305
Analog modules	Analog modules cannot be plugged in here.									
Modules	Addresses									

Figure 6-3 Fixed Slot Addressing in the Expansion Unit

6.2.2 Variable Slot Address Assignments

The S5-115U offers you the possibility of assigning an address to each slot. You can do this if an IM 306 interface module is plugged into the central controller and each expansion unit. For addressing purposes, it does not matter whether the module in question is plugged into a central controller or an expansion unit. Under a hinged cover on the right side of the interface module is an addressing panel. It has a DIP switch for each slot. Use the DIP switch to set the least significant byte number for a particular slot.

Addressing is independent of

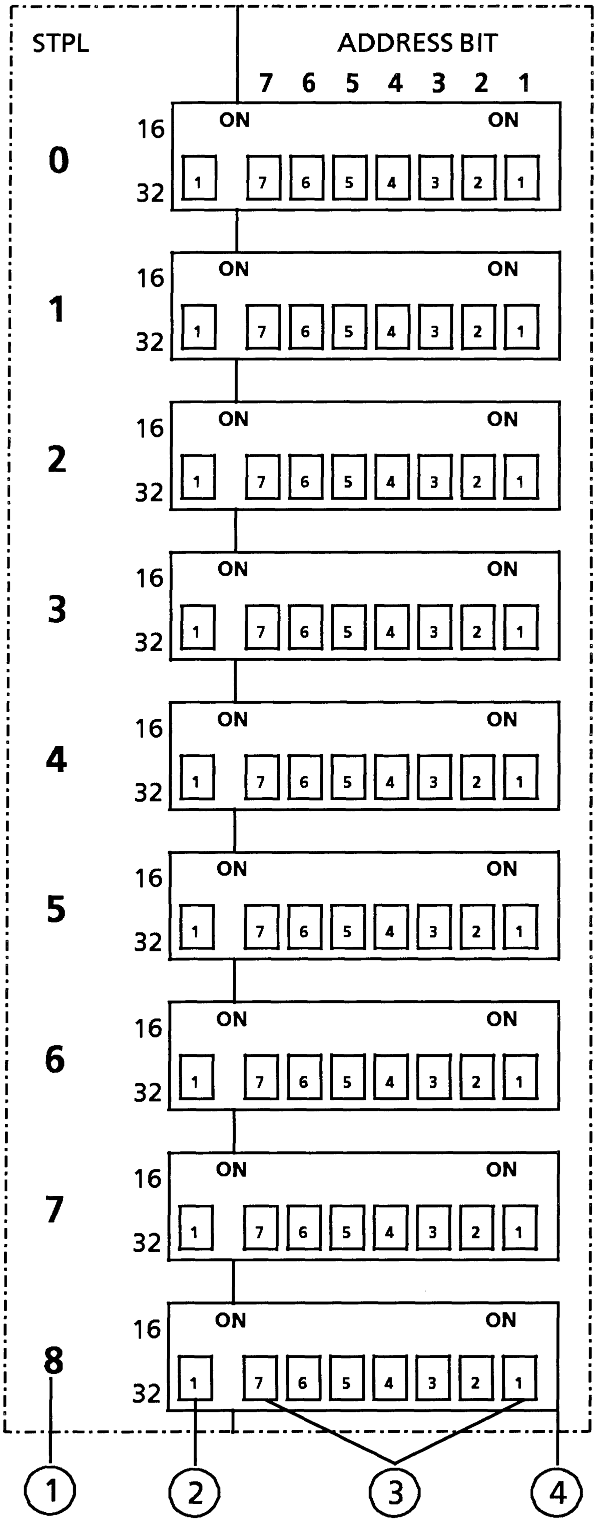
- whether the module is plugged into a central controller or an expansion unit
- whether you want to address the I/O in an expansion in the O peripheral area.

Note

Input and output modules in different slots can have the same address.
The slot addressing only applies to block-type digital and analog modules.

Note

Addresses for compact modules are set on the module!



Addresses for digital modules	Address switches (ON = 1, OFF = 0) 7	6	5	4	3	2	1
0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	1
4	0	0	0	0	0	1	0
6	0	0	0	0	0	1	1
8	0	0	0	0	1	0	0
10	0	0	0	0	1	0	1
12	0	0	0	0	1	1	0
14	0	0	0	0	1	1	1
16	0	0	0	1	0	0	0
18	0	0	0	1	0	0	1
20	0	0	0	1	0	1	0
22	0	0	0	1	0	1	1
24	0	0	0	1	1	0	0
26	0	0	0	1	1	0	1
28	0	0	0	1	1	1	0
30	0	0	0	1	1	1	1
32	0	0	1	0	0	0	0
34	0	0	1	0	0	0	1
36	0	0	1	0	0	1	0
38	0	0	1	0	0	1	1
40	0	0	1	0	1	0	0
42	0	0	1	0	1	0	1
44	0	0	1	0	1	1	0
46	0	0	1	0	1	1	1
48	0	0	1	1	0	0	0
50	0	0	1	1	0	0	1
52	0	0	1	1	0	1	0
54	0	0	1	1	0	1	1
56	0	0	1	1	1	0	0
58	0	0	1	1	1	0	1
60	0	0	1	1	1	1	0
62	0	0	1	1	1	1	1

Addresses for analog modules	Address switches 7	6	5	4	3	2	1
128	1	0	0	0	0	0	0
144	1	0	0	1	0	0	0
160	1	0	1	0	0	0	0
176	1	0	1	1	0	0	0
192	1	1	0	0	0	0	0
208	1	1	0	1	0	0	0
224	1	1	1	0	0	0	0
240	1	1	1	1	0	0	0

① : Slot number

② : Switch for setting the number of inputs or outputs per slot

③ : Address switches

④ : DIP switch

Figure 6-4 Setting Addresses on the Addressing Panel of the IM 306 Interface Module

Setting addresses

Use the left-hand switch (② in Figure 6-4) on the addressing panel of the IM 306 to indicate what type of module you have plugged into the slot. Proceed as follows:

Set the switch to OFF: for a 32-channel digital module or a 16-channel analog module.
Set the switch to ON: for a 16-channel digital module or an 8-channel analog module.

The following modules must also be set as 16-channel digital modules:

- 482-7 digital input/output module
- 434-7 digital input module with process interrupt.

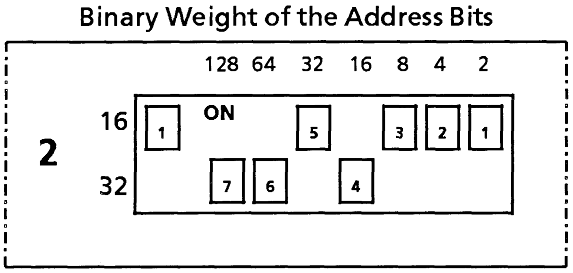
Use the seven address switches (③ in Figure 6-4) on the addressing panel of the IM 306 to indicate the least significant address (the address for channel "0") for the module in question. This setting establishes the addresses of the other channels in ascending order.

When setting start addresses, note the following:

- 32-channel digital modules can only have start addresses whose byte numbers are divisible by 4 (e.g., 0, 4, 8 ...).
- 16-channel digital modules can only have start addresses whose byte numbers are divisible by 2 (e.g., 0, 2, 4 ...).
- 16-channel analog modules can only have the start addresses 128, 160, 192 and 224.
- 8-channel analog modules can only have the start addresses 128, 144, 160 to 240.

Example: A 16-channel digital input module is plugged into slot 2. Assign it start address 46.0 by performing the following steps:

- Check to see if the byte number of the start address can be divided by 2 since you are dealing with a 16-channel digital module.
  46 : 2=23 Remainder 0
- Set the number of input channels (set switch to ON).
- Set the address switches on the DIP switch for slot number 2 as shown in Figure 6-5.



The address is equal to the sum of the weights selected by the individual coding switches, e.g.:

2+4+8+32=46

Figure 6-5. Setting a DIP Switch

The module is then addressed as follows:

Channel No.	0	1	2...	7	8	9	10...	15
Address	46.0	46.1		46.7	47.0	47.1		47.7

6.2.3 Addressing in the O Area

Modules can be addressed in the O area (see Section 6.3) only via a distributed link. You can address the O area on an expansion unit via the interface module of a distributed link (see Section 3.3.2).

For block-type modules, the addresses of the O area are set on the IM 306 interface module in exactly the same way as the addresses of the P area (see Section 6.2.2).

If you want to operate CPs and IPs in the O area, you must additionally set them to the O area on the module itself.

Modules cannot be addressed in the O area on the module.

6.3 Handling the Process Signals

With the CPU 945, you can additionally use the O area for addressing input and output modules. This area is not available on the CPUs 941 to 944.
The signal states of the input and output modules can be written to or read under the following addresses.

Table 6-1 Addresses of the Input and Output Modules

Address Area	Relative Byte Addresses	Absolute Addresses
Digital modules	0 : 127	00F000$_H$: 00F07F$_H$
Analog modules	128 : 255	00F080$_H$: 00F0FF$_H$
O peripherals	0 : 255	00F100$_H$: 00F1FF$_H$
IM3 area	0 : 255	00FC00$_H$: 00FCFF$_H$
IM4 area	0 : 255	00FD00$_H$: 00FDFF$_H$

Digital module signal states are also stored in a special memory area called the process image. The process image has two sections, namely the process input image (PII) and the process output image (PIQ). Table 6-2 shows where the process images are located in the program memory.

Table 6-2 Addresses of the Process I/O Images

Address Area	Relative Byte Addresses	Absolute Addresses
Process image of the inputs (PII)	0 : 127	020500$_H$: 02057F$_H$
Process image of the outputs (PIQ)	0 : 127	020580$_H$: 0205FF$_H$

Process signals can be read or output either via the process image or directly.

6.3.1 Accessing the PII

At the beginning of program execution, the input module signal states are written to the PII. The statements in the control program use a particular address to indicate what information is currently needed. The control logic then reads the data that was current at the beginning of program execution and works with it.

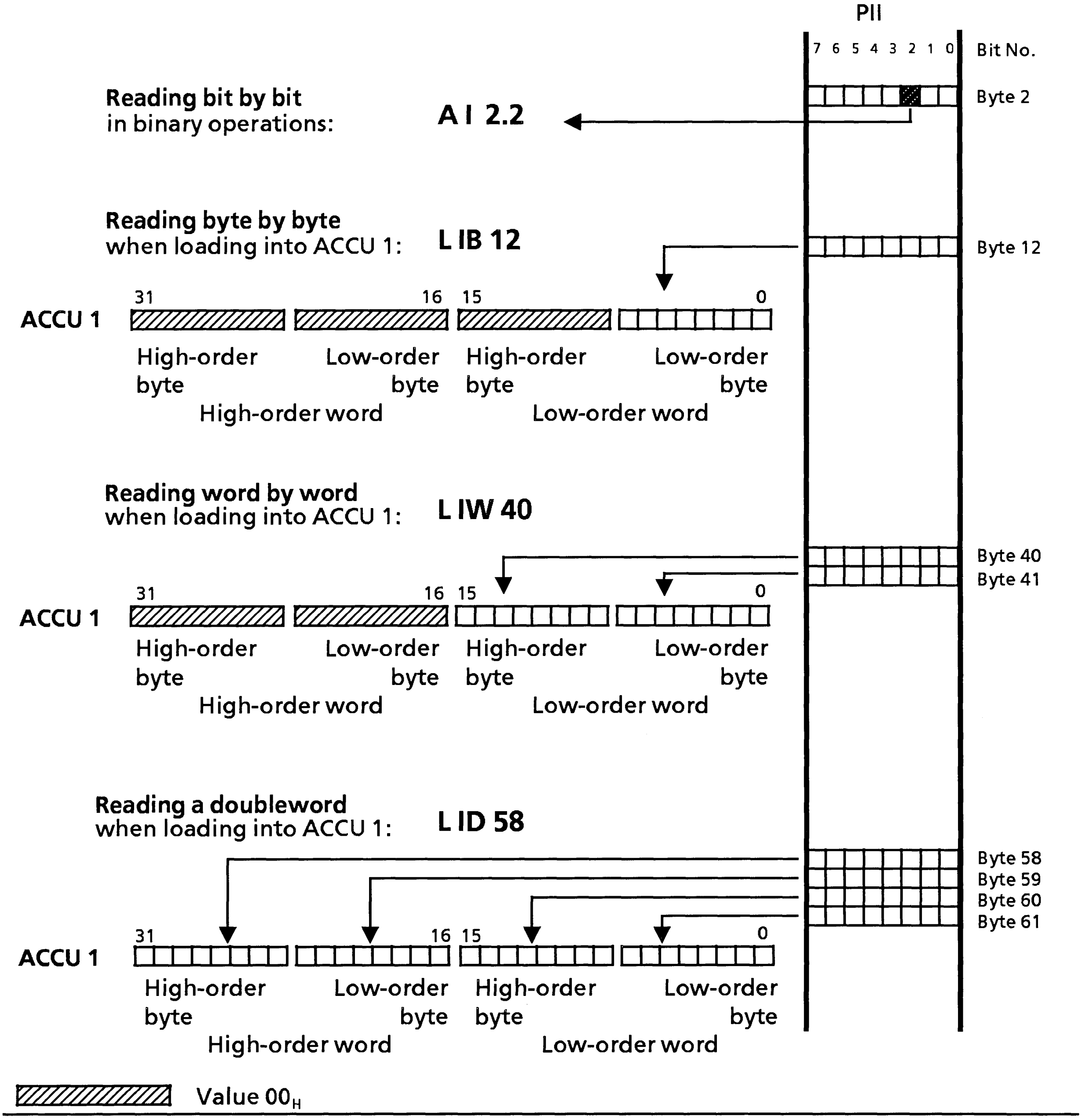


Figure 6-6. Accessing the PII

Reading of the PII can be inhibited.
To do so, it is necessary to act upon bit 1 of system data word 120 (E $10F0_H$) using system operations (Load and Transfer).

Bit 1="1":	Reading of inputs is inhibited.
Bit 1="0":	Reading of inputs is enabled.

The default setting is bit 1 = "0" (read enabled).

6.3.2 Accessing the PIQ

New signal states are entered in the PIQ during program execution. This information is transferred to the output modules at the end of each program scan.

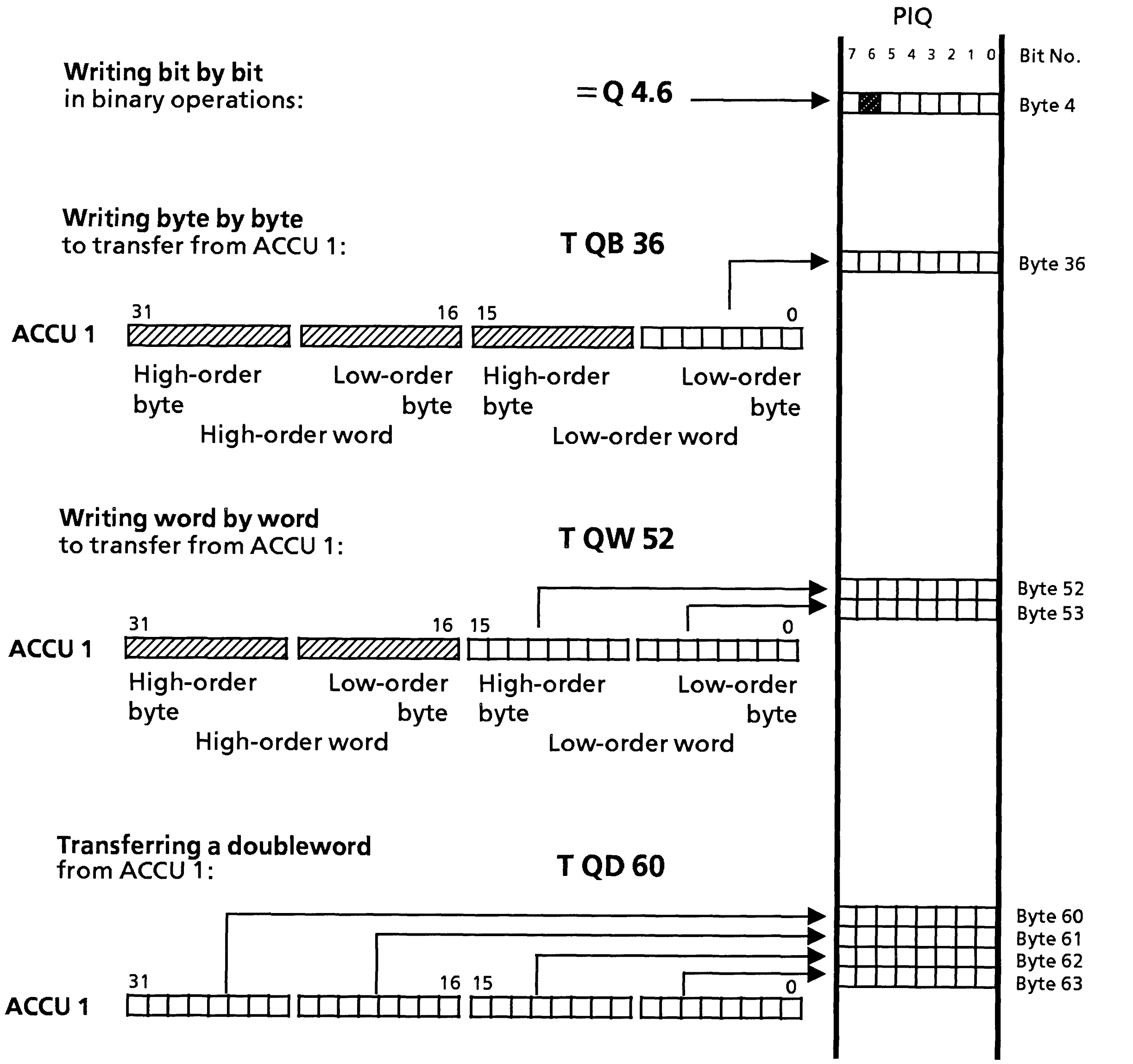


Figure 6-7. Accessing the PIQ

Output of the PIQ to the output modules can be inhibited by setting bit 2 in system data word 120 (E 10F0$_H$).

Bit 2="1": Output of the PIQ is inhibited.
Bit 2="0": Output of the PIQ is enabled.

The default setting is bit 2="0" (output of the PIQ enabled).

6.3.3 Direct Access

Analog module signal states are not written to the process image. They are read in or transferred to an output module directly with the "L PY x, L PW x, T PY x or T PW x / L OY , L OW x, T OY x or T OW x" statements.

You can also exchange information with digital modules directly. This is necessary when signal states have to be processed immediately in the control program. Figure 6-8 shows differences during the loading of signal states.

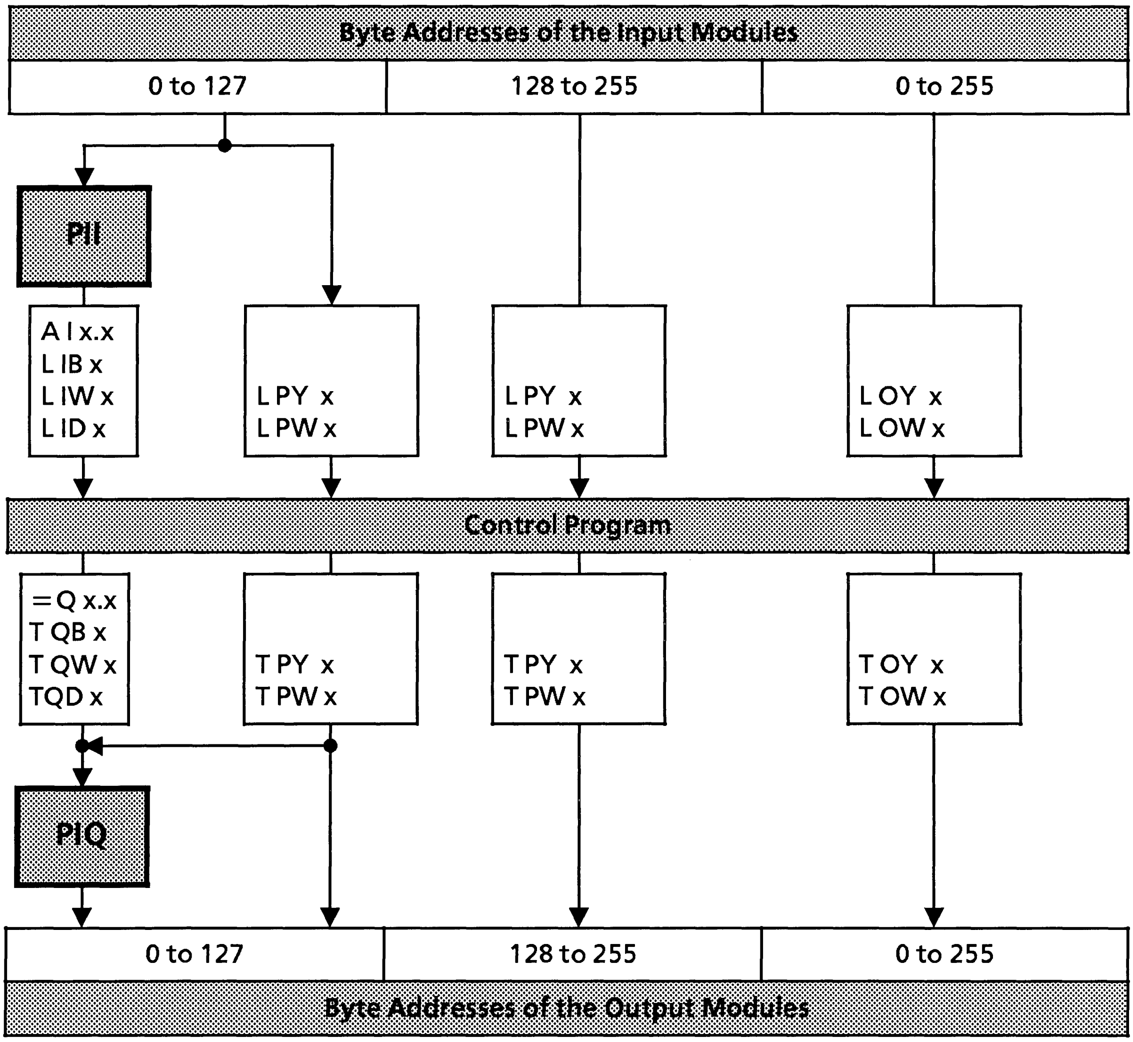


Figure 6-8 Loading Input/Output Modules

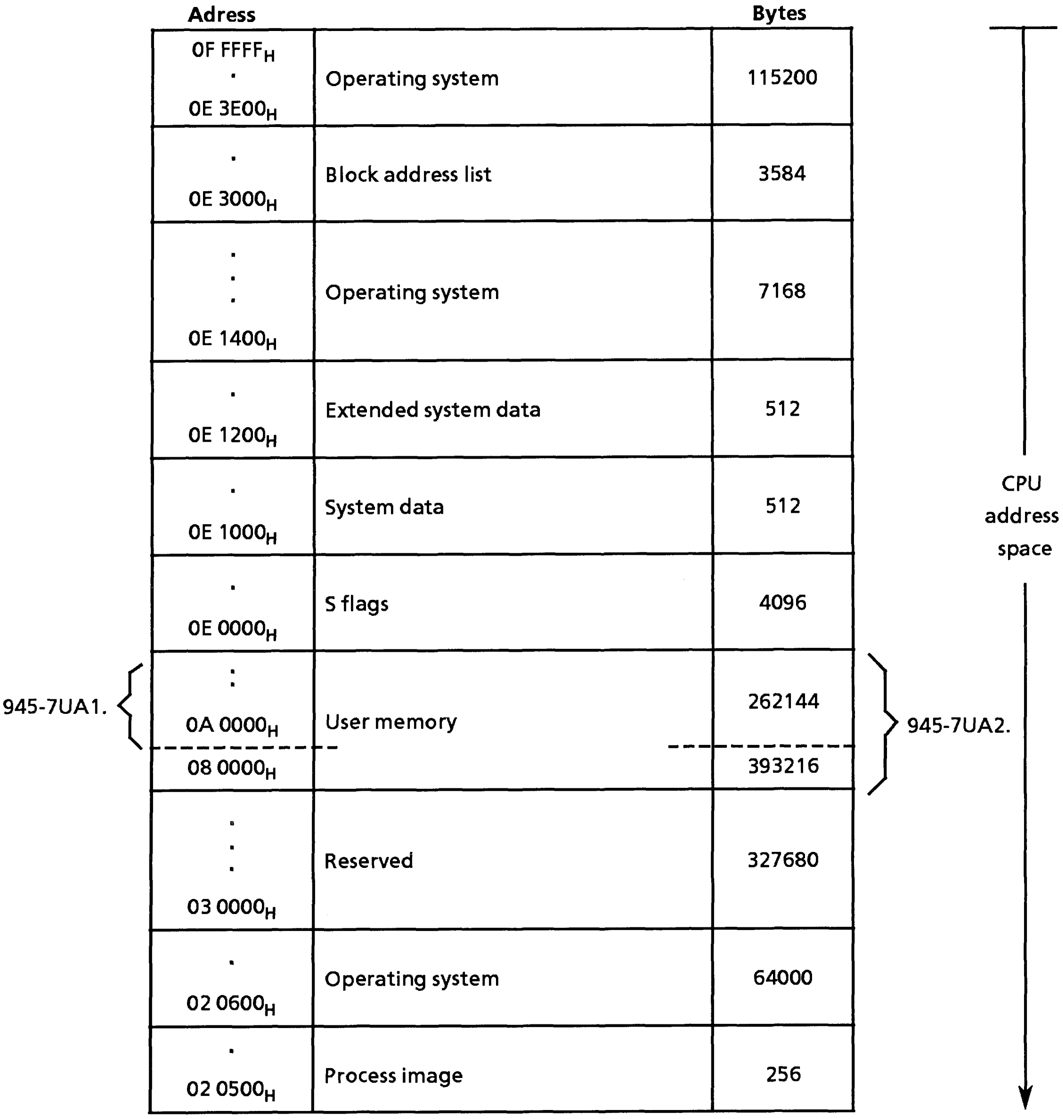
Note

If you use direct access to call an address whose slot is unoccupied, the CPU enters the STOP mode with error code "QVZ" (timeout) or the error DB is activated.

> **Note**
>
> The digital inputs can be read with OB254 and the PIQ output to the output modules with OB255, irrespective of the contents of system data word 120 (see Section 2.11.4 and 2.11.5).

6.4 Address Allocation on the CPU

You can see from Figure 6-9 how the CPU address space is allocated.



Adress		Bytes
0F FFFF$_H$ · 0E 3E00$_H$	Operating system	115200
· 0E 3000$_H$	Block address list	3584
⋮ 0E 1400$_H$	Operating system	7168
· 0E 1200$_H$	Extended system data	512
· 0E 1000$_H$	System data	512
· 0E 0000$_H$	S flags	4096
⋮ 0A 0000$_H$ 08 0000$_H$	User memory	262144 393216
⋮ 03 0000$_H$	Reserved	327680
· 02 0600$_H$	Operating system	64000
· 02 0500$_H$	Process image	256

945-7UA1.
945-7UA2.
CPU
address
space

Figure 6-9 Memory Allocation in the CPU

Adress		Bytes
· 02 0400$_H$	Flags	256
· 02 0200$_H$	Timers	512
· 02 0000$_H$	Counters	512
· · 01 0000$_H$	Reserved for operating system	65 536
· 00 FF00$_H$		256
00 FEFF$_H$	Page select register for CPs/IPs (can only be written to)	1
00 FE00$_H$		255
00 FD00$_H$	IM 4 area	256
00 FC00$_H$	IM 3 area	256
· · 00 F400$_H$	Pages (2 Kbytes)	2048
00 F300$_H$		256
00 F200$_H$	Interprocessor communication flags	256
00 F100$_H$	O peripherals	256
00 F080$_H$	Analog I/O	128
00 F000$_H$	Digital I/O	128
· 00 0000$_H$	Modules with absolute addresses	61440

CPU address area

S5 bus address area

Figure 6-9 Memory Allocation in the CPU (Continued)

The following table lists the system data of relevance to the user and indicates the sections which provide more detailed information.

Table 6-3 Address Allocation in the System Data Area

System Data Word	Address (hex.)	Description	Details in Section
8 - 12	E 1010 : E 1019	Integrated hardware clock; clock data area, status word, error codes, correction value	13
32 - 33	E 1040 : E 1043	Start address of internal RAM	
34 - 35	E 1044 : E 1047	End address of internal RAM	
36 - 37	E 1048 : E 104B	Address level indicator for internal RAM	
40 -45	E 1050 E 105B	CPU ID and firmware status in ASCII code	
46	E 105C E 105D	Driver number and error message (e.g. in the case of ASCII driver)	12
48 - 55	E 1060 : E 106F	Driver parameter block (e.g. for ASCII driver)	12
57 - 63	E 1072 : E 107F	SINEC L1 parameter field	12.3.3
64 - 79	E 1080 : E 109F	Address list for IPC output flags	12.2.1
80 - 95	E 10A0 : E 10BF	Address list for IPC input flags	12.2.1
96	E 10C0 E 10C1	Scan monitoring time (multiple of 10 ms)	2.9.5
97	E 10C2 E 10C3	Interval timer for OB 13 (in ms)	2.8.3
98	E 10C4 E 10C5	Interval timer for OB 12 (in ms)	2.8.3

Table 6-3 Address Allocation in the System Data Area (Continued)

System Data Word	Address (hex.)	Description	Details in Section
99	E 10C6 E 10C7	Interval timer for OB 11 (in ms)	2.8.3
100	E 10C8 E 10C9	Interval timer for OB 10 (in ms)	2.8.3
101	E 10CA E 10CB	Time (in ms) to calling OB6	2.8.5
102	E 10CC E 10CD	Error when copying the memory submodule	5.3
103 - 104	E 10CE : E 10D1	Address of the faulty module in the case of QVZ or error address in the case of address list generation	5.1
120	E 10F0 E 10F1	System characteristics: software protection disable reading in of PII disable output of PIQ retentive feature of flags, counters and timers parallel transfer of process image	 4.3.3 6.3.1 6.3.2 4.3.4 2.8.2
121	E 10F2 E 10F3	Current scan time	2.9.4
122	E 10F4 E 10F5	Maximum scan time	2.9.4
123	E 10F6 E 10F7	Minimum scan time	2.9.4
126	E 10FC E 10FD	Startup delay in ms	2.6.2
128 - 159	E 1100 : E 113F	List of all addressable I/O words	2.6.1
160 - 175	E 1140 : E 115F	Error buffer for system error handling	2.8.6
203 - 229	E 1196 : E 11CB	Interrupt stack	5

From RS 240 onward, the system data words are reserved for standard FBs.
The unnamed system data words between RS 0 and RS 239 are either used by the operating system of the CPU or they are reserved.

Table 6.4 Address Allocation in the Flag, Timer and Counter Areas

Memory Area		Description in Section	Abs. Address (hex.)
Flags (F)	FY 0 FY 1 : FY 255	8.1.2	2 0400 2 0401 : 2 04FF
Flags (S) (extended)	SY 0 SY 1 : SY 4095	8.1.2	E 0000 E 0001 : E 0FFF
Timers (T)	T 0 T 1 : T 255	8.1.4	2 0200,2 0201 2 0202,2 0203 : 2 03FE,2 03FF
Counters (C)	C 0 C 1 : C 255	8.1.5	2 0000,2 0001 2 0002,2 0003 : 2 01FE,2 01FF
System data RS area	RS 0 : : RS 255	8.2.1	E 1000,E 1001 E 1002,E 1003 : E 11FE,E 11FF
System data RT area	RT 0 : : RT 255	8.2.1	E 1200,E 1201 E 1202,E 1203 : E 13FE,E 13FF

Table 6-5 Block Address List

Block Type	Block Number	Abs. Address (hex.)
Organization blocks	OB0 OB1 : OB255	E 3000,E 3001 E 3002,E 3003 : E 31FE, E 31FF
Function blocks	FB0 FB1 : FB255	E 3200,E 3201 E 3202, E 3203 : E 33FE,E 33FF
Program blocks	PB0 PB1 : PB255	E 3400,E 3401 E 3402,E3403 : E 35FE, E35FF
Sequence blocks	SB0 SB1 : SB255	E 3600, E 3601 E 3602, E 3603 : E 37FE,E 37FF
Data blocks	DB0 DB1 : DB255	E 3800,E 3801 E 3802,E 3803 : E 39FE,E 39FF
Function blocks (extended)	FX0 FX1 : FX255	E 3A00,E 3A01 E 3A02,E 3A03 : E 3BFE,E 3BFF
Data blocks (extended)	DX0 DX1 : DX255	E 3C00,E 3C01 E 3C02,E 3C03 : E 3DFE,E 3DFF

Note

In the CPU 945, blocks are stored in program memory in such a way that the 4 least significant bits (bits 0 to 3) of the block start address (1st operation in block or 1st data word) are always zero. These 4 bits are not stored in the block address list and instead only the significant part of the address is stored.
Example:
Start address of FB1 is address $0A0040_H$ in the program memory. The value $A004_H$ is stored at address $E3202_H$ and $E3203_H$ in the block address list.

7 Introduction to STEP 5

Figures

Tables

7 Introduction to STEP 5

This chapter describes the registers of the CPU 945 and deals with programming of automation tasks with the CPU 945 of the S5-115U PLC. It explains how to generate programs and which blocks can be used to structure a program. In addition, you will find an overview of the different types of number representation available in the STEP 5 programming language.

The CPU 945 offers a STEP 5 which has been expanded compared to the CPUs 941 to 944.
The following are new additions in the CPU 945:

- New operations
  - Floating-point operations
  - Doubleword operations
  - Fixed-point multiplication and division

- Expanded register set
  - 32-bit ACCUs
  - BR register

- New operand areas
  - S flags
  - O peripherals (see Chap. 6)
  - Expanded system data

- New blocks
  - FX blocks
  - DX blocks

- A new integral organization block
  - OB250 for calling and parameterizing operating system services (see Section 2-10)

- New organization blocks for error handling
  - OB26, OB33, OB35

- Additional number representation methods
  - Processing of 32-bit fixed-point binary numbers (see Section 7-6)
  - Processing of 32-bit floating-point numbers (see Section 7-6)

7.1 The Registers of the CPU 945

The register set of the CPU 945 exists for every program execution level.

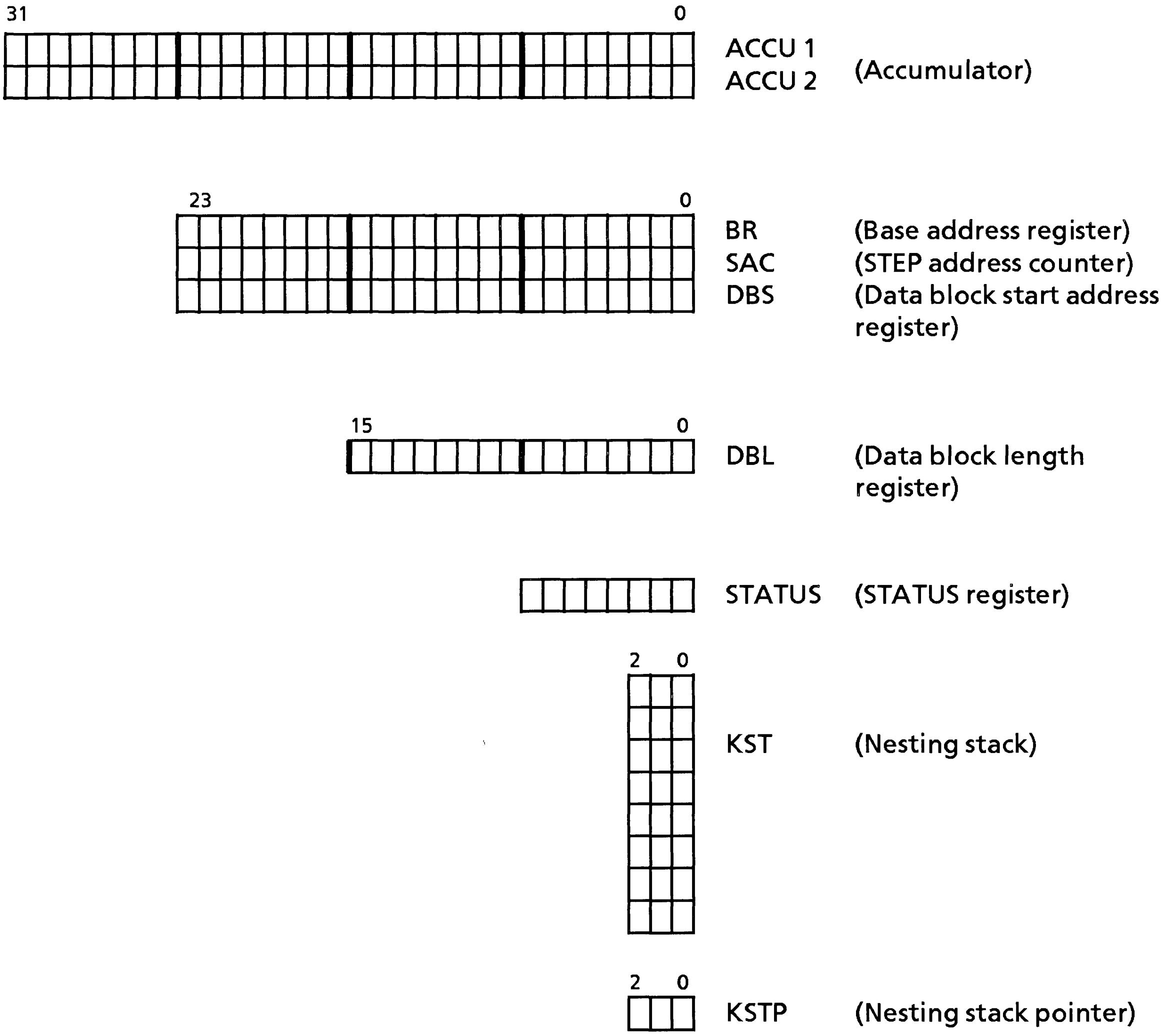


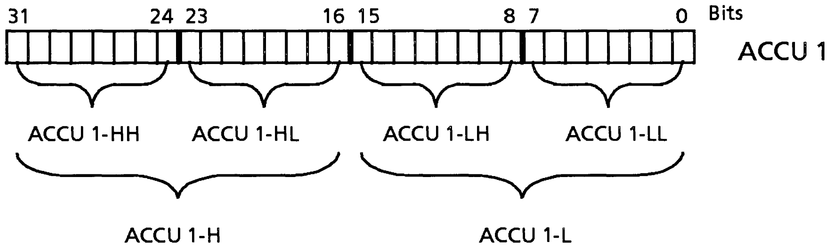
Figure 7-1 Registers of the CPU 945

ACCU 1

ACCU 1 is 32 bits wide. ACCU 1 is the main accumulator of the CPU 945 and is used as a working register for the operands. The operands can be processed in byte format, word format or doubleword format.

The load and transfer operations always refer to this ACCU.
ACCU 1 contains the destination address for the TNW and TNB operations.
ACCU 1 contains the operand address for the LLIR, TIR, LDI and TDI operations.

Descriptors:

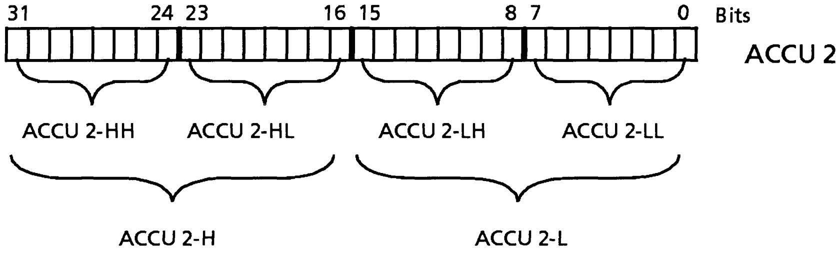


ACCU 2

ACCU 2 is also 32 bits wide. ACCU 2 contains the second operand for logic and arithmetic operations. The operands can be processed in byte format, word format or doubleword format.

ACCU 2 usually takes over the contents of ACCU 1 in the case of load and transfer operations.
ACCU 2 contains the source address for the TNW and TNB operations.

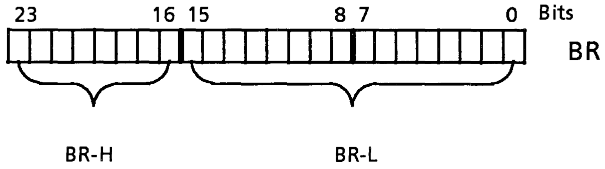
Descriptors:



Base address register (BR)

The base address register (BR) is 24 bits wide. This register is available for for address calculations in the case of absolute addressing.
The contents of the base address register is used as a byte address.

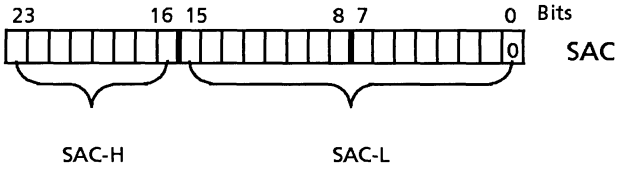
Descriptors:



STEP address counter (SAC)

The STEP address counter (SAC) is 24 bits wide. The SAC is responsible for addressing the operation store and always points to the address of the next operation to be executed.
The SAC contains even addresses since the STEP 5 operations are stored exclusively at even addresses and are always an even number of bytes long.
Bit 0 of the SAC is fixed at logic "0".

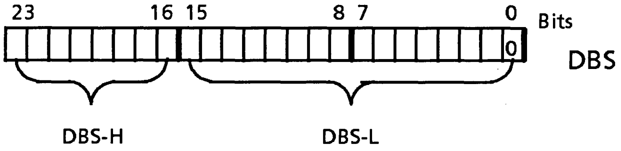
Descriptors:



Data block start address register (DBS)

The data block start address register (DBS) is 24 bits wide. THE DBS contains the start address of the data of the last opened data block.
The DBS contains even addresses.
Bit 0 of the DBS is fixed at logic "0".

Descriptors:



Data block length register (DBL)

The data block length register (DBL) is 16 bits wide. THE DBL contains the data length of the last opened data block. The data length is given in words.
The data length is the value taken over from the data block after the length of the data block header has been subtracted.

STATUS register (STATUS)

The status register is 8 bits wide.

Table 7-1 Bit Assignments in the STATUS Register

Bit	Assignment	
0	ERABX	Initial scan
1	RLO	Result of logic operation
2	STATUS	STATUS flag
3	OR	OR flag
4	OS	Latching overflow
5	OV	Overflow
6	CC0	Condition code bit 0
7	CC1	Condition code bit 1

Bits 0 to 7 are defined by the result of a logic or arithmetic operation.

Nesting stack (NST)

The nesting stack supports execution of nested operations. The intermediate results of logic operations are stored in the nesting stack in order to ensure the preset order of execution of the logic operations.
The nesting stack contains up to 8 entries.
Block calls do not affect the nesting stack.

Nesting stack pointer (NSTP)

The nesting stack pointer (NSTP) points to the current level of the nesting stack in each case. The NSTP is incremented by an "Open bracket" operation and decremented by a "Close bracket" operation.
Block calls do not affect the nesting stack pointer.

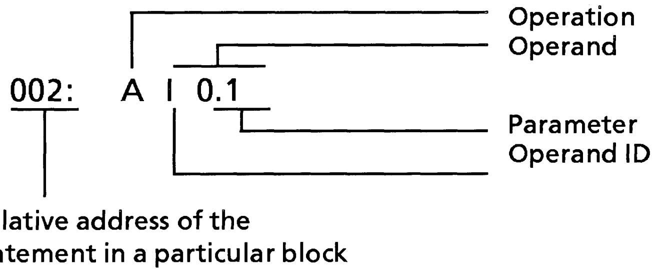
7.2 Generating a Program

In programmable controllers (PLCs), automation tasks are formulated as control programs. In the control program, the user draws up a series of statements which defines how the PLC is to perform its control task. So that the programmable controller (PLC) can "understand" the program, it must be written according to fixed rules in a specific language, the programming language. The STEP 5 programming language has been developed for the SIMATIC S5 family.

7.2.1 Methods of Representation

The following methods of representation are possible with the STEP 5 programming language:

- **Statement List (STL)**
  STL represents the program as a sequence of operation mnemonics. A statement has the following format:



  The operation instructs the PLC what to do with the operand. The parameter indicates the operand address.

- **Control System Flowchart (CSF)**
  CSF represents logic operations with symbols.

- **Ladder Diagram (LAD)**
  LAD represents control functions with circuit diagram symbols.

- **GRAPH 5**
  GRAPH 5 is a graphic representation of the structure of sequence controls.

Types of operation

The STEP 5 programming language has the following two operation types:

- basic
- supplementary

Table 7-2 provides further information on these operations.

Table 7-2 Comparison of Operation Types

STEP 5 PROGRAMMING LANGUAGE		
	Basic Operations	**Supplementary Operations**
Application	in all blocks	only in function blocks
Methods of Representation	STL, CSF, LAD	STL

Refer to Chapter 8 for a description of all operations and programming examples.

7.2.2 Operands and Blocks

The CPU 945 has the following new operands and blocks compared to the CPUs 941 to 944:

- S flags
- O peripherals
- FX blocks
- DX blocks

You can use the following operands in the CPU 945:

I	(inputs)	interfaces from the process to the PLC
Q	(outputs)	interfaces from the PLC to the process
F	(flags)	memory for intermediate results of binary operations
S	(S flags)	additional memory for intermediate results of binary operations
D	(data)	memory for intermediate results of digital operations
T	(timers)	memory for implementing timers
C	(counters)	memory for implementing counters
P, O	(peripherals)	interface between process and programmable controller
K	(constants)	defined numeric values
RS, RT	(system data)	Interface between operating system and control program

You can use the following blocks in the CPU 945:

OB, PB, SB, FB, FX	(code blocks)	aids for structuring the program
DB, DX	(data blocks)	blocks for storing and structuring data

You will find a complete list of operations, operands and the blocks already integrated on the CPU 945 in the enclosed Pocket Guide for the CPU 945.

7.3 Program Structure

An S5-115U program can be one of the two following types:

- linear
- structured

Sections 7.3.1 and 7.3.2 describe these program types.

7.3.1 Linear Programming

When processing simple automation tasks, it is enough to program the individual operations in one block.
In the case of the S5-115U, this is organization block 1 (OB1) (see Section 7.4.1). This block is scanned cyclically, i.e. after processing the last statement, the processor returns to the first statement.

Please note the following:

- Five words are assigned to the block header (see Section 7.4).
- Normally, a statement takes up one word in the program memory. Two word and three word statements also exist (e.g., with the operation "Load a constant"). Count these statements twice or three times when calculating the program length.
- Like all blocks, OB1 must be terminated by a Block End statement (BE).

7.3.2 Structured Programming

To solve complex tasks, it is advisable to divide an entire program into individual, self-contained program parts (blocks). This procedure has the following advantages:

This procedure has the following advantages:

- simple and clear programming, even for large programs
- capability to standardize program parts
- easy alteration
- simple program test
- simple startup
- subroutine techniques (block call from different locations)

The STEP 5 programming language has the following five block types:

- **Organization Block (OB)**
  Organization blocks manage the control program.

- **Program Block (PB)**
  Program blocks arrange the control program according to functional or technical aspects.

- **Sequence Block (SB)**
  Sequence blocks are special blocks that program sequence controls. They are handled like program blocks.

- **Function Block (FB/FX)**
  Function blocks are special blocks for programming frequently recurring or especially complex program parts (e.g., reporting and arithmetic functions). You can assign parameters to them. They have an extended set of operations (e.g., jump operations within a block).

- **Data Block (DB/DX)**
  Data blocks store data needed to process a control program. Actual values, limiting values, and texts are examples of data.

The program uses block calls to exit one block and jump to another. You can therefore nest program, function, and sequence blocks randomly in up to 50 levels (see Section 7.4).

Note

When calculating the nesting depth, you must take into account that in the event of specific events occurring, the system program can call an organization block (e.g. OB32) autonomously or it can interrupt the different execution levels (e.g. cyclic, time-controlled and interrupt-driven program execution) according to their priority.

The total nesting depth is the sum of the nesting depths of all programmed organization blocks. If nesting goes beyond 50 levels, the PLC goes into the STOP mode with the error message "STUEB" (block stack overflow) (see Section 5.2). Figure 7-2 illustrates the nesting principle.

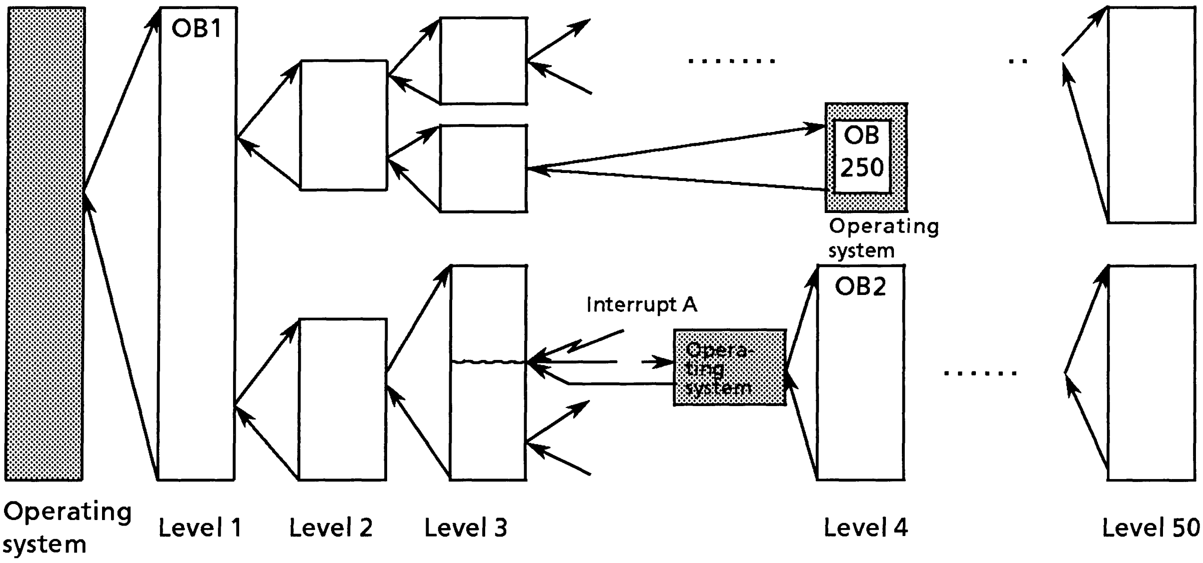


Figure 7-2 Nesting Depth

7.4 Block Types

You will find the most important characteristics of the individual block types in Table 7-3:

Table 7-3 Comparison of Block Types

	Number	Length (Max.)	Operation Set (Contents)	Methods of Representation	Block Header Length
Code blocks					
OB	256[1] OB0 to OB255	8 Kbytes	Basic operations	STL, CSF, LAD	5 words
PB	256 PB0 to PB255	8 Kbytes	Basic operations	STL, CSF, LAD	5 words
SB	256 SB0 to SB255	8 Kbytes	Basic operations	STL, CSF, LAD	5 words
FB	256[2] FB0 to FB255	8 Kbytes	Basic operations, supplementary operations, system operations	STL	5 words
FX	256 FX0 to FX255	8 Kbytes	Basic operations, supplementary operations, system operations	STL	5 words
Data blocks					
DB	256[3] DB0 to DB255	64K words[4]	Bit pattern Numbers Texts		5 words
DX	256 DX0 to DX255	64K words[4]	Bit pattern Numbers Texts		5 words

[1] Organization blocks have already been integrated into the operating system (see Chap. 2). A few OBs are called autonomously by the operating system (see Section 7.4.1)

[2] Function blocks have already been integrated into the operating system (see Section 2.11 and Chap. 12)

[3] Data block DB1 is reserved for parameterization of the CPU

[4] Can be accessed with "L DW", "T DW" up to DW 255
Only 2042 DWs can be entered with the programmer.

Block structure

Each block consists of the following:

- Block header specifying the block type, number, and length.
  The programmer generates the block header when it transforms the block.
- Block body with the STEP 5 program or data.

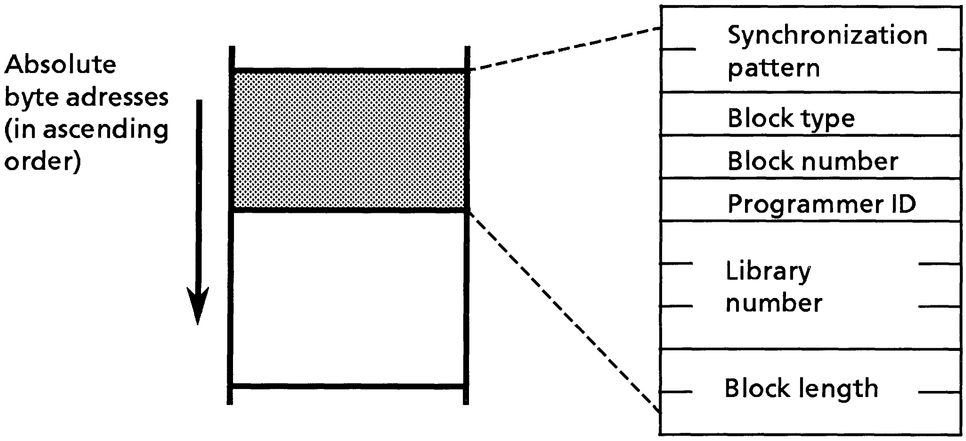


Figure 7-3 Structure of a Block Header

Programming

Blocks are programmed with the "LAD, CSF, STL" software package. See your programmer manual for details of programming individual blocks.

7.4.1 Organization Blocks (OBs)

Organization blocks are the interface between the operating system and the control program; they can be divided into three groups:

- An organization block is called cyclically by the operating system (OB1)
- Some organization blocks are event-driven or time-controlled; i.e. they are called by
  - STOP→RUN or POWER OFF→POWER ON transitions (OB21, OB22)
  - Interrupts (OB2 to OB6)
  - Programming errors or PLC faults (OB19, OB23, OB24, OB27, OB32, OB34)
  - Expiry of an interval (OB10 to OB13)
- Other organization blocks represent operating functions (similar to integral function blocks), which can be called from the control program (see Chapter 2).

Program execution by OBs takes place on different levels. The OBs have different interrupt characteristics depending on the execution levels.
You will find a detailed explanation of the program execution levels in Section 2.8.
Handling of programming errors and PLC faults via OBs is explained in Section 2.8.7.
Handling of system errors via OBs is explained in Section 2.8.6.

Table 7-4 Overview of the Organization Blocks

OB No.	Function	Section
OB called cyclically by the operating system		
OB1	Cyclic program execution	2.8.2
OBs for interrupt-driven and time-controlled program execution		
OB2	Interrupt A: Interrupt generated by the 434-7 and 487-7 modules and by IP	2.8.4
OB3	Interrupt B: Interrupt generated by IP	
OB4	Interrupt C: Interrupt generated by IP	
OB5	Interrupt D: Interrupt generated by IP	
OB10	Time-controlled program execution (variable in each case: 1 ms to 65535 ms)	2.8.3
OB11		
OB12		
OB13		
OBs for timed-interrupt-driven program execution		
OB6	Interrupt triggered by internal timer	2.8.5
OBs for controlling restart characteristics		
OB21	Cold restart after STOP-RUN transition	2.6.2
OB22	Cold restart after POWER ON	
OBs for handling programming errors and PLC faults		
OB19	When a block is called which has not been loaded	2.8.7
OB23	Timeout during direct I/O access	
OB24	Timeout during update of the process image and the interprocessor communication flags	
OB27	Substitution error	
OB32	Transfer error	
OB34	Battery failure	
OBs for handling system errors		
OB26	Scan time exceeded	2.8.6
OB33	Collision of two timed interrupts	
OB35	I/O error	
OBs which offer operating functions		
OB31	Scan time triggering	2.9.5
OB160	Programmable time loop	2.11.3
OB250	Operating system services	2.10
OB254	Read in process I/O image	2.11.4
OB255	Output process I/O image	2.11.5

7.4.2 Program Blocks (PB)

Self-contained program parts are usually programmed in blocks.

Special feature:
Control functions can be represented graphically in program blocks.

Call

Block calls JU and JC activate program blocks. You can program these operations in all block types except data blocks. Block call and block end cause the RLO to be reloaded. However, the RLO can be included in the "new" block and be evaluated there.

7.4.3 Sequence Blocks (SBs)

Sequence blocks are special program blocks that process sequence controls. They are treated and used like program blocks.

7.4.4 Function Blocks (FBs)

Frequently recurring or complex control functions are programmed in function blocks. Function blocks have the following special features:

You can also use FX blocks in the CPU 945. You can program and process FX blocks in the same way as FBs.

Function blocks have the following special features:

- FBs can be assigned parameters.
  Actual parameters can be assigned when the block is called.
- FBs have a supplementary set of operations not available to other blocks.
- The FB program can be written and documented in STL only.

The S5-115U has the following types of function blocks:

- FBs that you can program
- FBs that are integrated in the operating system (see Chapter 2 and Chapter 12)
- FBs that are available as software packages (Standard Function Blocks, see Catalog ST 57).

Block header

In contrast to other types of blocks, function blocks have other organization information in addition to the block header.

Its memory requirements consist of the following:

- block description as for other blocks (five words)
- block name (five words)
- block parameter for parameter assignment (three words per parameter).

Creating a function block

In contrast to other blocks, parameters can be assigned to FBs. To assign parameters, you must program the following block parameter information:

- ***Name*** of the block parameter (formal operands)
  Each formal operand receives a name (declaration "DECL"). The name can contain up to four characters and must begin with an alpha character. You can program up to 40 block parameters per function block.

- Block ***parameter type***
  You can enter the following parameter types:

- I	input parameters
- Q	output parameters
- D	data
- B	blocks
- T	timers
- C	counters

Output parameters are represented to the right of the function symbol in graphics representation (CSF). The other parameters are to the left.

- Block ***data type***
  You can specify the following data types:

- BI	for operands with bit address
- BY	for operands with byte address
- W	for operands with word address
- D	for operands with doubleword address
- K	for constants

You can specify the following types for parameter ***D***

- KM	for a bit pattern
- KY	for two numbers expressed in byte format
- KH	for a hexadecimal number
- KC	for two alphanumeric characters
- KF	for a fixed-point number
- KT	for a timer
- KZ	for a counter
- KG	for a floating-point number

Data type specification is not permissible with the ***B, T, C*** parameters.

Table 7-5 Block Parameter Types and Data Types with Permissible Actual Operands

Parameter Type	Data Type	Permissible Actual Operands *
I, Q	BI for an operand with bit address	I x.y inputs Q x.y outputs F x.y flags
	BY for an operand with byte address	IB x input bytes QB x output bytes FB x flag bytes DL x data bytes left DR x data bytes right PB x peripheral bytes
	W for an operand with word address	IW x input words QW x output words FW x flag words DW x data words PW x peripheral words
	D for an operand with doubleword address	ID x Input doubleword QD x Output doubleword FD x Flag doubleword DD x Data doubleword
D	KM for a binary pattern (16 digits) KY for two absolute numbers, one byte each, each in the range from 0 to 255 KH for a hexadecimal pattern (maximum 4 digits) KS for a character (maximum 2 alphanumeric characters) KT for a time (BCD-coded time) with time base 0.0 to 999.3 KC for a count (BCD-coded) 0 to 999 KF for a fixed-point number in the range from - 32768 to + 32767 KG for a floating-point number	Constants
B	Type designation not permitted	DB x Data blocks. The C DBx operation is executed. FB x Function blocks (permissible without parameters only) are called unconditionally (JU..x). PB x Program blocks are called unconditionally (JU..x). SB x Sequence blocks are called unconditionally (JU..x). OB x Organization blocks are called unconditionally (SPA..x).
T	Type designation not permitted	T Timer. The time should be assigned parameters as data or be programmed as a constant in the function block.
C	Type designation not permitted	C Counter. The count should be assigned parameters as data or be programmed as a constant in the function block.

* Note: S flags are not permissible as actual operands

Calling a function block

Function blocks are stored like other blocks under a specific number (e.g. FB47) in internal program memory. Numbers 238 to 251 are reserved for integral FBs and can therefore not be used for user-written FBs!
FB calls can be programmed in all blocks except data blocks.

A function block call consists of:

- Call statement
  - JU FBx Unconditional call of the FB x (*J*ump *U*nconditional...)
  - DOU FXa Unconditional call of the FXa
  - JC FBx Call if RLO = 1 (*J*ump *C*onditional...)
  - DOC FXa Call if RLO = 1
- Parameter list (only necessary if block parameters have been defined in the FB)

Function blocks can only be called if they have already been programmed. When a function block call is being programmed, the programmer automatically requests the parameter list for the FB, provided block parameters have been defined in the FB.

Assigning function block parameters

The program in the function block specifies how the formal operands (the parameters defined as "DECL") are to be processed.
As soon as you program a call statement (e.g. JU FB2), the programmer displays the ***parameter list***.
The parameter list consists of the names of the parameters each followed by a colon (:). Actual operands must now be assigned to the parameters. When the FB is called, the actual operands replace the formal operands defined in the FB so that the FB "actually" works with actual operands.
The parameter list may contain up to 40 parameters.

Example: The name (DECL) of a parameter is IN 1, the parameter type is I (input) and the data type is BI (bit).
The formal operand of the FB then has the form
DECL: IN1 I BI.
The parameter list in the calling block specifies which (actual) operand is to replace the formal operand in the event of the FB being called; in the example this is the operand "I 1.0".
The parameter list must therefore contain the entry
IN1: I 1.0.
When the FB is called, it replaces the formal operand "EIN1" with the actual operand "I 1.0".

Figure 7-4 contains a detailed example of the parameter assignment of a function block.

The function block call occupies three words in the internal program memory and each parameter a further memory word.

The required memory length of the standard function blocks as well as the execution time are specified in Catalog ST 57.

The identifiers for the inputs and outputs of the function block appearing on the programmer during programming are deposited, together with the name, in the function block itself. It is therefore necessary to transfer all required function blocks to the program diskette (in the case of off-line programming) or enter them directly into the program memory of the PLC before programming begins on the programming unit.

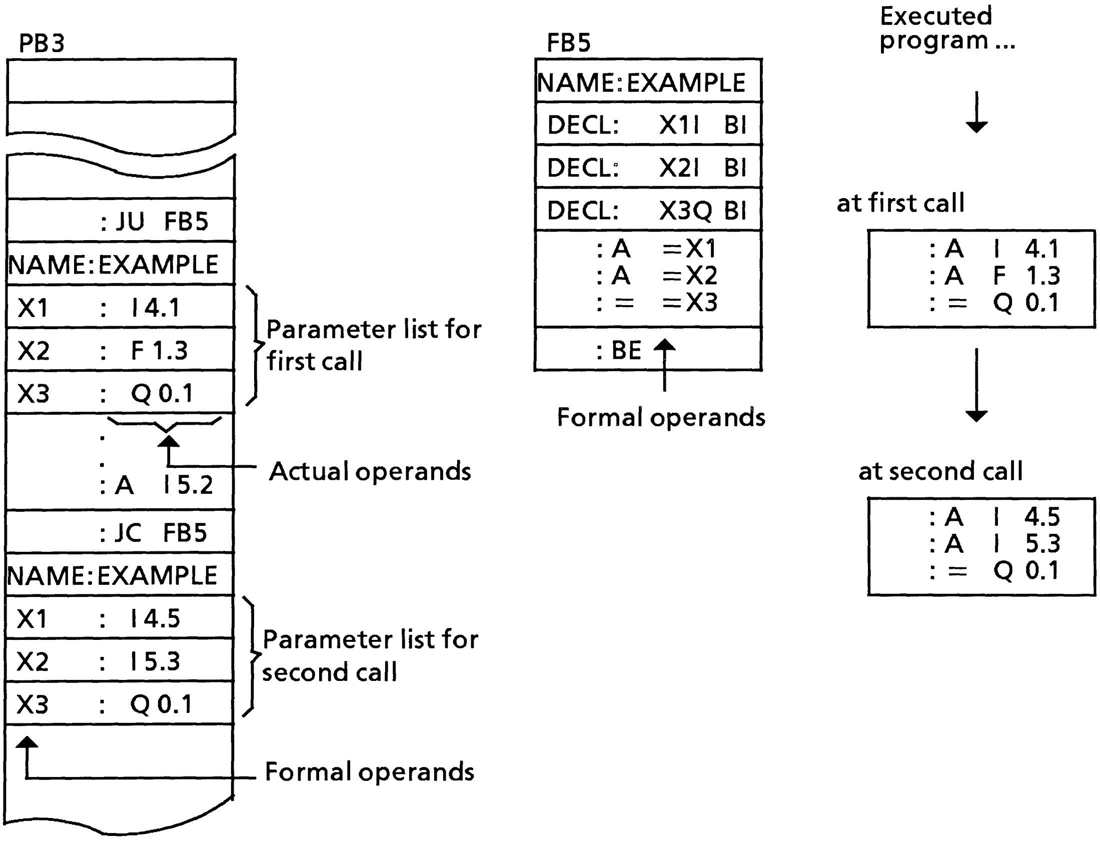


Figure 7-4 Assigning Function Block Parameters

7.4.5 Data Blocks (DBs/DXs)

The data which is to be processed in the program is stored in data blocks.
You can also use DX blocks in the CPU 945. You can program and use DX blocks in the same way as DBs.
For indexed calling of a DX block, you must use the operating system service No. 24 (OB 250) (see Section 2.10).

The following data types are permissible:

- bit pattern (representation of controlled system states)
- hexadecimal, binary or decimal numbers (times, results of arithmetic operations)
- alphanumeric characters (message texts)

Programming data blocks

Begin data block programming by specifying a block number between 2 and 255.

DB1 is reserved

- for parameterizing internal functions (see Chapter 11)
- for defining interprocessor communication flags (see Chapter 12.2.1)

The data is deposited in this block in words. If the information is less than 16 bits in volume, the higher-order bits are filled with zeros. Entry of the data begins at data word zero and is continued in ascending order. The data block can accommodate up to 65530 data words. Only 2042 of these 65530 data words can be entered with the programmer. Accessing is possible up to DW 255 using the "L DW" and "T DW" operations. Data words 256 to 65529 can only be accessed using the "LRW", "LRD", "TRW", "TRD", LIR", "TIR" und "TNB" operations.

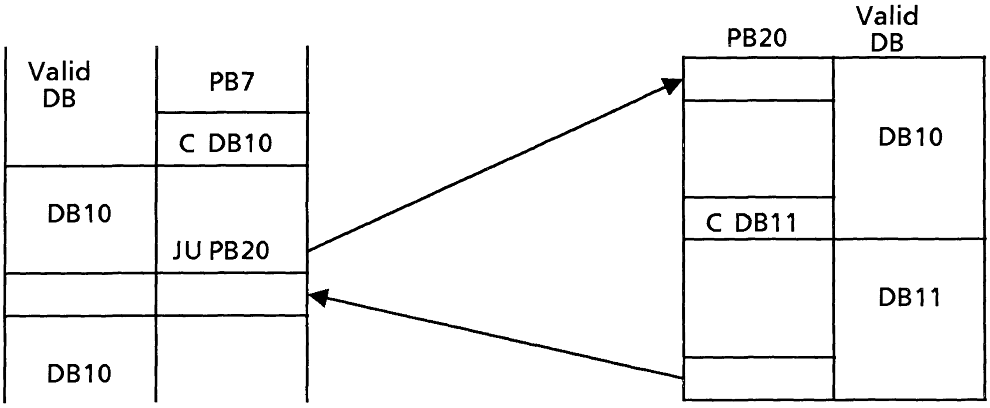
Input	Values stored	
0000 : KH = A13C	DW0	A13C
0001 : KT = 100.2	DW1	2100
0003 : KF = +21874	DW2	5572

Figure 7-5 Example of the Contents of a Data Block

Data blocks can also be generated or deleted in the control program (see Section 8.1.8).

Program processing with data blocks:

- A data block must be called in the program with the C DB x (x=0 to 255) or QX DXa (a=0 to 255) operation.
- Within a block, a data block remains valid until another data block is called.
- When the program jumps back into the higher-level block, the data block that was valid before the block call is again valid.
- In all organization blocks (OBs), the data blocks used by the user program must be opened with the relevant C DBxx operation.



When PB20 is called, the valid data area is entered into a memory.
When the program jumps back, this area is reopened.

Figure 7-6 Validity Areas of Data Blocks

7.5 Processing Blocks

The preceding sections have already described how to use blocks. In addition, Chapter 8 describes all operations necessary to work with blocks.

Of course, blocks that have already been programmed can be changed. Possibilities for changing blocks are described here only briefly. The operating instructions for the particular programmer used explain the necessary steps in detail.

7.5.1 Modifying the Program

You can modify the program, regardless of block type, with the following programmer functions:

- INPUT
- OUTPUT
- STATUS (see Chapter 4.2.3)

With the above functions, you can make the following changes:

- insert, delete, or overwrite statements
- insert or delete segments.

7.5.2 Modifying Blocks

Program modifications relate to the contents of a block. You can also delete or overwrite entire blocks. However, this does not delete the blocks in the program memory. Instead, it simply invalidates the blocks. The memory locations of these blocks cannot be written to again. As a result, new blocks might not be accepted. The programmer reports the error message "No memory space".
Eliminate this by compressing the PLC memory.

7.5.3 Compressing the Program Memory

Figure 7-7 explains compressing. Internally one block is shifted per cycle.

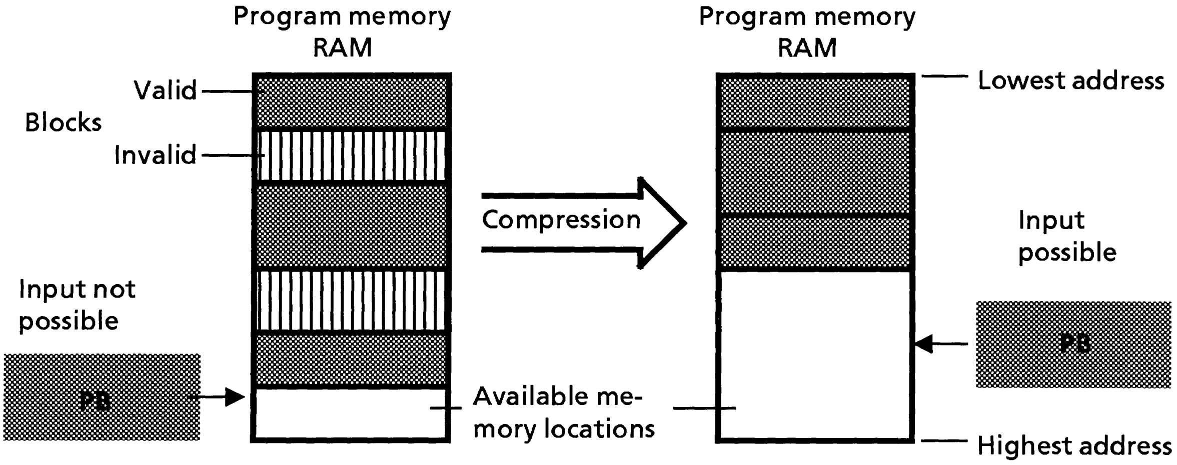


Figure 7-7 Compression Process

You can compress the internal program memory in the following ways:

- With the COMPRESS programmer function
  or
- With the integral FB238 (COMPR, see Chapter 2.11.1).

Shifting of blocks cannot be interrupted.

7.6 Number Representation

STEP 5 allows you to work with numbers in different representations:

- Constant bytes (KB, KY)
- Fixed-point numbers (KF)
- Hexadecimal numbers (KH, DH)
- Floating-point numbers (KG)
- BCD-coded numbers
- Bit patterns (KM)

Table 7-6 gives you an overview of the different number representations.

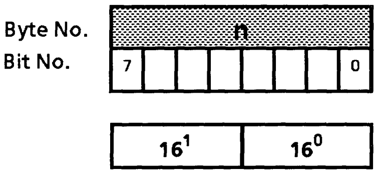
Table 7-6 Number Representations in the CPU 945

Number Representation	Identifier	Bits	Number Range
Constant byte	KB KY	8 16	0 to 255 0.0 to 255.255
Fixed-point number	KF	16	– 32768 to 32767
Hexadecimal number	KH DH	16 32	0000 to FFFF 0000 0000 to FFFF FFFF
BCD-coded number		16 (4 Tetrads) 32 (8 Tetrads)	– 999 to + 999 (KH: F999 to 0999) – 999 9999 to + 999 9999 (DH: F999 9999 to 0999 9999)
Floating-point number	KG	32	± 0.1469368 × 10^{-38} ... ± 0.1701412 × 10^{39} und 0.0 × 10^{0}

The following figures show you the representation inside the CPU for

- 8-bit numbers
- 16-bit numbers
- 32-bit numbers

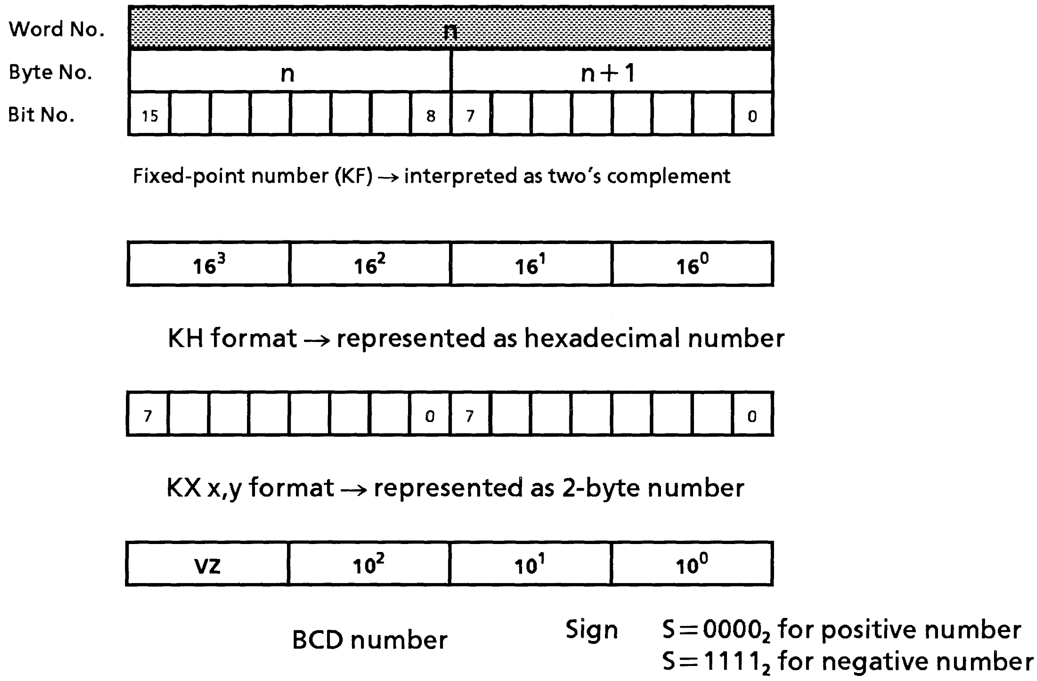
8-bit numbers



KB format → interpreted as number

Figure 7-8 8-Bit Number Representation

16-bit numbers



Fixed-point number (KF) → interpreted as two's complement

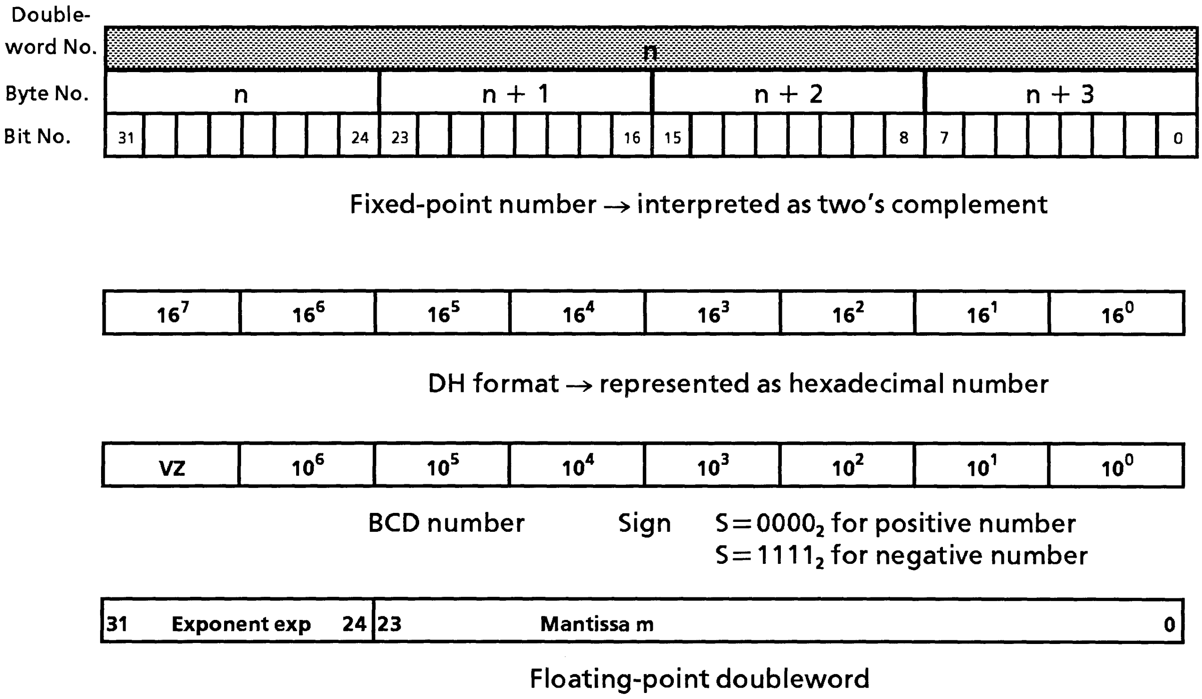
KH format → represented as hexadecimal number

KX x,y format → represented as 2-byte number

BCD number Sign $S = 0000_2$ for positive number
$S = 1111_2$ for negative number

Figure 7-9 16-Bit Number Representation

32-bit numbers



Fixed-point number → interpreted as two's complement

DH format → represented as hexadecimal number

BCD number Sign $S = 0000_2$ for positive number
$S = 1111_2$ for negative number

Floating-point doubleword

Figure 7-10 32-Bit Number Representation

Floating-point numbers

A floating-point number consists of

- the mantissa m (24 bits)
  and
- the exponent exp (8 bits) (see Figure 7-11)

The value of the floating-point number is derived from: $G = m \times 2^{exp}$.

The exponent is a whole binary number in two's complement representation.
Value range: 128 ≤ 127

The mantissa is a scaled broken binary number in two's complement representation (see Figure 7-11). Value range: m=0
+0.5≤m<+1
-1≤m<-0.5

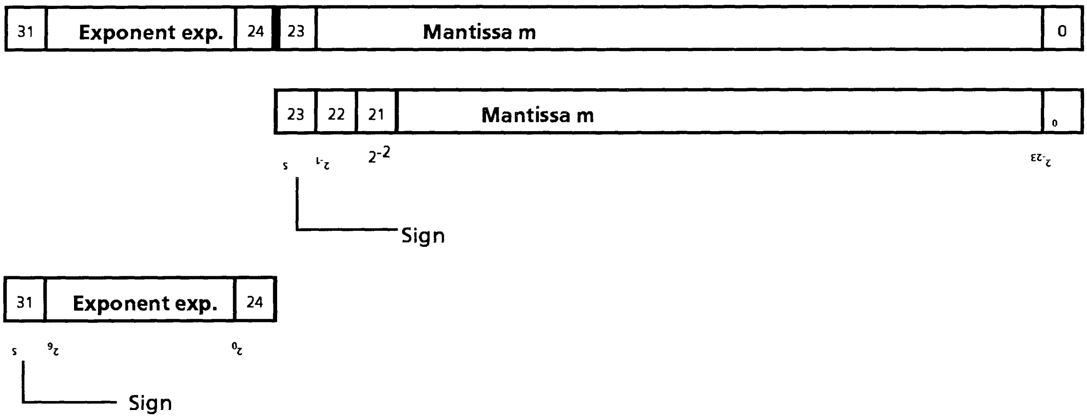
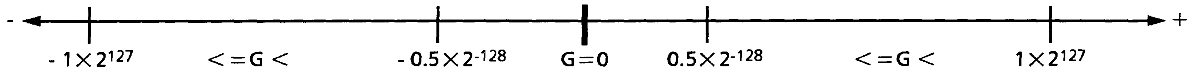


Figure 7-11 Floating-Point Numbers - Values of the Bits

Floating-point numbers can be represented in the following value range



Entering floating-point numbers Z with the programmer

Z = 12.34567

$$\text{L KG }\underbrace{+1234567}_{\text{Mantissa}\,[1]}\ \underbrace{+02}_{\text{Exponent (base 10) with sign}}$$

$Z = +0.1234567 \times 10^{+2} = 12.34567$

Z = -0.005

$$\text{L KG }\underbrace{-5000000}_{\text{Mantissa}\,[1]}\ \underbrace{-2}_{\text{Exponent (base 10) with sign}}$$

$Z = -0.5 \times 10^{-2} = -0.005$

Use floating-point numbers for large calculations, especially

- Multiplication and division
  and
- if you are working with very large or very small numbers.

[1] The entry is interpreted as a value after the decimal point (+0.1234567 or -0,.000 000)

8 STEP 5 Operations

Figures

Tables

Tables

8 STEP 5 Operations

This chapter describes the operations of the STEP 5 programming language.
Compared with CPUs 941 to 944, the operation set of CPU 945 has been considerably extended.

The following operations, for example, are new in the CPU 945:

- Floating-point arithmetics
- Conversion operations
- Fixed-point multiplication and division
- Bit operations with data blocks
- S flag operations
  and
- Additional system operations

For an overview of the operations which can be used in the CPU 945 also refer to the CPU 945 Pocket Guide.

The STEP 5 programming language has the following three operation types:

- Basic Operations include functions that can be executed in organization, program, sequence, and function blocks. Except for the addition (+F), subtraction (-F), and organizational operations, the basic operations can be input and output in the statement list (STL), control system flowchart (CSF), or ladder diagram (LAD) methods of representation.

- Supplementary Operations include complex functions such as substitution statements, test functions, and shift and conversion operations.
  They can be input and output in STL form only.

- System operations usually involve absolute addressing and therefore require exact knowledge of the hardware, address space, etc. Only an experienced programmer should use them.
  System operations can be input and output in STL form only.

Table 8.1 Overview of operation types

Operations	Executed in			
	OB	FB/FX	PB	SB
Basic operations	X	X	X	X
Supplementary operations		X		
System operations		X		

8.1 Basic Operations

Sections 8.1.1 through 8.1.9 describe the basic operations.

8.1.1 Boolean Logic Operations

Table 8.2 provides an overview of Boolean logic operations.

Table 8.2 Overview of Boolean Logic Operations

Operation	Operand		Meaning
O			**Combine AND operations through logic OR.** Combine the result of the next AND logic operation (RLO) with the previous RLO through logic OR.
A(			**Combine expression enclosed in parentheses through logic AND.** Combine the RLO of the expression enclosed in parentheses with the previous RLO through logic AND. [1]
O(			**Combine expression enclosed in parentheses through logic OR.** Combine the RLO of the expression enclosed in parentheses with the previous RLO through logic OR. [1]
)			**Close parenthesis.** Conclude the expression enclosed in parentheses.
A	□	□	**Scan operand for "1" and combine with RLO through logic AND.** The result is "1" when the operand in question carries signal state "1". Otherwise the scan results in "0". Combine this result with the RLO in the processor through logic AND. [1]
O	□	□	**Scan operand for "1" and combine with RLO through logic OR.** The result is "1" when the operand in question has signal state "1". Otherwise the scan results in "0". Combine this result with the RLO in the processor through logic OR. [1]
AN	□	□	**Scan operand for "0" and combine with RLO through logic AND.** The result is "1" when the operand in question has signal state "0". Otherwise the scan results in "0". Combine this result with the RLO in the processor through logic AND. [1]
ON	□ ↑	□ ↑	**Scan operand for "0" and combine with RLO through logic OR.** The result is "1" when the operand in question has signal state "0". Otherwise the scan results in "0". Combine this result with the RLO in the processor through logic OR. [1]
ID	 I Q F S D T C	**Parameter** 0.0 to 127.7 0.0 to 127.7 0.0 to 255.7 0.0 to 4095.7 0.0 to 255.15 0 to 255 0 to 255	

[1] If the scan follows an RLO limiting operation directly (first scan), the scan result is reloaded as a new RLO.

8.1.2 Set/Reset Operations

Set/reset operations store the result of logic operation (RLO) formed in the processor. The stored RLO represents the signal state of the addressed operand. Storage can be dynamic (assignment) or static (set and reset). Table 8.3 provides an overview of the set/reset operations. Examples follow the table.

Table 8.3 Overview of the Set/Reset Operations

Operation	Operand		Meaning
S	□	□	**Set** The first time the program is scanned with RLO= "1", signal state "1" is assigned to the addressed operand. An RLO change does not affect this status.
R	□	□	**Reset** The first time the program is scanned with RLO= "0", signal state "0" is assigned to the addressed operand. An RLO change does not affect this status.
=	□ ↑	□ ↑	**Assign** Every time the program is scanned, the current RLO is assigned to the addressed operand.
ID		**Parameter**	
	I	0.0 to 127.7	
	Q	0.0 to 127.7	
	F	0.0 to 255.7	
	S	0.0 to 4095.7	
	D	0.0 to 255.15	

8.1.3 Load and Transfer Operations

Use load and transfer operations to do the following:

- exchange information between various operand areas
- prepare times and counts for further processing
- load constants for program processing.

Information flows indirectly via accumulators (ACCU 1 and ACCU 2). The accumulators are special registers in the CPU for temporary storage. In the CPU 945 they are each 32 bits long. The accumulators are structured as shown in Figure 8.1.

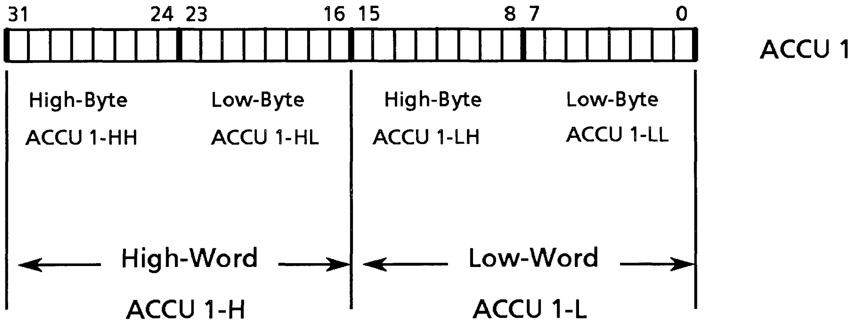


Figure 8.1 Accumulator Structure

When processing 16-bit commands, the CPU 945 is fully compatible with CPUs 941 to 944.

You can load and transfer permissible operands in bytes or words. For exchange in bytes, information is stored right-justified, i.e., in ACCU-LL.
The remaining bits are set to zero.
For exchange in words, information is stored right-justified, i. e. in ACCU 1-L. The remaining bits are set to zero (not for LIR command).
You can process the information in the two accumulators using various operations.
You can program load and transfer operations graphically only in combination with timer or counter operations; otherwise you can represent them only in STL form.

The various operations are listed in the following table.

Table 8.4 Overview of Load and Transfer Operations

Operation	Operand		Meaning
L	□	□	**Load** ACCU 1 is loaded into ACCU 2. The operand contents are copied into ACCU 1 regardless of the RLO. The RLO is not affected.
T	□ ↑	□ ↑	**Transfer** The contents of ACCU 1 are assigned to an operand regardless of the RLO. The RLO is not affected.
ID			**Parameter**
	IB		0 to 127
	IW		0 to 126
	ID		0 to 124
	QB		0 to 127
	QW		0 to 126
	QD		0 to 124
	PY[2]		0 to 127
			128 to 255
	PW[2]		0 to 126
			128 to 254
	OY		0 to 255
	OW		0 to 254
	FY		0 to 255
	FW		0 to 254
	FD		0 to 252
	SY		0 to4095
	SW		0 to4094
	SD		0 to4092
	DR		0 to 255
	DL		0 to 255
	DW		0 to 255
	DD		0 to 254
	T[1]		0 to 255
	C[1]		0 to 255
	KM[1]		random bit pattern (16 bits)
	KH[1]		0 to FFFF
	KF[1]		- 32768 to + 32767
	KY[1]		0 to 255 per byte (2 bytes)
	KB[1]		0 to 255
	KS[1]		any 2 alphanumeric characters
	KT[1]		0.0 to 999.3
	KC[1]		0 to 999
	DH[1]		0 to FFFF FFFF
	KG[1]		$\pm 0.1469368 \times 10^{-38}$ to $\pm 0.1701412 \times 10^{39}$; 0.0×10^{0}
LD	□ ↑	□ ↑	**Load in BCD** Binary times and counts are loaded into ACCU 1 in BCD code regardless of the RLO.
ID			**Parameter**
	T		0 to 255
	C		0 to 255

[1] These operands cannot be used for transfer

[2] For "transfer" the process image is updated

Load Operation:

During loading, information is copied from a memory area, e.g., from the PII, into ACCU 1.
The previous contents of ACCU 1 are shifted to ACCU 2.
The original contents of ACCU 2 are lost.

Example: An input byte, an input word and an input double word are loaded consecutively from the PII into the accumulator.
Loading does not change the PII (see Figure 8.2).

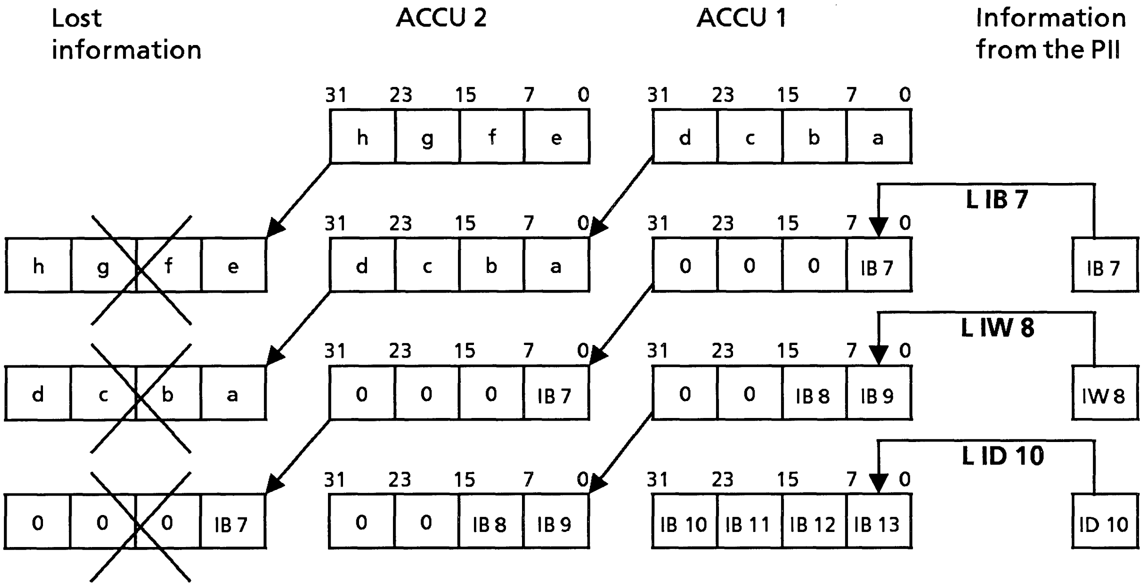


Figure 8.2 Execution of the Load Operation

Transfer Operation:

During transfer, information from ACCU 1 is copied into the addressed memory area, e.g., into the PIQ.
This transfer does not affect the contents of ACCU 1.
When transfer to the digital output area (T PY 0 to 127, T PW 0 to 126) occurs, the byte or word in question in the PIQ is automatically updated.

Example: Figure 8.3 shows how byte a, the low byte of the low word in ACCU 1, is transferred to QB 5.

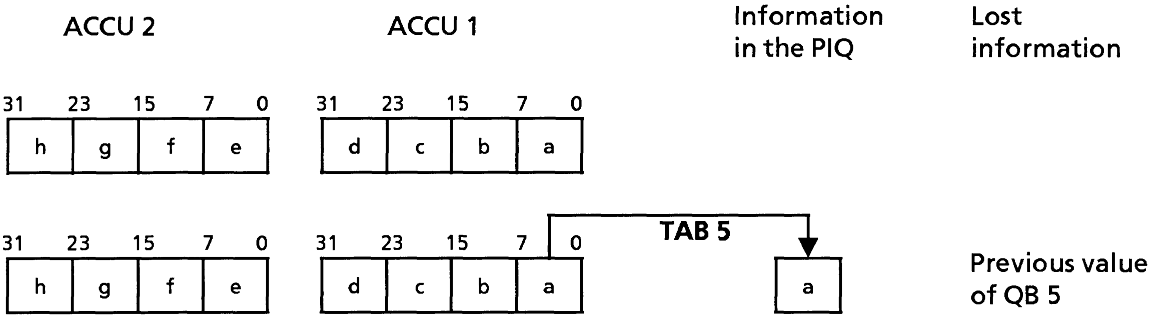
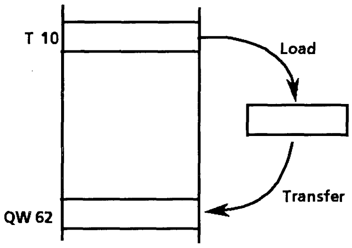


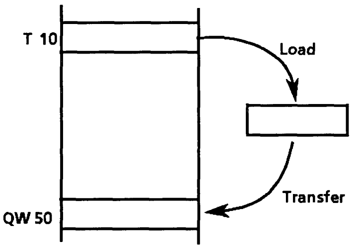
Figure 8.3 Transferring a Byte

Loading and Transferring a Time (See also Timer and Counter Operations)

Example	Representation	STL
During graphic input, QW 62 is assigned to output BI of a timer. The programmer automatically stores the corresponding load and transfer operation in the control program. Thus the contents of the memory location addressed with T 10 are loaded into ACCU 1. Afterwards, the contents of the accumulator are transferred to the process image addressed with QW 62. In this example, you can see timer T 10 at QW 62 in binary code. Outputs BI and DE are digital outputs. The time at output BI is in binary code. The time at output DE is in BCD code with time base.		A I 5.0 L IW 22 SP T 10 L T 10 T QW 62

8

Loading and Transferring a Time (Coded)

Example	Representation	STL
The contents of the memory location addressed with T 10 are loaded into the accumulator in BCD code. Then a transfer operation transfers the accumulator contents to the process image memory location addressed by QW 50. A coding operation is possible only indirectly for the graphic representation forms LAD and CSF by assigning an address to output DE of a timer or counter location. However, with STL, this operation can be entered with a separate statement.		A I 5.0 L IW 22 SP T 10 LC T 10 T QW 50

8.1.4 Timer Operations

The program uses timer operations to implement and monitor chronological sequences. The following table provides an overview of timer operations. Examples follow the table.

Table 8.5 Overview of Timer Operations

Operation *	Operand		Meaning
SP 1_-_	□	□	**Pulse Timer** The timer is started on the leading edge of the RLO. When the RLO is "0", the timer is set to "0". Scans result in signal state "1" as long as the timer is running.
SE 1_-_V	□	□	**Extended Pulse Timer** The timer is started on the leading edge of the RLO. When the RLO is "0", the timer is not affected. Scans result in signal state "1" as long as the timer is running.
SD T!-!0	□	□	**On-Delay Timer** The timer is started on the leading edge of the RLO. When the RLO is "0", the timer is set to "0". Scans result in signal state "1" when the timer has run out and the RLO is still pending at the input.
SS T!-!S	□	□	**Stored On-Delay Timer** The timer is started on the leading edge of the RLO. When the RLO is "0", the timer is not affected. Scans result in signal state "1" when the timer has run out. The signal state becomes "0" when the timer is reset with the "R" operation.
SF 0!-!T	□	□	**Off-Delay Timer** The timer is started on the trailing edge of the RLO. When the RLO is "1", the timer is set to its initial value. Scans result in signal state "1" as long as the RLO at the input is "1" or the timer is still running.
R	□ ↑	□ ↑	**Reset Timer** The timer is reset to its initial value as long as the RLO is "1". When the RLO is "0", the timer is not affected. Scans result in signal state "0" as long as the timer is reset or has not been started yet.
ID	T	**Parameter** 0 to 255	

* The corresponding graphical PG symbol is indicated below the operations.

Loading a Time

Timer operations call internal timers.
When a timer operation is started, the word in ACCU 1-L is used as a time value. You must therefore first specify time values in the accumulator.

You can load a constant time value with:

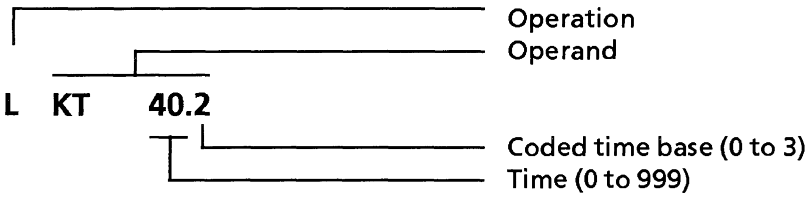
L KT constant time value
or
any other load
command

} These data types must be in BCD code with the corresponding time base.

A timer can also be loaded with random ACCU contents. However, the correct format of the time value must be observed.

Loading a Constant Time Value:

The following example shows how you can load a time value of 40 sec.



Key for Time Base:

Base	0	1	2	3
Factor	0.01 s	0.1 s	1 s	10 s

Example: KT 40.2 corresponds to 40 x 1 sec.

Tolerance:

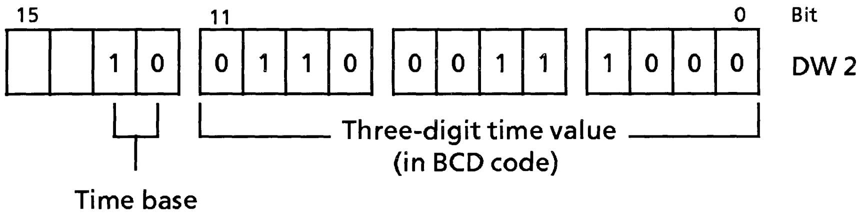
The time tolerance is equivalent to the time base. Therefore, always use the smallest time base possible.

Examples	Operand	Minimum times	Time Interval
Possible settings for the time 40 sec.	KT 400.1	400 × 0.1 sec. - 0.1 sec.	39.9 sec. to 40 sec.
	KT 40.2	40 × 1 sec. - 1 sec.	39 sec. to 40 sec.
	KT 4.3	4 × 10 sec. - 10 sec.	30 sec. to 40 sec.

Loading a Time as Input, Output, Flag, or Data Word

Load Statement: **L DW 2**

The time 638 sec. is stored in data word DW 2 in BCD code.
Bits 14 and 15 are insignificant for the time value.



Key for Time Base:

Base	0 0	0 1	1 0	1 1
Factor	0.01 s	0.1 s	1 s	10 s

You can also use the control program to write to data word DW 2.

Example: Store the value 270 x 100 msec. in data word DW 2 of data block DB3.

```
C   DB   3
L   KT  270.1
T   DW   2
```

Output of the Current Time [1]

You can use a load operation to put the current time into ACCU 1 and process it further from there (see Figure 8.4).
Use the "Load in BCD" operation for digital display output.

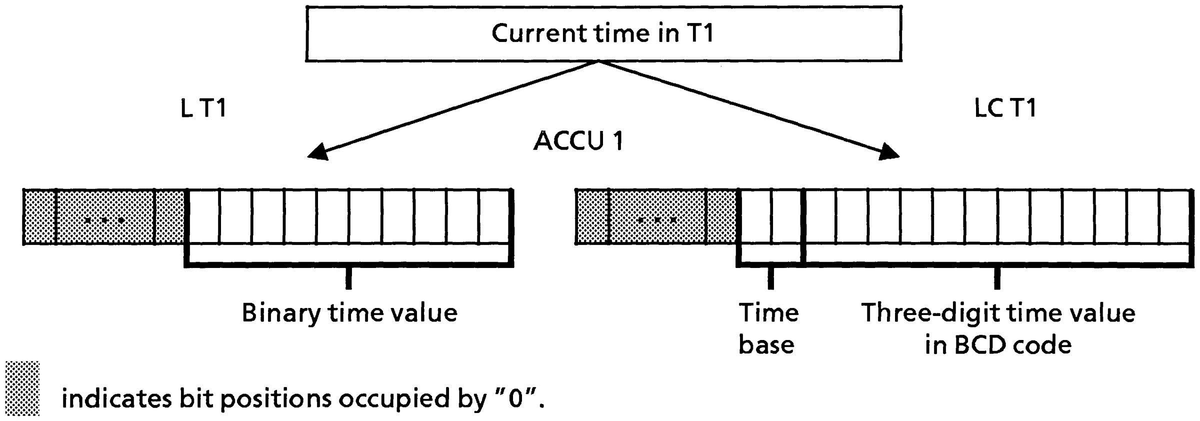

Current time in T1
L T1
LC T1
ACCU 1
...
...
Binary time value
Time base
Three-digit time value in BCD code
indicates bit positions occupied by "0".

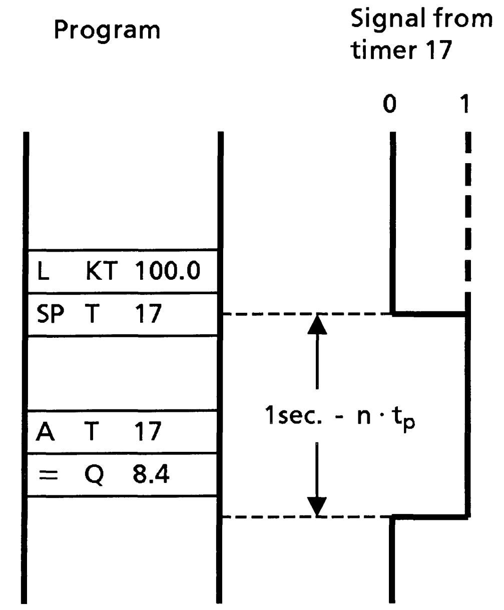
Figure 8.4 Output of the Current Time (Example)

[1] The current time is the time value in the addressed timer

Starting a Timer

In the PLC, timers run asynchronously to program execution. The time that has been set can run out during a program execution cycle. It is evaluated by the next time scan. In the worst case, an entire program execution cycle can go by before this evaluation. Consequently, timers should not activate themselves.

Example:

Schematic Representation	Explanation
 Program Signal from timer 17 0 1 L KT 100.0 SP T 17 A T 17 = Q 8.4 1sec. - n · t_p n: number of program execution cycles t_p: program scan time	The schematic shows the "nth + 1" processing cycle since timer T 17 * was started. Although the timer ran out shortly after the statement "=Q 8.4", output Q 8.4 remains set. The change is not considered until the next program execution cycle. * KT 100.0 is equal to 1 sec.

Except for "Reset timer," all timer operations are started only on an edge of the RLO. (The RLO alternates between "0" and "1".)

After being started, the loaded time is decremented in units corresponding to the time base until it reaches zero.

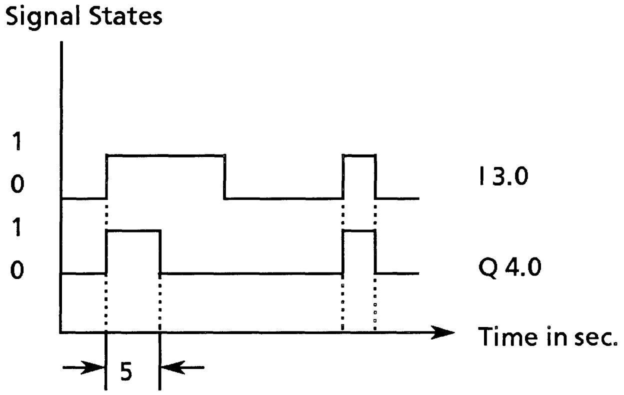
If there is an edge change while the timer is running, the timer is reset to its initial value and restarted.

The signal state of a timer can be interrogated with Boolean logic operations.

Pulse

Example:
Output Q 4.0 is set when the signal state at input I 3.0 changes from "0" to "1". However, the output should not remain set longer than 5 sec.

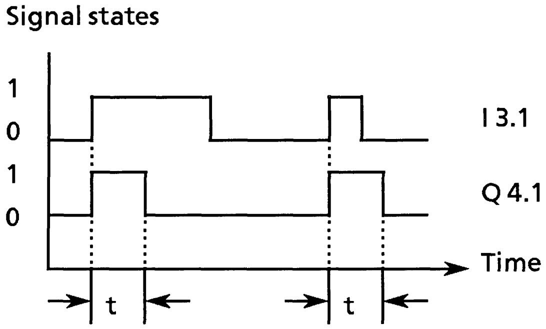
Timing Diagram	STL

	`A  I   3.0`
	`L  KT  50.1`
	`SP T   1`
	`A  T   1`
	`=  Q   4.0`

Extended pulse

Example:
Output Q 4.1 is set for a specific time when the signal at input I 3.1 changes to "1". The time is indicated in IW 15.

Timing Diagram	STL

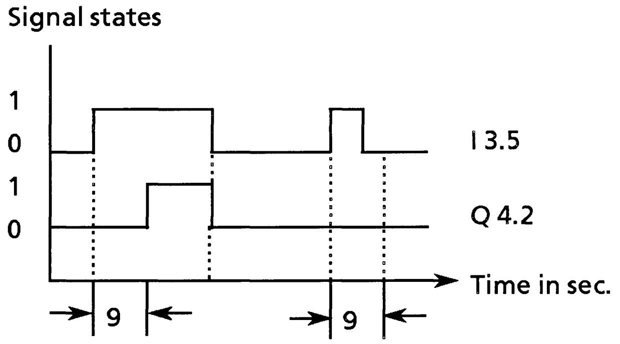
	`A  I   3.1`
	`L  IW  15`
	`SE T   2`
	`A  T   2`
	`=  Q   4.1`

Note

The time tolerance is equivalent to the time base.

On-delay

Example:
Output Q 4.2 is set 9 sec. after input I 3.5. It remains set as long as the input is "1".

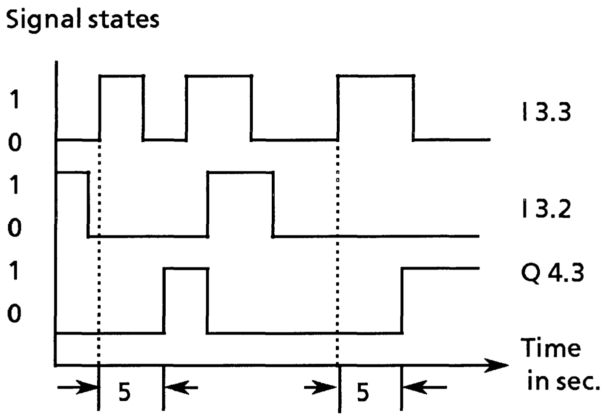
Timing Diagram	STL
 Signal states 1 0 1 0 I 3.5 Q 4.2 Time in sec. 9 9	A I 3.5 L KT 9.2 SD T 3 A T 3 = Q 4.2

Note

The time value "9 sec." will have a sharper tolerance if you load the timer with the statement "L KT 900.0".

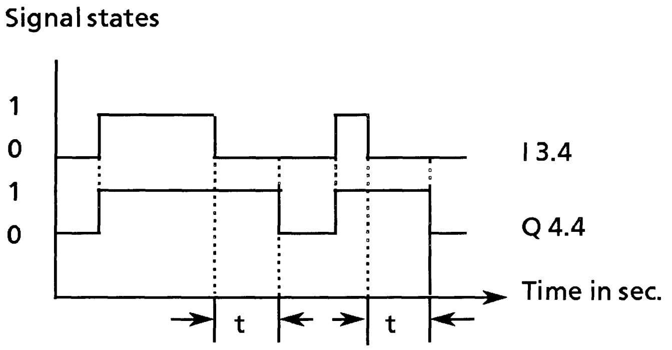
Stored On-Delay and Reset

Example:
Output Q 4.3 is set 5 sec. after I 3.3.
Further changes in the signal state at input I 3.3 do not affect the output.
Input I 3.2 resets timer T 4 to its initial value and sets output Q 4.3 to zero.

Timing Diagram	STL
 Signal states 1 0 1 0 1 0 I 3.3 I 3.2 Q 4.3 Time in sec. 5 5	A I 3.3 L KT 50.1 SS T 4 A I 3.2 R T 4 A T 4 = Q 4.3

Off-Delay

Example:
When input I 3.4 is reset, output Q 4.4 is set to zero after a certain delay (t). The value in FW 13 specifies the delay time.

Timing Diagram	STL
	A I 3.4 L FW 13 SF T 5 A T 5 = Q 4.4

8.1.5 Counter Operations

The CPU uses counter operations to handle counting jobs directly. Counters can count up and down. The counting range is from 0 to 999 (three decades). The following table provides an overview of the counter operations. Examples follow the table.

Table 8.6 Overview of Counter Operations

Operation	Operand		Meaning
S	□	□	**Set Counter** The counter is set on the leading edge of the RLO.
R	□	□	**Reset Counter** The counter is set to zero as long as the RLO is "1".
CU	□	□	**Count Up** The count is incremented by 1 on the leading edge of the RLO. When the RLO is "0", the count is not affected.
CD	□ ↑	□ ↑	**Count Down** The count is decremented by 1 on the leading edge of the RLO. When the RLO is "0", the count is not affected.
ID	Z	Parameter 0 to 255	

Loading a Count

Counter operations call internal counters.
When a counter is set, the word in ACCU 1 is used as a count. You must therefore first store counts in the accumulator.

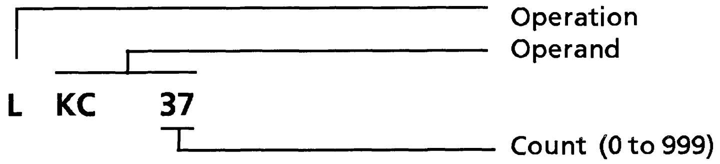
A constant count is loaded with:

L KC constant count
or
any other load command } The data for these words must be in BCD code.

A count can also be loaded with random ACCU contents. However, the correct format of the count must be observed.

Loading a Constant Count:

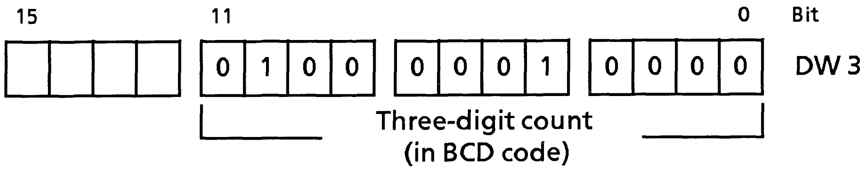
The following example shows how the count 37 is loaded.



Loading a Count as Input, Output, Flag, or Data Word

Load statement: **L DW 3**

The count 410 is stored in data word DW 3 in BCD code.
Bits 12 to 15 are insignificant for the count.



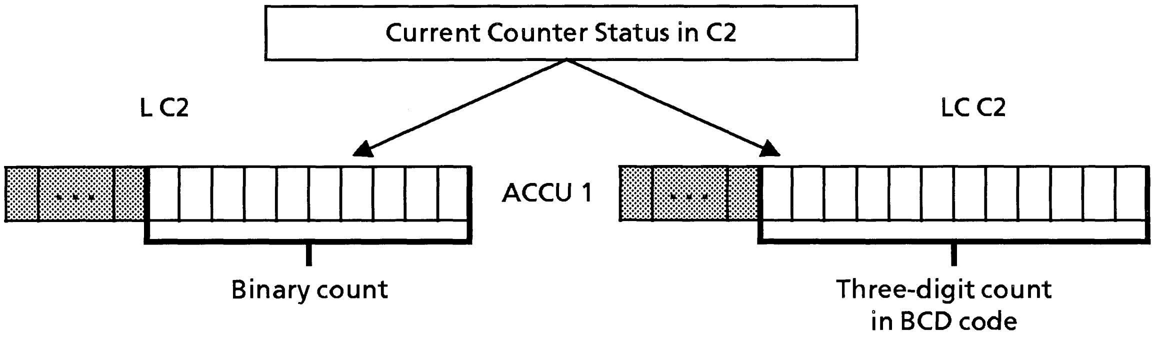
15				11											0	Bit
				0	1	0	0	0	0	0	1	0	0	0	0	DW 3

Scanning the Counter

Use Boolean logic operations to scan the counter status (e.g., A Cx). As long as the count is not zero, the scan result is signal state "1".

Outputting the Current Counter Status

You can use a load operation to put the current counter status into ACCU 1 and process it further from there. The "Load in BCD" operation outputs a digital display (see Figure 8.5).
The "Load in BCD" operation is suitable for output via a numeric display.

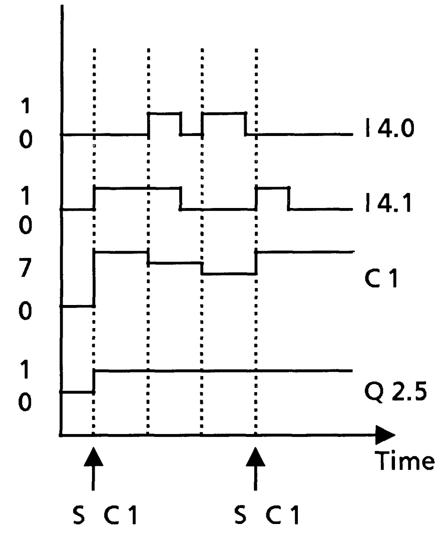


▨ indicates bit positions occupied by "0".

Figure 8.5 Outputting the Current Counter Status (example)

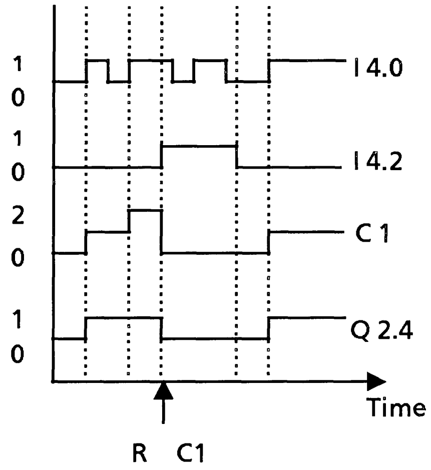
Setting a Counter "S" and Counting Down "CD"

Example:
When input I 4.1 is switched on (set), counter 1 is set to the count 7. Output Q 2.5 is now "1".
Every time input I 4.0 is switched on (count down), the count is decremented by 1.
The output is set to "0" when the count is "0".

Timing Diagram	STL
	A I 4.0 CD C 1 A I 4.1 L KC 7 S C 1 A C 1 = Q 2.5

Resetting a Counter "R" and Counting Up "CU"

Example:
When input I 4.0 is switched on, the count in counter 1 is incremented by 1. As long as a second input (I 4.2) is "1", the count is reset to "0".
The A C1 operation results in signal state "1" at output Q 2.4 as long as the count is not "0".

Timing Diagram	STL
 1 0 I 4.0 1 0 I 4.2 2 0 C 1 1 0 Q 2.4 Time R C1	A I 4.0 CU C 1 A I 4.2 R C 1 A C 1 = Q 2.4

8.1.6 Comparison Operations

Processing a Comparison Operation

When using comparison operations, make sure that the operands have the same number format. Execution of the operations is independent of the RLO. The result is binary and is available as RLO for further program execution. If the comparison is satisfied, the RLO is "1", otherwise it is "0". Executing the comparison operations sets the condition codes (see Section 8.4).

Example: The values of the input words 18 and 20 are compared. If they are equal, output Q3.0 is set.

```
L   IW  18
L   IW  20
!=F
=   Q     3.0
```

Comparison Operations

Fixed-point comparison operations compare the contents of ACCU 2-L and ACCU 1-L. The contents of the accumulators are interpreted as two's complement and remain unchanged. The comparison takes into account the number notation of the operands, i.e. the contents of ACCU 1-L and ACCU 2-L are interpreted as fixed-point numbers.

Table 8.7 Overview of Fixed-Point Comparison Operations (16-Bit Operations)

Operation	Operand		Meaning
! = F			**Comparison for equal to** Compare two fixed-point numbers for equal to: if ACCU 2-L = ACCU 1-L, the RLO is "1".
> < F			**Comparison for not equal to** Compare two fixed-point numbers for not equal to: if ACCU 2-L≠ACCU 1-L, the RLO is "1".
> F			**Comparison for greater than** Compare two fixed-point numbers for greater than: if ACCU 2-L > ACCU 1-L, the RLO is "1".
> = F			**Comparison for greater than or equal to** Compare two fixed-point numbers for greater than: if ACCU 2-L≥ACCU 1-L, the RLO is "1".
< F			**Comparison for less than** Compare two fixed-point numbers for less than: if ACCU 2-L < ACCU 1-L, the RLO is "1".
< = F			**Comparison for less than or equal to** Compare two fixed-point numbers for less than or equal to: if ACCU 2-L≤ACCU 1-L, the RLO is "1".

The comparison operations are also possible as 32-bit operations.
32-bit operations compare the whole acccumulator contents.
The fixed-point double word comparison operations interpret the contents of the accumulators as two's complement. The floating-point comparison operations interpret the contents of the accumulators as floating-point numbers.

Table 8.8 Overview of Fixed-Point Double-Word Comparison Operations (32-Bit Operations)

Operation	Operand		Meaning
! = D			**Comparison for equal to** Compare two fixed-point double words for equal to: if ACCU 2=ACCU 1, the RLO is "1".
>< D			**Comparison for not equal to** Compare two fixed-point double words for not equal to: if ACCU 2≠ACCU 1, the RLO is "1".
> D			**Comparison for greater than** Compare two fixed-point double words for greater than: if ACCU 2 > ACCU 1, the RLO is "1".
>= D			**Comparison for greater than or equal to** Compare two fixed-point double words for greater than or equal to: if ACCU 2≥ACCU 1, the RLO is "1".
< D			**Comparison for less than** Compare two fixed-point double words for less than: if ACCU 2 < ACCU 1, the RLO is "1".
<= D			**Comparison for less than or equal to** Compare two fixed-point double words for less than or equal to: if ACCU 2≤ACCU 1, the RLO is "1".

Table 8.9 Overview of Floating-Point Comparison Operations (32-Bit Operations)

Operation	Operand		Meaning
! = G			**Comparison for equal to** Compare two floating-point numbers for equal to: if ACCU 2=ACCU 1, the RLO is "1".
>< G			**Comparison for not equal to** Compare two floating-point numbers for not equal to: if ACCU 2≠ACCU 1, the RLO is "1".
> G			**Comparison for greater than** Compare two floating-point numbers for greater than: if ACCU 2 > ACCU 1, the RLO is "1".
>= G			**Comparison for greater than or equal to** Compare two floating-point numbers for greater than or equal to: if ACCU 2≥ACCU 1, the RLO is "1".
< G			**Comparison for less than** Compare two floating-point numbers for less than: if ACCU 2 < ACCU 1, the RLO is "1".
<= G			**Comparison for less than or equal to** Compare two floating-point numbers for less than or equal to: if ACCU 2≤ACCU 1, the RLO is "1".

8.1.7 Arithmetic Operations

Arithmetic operations interpret the contents of the accumulators as

- fixed-point numbers (16 bits)
- fixed-point numbers (32 bits)
  or as
- floating-point numbers (32 bits)

and manipulate them. The result is stored in ACCU 1. The operations are listed in the following table and explained in an example following the table. For condition code generation and overflow and underflow behaviour refer to Section 8.4.

"F" and "D" operations expect fixed-point numbers in two's complement notation in ACCU 1-L or ACCU 2-L and ACCU 1 or ACCU 2.

After the operation, the sign of " + F" operations and "-F" operations is also in
ACCU 1-H: - : FFFF$_H$
+ : 0000$_H$

"G" operations expect normalized floating-point notations in ACCU 1 or ACCU 2 (for SIMATIC floating-point format refer to Section 7.6).

Table 8.10 Overview of Arithmetic Operations

Operation	Operand		Meaning
			Addition
			Add two fixed-point numbers
+ F			ACCU 1-L=(ACCU 2-L)+(ACCU 1-L)
+ D			ACCU 1=ACCU 2+ACCU 1
			Add two floating-point numbers
+ G			ACCU 1=ACCU 2+ACCU 1
			Subtraction
			Subtract two fixed-point numbers
- F			ACCU 1-L=(ACCU 2-L) - (ACCU 1-L)
- D			ACCU 1=ACCU 2 - ACCU 1
			Subtract two floating-point numbers
- G			ACCU 1=ACCU 2 - ACCU 1
			Multiplication
			Multiply two fixed-point numbers
x F			ACCU 1=(ACCU 2-L) x (ACCU 1-L)
			Multiply two floating-point numbers
x G			ACCU 1=ACCU 2 x ACCU 1
			Division
			Divide two fixed-point numbers
: F			ACCU 1=(ACCU 2-L) : (ACCU 1-L)
			In ACCU 1-L: result; in ACCU 1-H: remainder
			Divide two floating-point numbers
: G			ACCU 1=ACCU 2 : ACCU 1

Processing an Arithmetic Operation

Before an arithmetic operation is executed, both operands must be loaded into the accumulators.

Note

When using arithmetic operations, make sure the operands have the same number format.

Arithmetic operations are executed independently of the RLO. The result is available in ACCU 1 for further processing. The contents of ACCU 2 are not changed. These operations do not affect the RLO. The condition codes are set according to the results.

STL	Explanation
L C 3	The value of counter 3 is loaded into ACCU 1.
L C 1	The value of counter 1 is loaded into ACCU 1. The previous contents of ACCU 1 are shifted to ACCU 2.
+F	The contents of the two accumulators are interpreted as 16-bit fixed-point numbers and added.
T QW 12	The result, contents of ACCU 1, is transferred to output word QW 12.

Numeric Example

	15		0	
876	00000011		01101100	ACCU 2
+			+ F	
668	00000010		10011100	ACCU 1
=				
1544	00000110		00001000	ACCU 1

8.1.8 Block Call Operations

Block call operations specify the sequence of a structured program. Tables 8.11 and 8.12 provide an overview of the block call operations. Examples follow the table.

Table 8.11 Overview of Code Block Call Operations

Operation	Operand		Meaning
JU	□	□	**Jump unconditionally** Program scanning continues in a different block regardless of the RLO. The RLO is not affected.
JC	□	□	**Jump conditionally** Program scanning jumps to a different block when the RLO is "1". Otherwise program execution continues in the previous block. The RLO is set to "1".
ID	OB PB FB SB	Parameter	0 to 255 * 0 to 255 0 to 255 * 0 to 255
DOU	□	□	**Jump unconditionally** A function block (extension) is called regardless of the RLO.
DOC	□	□	**Jump conditionally** A function block (extension) is called conditionally when the RLO is "1". Otherwise, program execution continues in the previous block. The RLO is set to "1".
ID	FX	Parameter	0 to 255

* Please note that certain OBs/FBs are reserved by the operating system.

Table 8.11 Overview of Block Call Operations (continued)

Operation	Operand		Meaning
BE	□	□	**Block end** The current block is terminated regardless of the RLO. Program scanning continues in the block in which the call originated. The RLO is "carried along" but not affected.
BEU	□	□	**Block end, unconditional** The current block is terminated regardless of the RLO. Program scanning continues in the block in which the call originated. The RLO is "carried along" but not affected.
BEC	□	□	**Block end, conditional** When the RLO is "1", the current block is terminated. Program scanning continues in the block in which the call originated. During the block change, the RLO remains "1". If the RLO is "0", the operation is not executed. The RLO is set to "1" and linear program execution continues.

Table 8.12 Overview of Data Block Call Operations

Operation	Operand		Meaning
C	DB	□	**Call a data block** A data block (DB) is called regardless of the RLO. Program scanning is not interrupted. The RLO is not affected.
CX	DX	□	**Call a data block** A data block (DX) is called regardless of the RLO.
G	DB	□	**Generate and delete a data block** * A data block to store data is set up regardless of the RLO.
GX	DX	□ ↑	**Generate and delete a data block** * A data block (extension) is generated regardless of the RLO.
		Parameter 0 to 255 **	

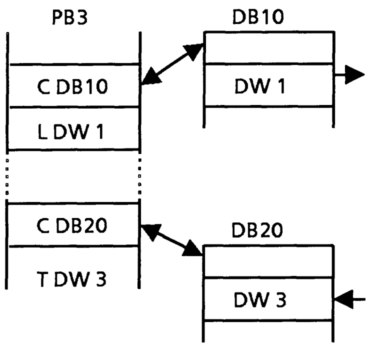
* The length of the DB/DX must be deposited in ACCU 1 prior to execution of the operation. In the case of length 0, the DB is deleted.

** Data block DB1 is reserved.

Call a Data Block "C DB"

Data blocks are always called unconditionally. All data processed following the call refers to the data block that has been called.
This operation cannot generate new data blocks. Blocks that are called must be loaded or generated prior to program execution.

Example: Program block PB3 needs information that has been programmed as data word DW 1 in data block DB10. Other data, e.g., the result of an arithmetic operation, is stored as data word DW 3 in data block DB20.

Program Sequence	STL	Explanation
 PB3 DB10 C DB10 DW 1 L DW 1 C DB20 DB20 T DW 3 DW 3	C DB 10 L DW 1 . . . . C DB 20 T DW 3	The information from data word DW 1 in data block DB 10 is loaded into the accumulator. The contents of ACCU 1 are stored in data word DW 3 of data block DB20.

The "CX DX" operation calls a Dx data block and operates like the "C DB" operation.

Generating and Deleting a Data Block

The "G DB x" statement does not call a data block. Instead, it generates a new block. If you want to use the data in this data block, call it with the "C DB" statement.
Before the "G DB" statement, indicate in ACCU 1-L the number of data words the block is to have (see the example below).

A data block generated with the "G DB" statement can have 2 to 65530 data words. If you wish to generate a DB with n data words, you must load n-1 into ACCU 1-L.

If you specify zero as the data block length, the data block in question is deleted, i.e., it is removed from the address list. It is considered nonexistent (see sections 2.11.1 and 2.11.2).

The length of the data block set up is optional. It can be set up in the CPU 945 with a maximum length of 64k words. However, please note that programmers can process blocks of limited length only.

The following overview shows the response of the CPU to various events during the generation of a new DB.

Operation	Event	Response of the CPU
G DB	- DB is to be deleted	DB is deleted
	- DB is to be deleted, but is not existing	None
	- DB is to be deleted, but cannot be deleted as it is stored in the EPROM	None
	- DB is to be regenerated	DB is generated.
	- DB is to be regenerated, but DB is already existing	None
	- Memory space is not sufficient	The CPU goes into STOP with "TRAF" or jumps to the relevant error response OB and does not generate a DB.
	- Length in ACCU 1-L > $FFF9_H$	The CPU goes into STOP with "TRAF" or jumps to the relevant error response OB and does not generate a DB.

The "GX DX" operation generates a DX data block and operates like "G DB".

Note

The block is retained as a "dead" block until the PLC memory is compressed (see Section 7.5.3).

Generating a Data Block

Example	STL	Explanation
Generate a data block with 128 data words without the aid of a programmer.	L KB 127 G DB 5	The constant fixed-point number +127 is loaded into ACCU 1. At the same time, the old contents of ACCU 1 are shifted to ACCU 2. Data block 5 is generated with a length of 128 data words (0000) in the RAM of the PC and entered in the block address list. The next time the "G DB5" operation is processed, it has no effect if the contents of ACCU 1 are not 0.

Deleting a Data Block

Example	STL	Explanation
Delete a data block that is no longer needed.	L KB 0 G DB 5	The constant fixed-point number +0 is loaded into ACCU 1. At the same time, the old contents of ACCU 1 are shifted to ACCU 2. Data block 5, which is in the RAM of the PC, is declared invalid and removed from the block address list.

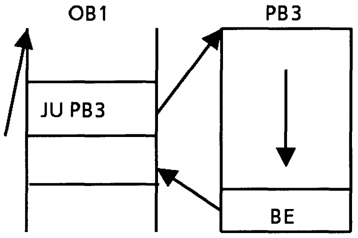
Block End "BE"

The "BE" operation terminates a block. Data blocks do not need to be terminated. "BE" is always the last statement in a block.

In structured programming, program execution jumps back to the block where the call for the current block was made.

Binary logic operations cannot be continued in a higher-order block.

Example: Program block PB3 is terminated by the "BE" statement.

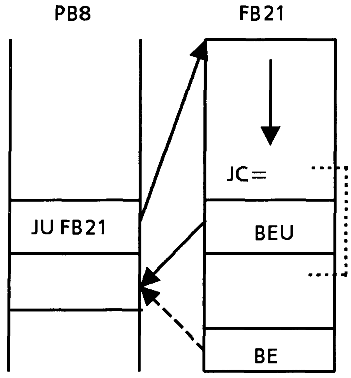
Program Sequence	STL	Explanation
	. . . . . . BE	The "BE" statement terminates program block PB 3 and causes program execution to return to organization block OB1.

Unconditional Block End "BEU"

The "BEU" operation causes a return within a block. However, jump operations can by-pass the "BEU" operation in function blocks (see Sections 8.2.10 and 8.3.4).

Binary logic operations cannot be continued in a higher-order block.

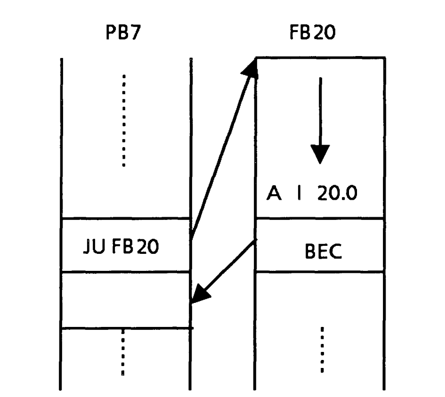
Example: Scanning of function block FB21 is terminated regardless of the RLO.

Program Sequence	STL	Explanation
	. . . . JC= BEU . . . . BE	The "BEU" statement causes program execution to leave function block FB21 and return to program block PB8.

Conditional Block End "BEC"

The "BEC" operation causes a return within a block if the previous condition has been satisfied (RLO = 1).
Otherwise, linear program execution is continued with RLO "1".

Example: Scanning of program block FB 20 is terminated if the RLO = "1".

Program Sequence	STL	Explanation
	. . . . S Q 1.0 A I 20.0 BEC	The "BEC" statement causes program execution to return to program block PB7 from function block FB20 if input I 20.0 is "1."

8.1.9 Other Operations

These operations can be programmed in STL form only.

Table 8.13 Overview of Other Operations

Operation	Operand		Meaning
NOP 0			**"No" Operation**
NOP 1			**"No" Operation**
BLD	□	□	Display Generation Commands for the Programmer
ID		**Parameter**	
		0 to 255	Is treated by the CPU like a no-operation
		130	Generate space line by carriage return
		131	Switch over to statement list (STL)
		132	Switch over to control system flowchart (CSF)
		133	Switch over to ladder diagram (LAD)
		255	Terminate segment

Display generation operations

Within a block, program parts are subdivided by "BLD" display generation operations into segments.
The no-operations and display generation operations are only relevant for the programmer when representing the STEP 5 program.
During the execution of these statements, the CPU executes no operation.

8.2 Supplementary Operations

Supplementary operations extend the operations set. However, compared to basic operations, which can be programmed in all blocks, supplementary operations have the following limitations:

- They can be programmed in function blocks only.
- They can be represented in STL form only.

Sections 8.2.1 through 8.2.11 describe the supplementary operations.

8.2.1 Load Operation

As with the basic load operations, the supplementary load operation copies information into the accumulator. Table 8.14 explains the load operation.

Table 8.14 Load Operation

Operation	Operand		Meaning
L	□	□	**Load** A word from the system data is loaded into ACCU 1 regardless of the RLO.
ID		Parameter	
	RS	0 to 255	
	RT	0 to 255	

Example	STL	Explanation
In the event of a time-out, the error address is stored in SD 103 and SD 104. An important output module is plugged in at start address 4. If the time-out is triggered by this address, the CPU is to go into the STOP mode. Otherwise a signal is to be given and program execution is to continue. You can program this example in OB24.	:L RS 103 :L KH 0000	The contents of SD 103 and the high word of the address of the important module are loaded into the accumulators.
	:><F :JC L1	If the two values are not equal, output Q 12.0 is set.
	:L RS 104 :L KH F004	The contents of SD 104 and the low word of the address of the important module are loaded into the accumulators.
	:><F L1 := Q 12.0	If the two values are not equal, output Q 12.0 is set again.
	:BEC	Program scanning continues in the block where the call originated.
	:STP	If the two values are equal, the CPU goes into the STOP mode.

8.2.2 Enable Operation

Use the enable operation (FR) to execute the following operation even without edge change:

- start a timer
- set a counter
- count up and down.

Table 8.15 presents the enable operation. An example follows the table.

Table 8.15 Enable Operation

Operation	Operand		Meaning
FR	□	□	**Enable a Timer/Counter** Timers and counters are enabled on the leading edge of the RLO. This operation restarts a timer, sets a counter, or causes a counter to count up or down when the RLO "1" is pending at the "Start" operation.
ID			**Parameter**
	T		0 to 255
	Z		0 to 255

Example	STL	Explanation
Input I 2.5 starts a timer T 2 as extended pulse (pulse width 50 sec.). This timer sets output Q 4.2 for the duration of the pulse.	A I 2.5 L KT 5.3 * SE T 2 A T 2 = Q 4.2	Start a timer T 2 as extended pulse. Output Q 4.2 is set for 50 sec.
. . . .	. . . .	
If output Q 3.4 is reset repeatedly, the timer should also be restarted repeatedly.	A Q 3.4 **FR** T 2 BE	If output Q 3.4 is set (positive edge change of the RLO) during the time in which input I 2.5 is set, the timer T 2 is restarted. Output Q 4.2 therefore remains set at the restarted time or is reset. If input I 2.5 is not set during the edge change of output Q 3.4, the timer is not restarted.

* This time value has a tolerance of ± 10 sec. Use a smaller time base if necessary.

8.2.3 Bit Test Operations and Bit Setting Operations

Bit test and bit setting operations scan digital operands bit by bit and affect them. Bit test operations must always be at the beginning of a logic operation. Table 8-12 provides an overview of these operations.

Table 8.16 Overview of Bit Operations

Operation	Operand		Meaning
TB	□	□	**Test a bit for signal state "1"** A single bit is scanned regardless of the RLO. The RLO is affected according to the bit's signal state (see Table 8.13).
TBN	□	□	**Test a bit for signal state "0"** A single bit is scanned regardless of the RLO. The RLO is affected according to the bit's signal state (see Table 8.13).
SU	□	□	**Set a bit unconditionally** The addressed bit is set to "1" regardless of the RLO. The RLO is not affected.
RU	□ ↑	□ ↑	**Reset a bit unconditionally** The addressed bit is set to "0" regardless of the RLO. The RLO is not affected.
ID		Parameter	
	I		0.0 to 127.7
	Q		0.0 to 127.7
	F		0.0 to 255.7
	D		0.0 to 255.15
	T		0.0 to 255.15
	C		0.0 to 255.15
	RS		0.0 to 255.15
	RT		0.0 to 255.15

Table 8.17 shows how the RLO is formed during the bit test operations "TB" and "TBN". An example for applying the bit operations follows the table.

Table 8.17 Effect of "TB" and "TBN" on the RLO

Operation	P		PN	
Signal state of the bit in the operation indicated	0	1	0	1
Result of logic operation	0	1	1	0

Example	STL	Explanation
A photoelectric barrier that counts piece goods is installed at input I 2.0. After every 100 pieces, the program is to jump to FB5 or FB6. After 800 pieces, counter 10 is to be reset automatically and start counting again.	C DB 10 A I 2.0 CU C 10 A I 3.0 L KC 0 S C 10 O I 4.0 O F 5.2 R C 10 LC C 10 T DW 12	Call data block 10. Input I 3.0 loads the count of counter 10 with the constant 0. With each positive edge change at I 2.0, the counter is incremented by 1. The counter is reset by either input I 4.0 or flag F 5.2. The current count of the counter is stored in data word DW 12 in BCD code.
	TBN D 12.8 JC FB 5 **TB D 12.8** JC FB 6 **TB D 12.11** = F 5.2	As long as bit 8 of data word DW 12 is zero, program processing jumps to function block FB5. This is the case for the first, third, fifth etc. batch of 100 pieces. As long as bit 8 of data word DW 12 is "1", program execution jumps to function block FB6. This is the case for the second, fourth, sixth etc. batch of 100 pieces. When data bit 11 of data word DW 12 becomes "1" (the count is then 800), flag F 5.2 is set conditionally.
A photoelectric barrier that counts piece goods is installed at input I 10.0. After every 256 pieces, the counter is supposed to be reset and start counting again.	:A I 10.0 :CU C 20 :A I 11.0 :L KC 0 :S C 20 **:TB C 20.8** :JC =FULL :BEU	Input I 11.0 loads the count of counter 20 with the constant 0. The count is incremented by 1 with each positive edge change at input I 10.0. If the count has reached 256=100H (bit 8 is "1"), program execution jumps to the label "FULL". Otherwise the block is terminated.
	FULL :RU C 20.8 :BE	Bit 8 of counter C 20 is set to "0" unconditionally. Then the count is again 000H.

NOTE

Times and counts are stored in the timer/counter word in hexadecimal notation in the 10 least significant bits (bits 0 to 9).
The time base is stored in bits 12 and 13 of the timer word.

8.2.4 Digital Logic Operations

Digital logic operations combine the contents of ACCU 1-L and ACCU 2-L logically bit by bit. The result is deposited in ACCU 1-L; the contents of ACCU 1-H and ACCU 2 are retained.
Table 8.18 provides an overview of these digital logic operations. Examples follow the table.

Table 8.18 Overview of Digital Logic Operations

Operation	Operand	Meaning
AW		Combine bit by bit through logic AND
OW		Combine bit by bit through logic OR
XOW		Combine bit by bit through EXCLUSIVE OR

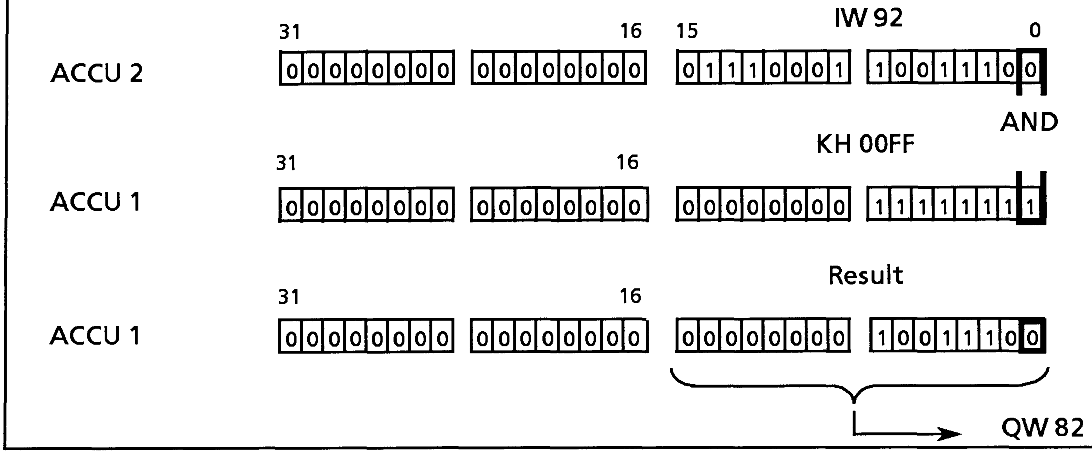
Processing a Digital Logic Operation

A digital logic operation is executed regardless of the RLO. It also does not affect the RLO. However, it sets condition codes according to the result of the arithmetic operation (see Section 8.4). The result of the arithmetic operation is available in ACCU 1 for further processing. The contents of ACCU 2 are not affected.

STL	Explanation
`L    IW   92`	Load input word IW 92 into ACCU 1.
`L    KH 00FF`	Load a constant into ACCU 1. The previous contents of ACCU 1 are shifted to ACCU 2.
`AW`	Combine the contents of both accumulators bit by bit through logic AND.
`T    QW   82`	Transfer the resulting contents from ACCU 1 to output word QW 82.

Numeric Example

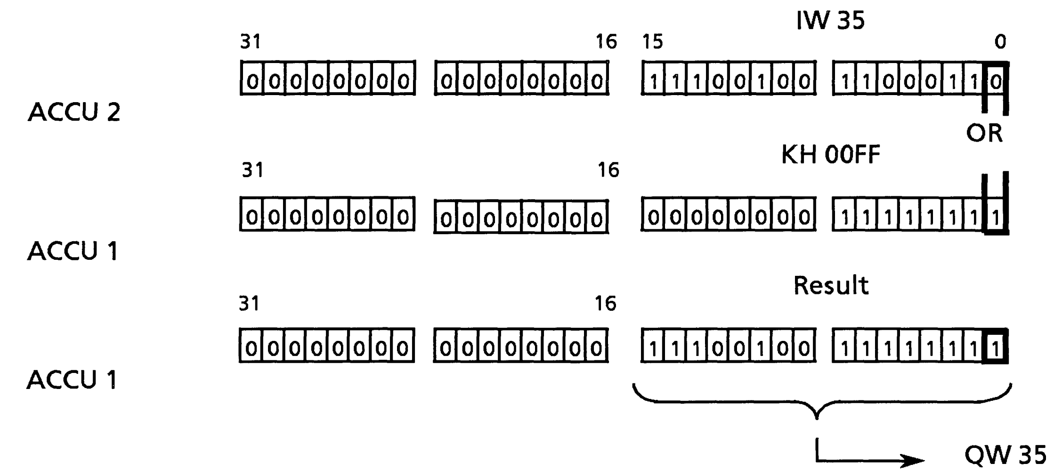
Set the 8 high-order bits in input word IW 92 to "0".
Compare both words bit by bit. If corresponding bits are both "1", the result bit is set to "1".



STL	Explanation
L IW 35	Load input word IW 35 into ACCU 1.
L KH 00FF	Load a constant into ACCU 1. The previous contents of ACCU 1 are shifted to ACCU 2.
OW	Combine the contents of both ACCU-L bit by bit through logic OR.
T QW 35	Transfer the result (contents of ACCU 1-L) to output word QW 35.

Numeric Example

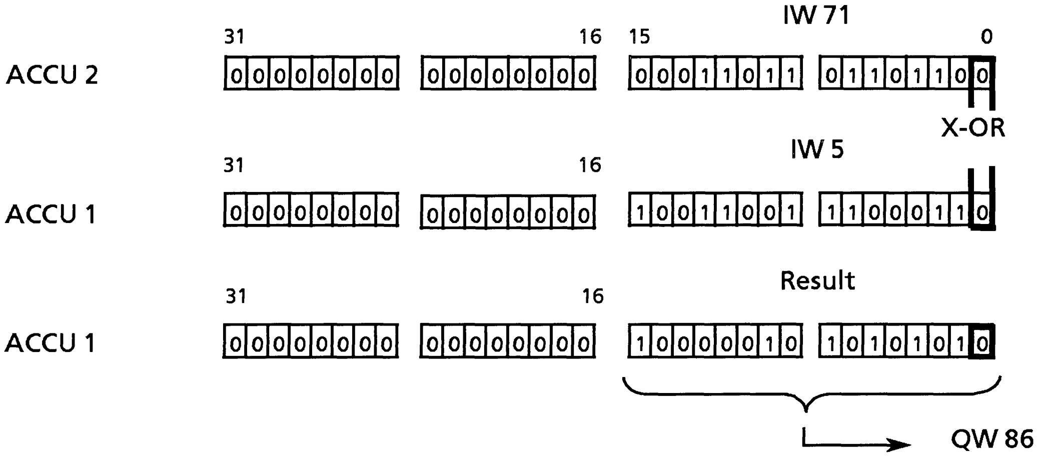
Set the 8 low-order bits in input word IW 35 to "1". Compare both words bit by bit. If corresponding bits are both "1", a "1" is set in the result word.



STL	Explanation
L IW 71	Load input word IW 71 into ACCU 1.
L IW 5	Load input word IW 5 into ACCU 1. The previous contents of ACCU 1 are shifted to ACCU 2.
XOW	Combine the contents of both accumulators bit by bit through EXCLUSIVE OR.
T QW 86	Transfer the result (contents of ACCU 1) to output word QW 86.

Numeric Example

Check to see if input words IW 71 and IW 5 are equal.
The result bit is set to "1" only if corresponding bits in ACCU 1 and ACCU 2 are unequal.



8.2.5 Shift Operations and Rotate Operations

These operations shift a bit pattern in ACCU 1. The contents of ACCU 2 are not affected. Table 8.19 provides an overview of these operations. Examples follow the table.

Table 8.19 Overview of Shift Operations

Operation	Operand		Meaning
SLW		□	**Shift to the left** The bit pattern (16 bits) in ACCU 1-L is shifted to the left. Positions becoming vacant are padded with zeros. The result is stored in ACCU 1-L. The contents of ACCU 1-H and ACCU 2 are not affected.
SRW		□	**Shift to the right** The bit pattern (16 bits) in ACCU 1-L is shifted to the right. Positions becoming vacant are padded with zeros. The result is stored in ACCU 1-L. The contents of ACCU 1-H and ACCU 2 are not affected.
SSW		□	**Shift to the right (with sign)** The bit pattern (16 bits) with sign in ACCU 1-L is shifted to the right. Positions becoming vacant are padded with the sign (bit 15). The result is stored in ACCU 1-L. The contents of ACCU 1-H and ACCU 2 are not affected.
		↑ Parameter	0 to 15
SLD		□	**Shift to the left** The bit pattern (32 bits) in ACCU 1 is shifted to the left. Positions becoming vacant are padded with zeros. The result is stored in ACCU 1. ACCU 2 is not affected.
SSD		□	**Shift to the right (with sign)** The bit pattern (32 bits) with sign in ACCU 1 is shifted to the right. Positions becoming vacant are padded with the sign (bit 31). The result is stored in ACCU 1. ACCU 2 is not affected.
RLD		□	**Rotate to the left (32 bits)** The bit pattern in ACCU 1 is rotated to the left. Positions becoming vacant are padded with the bits shifted. The result is stored in ACCU 1. ACCU 2 is not affected.
RRD		□	**Rotate to the right (32 bits)** The bit pattern in ACCU 1 is rotated to the right. Positions becoming vacant are padded with the bits shifted. The result is stored in ACCU 1. ACCU 2 is not affected.
		↑ Parameter	0 to 32

Processing a Shift Operation

Execution of shift operations is unconditional. The RLO is not affected.
However, shift operations set condition codes. CC 1 indicates the status of the last bit shifted out of the accumulator, CC 0 = "0". The condition codes can be scanned with jump functions.
The shift statement parameter indicates the number of bit positions by which the contents of ACCU 1 are to be shifted to the left (SLW) or to the right (SRW). Bit positions vacated during shifting are assigned zeros/sign.
The contents of the bits that are shifted are lost. Following execution of the operation, the status of the last bit shifted out of the accumulator is indicated in the CC 1 bit. This bit can be evaluated.
A shift operation with parameter "0" is handled like a "NOP" operation. The central processor processes the next STEP 5 statement with no further reaction.
Before executing a shift operation, load the operand to be processed into ACCU 1.
The altered operand is available there for further processing.

STL	Explanation
`L DW 2`	`Load the contents of data word DW 2 into ACCU 1.`
`SLW 3`	`Shift the bit pattern in ACCU 1 three positions to the left.`
`T DW 3`	`Transfer the result (contents of ACCU 1) to data word DW 3.`

Numeric Example

The value 464_{10} is stored in data word DW 2.
Multiply this value by $2^3=8$. Do so by shifting the bit pattern of DW 2 in ACCU 1 three positions to the left.

```
                                                  464₁₀      (DW 2)
          31                     16   15                              0
ACCU 1    0 0 0 0 0 0 0 0  0 0 0 0 0 0 0 0   0 0 0 0 0 0 0 1  1 1 0 1 0 0 0 0

                                                          <---- SLW 3

                                                  3712₁₀
          31                     16   15                              0
ACCU 1    0 0 0 0 0 0 0 0  0 0 0 0 0 0 0 0   0 0 0 0 1 1 1 0  1 0 0 0 0 0 0 0
```

STL	Explanation
`L   IW 128`	Load the value of input word IW 128 into ACCU 1.
`SRW 4`	Shift the bit pattern in ACCU 1-L four positions to the right.
`T   QW 160`	Transfer the result (contents of ACCU 1-L) to output word QW 160.

Numeric Example

The value 352_{10} is stored in IW 128.
Shift the corresponding bit pattern in ACCU 1 four positions to the right to divide the value 352_{10} by $2^4 = 16$ (without remainder).

352_{10} (IW 128)

	31	16	15	0
ACCU 1	00000000	00000000	00000001	01100000

SRW 4 →

22_{10}

	31	16	15	0
ACCU 1	00000000	00000000	00000000	00010110

STL	Explanation
`L   KF-64`	Load the fixed-point number -64 into ACCU 1.
`SSW 4`	Shift the bit pattern in ACCU 1-L four positions to the right.
`T   QW 10`	Transfer the result (contents of ACCU 1-L) to output word QW 10.

Numeric Example

The fixed-point number -64 is to be divided by 16 (with sign, without remainder).
Therefore, the bit pattern is shifted four positions to the right (SSW 4).

	31	16	15	0
ACCU 1	00000000	00000000	11111111	11000000

SSW 4 →

	31	16	15	0
ACCU 1	00000000	00000000	11111111	11111100

STL	Explanation
L DH FFFF FFCO	Load the hexadecimal number FFFF FFCO (≙-64) into ACCU 1.
SSD 3	Shift the bit pattern in ACCU 1 three positions to the right.
T QD 20	Transfer the result (contents of ACCU 1) to the output double word 20.

Numeric example

The fixed-point double word (-64) is to be divided by 8 (with sign, without remainder). Therefore, the bit pattern is shifted 3 positions to the right (SSD 3).

```
          31              16  15               0
ACCU 1    11111111 11111111   11111111 11000000

          SSD 3 ──▶

          31              16  15               0
ACCU 1    11111111 11111111   11111111 11111000
```

Processing a Rotate Operation

Execution of rotate operations is unconditional. The RLO is not affected. However, rotate operations set condition codes. CC1 indicates the status of the last bit shifted out of the accumulator, CC0 = "0". The condition codes can be scanned with jump functions.

The rotate statement parameter indicates the number of bit positions by which the contents of ACCU 1 are to be rotated to the left or right. Bit positions vacated during rotating are padded with the bits shifted.

Following execution of the operation, the status of the last bit shifted out of the accumulator is indicated in the CC1 bit. This bit can be evaluated.

Before executing a rotate operation, load the operand to be processed into ACCU 1.

The altered operand is available there for further processing.

STL	Explanation
L ID 12	Load the input double word 12 into ACCU 1.
RLD 5	Rotate the bit pattern in ACCU 1 five positions to the left.
T QD 16	Transfer the result (contents of ACCU 1) to the output double word 16.
Numeric example	
ACCU 1	10111011 10010000 11111010 00001111
	RLD 5
ACCU 1	11110010 00011111 01000001 11110110

8.2.6 Conversion Operations

Conversion operations convert the values in ACCU 1. ACCU 2 is unaffected. The individual operations are listed in Table 8.20 and explained in examples following the table. Execution of these operations does not depend on the RLO.
The effects on the condition codes and the behaviour in the case of non-convertibility are explained in Section 8.4.

Table 8.20 Overview of Conversion Operations

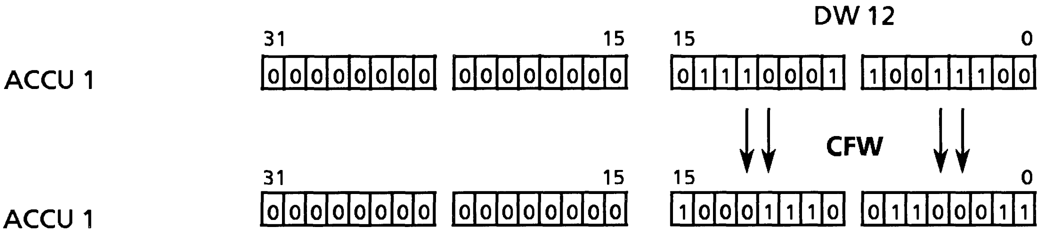
Operation	Operand	Meaning
CFW		**One's complement (16 bits)** The contents of ACCU 1 are inverted bit by bit. Result in ACCU 1-L, ACCU 1-H is unaffected.
CSW		**Two's complement (16 bits)** The contents of ACCU 1-L are inverted bit by bit. Afterwards the word 0001_H is added. Result in ACCU 1-L, ACCU 1-H is extended by the correct sign. 0000_H: result ≥ 0; $FFFF_H$: result < 0
CSW		**Two's complement (32 bits)** The contents of ACCU 1 are inverted bit by bit. Afterwards the double word 0000 0001_H is added. Result in ACCU 1.
DEF		**Fixed-point conversion (16 bits) BCD → binary** The contents of ACCU 1-L are interpreted as a three-digit BCD number with sign (see Section 7.6) and converted into a 16-bit fixed-point number. Result in ACCU 1-L, ACCU 1-H is unaffected.
DUF		**Fixed-point conversion (16 bits) binary → BCD** The contents of ACCU 1-L are interpreted as a 16-bit fixed-point number and converted into a three-digit BCD number with sign. Result in ACCU 1-L, ACCU 1-H is unaffected.
DED		**Double word conversion (32 bits) BCD → binary** The contents of ACCU 1 are interpreted as a seven-digit BCD number and converted into a 32-bit fixed-point number. Result in ACCU 1.
DUD		**Double word conversion (32 bits) binary → BCD** The contents of ACCU 1 are interpreted as a 32-bit fixed-point number and converted into a BCD number. Result in ACCU 1.
FDG		**Conversion of fixed-point number into floating-point number** The contents of ACCU 1 are interpreted as a 32-bit fixed-point number and converted into a floating-point number. Result in ACCU 1.
GFD		**Conversion of floating-point number into fixed-point number** The contents of ACCU 1 are interpreted as a floating-point number and converted into a 32-bit fixed-point number. Result in ACCU 1.

Example

STL	Explanation
L DW 12	Load the contents of data word DW 12 into ACCU 1.
CFW	Invert all bits in ACCU 1.
T QW 20	Transfer the new contents of ACCU 1 to output word QW 20.

Numeric Example

In a system, normally open contacts have been replaced by normally closed contacts. If the information in data word DW 12 is to maintain its previous effect, DW 12 must be inverted.

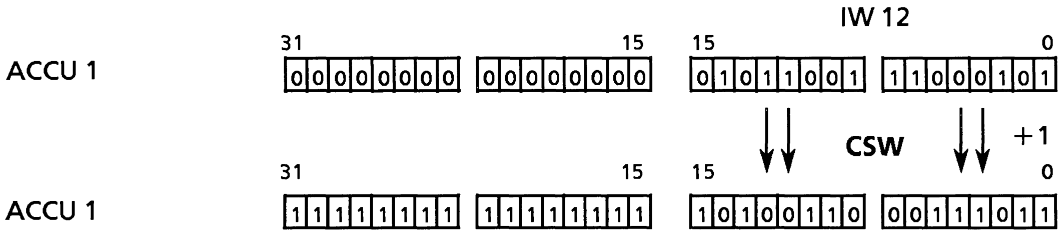


```
                                              DW 12
         31               15   15                      0
ACCU 1   00000000  00000000    01110001  10011100
                                   ↓↓   CFW   ↓↓
         31               15   15                      0
ACCU 1   00000000  00000000    10001110  01100011
```

STL	Explanation
L IW 12	Load the contents of input word IW 12 into ACCU 1.
CSW	Invert all bits. Add a "1" at the least significant position.
T DW 100	Transfer the altered word to data word DW 100.

Numeric Example

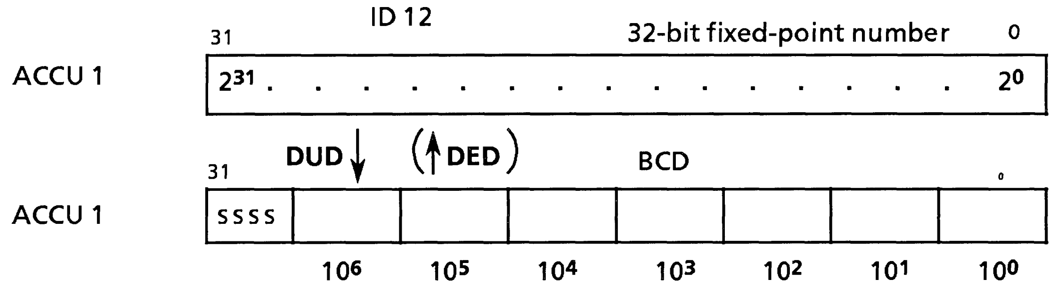
Form the negative value of the value in input word IW 12.



```
                                              IW 12
         31               15   15                      0
ACCU 1   00000000  00000000    01011001  11000101
                                   ↓↓   CSW   ↓↓  +1
         31               15   15                      0
ACCU 1   11111111  11111111    10100110  00111011
```

STL	Explanation
L ID 12	Load the contents of ID 12 into ACCU 1.
DUD (DED)	Conversion of accumulator contents; BINARY → BCD (BCD → BINARY)
SPS=FEHL	Evaluation of the latching overflow bit in case of non-convertibility
T DD 100	Transfer the altered word to DD 10.
BEU	
FEHL:	Error evaluation
:	
:	

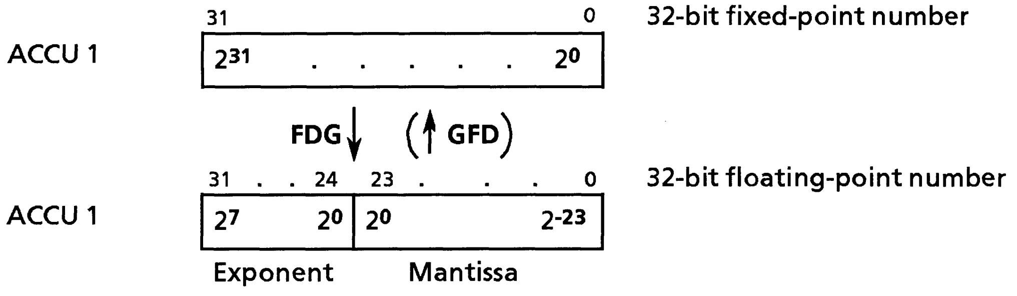
Conversion



S (sign): 0 = positive
1 = negative

STL	Explanation
L ID 12	Load the contents of ID 12 into ACCU 1.
FDG (GFD)	Conversion of accumulator contents; fixed-point number → floating-point number (floating-point number → fixed-point number)
SPS=FEHL	Evaluation of the latching overflow bit in case of non-convertibility
T DD 100	Transfer the altered accumulator contents to DD 100.
BEU	
FEHL:	Error evaluation
:	
:	

Conversion



Accuracy of GFD Conversion

Exact conversion is possible with integer floating-point numbers only.
Floating-point numbers with fractions of 1 are converted to the next-smaller integer value.
An exception are the floating-point numbers in the range of (-1 < G < 0).
The floating-point numbers of this range are converted to a fixed-point zero.
The floating-point number -1 is converted to the fixed-point number -1.
The absolute conversion error is < 1 (see Fig. 8.6).

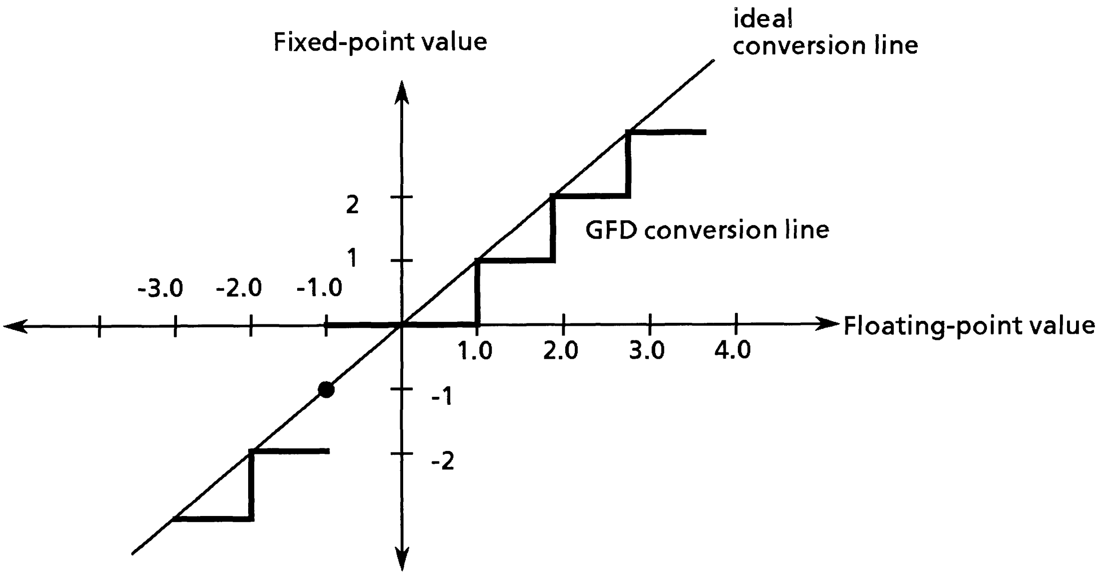


Figure 8.6 Conversion Characteristics of GFD Conversion

8.2.7 Decrement/Increment

The decrement/increment operations change the data loaded into ACCU 1-LL. Table 8.21 provides an overview of the decrement/increment operations. An example follows the table.

Table 8.21 Decrement/Increment Operations

Operation	Operand		Meaning	
D		□	**Decrement** Decrement ACCU 1-LL	The contents of ACCU 1-LL are either decremented or incremented by the number indicated in the parameter (without carry). Execution of the operation is unconditional. ACCU 1-H, ACCU 1-LH and ACCU 2 are unaffected.
I		□ ↑	**Increment** Increment ACCU 1-LL	
		Parameter 0 to 255		

Processing

Execution of the decrement and increment operations is independent of the RLO and does not affect the RLO or the condition codes.
The parameter indicates the value by which the contents of ACCU 1 are to be changed.
The operations refer to decimal values; however, the result is stored in ACCU 1 in binary form.
Changes relate only to ACCU 1-LL.

Example	STL	Explanation
Increment the hexadecimal constant 1010_H by 16 and store the result in data word DW 8.	C DB 6	Call data block DB6.
	L KH 1010	Load hexadecimal constant 1010_H into ACCU 1-L.
	I 16	Increment the low byte of ACCU 1 by 16. The result, 1020_H, is located in ACCU 1.
In addition, decrement the incrementation result by 33 and store the new result in data word DW 9.	T DW 8	Transfer the contents of ACCU 1 (1020_H) to data word DW 8. Since the incrementation result is still in ACCU 1, you can decrement by 33 directly. The result would be FFF_H.
	D 33	However, since the high byte of ACCU 1-L is not decremented along with the low byte, the result in ACCU 1 is $10FF_H$. The contents of ACCU 1-L are
	T DW 9	transferred to DW 9 ($10FF_H$).

8.2.8 Disable/Enable Interrupt

Global disabling/enabling of interrupts

The disable/enable interrupt operations affect interrupt and time-controlled program execution. They prevent process or time interrupts from interfering with the processing of a sequence of statements or blocks. Table 8.22 lists the disable/enable interrupt operations. An example follows the table.

Table 8.22 Disable/Enable Interrupt Operations

Operation	Operand		Meaning
IA			**Disable interrupt**
RA			**Enable interrupt**
			IA/RA applies for OBs 2, 3, 4, 5, 6, 10, 11, 12, 13. IA/RA does ***not*** apply for OBs 26, 33, 35. (System error level)

Processing

Execution of the disable/enable interrupt operations does not depend on the RLO. These operations do not affect the RLO or the condition codes. After the "IA" statement is processed, no more interrupts are executed. The "RA" statement cancels the effect of "IA".

Example	STL	Explanation
Disable interrupt processing in a specific program section and then enable it again.	.	
	.	
	.	
	.	
	.	
	`= Q 7.5`	
	`IA`	Disable interrupt.
	`A I 2.3`	
	.	
	.	
	.	
	`JU FB 3`	If an interrupt occurs, the program section between the operations IA and RA is scanned without interruption (unless "RA" has been programmed in FB3).
	.	
	.	
	.	
	`RA`	Enable interrupt. Interrupts that occurred in the meantime are processed after the "RA" operation.[1]
	.	
	.	
	.	

[1] Only one interrupt can be stored for each interrupt line.

Selective disabling/enabling of interrupts

The SIM operation sets the interrupt mask. With this interrupt mask, each interrupt can be disabled/enabled individually.
The LIM operation is used to read the current interrupt mask. Execution of these operations does not depend on the RLO. These operations do not affect the RLO and the condition codes.

Table 8.23 Selective Disabling/Enabling of Interrupts

Operation	Operand	Meaning
SIM		**Set interrupt mask** (32 bits) The system error level and the time and process interrupt OBs are selectively disabled/enabled. Time and process interrupts (OBs 2 to 6, 10 to 13) enabled in the interrupt mask are not processed, if the interrupts have been disabled by the IA operation. After interrupt enabling with RA, the time and process interrupts enabled in the interrupt mask are processed. IA/RA does not affect processing of the system error OBs (OBs 26, 33, 35). OBs 26, 33, 35 can be enabled/ disabled only by setting/resetting of bit 0 in the interrupt mask.
LIM		**Read interrupt mask** (32 bits) The current interrupt mask can be read in ACCU 1.

Structure of the interrupt mask

In the interrupt mask, each time and process interrupt OB and each system error OB is assgined one bit (see Fig. 8.7).
If a bit is set, the corresponding OB is enabled.
If a bit is not set, the OB is disabled.
Bits 0 to 9 are defaulted with "1", i.e. the interrupts are enabled.

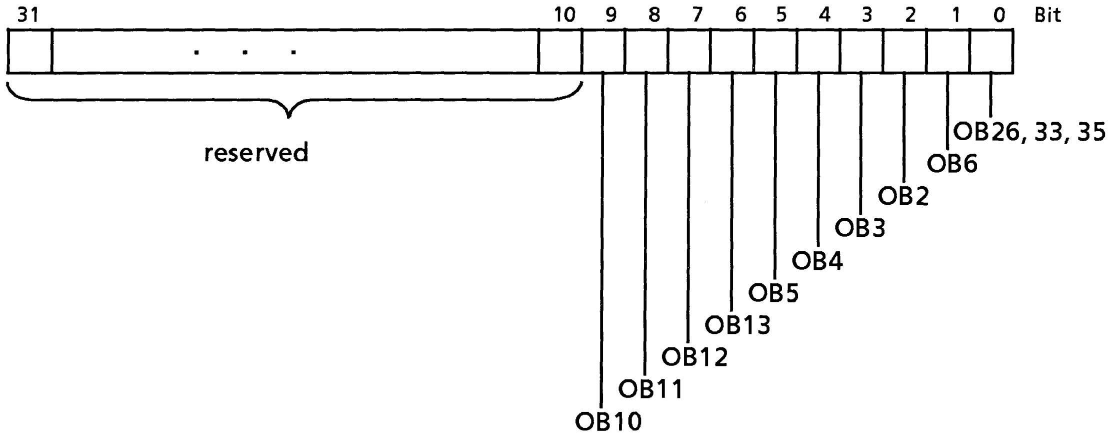


Figure 8.7 Structure of the Interrupt Mask

Example	STL	Explanation
Processing of OB 4 and OB 13 is to be disabled. Otherwise, the interrupt mask should remain unaffected.	LIM	Load the interrupt mask into ACCU 1
	L KM 11111111 10101111	Bit pattern with marking of OB4 and OB13 bits
	AW	
	SIM	Set interrupt mask.
	:	

Example	STL	Explanation
System error processing, OB2 processing and OB13 processing are to be enabled. Otherwise, the interrupt mask should remain unaffected.	LIM	Load the interrupt mask into ACCU 1
	L KM 00000000 01000101	Load bit pattern with bits set for system error OBs, OB2 and OB13
	OW	
	SIM	Set interrupt mask.
	:	

8.2.9 Processing Operation

The processing operation (DO) can handle STEP 5 statements in "indexed" form. Use it to change the parameter of an operand while the control program is being scanned. Table 8.24 and the example that follows explain the processing operation.

Table 8.24 Processing Operations

Operation	Operand		Meaning
DO	□	□	**Process a flag or data word.**
ID	↑ FW DW	↑ **Parameter** 0 to 254 0 to 255	

Processing

The statement "Process flag or data word x" is a two-word statement that is executed independently of the RLO.

It consists of the following two related statements:

- The first statement contains the processing operation and specifies a flag or data word.
- The second statement specifies the operation and the operand ID that the control program is to process. Enter 0 or 0.0 here as parameter.

Note

If a value other than 0 or 0.0 is specified in the CPU 944, this value is replaced by 0 or 0.0.

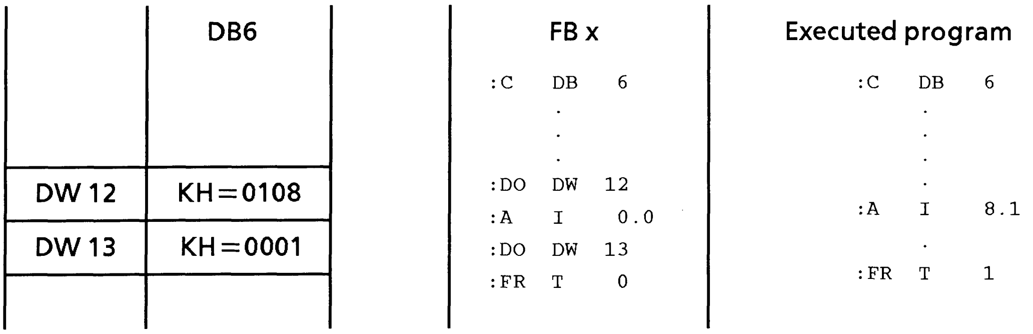
The control program works with the parameter that is stored in the flag or data word called by the first statement. If you are indexing binary operations, inputs, outputs, or flags, indicate the bit address in the high byte of this word and the byte address in the low byte.
In all other cases, the high byte must be "0".

The following operations can be combined with the processing statement:

Operations	Explanation
A, AN, O, ON* (with operands I, Q, T, C, F)	Binary logic operations
S, R, = (with operands I, Q, F)	Set/reset operations
FR T, RT, SF T, SD T, SI T, SS T, SE T	Timer operations
FR C, RC, SC, CD C, CU C	Counter operations
L IB, IW, ID, QB, QW, QD, FY, FW, FD, PY, PW, OY, OW, DL, DR, DW, DD, T, C, RS, RT	Load operations
LC T, C	Load operations in BCD code
T IB, IW, ID, QB, QW, QD, FY, FW, FD, PY, PW, OY, OW, DL, DR, DW, DD, RS, RT	Transfer operations
JU=, JC=, JZ=, JN=, JP=, JM=, JO=	Jump operations
SLW, SRW, SLD, RLD, SSD, RRD	Shift operations
D, I, ADD BN	Decrement and increment
C DB, JU, JC	Block calls
TNB, TNW	Block transfer

* The A I, AN I, O I, ON I operations become the A Q, AN Q, O Q, ON Q operations in combination with DO FW or DO DW if the byte address in the data or flag word is greater than 127.

Figure 8.8 shows how data word contents specify the parameter of the next statement.



	DB6
DW 12	KH=0108
DW 13	KH=0001

FB x

```
:C   DB   6
      .
      .
      .
:DO  DW  12
:A   I    0.0
:DO  DW  13
:FR  T    0
```

Executed program

```
:C   DB   6
      .
      .
      .
      .
:A   I    8.1
      .
:FR  T    1
```

Figure 8.8 Effect of the Processing Operation

The following example shows how new parameters are generated each time the program is scanned.

Example	STL	Explanation
Set the contents of data words DW 20 to DW 100 to signal state "0". The "index register" for the parameter of the data words is DW 1.	:C DB 202	Call data block 202.
	:L KB 20	Load constant 20 into ACCU 1.
	:T DW 1	Transfer contents of ACCU 1 to DW 1.
	L1 :L KH 0	Load hexadecimal constant 0 into ACCU 1.
	:DO DW 1	Process data word 1.
	:T DW 0	Transfer the contents of ACCU 1 to the data word whose address is stored in data word DW 1.
	:L DW 1	Load data word DW 1 into ACCU 1.
	:L KB 1	Load constant 1 into ACCU 1. Data word DW 1 is shifted to ACCU 2.
	:+F	ACCU 2 and ACCU 1 are added and the result is stored in ACCU 1 (incrementing the data word address).
	:T DW 1	Transfer the contents of ACCU 1 to data word DW 1 (new data word address).
	:L KB 100	The constant 100 is loaded into ACCU 1 and the new data word address is shifted to ACCU 2.
	:<=F	The accumulators are compared to see if ACCU 2-L ACCU 1-L.
	:JC =L1	Jump conditionally to label L1 if ACCU 2-L ACCU 1-L.

8.2.10 Jump Operations

Table 8.25 provides an overview of the jump operations. An example follows the table.

Table 8.25 Overview of Jump Operations

Operation	Operand		Meaning
JU =	□		**Jump unconditionally** The unconditional jump is executed independently of conditions.
JC =	□		**Jump conditionally** The conditional jump is executed if the RLO is "1". If the RLO is "0", the statement is not executed and the RLO is set to "1".
JZ =	□		**Jump if the result is "zero"** The jump is executed only if CC 1=0 and CC 0=0 The RLO is not changed.
JN =	□		**Jump if the result is "not zero"** The jump is executed only if CC 1≠CC 0 The RLO is not changed.
JP =	□		**Jump if the result is positive** The jump is executed only if CC 1=1 and CC 0=0 The RLO is not changed.
JM =	□		**Jump if the result is negative** The jump is executed only if CC 1=0 and CC 0=1 The RLO is not changed
JO =	□		**Jump on overflow (OV=1)** The jump is executed if an overflow occurs and the OV bit is set to 1. Otherwise the jump is not executed. The RLO is not changed.
JOS =	□		**Jump on overflow/latching (OS=1)** The jump is executed if an overflow occurs and the OS bit is set to 1. The RLO ist not changed.
ID Jump label (up to 4 characters)			
JUR			**Jump relative** Linear program processing is interrupted and continued at the position which has been determined by the jump displacement.
Jump displacement in words: - 32768 to +32767			

Processing the Jump Operations

A symbolic jump destination (jump label) must always be entered next to a jump operation. This jump label can have up to four characters, the first of which must be a letter of the alphabet.

When programming, please be aware of the following:

- The absolute jump displacement cannot exceed + 127 or - 128 words in the program memory. Some statements take up two words (e.g., "Load a constant"). For long jumps, insert an intermediate destination.
- Jumps can be executed only within a block.
- Jumping over segment boundaries ("BLD 255") is not permitted.

Example	STL	Explanation
If no bit of input word IW 1 is set, program execution jumps to the label "AN 1". If input word IW 1 and output word QW 3 do not agree, program processing jumps back to the label "AN 0". Otherwise input word IW 1 and data word DW 12 are compared. If input word IW 1 is greater than or less than data word DW 12, program execution jumps to the "Destination" label.	AN0 :L IW 1 :L KH 0000 :+F **:JZ =AN1** :A I 1.0	Load input word IW 1 into ACCU 1. If the contents of ACCU 1 equal zero[1], jump to the label "AN 1". Otherwise process the next statement (A I 1.0).
	AN1 :L IW 1 :L QW 3 :XOW	Compare input word IW 1 and output word QW 3. If they are not equal, set individual bits in ACCU 1.
	:JN =AN0	If the contents of ACCU 1 are not zero, jump to the label "AN 0". Otherwise process the next statements.
	:L IW 1 :L DW 12 :><F	Compare input word IW 1 and data word DW 12. If they are not equal, set RLO to "1".
	:JB = Destination	If the RLO="1", jump to the "Destination" label. If the RLO="0", process the next statement.
	. . . Dest. :A I 12.2 . .	

[1] The "L..." statement does not affect the condition codes. An addition (+F) is executed with the constant 0000$_H$ so that the "JZ" operation can evaluate the contents of the accumulator.

Processing the "JUR" Operation

Execution of the "JUR" operation is independent of the RLO.
The parameter specifies the jump displacement directly. For example, parameter "1" means that processing will continue with the next one-word statement. Parameter "2" means processing will continue with the one-word statement directly after the next one-word statement.

Such labeling includes the following special features:

The jump displacement is not corrected automatically. If changes are made in the Section of the program that is jumped over, the jump destination can be displaced.

Caution

Avoid jumps over block boundaries since you have no control over the absolute location of blocks in the internal user memory.

8.2.11 Substitution Operations

If you plan to process a program with various operands, i. e. independent of the current operand, it is advisable to assign parameters to individual operands (see Section 7.4.4). Substitution operations are possible in function blocks only. If you have to change the operands, you only need to reassign the parameters in the function block call.
These parameters are processed in the program as "formal operands".
Special operations are necessary for this processing. However, these special operations are no different in their effect than operations without substitution. A brief description of these operations and examples follow.

Binary Logic Operations

Table 8.26 provides an overview of binary logic operations.

Table 8.26 Overview of Binary Logic Operations

Operation	Operand	Meaning
A =	□	**AND operation, scan for signal status "1"** The scan result is "1", if the corresponding actual operand is in signal status "1". Otherwise, the scan result is "0". This result is combined with the RLO in the processor through logic AND. [1]
O =	□	**OR operation, scan for signal status "1"** The scan result is "1", if the corresponding actual operand is in signal status "1". Otherwise, the scan result is "0". This result is combined with the RLO in the processor through logic OR. [1]
AN =	□	**AND operation, scan for signal status "0"** The scan result is "1", if the corresponding actual operand is in signal status "0". Otherwise, the scan result is "0". This result is combined with the RLO in the processor through logic AND. [1]
ON =	□	**OR operation, scan for signal status "0"** The scan result is "1", if the corresponding actual operand is in signal status "0". Otherwise, the scan result is "0". This result is combined with the RLO in the processor through logic OR. [1]

	Actual Operands Permitted	Parameter	
		Type	Type
Formal operand	Inputs, outputs, and flags addressed in binary form Timers and counters	I, Q T, C	BI

[1] If the scan follows immediately after an operation which reloads the RLO (first input bit scan), the scan result is taken to form the new RLO.

Set/Reset Operations

Table 8.27 provides an overview of the set/reset operations.

Table 8.27 Overview of Set/Reset Operations

Operation	Operand	Meaning
S =	□	**Set a formal operand (binary).** First program execution with RLO = "1" assigns signal status "1" to the actual operand addressed. Changes of the RLO do not affect this status.
RB =	□	**Reset a formal operand (binary).** First program execution with RLO = "1" assigns signal status "1" to the actual operand addressed. A change of the RLO does not affect this status.
= =	□	**Assign** Each program execution assigns the current RLO to the format operand addressed.

	Actual Operands Permitted	Parameter	
		Type	Type
Formal operand	Inputs, outputs, and flags addressed in binary form	I, Q	BI

Example: FB30 is assigned parameters in OB1.

Call in OB1	Program in FB30	Executed Program
:JU FB 30	:A =ON1	:A I 2.0
NAME :LOGIC	:AN =ON2	:AN I 2.1
ON1 : I 2.0	:O =ON3	:O I 2.2
ON2 : I 2.1	:S =MOT5	:S Q 7.3
ON3 : I 2.2	:= =OFF1	:= Q 7.1
VAL1 : I 2.3	:A =VAL1	:A I 2.3
OFF1 : Q 7.1	:A =ON2	:A I 2.1
OFF2 : Q 7.2	:ON =ON3	:ON I 2.2
MOT5 : Q 7.3	:RB =MOT5	:R Q 7.3
:BE	:= =OFF2	:= Q 7.2
	:BE	:BE

Load and Transfer Operations

Table 8.28 provides an overview of the load and transfer operations. An example follows the table.

Table 8.28 Overview of Load and Transfer Operations

Operation	Operand	Meaning
L =	□	Load a formal operand.
LC =	□	Load a formal operand in BCD code.
LW =	□	Load the bit pattern of a formal operand.
LD =	□	Load the bit pattern of a formel operand.
T =	□	Transfer to a formal operand.

Formal operand	Actual Operands Permitted	Parameter	
		Type	
for L =	FY, FW, FD IB, IW, ID, QB, QW, QD, PY, PW, DL, DR, DW, DD, RS, RT, C, T	I, Q T, C	BY, W, D
for LC =	Timers and counters	T, C	
for LW =	Bit pattern	D	KF, KH, KM, KY, KS, KT, KC
for LD =	Bit pattern (32 bits)	D	KG
for T =	FD, FY, FW, IB, IW, ID QB, QW, QD, PY, PW, DL, DR, DW, DD, RS, RT	I, Q	BY, W, D

Example: FB34 is assigned parameters in PB1.

Call in PB1	Program in FB34	Executed Program
	:A =I0	:A I 2.0
:JU FB 34	**:L =L1**	:L FW 10
NAME :LOAD/TRAN	:S C 6	:S C 6
I0 : I 2.0	:A =I1	:A I 2.1
I1 : I 2.1	**:LW =LW1**	:L KC 140
L1 : FW 10	:S C 7	:S C 7
LW1 : KC 140	:A I 2.2	:A I 2.2
LC1 : C 7	:CU C 6	:CU C 6
T1 : QW 4	:CU C 7	:CU C 7
LW2 : KC 160	**:LC =LC1**	:LC C 7
:BE	**:T =T1**	:T QW 4
	:A I 2.7	:A I 2.7
	:R C 6	:R C 6
	:R C 7	:R C 7
	:LW =LW2	:L KC 160
	:LC =LC1	:LC C 7
	:!=F	:!=F
	:R C 7	:R C 7
	:BE	:BE

Timer and Counter Operations

Table 8.29 provides an overview of timer and counter operations. Examples follow the table (also refer to Sections 8.1.4 and 8.1.5).

Table 8.29 Overview of Timer and Counter Operations

Operation	Operand	Meaning
FR =	□	**Enable** a formal operand for cold restart. (For a description, see "FT" or "FC", according to the formal operand).
RD =	□	**Reset** a formal operand (digital).
SP =	□	**Start** a pulse timer specified as a formal operand using the value stored in the accumulator.
SD =	□	**Start** an on-delay timer specified as a formal operand using the value stored in the accumulator.
SEC =	□	**Start** an extended pulse timer specified as a formal operand using the value stored in the accumulator or **set** a counter specified as a formal operand using the count specified in the accumulator.
SSU =	□	**Start** a stored on-delay timer specified as a formal operand using the value stored in the accumulator or start the **count up** of a counter specified as a formal operand.
SFD =	□	**Start** an off-delay timer specified as a formal operand using the value stored in the accumulator or start the **count down** of a counter specified as a formal operand.

Formal operand	Actual Operands Permitted	Parameter Type	Type
	Timers and counters[1]	T, C[1]	

[1] "SP" and "SD" do not apply to counters

Specifying Times and Counts

As with the basic operations, you can specify a time or count as a formal operand. In this case, you must distinguish as follows whether the value is located in an operand word or is specified as a constant.

- Operand words can be of parameter type "I" or "Q" and of data type "W". Use the "L=" operation to load them into the accumulator.
- Constants can be of parameter type "D" and of data type "KT" or "KC". Use "LW=" to load these formal operands into the accumulator.

The following examples show how to work with timer and counter operations.

Example 1:

Function Block Call	Program in Function Block (FB32)	Executed Program
	:AN =I5	:AN I 2.5
:JU FB 32	:A =I6	:A I 2.6
NAME :TIME	:L KT 5.2	:L KT 5.2
I5 : I 2.5	**:SFD =TIM5**	:SF T 5
I6 : I 2.6	:A =I5	:A I 2.5
TIM5 :T 5	:AN =I6	:AN I 2.6
TIM6 :T 6	:L KT 5.2	:L KT 5.2
OUT6 : Q 7.6	**:SSU =TIM6**	:SS T 6
:BE	:A =TIM5	:A T 5
	:O =TIM6	:O T 6
	:= =OUT6	:= Q 7.6
	:A I2.7	:A I 2.7
	:RD =TIM5	:R T 5
	:RD =TIM6	:R T 6
	:BE	:BE

Example 2:

Function Block Call	Program in Function Block (FB33)	Executed Program
STL	:A =I2	:A I 2.2
:JU FB 33	:L KC 17	:L KC 17
NAME :COUNT	**:SEC =COU5**	:S C 5
I2 : I 2.2	:A =I3	:A I 2.3
I3 : I 2.3	**:SSU =COU5**	:CU C 5
I4 : I 2.4	:A =I4	:A I 2.4
COU5 : C 5	**:SFD =COU5**	:CD C 5
OUT3 : Q 7.3	:A =COU5	:A C 5
:BE	:= =OUT3	:= Q 7.3
	:A I 2.7	:A I 2.7
	:RD =COU5	:R C 5
	:BE	:BE

Processing Operation

Table 8.30 and the example that follows explain the processing operation.

Table 8.30 Processing Operation

Operation	Operand	Meaning		
DO =	□	**Process formal operand** The substituted blocks are called unconditionally.		
Formal operands		Actual Operands Permitted	Parameter TYPE	 TypR
		DB, PB, SB, FB^1, OB	B	

[1] As actual operands, function blocks cannot have block parameters

Example:

Function Block Call	Program in Function Block FB35	Executed Program
STL		
:JU FB 35	:DO =D5	:C DB 5
NAME :PROCES	:L =DW2	:L DW 2
D5 : DB 5	:DO =D6	:C DB 6
DW2 : DW 2	:T =DW1	:T DW 1
D6 : DB 6	:T =Q4	:T QW 4
DW1 : DW 1	:DO =MOT5	:JU FB 36
Q4 : QW 4	:BE	:BE
MOT5 : FB 36		
:BE		

8.3 System Operations

System operations and supplementary operations have the same limitations.

You can program them only as follows:

- in function blocks
- in the STL method of representation

System operations involve absolute addressing and therefore require exact knowledge

- of the hardware
- of the CPU address space
- of the system data area, etc.

Only experienced programmers should use them.

8.3.1 Load and Transfer Operations

Use these load and transfer operations to address the entire program memory of the CPU. They are used mainly for data exchange between the accumulator and memory locations that cannot be addressed by operands. Table 8.31 provides an overview of the load and transfer operations.

Caution

The TIR, TDI, TNB, TNW, TRW, TRD operations are memory changing operations with which you can accesss the program memory of the CPU. These accesses are not monitored by the operating system. Improper use of the operations can lead to changes in the program and to a CPU crash or dangerous statuses.

Table 8.31 Overview of Load and Transfer Operations

Operation	Operand		Meaning
LIR		□	**Load the register indirectly** Load the contents of a 16-bit wide memory location (addressed by ACCU 1) into the specified 16-bit register.
TIR		□	**Transfer the register indireclty** Transfer the contents of a 16-bit wide memory location (addressed by ACCU 1) to the specified 16-bit register.
		Parameter (register number) 0 ACCU 1-L 1 ACCU 1-H 2 ACCU 2-L 3 ACCU 2-H 8 DBL	
LDI	□		**Load the register indirectly** Load the specified 32-bit register with the contents of a double word, which is addressed by ACCU 1.
TDI	□		**Transfer the register indirectly** Load the contents of the specified 32-bit register into a double word, which is addressed by ACCU 1.
Code (register) A1 ACCU 1 A2 ACCU 2 SA* SAC=Step address counter BR Base address register			
TNW		□	**Transfer a data block (word by word)** Transfer a memory area block by block into another memory area. End address destination area: ACCU 1 End address source area: ACCU 2 Use even addresses (20 bits wide) only. After the execution of the operation, the accumulators are decremented by 2 x the number of words
		Parameter (Number of words to be transferred) 0 to 255	

* TDI SA: The absolute address of the operation following the TDI operation is transferred to the double word addressed by ACCU 1.

Note

Operations LIR 0 to LIR 3 change only the specified 16 bits of the ACCUs. The othe 16 bits remain unchanged.

Table 8.31 Overview of Load and Transfer Operations (continued)

Operation	Operand		Meaning
TNB		□ ↑	**Transfer a data block (byte by byte)** Transfer a memory area block by block into another memory area. End address destination area (20 bits wide): ACCU 1 End address source area (20 bits wide): ACCU 2 After the execution of the operation, the accumulators are decremented by the number of bytes
		Parameter (Number of bytes to be transferred) 0 to 255	
T	□	□	**Transfer** Transfer a word into the system data area
L	□ ↑	□ ↑	**Load** Load a word from the system data area into ACCU 1-L.
ID	RS RT	**Parameter** 0 to 255 0 to 255	
MBR		□ ↑	**Load the BR register** The base address register is loaded with a 20-bit constant.
		Parameter 0 to F FFFF$_H$	
ABR		□	**Add offset to the BR register** The offset is multiplied by 2 (word offset) and the result added to the contents of the BR register. The result is available in the BR register.
LRW		□	**Load the word addressed by the BR register + offset** The offset is multiplied by 2 (word offset) and the result added to the contents of the BR register. The word with this address is loaded into ACCU 1-L. The BR register is not changed. The original ACCU 1 value is available in ACCU 2.
LRD		□ ↑	**Load the double word addressed by the BR register + offset** The specified offset is multiplied by 2 (word offset) and the result added to the contents of the BR register. The double word with this address is loaded into ACCU 1. The BR register is not changed. The original ACCU 1 value is available in ACCU 2.
ID		**Parameter (offset)** -32768 to 32767	

Table 8.31 Overview of Load and Transfer Operations (continued)

Operation	Operand		Meaning
TRW		□	**Transfer ACCU 1-L into the word addressed by the BR register + offset** Multiply the offset specified by 2 and add the result to the contents of the BR register and transfer the contents of ACCU 1-L to the word with this address. The BR register is unchanged.
TRD		□	**Transfer ACCU 1 into the double word addressed by the BR register + offset** Mulitply the offset specified by 2 and add the result to the contents of the BR register and transfer the contents of ACCU 1 to the double word with this address. The BR register is unchanged.
		↑ **Parameter (offset)**	- 32768 to +32767
MAS			**Transfer the contents of ACCU 1 into SAC** Transfer the contents of ACCU 1 (bits 31 to 20 must be zero) into the step address counter (SAC).
MAB			**Transfer the contents of ACCU 1 into BR** Transfer the contents of ACCU 1 (bits 31 to 20 must be zero) into the base address counter (SAC).
MSA			**Transfer the contents of SAC into ACCU 1** Transfer the contents of the step address counter into ACCU 1 (bits 31 to 20 are set to zero) (address of next command).
MSB			**Transfer the contents of SAC into BR** Transfer the contents of the step address counter into the base address register (address of next command).
MBA*			**Transfer the contents of the BR into ACCU 1** Transfer the contents of the base address register into ACCU 1 (bits 31 to 20 are set to zero).
MBS			**Transfer the contents of the BR into the SAC** Transfer the contents of the base address register into the step address counter.

* If the DBS register is set to an address between 020000_H to $0205FF_H$ and an "MBA" command is executed, data may be falsified in this range. In order to avoid this effect, use the command sequence "ABR+0, MBA" instead of the "MBA" command.

⚠ Caution

Execution of the MAS, MBS operations is continued with the address stored in the accumulator or in the BR register.

Loading and Transferring Register Contents

STL	Explanation
. .	
L DH 0008 0010	Load the address 8 0010_H into ACCU 1.
LIR 0	Load the information from the memory location with the address 8 0010_H and 8 0011_H into ACCU 1-L.

Example: Load the contents of the double word with the absolute address E $10CE_H$ into ACCU 2. The area E 10CE to E 10D1 corresponds to RS 103 and RS 104.

Contents of memory area: 0008 1122

STL	Explanation
L DH 000E 10CE	Load the constant E $10CE_H$ into ACCU 1.
LDI A2	After this operation, ACCU 2 contains the contents of memory locations E $10CE_H$ to E $10D1_H$, i.e. 0008 1122_H.

Example: Transferring the values 2244_H and $AACD_H$ to memory locations E $10F4_H$ and E $10F7_H$ **(corresponds to RS 122, 123).**

STL	Explanation
L DH 2244AACD	Load the constant 2244 $AACD_H$ into ACCU 1.
L DH 000E10F4	After this operation, ACCU 1 contains 000E 10F4 and ACCU 2 contains 2244 AACD.
TDI A2	After the transfer operation, memory locations E 10F4 to E 10F7 contain the value 2244 $AACD_H$.

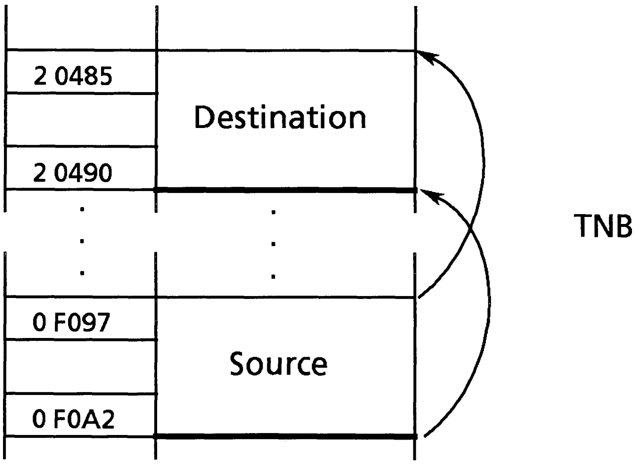
Processing a Field Transfer

A field transfer is processed independently of the RLO.
The parameter indicates the length of the data field (in bytes) that is to be transferred.

The maximum field length is:

- For TNB 255 bytes
- For TNW 255 words.

The end address of the source field is in ACCU 2. The end address of the destination field is in ACCU 1.
The highest address of each field must be specified. The bytes in the destination field are overwritten during the transfer.

Example	Representation
Transfer a 12-byte data field from address 0 F097 to 0 F0A2 to address area 2 0485 to 2 0490.	 2 0485 Destination 2 0490 0 F097 Source 0 F0A2 TNB
STL	**Explanation**
:L DH 0000 F0A2	Load the upper address of the source field into ACCU 1
:L DH 0002 0490	Load the end address of the destination field into ACCU 1. The source address is shifted to ACCU 2.
:TNB 12	Transfer the data field to the destination field. Contents of the accumulators after TNB: ACCU 1: 2 0484 ACCU 2: 0 F096

Transferring to the system data area

Example: Set the scan monitoring time to 1000 msec. after each mode change from STOP to RUN. Program the time as a multiple of ten in system data word 96. The following function block can be called from OB21, for example.

STL	Explanation
`FB 11`	Type and number of block.
`L   KF +100`	Load ACCU 1 with the factor 100.
`T   BS  96`	Transfer this value to system data word 96.
`BE`	

Caution

The TIR, TDI, TBS und TNB, TNW, TRW, TRD operations are memory changing operations with which you can access the user memory and the system data area. These accesses are not monitored by the operating system. Improper use of the operations can lead to changes in the program and to a CPU crash.
The MAS, MBS operations continue processing at another point in the program.

8.3.2 Arithmetic Operations

These operations add the value specified to the contents of ACCU 1. This value is represented by the parameter as a positive or negative decimal number (refer to Table 8.32).

Table 8.32 Arithmetic Operations

Operation	Operand		Meaning
ADD	□	□	**Add a constant** Byte or word constants or double words (as hexadecimal number) can be added.
ID	↑	↑ **Parameter**	
	BN		- 128 to+ 127
	KF		- 32768 to+ 32767
	DH		0 to FFFF FFFF

Processing:

The operation is executed independent of the RLO. Neither the RLO nor the condition codes are affected.
Subtractions can be executed by entering negative parameters.
With ADD BN there is no carry to ACCU 1-H.
With ADD KF there is no carry to ACCU 1-H.
With ADD DH there is no carry to ACCU 2.

Example	STL	Explanation
Decrement the constant 1020_H by 33 and store the result in flag word FW 28. Afterwards add the constant 256 to the result and store the sum in flag word FW 30.	`L   KH 1020`	The constant 1020_H is loaded into ACCU 1.
	`ADD BN -33`	The constant -33_{10} is added to the ACCU contents.
	`T   FW  28`	The new ACCU contents are 0000 $0FFF_H$. ACCU 1-L is stored in flag word FW 28.
	`ADD KF +256`	The constant 256_{10} is added to the last result.
	`T   FW  30`	The new ACCU contents are 0000 $10FF_H$. AKKU 1-L is stored in flag word FW 30.

8.3.3 Other Operations

Table 8.33 provides an overview of the remaining system operations.

Table 8.33 System Operations

Operation	Operand	Meaning
TAK		**Swap accumulator contents** Swap the contents of ACCU 1 and ACCU 2. The RLO and the condition codes are not affected.
STS		**Stop immediately** The CPU goes into STOP immediately.
STP		**Stop at the end of program execution** Current program processing is terminated; the PIQ is output. Then the PLC goes into STOP.

Processing of STS operation:

When executing the STS operation, the CPU goes into STOP immediately. Program processing is interrupted at this position. The STOP status can be exited only manually (mode selector switch) or with the programmer function "PLC start".

Processing of STP operation:

The STP operation is used to bring the PLC into the STOP status at the cycle checkpoint. This may be necessary in the case of time-critical plant statuses or when a device error occurs.
After execution of the operation, the control program is executed up to its end. Afterwards, the PLC goes into STOP with the "STS" error code. Restart is possible via the mode selector switch (STOP → RUN) or with the programmer.

8.4 Condition Code Generation

The processor of the S5-115U programmable controller has the following four condition codes:

- CC 0
- CC 1
- OS (latching overflow)
- OV (overflow)

The following operations affect the condition codes:

- comparison operations
- arithmetic operations
- shift operations
- some conversion operations.

The OS bit is reset by transfer operations.
The condition codes can be evaluated with conditional jump operations.

Condition Code Generation for Comparison Operations

Execution of comparison operations sets condition codes CC 0 and CC 1 (see Table 8.34). The overflow condition code is not affected. However, comparison operations affect the RLO. When a comparison is satisfied, the RLO is 1. Consequently, the conditional jump operation "JC" can also be used after a comparison operation.

Table 8.34 Condition Code Settings for Comparison Operations

Contents of ACCU 2 as Compared to Contents of ACCU 1	Condition Codes				Possible Jump Operations
	CC 1	CC 0	OV	OS	
Equal to	0	0			JZ, JC, JN
Less than	0	1			JM, JC, (JN)
Greater than	1	0			JP, JC, (JN)

Condition Code Generation for Arithmetic Operations

Execution of arithmetic operations sets all condition codes according to the result of the arithmetic operation (see Tables 8.35 to 8.39).

Table 8.35 Condition Code Settings for " + F"/"-F" Operations

Result after Arithmetic Operation is Executed	Condition Codes				Possible Jump Operations
	CC 1	CC 0	OV	OS	
x=0	0	0	0	-	JZ, (JN)
- 32768≤x<0	0	1	0	-	JM, (JN)
32767≥x>0	1	0	0	-	JP, (JN)
x=-65536*	0	0	1	1	JO, JOS, (JZ)
65534≥x>32767	0	1	1	1	JO, JOS, (JN, JM)
-65535≤x<-32768	1	0	1	1	JO, JOS, (JN, JP)

* This number is the result of the calculation - 32768 - 32768

Table 8.36 Condition Code Settings for " × F" Operation

Result after Arithmetic Operation is Executed	Condition Codes				Possible Jump Operations
	CC 1	CC 0	OV	OS	
x=0	0	0	0	-	JZ, (JN)
- 32768≤x<0	0	1	0	-	JM, (JN)
32767≥x>0	1	0	0	-	JP, (JN)
1073741824≥x>32767	1	0	1	1	JO, JOS, (JN, JP)
-1073709056≤x<-32768	0	1	1	1	JO, JOS, (JN, JM)

Table 8.37 Condition Code Settings for ":F" Operation

Result after Arithmetic Operation is Executed	Condition Codes				Possible Jump Operations
	CC 1	CC 0	OV	OS	
x = 0*	0	0	0	-	JZ, (JN)
-32768≤x<0	0	1	0	-	JM, (JN)
32767≥x>0	1	0	0	-	JP, (JN)
x = 32768	1	0	1	1	JO, JOS, (JN)
Division by 0	1	1	1	1	JO, JOS

* x=0 occurs, if the dividend = 0 and the divisor ≠ 0 or if the absolute value of the dividend is smaller than the absolute value of the divisor.

Table 8.38 Condition Code Settings for " + D"/"-D" Operation

Result after Arithmetic Operation is Executed	Condition Codes				Possible Jump Operations
	CC 1	CC 0	OV	OS	
x = 0	0	0	0	-	JZ, (JN)
-2147483648≤x<0	0	1	0	-	JM, (JN)
2147483647≥x>0	1	0	0	-	JP, (JN)
x = -4294967296*	0	0	1	1	JO, JOS, (JZ)
4294967294≥x>2147483647	0	1	1	1	JO, JOS, (JN, JM)
-4294967295≤x<-2147483648	1	0	1	1	JO, JOS, (JN, JP)

* This number is the result of the calculation -2147483648 -2147483648

Table 8.39 Condition Code Settings for floating-point arithmetics

Result after Arithmetic Operation is Executed	Condition Codes				Possible Jump Operations
	CC 1	CC 0	OV	OS	
$x=0,0\times 10^{0}$	0	0	0	-	JZ, (JN)
$-0,1701412\times 10^{39}\leq x\leq$ $-0,1469368\times 10^{-38}$	0	1	0	-	JM, (JN)
$0,1469368\times 10^{-38}\leq x\leq$ $0,1701412\times 10^{39}$	1	0	0	-	JP, (JN)
$0<\lvert x\rvert<0,1469368\times 10^{-38}$	0	0	1	1	JO, JOS, (JN, JZ)
$x<-0,1701412\times 10^{39}$	0	1	1	1	JO, JOS, (JN, JM)
$x>0,1701412\times 10^{39}$	1	0	1	1	JO, JOS, (JN, JP)
Division by 0	1	1	1	1	JO, JOS

Condition Code Generation for Digital Logic Operations

Digital logic operations set CC 0 and CC 1. They do not affect the overflow condition code (see Table 8.40). The setting depends on the contents of the ACCU after the operation has been processed.

Table 8.40 Condition Code Settings for Digital Logic Operations

Contents of the ACCU	Condition Codes				Possible Jump Operations
	CC 1	CC 0	OV	OS	
Zero (KH=0000)	0	0			JZ, (JN)
Not zero	1	0			JP, (JN)

Condition Code Generation for Shift Operations and Rotate Operations

Execution of the shift and rotate operations affects condition codes CC0 and CC1. The overflow condition code is not affected (see Table 8.41).

Setting of CC1 depends on the status of the last bit shifted.

Table 8.41 Condition Code Setting for Shift Operations

Value of the Last Bit Shifted	Condition Codes			Possible Jump Operations
	CC 1	CC 0	OV	
"0"	0	0		JZ, (JN)
"1"	1	0		JP, (JN)

Condition Code Generation for Conversion Operations

The formation of the two's complement (CSW, CSD) sets all condition codes (see Table 8.42). The state of the condition codes is based on the result of the conversion function.
The CFW operation does not set any condition codes.

Table 8.42 Condition Code Settings for Digital Logic Operations (16-bit arithmetics)

Result after Arithmetic Operation is Executed	Condition Codes				Possible Jump Operations
	CC 1	CC 0	OV	OS	
x=-32768*	0	1	1	1	JM, JO, (JN)
-32767≤x≤-1	0	1	0	0	JM, (JN)
x=0	0	0	0	0	JZ
32767≥x>=1	1	0	0	0	JP, (JN)

* This number is the result of the conversion of KH = 8000 (KF = -32768)

Table 8.43 Condition Code Settings for Digital Logic Operations (32-bit arithmetics)

Result after Arithmetic Operation is Executed	Condition Codes				Possible Jump Operations
	CC 1	CC 0	OV	OS	
x=2147483648	0	1	1	1	JO, JM, (JN)
-2147483648≤ x≤-1	0	1	0	-	JM, (JN)
1≤ x≤2147483647	1	0	0	-	JP, (JN)
x=0	0	0	0	-	JZ

DEF, DED, DUD, DUF, GFD Conversion Operations

Normally, these operations do not affect the condition codes. Only in case of non-convertibility, the condition code OS is set.
The accumulators are unaffected.

Special case: FDG is always convertible.

9 Interrupt Processing

9 Interrupt Processing

This chapter describes the following:

- Which blocks are provided for handling process interrupts in the S5-115U when the CPU 945 is used
- How to start up the 434-7 digital input module (with process interrupt)

9.1 Programming Interrupt Blocks

Each of these interrupts causes the operating system of the CPU to interrupt the cyclic or time-controlled program and to call an interrupt OB:

OB2 in the case of interrupt A	(interrupt A is triggered by the 434-7 DI module, the 485-7 DI/DQ module, by some CPs or by IPs)
OB3 in the case of interrupt B	(interrupt B is triggered by some CPs or by IPs)
OB4 in the case of interrupt C	(interrupt C is triggered by some CPs or by IPs)
OB5 in the case of interrupt D	(interrupt D is triggered by some CPs or by IPs)

Depending on the priorities of the individual alarms, the interrupt OBs have different interrupt behaviour.

For a description

- of the priorities of the interrupts and the interrupt behaviour of the interrupt OBs and
- notes on the calculation of interrupt response times

see Section 2.8.4 "Interrupt-Driven Program Processing".

9.1.1 Usable Modules

You can use interrupt-initiating modules in the S5-115U (e.g. intelligent I/O modules or the 434-7 digital input module or the 485-7 digital input/output module). These modules activate the CPU over an interrupt line in the I/O bus (S5 backplane bus). The CPU distinguishes between A, B, C or D interrupts depending on which interrupt line has been activated.
Exception: You must not use the digital inlut module 6ES5 432-4UA11 in the S5-115U.

9.2 Process Interrupt Generation with the 434-7 Digital Input Module

The 434-7 is a digital input module with programmable interrupt generation.

9.2.1 Function Description

The process interrupts are processed in two different ways:

- Interrupt-initiating inputs can be identified by the control program.
- A yellow LED lights up on the module and a relay contact is closed (the relay contact can be accessed externally via the "MELD" outputs). This signal remains even in the event of power failure and can be reset by applying 24 V to the 24 V RESET input.

Although the 434-7 digital input module has only eight inputs, it occupies two bytes in the input I/O area and two bytes in the output I/O area, i.e. you can access two bytes of inputs and two bytes of outputs (input byte and output byte each have the same address). Because the 434-7 digital input module occupies two I/O bytes, the IM 306 has to be set to 16 channels for this module.

The addresses of the two consecutive I/O bytes occupied by the 434-7 are referred to in the following as "module address" and "module address + 1".

- Use the two bytes of outputs in the restart OB for parameterizing the module (the "module address" byte indicates which input triggers the interrupt and the "module address + 1" byte determines the type of the interrupt-initiating edge)
- You must use the two bytes of inputs when
  - you want to scan the status of the inputs (scan the "module address" byte)
  - you want to indentify inputs which have triggered the interrupt (scan the "module address + 1" byte; only in interrupt program).

The status of the inputs must be scanned direct (L PY) since it is ***not*** transferred to the PII.

9.2.2 Startup

▸ Assign a slot address to the module; the IM 306 interface module is to be set to 16 channels for the 434-7 digital input module!

9.2.3 Initialization in Restart OBs

The following must be programmed in the RESTART blocks OB21 and OB22:

- Which inputs are to trigger an interrupt
- Whether the interrupt is to be triggered by a rising or falling edge.

This information is stored in two bytes which the program in OB21 or OB22 transfers to the module.
In the "module address" byte, mark which inputs are to trigger an interrupt, and in the "module address + 1" byte, mark which edge is to trigger the interrupt.

Programming the RESTART Blocks

STL	Description
`L KM a b`	Load a two-byte bit pattern into ACCU 1. (a: Bit pattern of the interrupt enable; b: Bit pattern of the edge initiating the interrupt)
`T PW x`	Transfer the information from ACCU 1 to the module (x is the module start address).

The bits in the high-order byte (byte a in this example) that was loaded into ACCU 1 with the statement "L KM ab" correspond to the bit addresses of the eight input channels. If a bit is set to "1", the interrupt is enabled for this channel.

The bits in the low-order byte indicate whether the interrupt on this channel is triggered on a leading edge ("0") or on a trailing edge ("1").

Example: Triggering inputs 2, 4, and 6 on a leading edge. Triggering inputs 1, 3, and 5 on a trailing edge.

	Interrupt enable								Interrupt-generating edge							
Bit address of the input	7							0	7							0
	0	1	1	1	1	1	1	0	▨	0	1	0	1	0	1	▨
	High-order byte								Low-order byte							

▨ = indicates irrelevant bits, since the corresponding bits in the high-order byte are set to "0" (no interrupt).

9.2.4 Reading in the Process Signals

The module offers a choice of two bytes for reading in the process signals:

- The "module address" byte reproduces the status of the inputs (regardless of whether the inputs have been parameterized for interrupt processing).
- In the "module address+1" byte, the bits assigned to the interrupt-initiating input are set after an interrupt, regardless of the type of initiating edge! (The module has to be parameterized at restart).

Example: The 434-7 digital input module has starting address 8; it occupies I/O bytes 8 and 9. At startup, only bit 0 has been enabled for interrupt initiation. The interrupt is to be triggered by a falling edge. In the event of an interrupt, bytes 8 and 9 have the following values (provided the status of input 8.0 has not changed after edge change):

Status of the inputs (8.0 to 8.7)

Bit address of the input	7							0
	x	x	x	x	x	x	x	0

Module address (byte 8)

Interrupt trigger

7							0
0	0	0	0	0	0	0	1

Module address+1 (byte 9)

x=Status of the inputs (0 or 1)

There are two ways of evaluating the input signals with bytes 8 and 9:

- You can read the status of the inputs with direct I/O access (L PY 8) at any point in your control program. It is irrelevant whether the status of the inputs is read in the cyclic, time-controlled or interrupt-processing program.

- If you have parameterized inputs at restart as interrupt-triggering inputs, you must program a specific interrupt response in OB2:

  - Acknowledge interrupt by reading the "module address+1" byte (in the example: byte 9; L PY 9)
  - Transfer the byte read to the PII (in the example: T IB 9)
  - Evaluate all inputs enabled for interrupt
  - Trigger interrupt response.

  After the byte "module address+1" (byte 9 in the example) has been loaded into the ACCU, is automatically reset on the module! The module is therefore in a position to trigger another interrupt and so set another bit in this byte! This means that the "module address+1" byte can be read out only once after an interrupt in order to identify the "interrupt trigger".

9.2.5 Programming Example for Interrupt Processing

Task

A tray is to be accurately positioned at two points:

Position 1 is determined by terminating switch 1.
When the signal status of limit switch 1 changes from 0 to 1 (positive edge), drive 1 is to be switched off.

Position 2 is determined by limit switch 2.
When the signal status of limit switch 2 changes from 0 to 1 (negative edge), drive 2 is to be switched off.

The status of the limit switches is to be indicated by two LEDs:

LED 1 for "Signal status of limit switch 1"
LED 2 for "Signal status of limit switch 2"

Implementation

The 434-7 module has starting address 8. The IM 306 is set to 16 channels for the 434-7.
Limit switch 1 is assigned to channel 0 of the module, limit switch 2 is assigned to channel 1 of the module.

The OB21 and OB22 restart programs have the task of parameterizing the module:

STL OB21/OB22	Meaning
`L KM 0000 0011 0000 0010`	Initialization of the interrupt inputs: Enable channel 0: positive edge
`T PW 8`	Enable channel 1: negative edge
`BE`	

The interrupts are evaluated in OB2:

Drive 1 is switched off by resetting output Q 0.0
Drive 2 is switched off by resetting output Q 0.1.

The status of the LEDs is updated in the cyclic program section:

When output Q 1.0 is set, LED 1 lights up
When output Q 1.1 is set, LED 2 lights up.

Evaluating the interrupt request in OB2:

STL OB2	Bedeutung
L PY 9 T IB 9	Acknowledge interrupt by loading the "mod. addr. +1" byte; Transfer to PII
A I 9.0 R Q 0.0	Scan: Did limit switch 1 trigger the interrupt? If yes, reset output Q 0.0 (switch off drive 1)
A I 9.1 R Q 0.1	Scan: Did limit switch 2 trigger the interrupt? If yes, reset output Q 0.1 (switch off drive 2)
L QB 0 T PY 0 BE	Transfer updated output byte QB 0 direct to the output module (direct I/O access to minimize response time!)

Updating the LED statuses in the cyclic program:

STL OB1	Bedeutung
:	
L PY 8 T IB 8	Load the status of the inputs (low byte) Transfer low byte to PII
A I 8.0 = Q 1.0	Transfer the status of limit switch 1 to the LED
A I 8.1 = Q 1.1 : BE	Transfer the status of limit switch 2 to the LED

Estimating the interrupt response time

(Prerequisite: no interrupts have been disabled with "IA")

The response time (i.e. the time between energizing the limit switch and switching off the drive) can be estimated as follows:

  Signal delay of the 434-7 DI (approx. 1 ms)
+ Response time of the CPU (see Section 2.8.4)
+ Execution time of OB2 (=sum of all operation execution times)

= Total response time

10 Analog Value Processing

Figures

Figures

Tables

Tables (Continued)

10 Analog Value Processing

Analog input modules convert analog process signals to digital values that the CPU can process. Analog output modules convert the digital values processed by the CPU to analog process signals.

10.1 Analog Input Modules

The analog measured value is digitized and stored in a data register on the module. It can then be read and processed further by the CPU.

Signal Interchange Between Module and CPU

The CPU can read the digitized value from the module's RAM in the following two ways:

- Via an integrated FB
- Via a load operation (L PW).

In the CPU 945, an integrated FB is available for each analog input module for reading of analog values.

• 460/465 analog input module:	FB250
• 463 analog input module:	FB241
• 464-8 analog input module (if ET 100/ET 200 are used):	FB242
• 466 analog input module:	FB243

The complete measured value (2 bytes) is stored in the CPU.

460, 463, 464-8, 465 and 466 Analog Input Modules

Five analog input modules with the following characteristics are available:

6ES5 460-7LA12/LA13
- Galvanically isolated
- 8 channels
- Encoding time: max. 60 ms per channel
  (worst case max. 400 ms with cyclic scanning)
- 2 range cards
- Max. permissible isolating voltage 60 V AC/75 V DC; between the channels and ground (M) in each case and between the channels themselves

6ES5 463-4UA11/-4UA12/-4UB11/-4UB12
- Galvanically isolated
- 4 channels
- Encoding time: max. 20 ms per channel
- Max. permissible isolating voltage 25 V AC/60 V DC; between the channels and ground (M) in each case and between the channels themselves

The **464-8** analog input module can be used in conjunction with ET 100/ET 200 applications.

6ES5 465-7LA13

- Non-isolated
- 8/16 channels (selectable)
- Encoding time: max. 60 ms per channel
  (worst case max. 960 ms with cycle scanning)
- 2/4 range cards
- 1 V max. permissible voltage between a channel and ground as well as between channels

6ES5 466-3LA11

- Floating
- 8/16 channels (switchable)
- Short coding times: 2 ms (8 channels) or 4 ms (16 channels)
- 12 different measuring ranges can be set using switches on the module
- Choice between common-reference measurement (16 channels) or differential measurement (8 channels)
- All operating modes can be set using switches on the module
- Maximum permissible isolation voltage V_{ISO}: 60 V AC/75 V DC; between the channels and ground (M) in each case; however, not between the channels themselves!

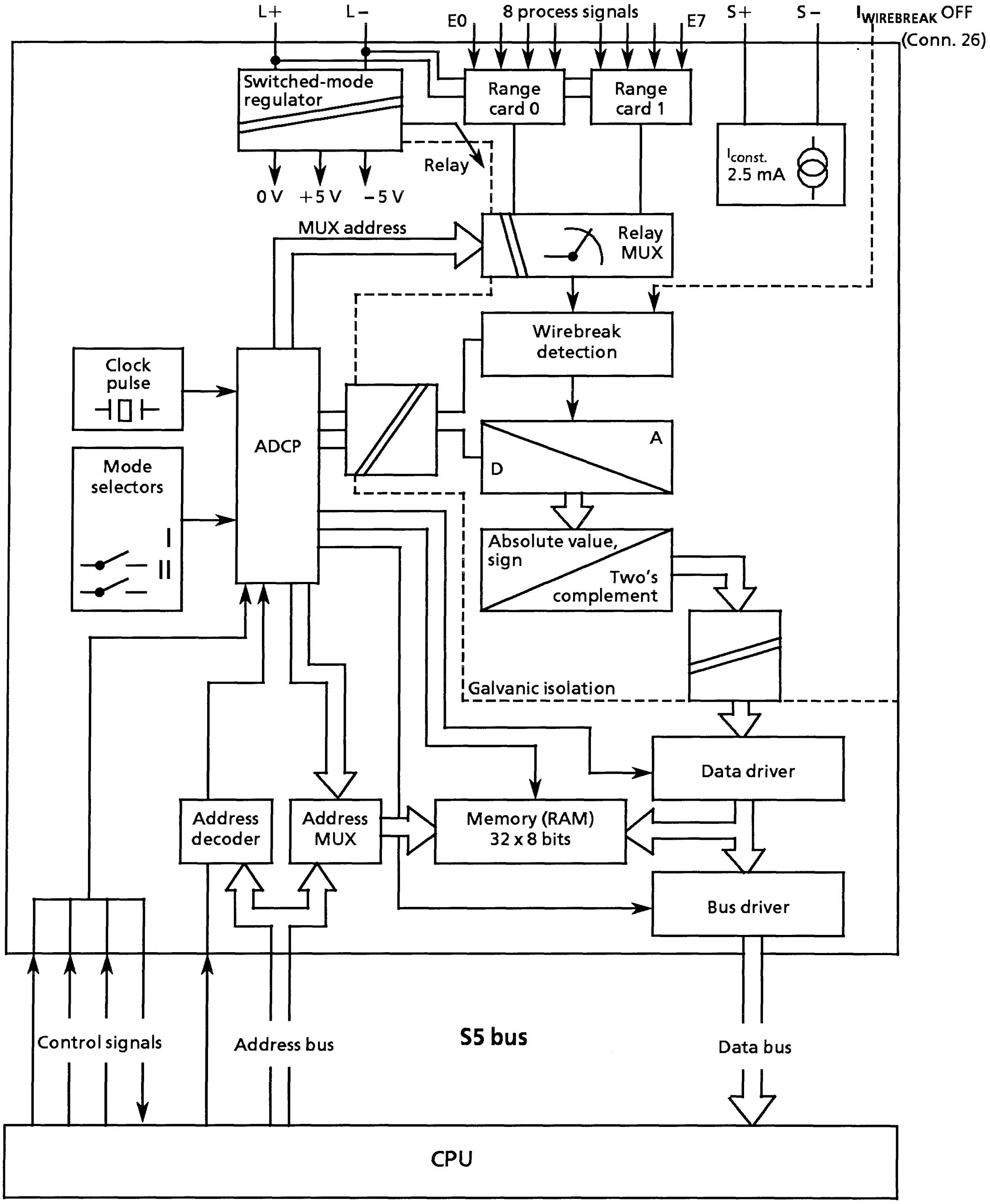
The block diagrams (Figures 10.1, 10.12, 10.17 and 10.22) illustrate the method of operation as well as the signal interchange between the analog input modules and the CPU.

In the case of the **460** and **465** modules, a processor (ADCP) controls the multiplexer, analog-digital conversion and the forwarding of the digitized measured values to the memory or to the data bus of the programmable controller. The controller takes account of the module's operating mode, which is set at the relevant switch.
The process signals must be matched to the input level of the analog digital converter (ADC) to suit the application. You can match the signals with the 460 and 465 modules by plugging a suitable range card (voltage divider or shunt) into the receptacle on the frontplate of the analog input module.

In the case of the **463** module the analog input signals are digitized by voltage-frequency converters and written into a counter via rapid optocouplers. Presetting of the counters, the duration of the integration time and the transfer of the counting result of the four input channels into the four measured value memories are coordinated by sequential control.
The measuring range of each channel is adjusted via the connection of sensors and through jumpers in the front connector of the module.

In the case of the **466** module an internal controller handles all required functions.
You can adapt the process signals to the input level of the analog input modules in the case of the 466 module by specific settings of the measuring range switches.

10.2 Analog Input Module 460-7LA12

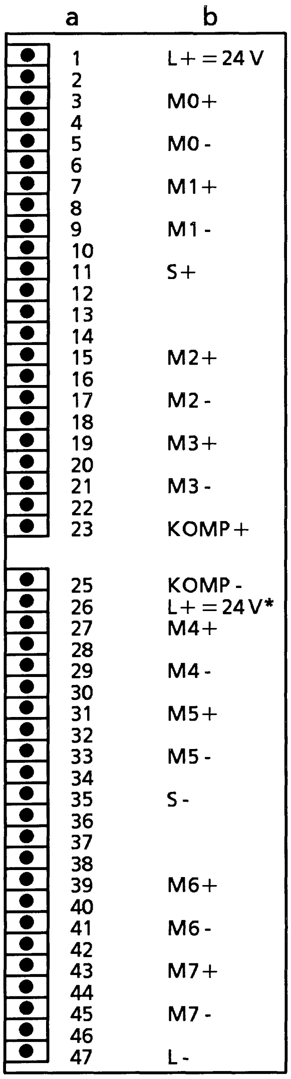


A/D	Analog-digital converter (ADC)
ADUS	ADU processor
MUX	Multiplexer

Figure 10-1 Block Diagram with Signal Interchange between the 460 Analog Input Module and the CPU

10.2.1 Connecting Transducers to the 460-7LA12 Analog Input Module

Pin assignments of the front connector



a	b
1	L+ = 24 V
2	
3	M0+
4	
5	M0 -
6	
7	M1+
8	
9	M1 -
10	
11	S+
12	
13	
14	
15	M2+
16	
17	M2 -
18	
19	M3+
20	
21	M3 -
22	
23	KOMP+
25	KOMP -
26	L+ = 24 V*
27	M4+
28	
29	M4 -
30	
31	M5+
32	
33	M5 -
34	
35	S -
36	
37	
38	
39	M6+
40	
41	M6 -
42	
43	M7+
44	
45	M7 -
46	
47	L -

460-7LA12

a = Pin No.
b = Assignment

* Switching off the test current in the case of non-activated wirebreak signal

Figure 10.2 Pin Assignments for the 460 Analog Input Module

Certain precautionary measures must be taken in order to make sure that potential difference V_{CM} is not exceeded. Different measures are required for isolated and non-isolated transducers.

Isolated Transducers

When isolated transducers are used, the measuring circuit can assume a potential to earth that exceeds the permissible potential difference U_{CM} (refer to the maximum values for the various modules).
To prevent this, the transducer's negative potential must be connected to the module's reference potential (reference bus).

Example: Measuring temperature on a busbar with an isolated thermocouple.
In a worst-case situation, the measuring circuit can assume a potential that would destroy the module; this must be prevented through the use of an equipotential bonding conductor (see Figure 10-3).

Possible causes:

- Static charge
- Contact resistors through which the measuring circuit assumes the potential of the busbar (e.g. 220 V AC).

Non-isolated Transducers

When using non-isolated transducers, the permissible potential difference U_{CM} between the inputs and the reference bus must not be exceeded.

Example: Measuring the temperature of the busbar of an electroplating bath with a non-isolated thermocouple. The difference between the potential of the busbar and the reference potential of the module is max. 24 V DC. A 460 analog input module with floating input (permissible U_{CM} is 60 V AC/75 V DC) is to be used.

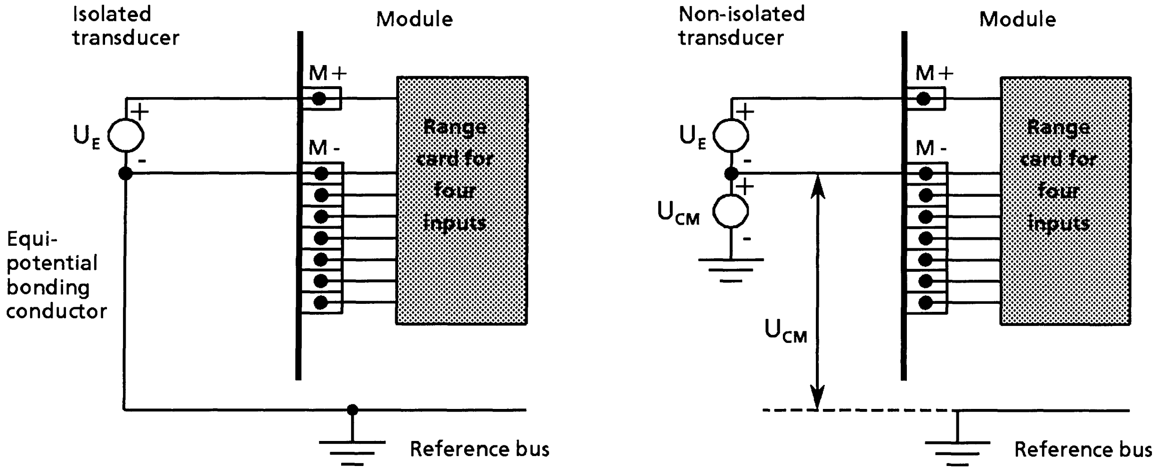


Figure 10.3 Connecting Transducers

You must observe various conditions when connecting current or voltage sensors to analog input modules, depending on what type of sensors are used.

Note

Detailed information on address assignment for analog modules is presented in Chapter 6 (Addressing/Address Assignments). Please observe the information regarding the overall structure (see Sections 3.5.2 to 3.5.4).

Note

Unused inputs must be terminated with a voltage divider or shunt (see Table 10-1).
In the case of the 498-1AA11 module, the unused inputs must be short-circuited (M+ with M - in each case).
Other modules require no additional wiring.
The galvanic isolation between the analog inputs and L+or L - is nullified when using the 498-1LAA51 module for a 2-wire transducer!

Connecting Thermocouples with Compensating Box

The influence of the temperature on the reference junction (in the terminal box, for instance) must be equalized using a compensating box. Please observe the following:

- The compensating box must have an isolated power supply.
- The power supply unit must have a grounded shielding winding.

Compensate as follows when all thermocouples connected to the module's inputs have the same reference junction:

- Provide a separate compensating box for each analog input module
- Bring the compensating box into thermal contact with the terminals
- Apply compensating voltage to pins 23 and 25 (KOMP+and KOMP -) on the analog input module (Figure 10-4)
- Set Function Select switch II on the module for operating a compensating box (see also Table 10-2)

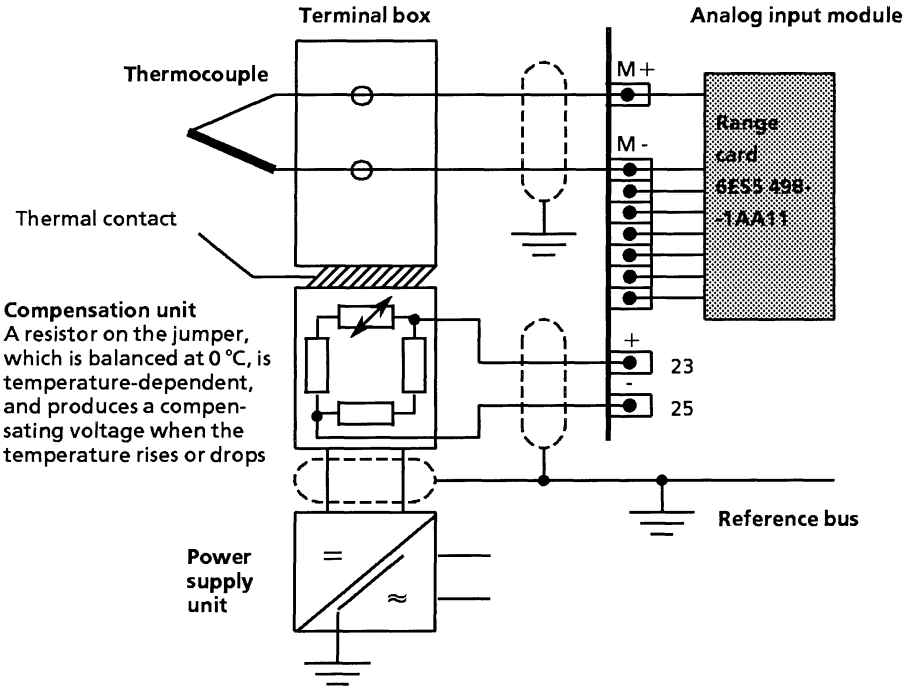

Terminal box
Analog input module
Thermocouple
M+
M -
Range card 6ES5 498- -1AA11
Thermal contact
Compensation unit
A resistor on the jumper, which is balanced at 0 °C, is temperature-dependent, and produces a compensating voltage when the temperature rises or drops
+
-
23
25
Reference bus
Power supply unit
=
≈

Figure 10-4 Connecting Thermocouples

Detailed information on thermocouples and compensating boxes can be found in Catalog MP 19.

When several thermocouples are distributed over areas with different temperature ranges, it is often advantageous to acquire different reference junction temperatures. In this case, the central compensating input is no longer used. A separate compensating box is used for each analog input channel to be compensated. KOMP+and KOMP - remain unconnected.

- Connect the relevant thermocouple in series with the compensating box.
- Run the remaining terminal leads from compensating box and thermocouple to the analog module (terminal M+and M - see Figure 10-5).
- Set Function Select switch II on the module to "Without reference junction compensation".

Compensation, i.e. correction of the temperature error, subsequently takes place in the compensating box rather than on the module.

The corrected value is thus available at terminals M+and M - of the relevant analog input channels, and is then converted into a digital value.

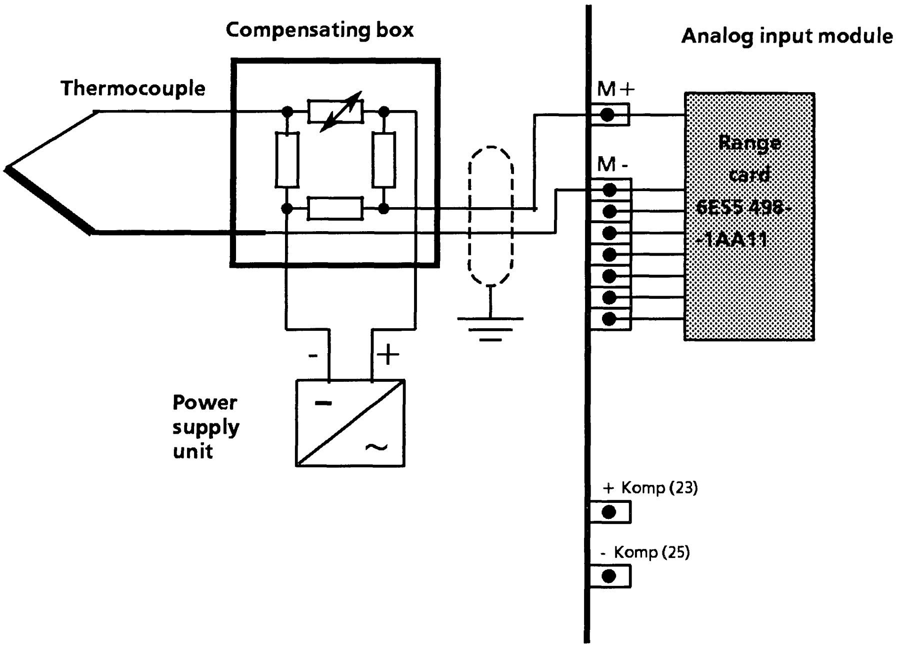


Figure 10.5 Connecting a Compensating Box to the Input of an Analog Input Module

Connecting Resistance Thermometers (e.g. PT 100) with 6ES5 460-7LA12

A constant-current generator supplies the series-connected resistance thermometers (max. 8 PT 100s) with a current of 2.5 mA over pins "S+" and "S -".
If you use the 498-1AA11 submodule, you need not terminate the unused input channels with a short-circuiting jumper (see Figure 10-6, range card 2, channel 5 and 6).

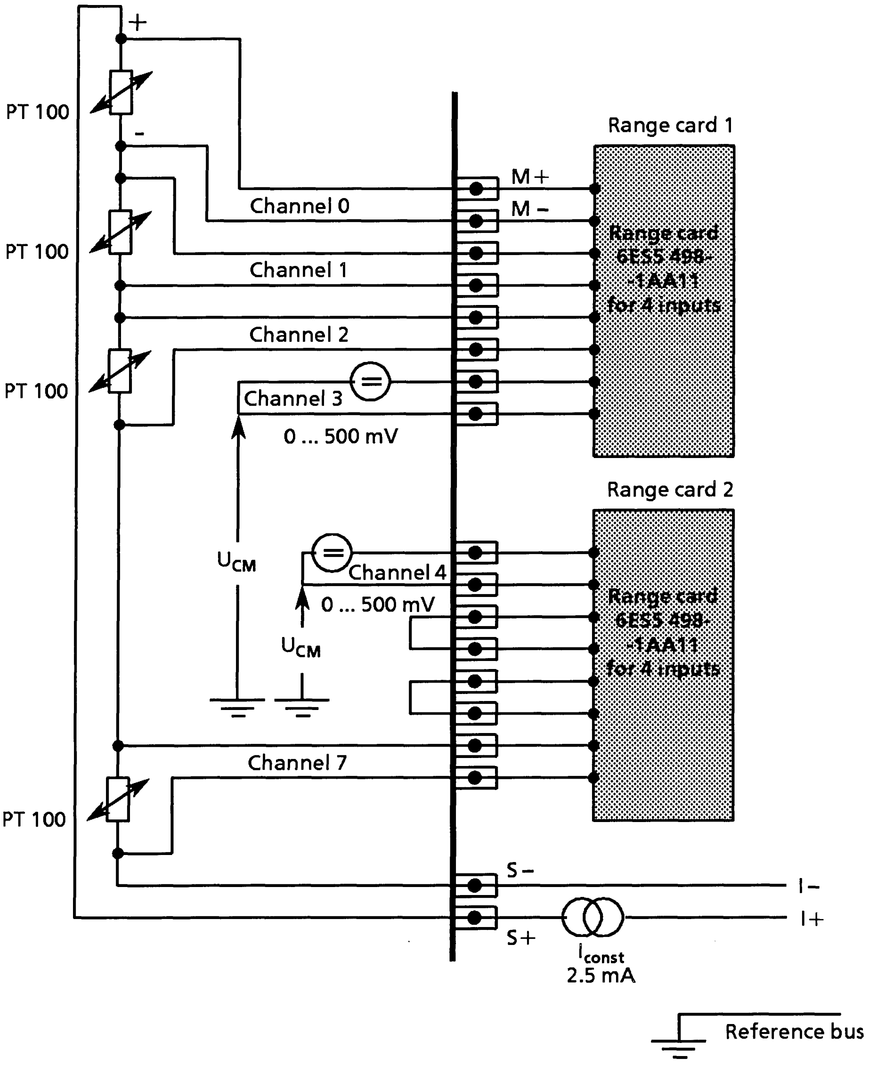
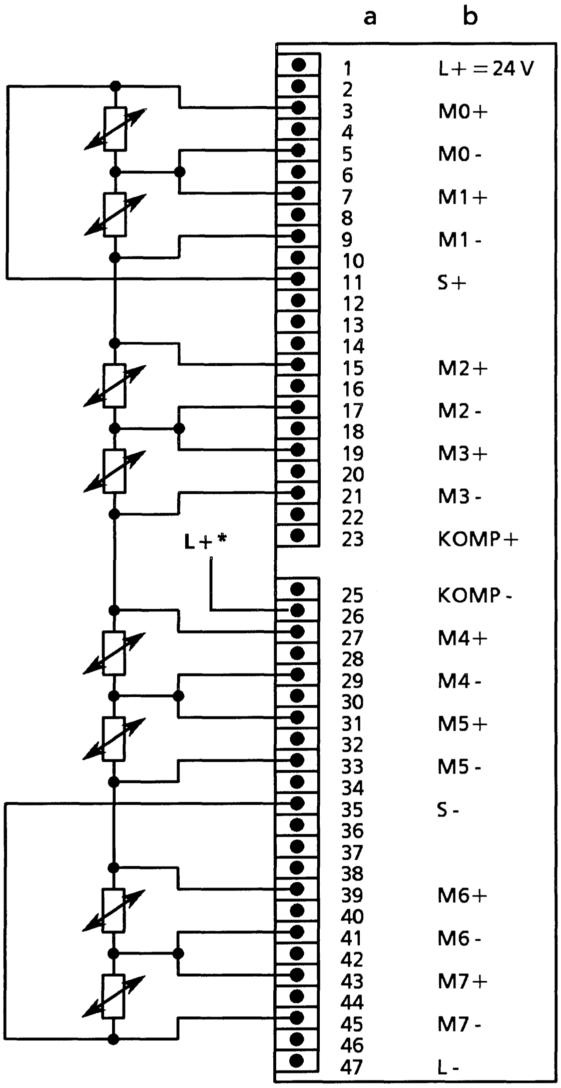


Figure 10-6 Connecting Resistance Thermometers (PT 100s) to a 460 Analog Input Module

If no PT 100 is connected to input channels 4 to 7, other voltages and currents can be measured on these channels using range card 498-1AA21, -1AA31, -1AA41, -1AA51, -1AA61 or -1AA71.

The diagram below shows the pin assignments for resistance thermometers used on analog input module 460.



6ES5 460-7LA12

a = Pin No.
b = Assignment

* Required only for disconnecting the test current when the wirebreak signal is not activated

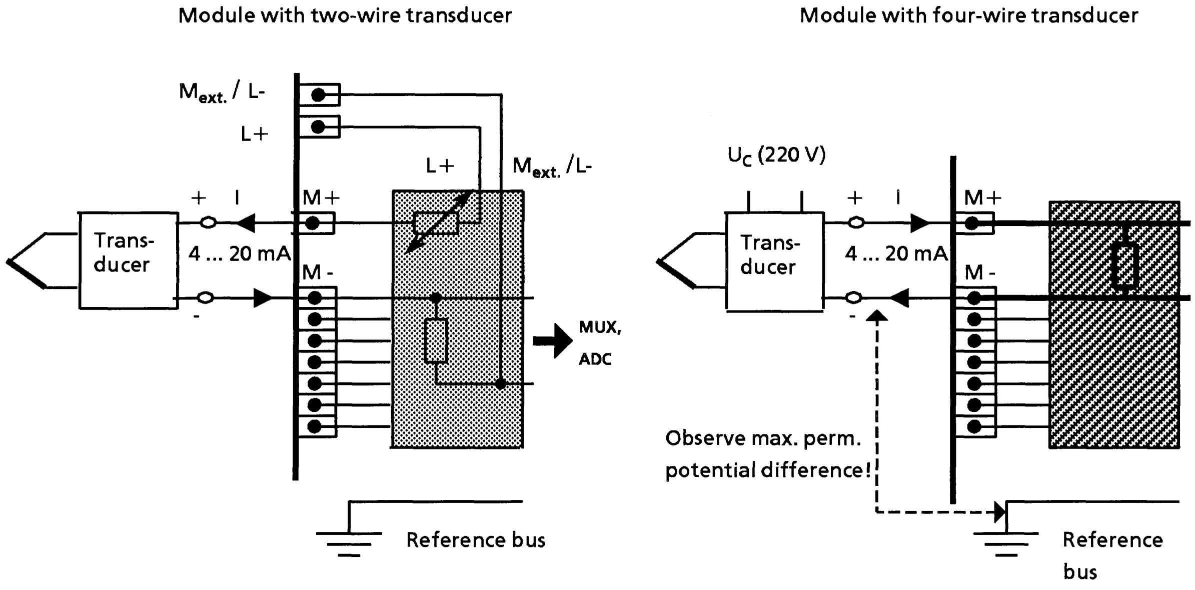
Figure 10.7 Pin Assignments for Analog Input Modules

Connecting Transducers with Module 460-7LA12

The inherently short-circuit-proof supply voltage is fed to the two-wire transducer over the range card.

Four-wire transducers have a separate power supply.

The diagram below shows how to connect two-wire and four-wire transducers.

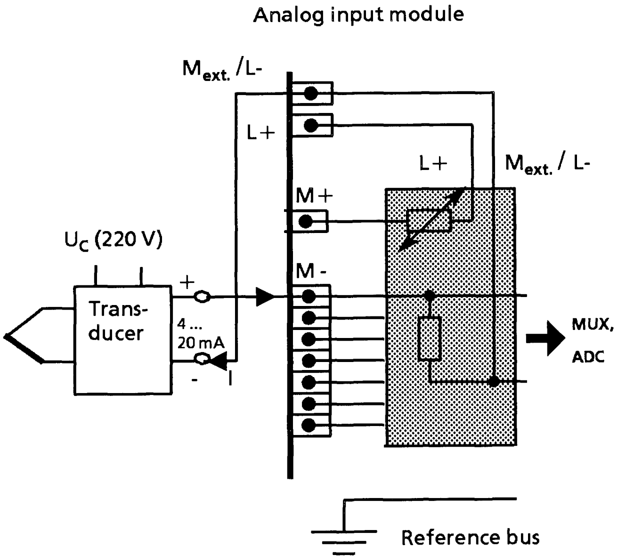


▒ 6ES5 498-1AA51 range card (with internal circuitry)
Galvanic isolation between the analog input and L+/L- is removed.

▨ 6ES5 498-1AA71 range card (with internal circuitry)

Figure 10.8 Connecting Transducers

The diagram below shows how to connect a four-wire transducer to a two-wire transducer range card (498-1AA51).



▨ 6ES5 498-1AA51 range card
(with internal circuitry)

Figure 10.9 Connecting Transducers (Four-Wire Transducer to a Two-Wire Range Card)

10.2.2 Startup of Analog Module 460-7LA12

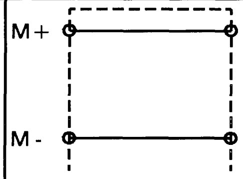
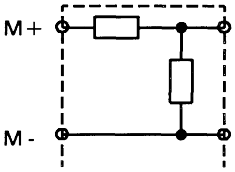
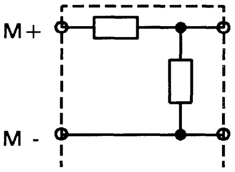
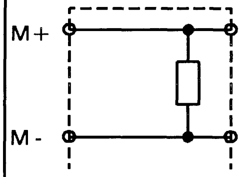
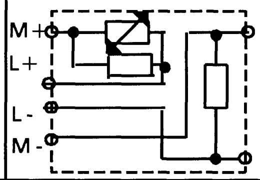
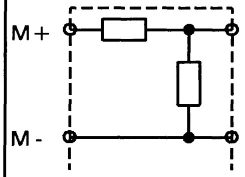
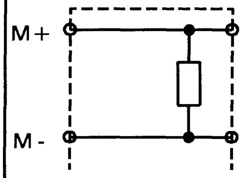
Voltage dividers or shunt resistors can be plugged into the input modules as cards (see Table 10-1).

They match the process signals to the input level of the module.
These cards make it possible to set different measuring ranges.

Connecting Range Cards

Two range cards can be plugged into the 460 analog input module. One card specifies the measuring range of four inputs.
We offer voltage dividers, shunts and through-connection cards (see Table 10-1).

Table 10.1 Range Cards

Range card 6ES5 498-	Circuitry (4 times each)	Function 500 mV/mA/PT100	Function 50 mV
- 1AA11		±500 mV; PT 100	±50 mV
- 1AA21		±1 V	±100 mV *
- 1AA31		±10 V	±1 V *
- 1AA41		±20 mA	±2 mA *
- 1AA51**		+4 ... +20 mA two-wire transducer	
- 1AA61		±5 V	±500 mV *
- 1AA71		+4 ... +20 mA four-wire transducer	

* Possible measuring range for "50 mV" setting, but with higher incidence of error

** When a -1AA51 range card is used, there is no longer any galvanic isolation between analog inputs and L+ !

Note

Jumpers must be set in the front connector in the case of through-connection card 1AA11. Unused inputs need *not* be short-circuited in the case of voltage dividers or shunts.

You can set various functions on an input module by setting the Function Select switches on the rear of the module accordingly (see Table 10-2).

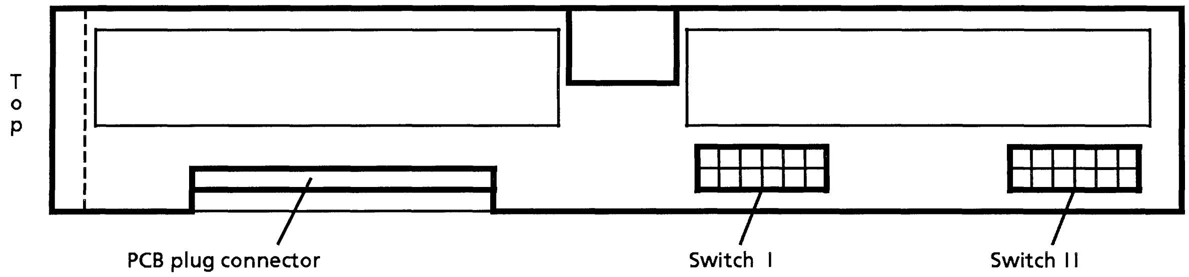
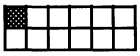
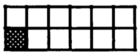
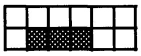
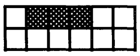
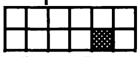
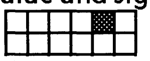
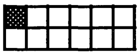
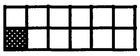
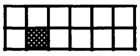
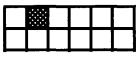
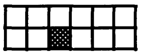
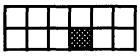
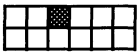
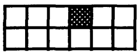


Figure 10.10 Position of the Function Select Switches of the 460-7LA12 Analog Input Module

Note

Selection of a function entails the setting of all switches.

Table 10.2 Setting Functions on the 6ES5 460-7LA12 Module

Function	Setting on Switch I		Setting on Switch Π	
Reference junction compensation			Yes 	No 
Measuring range* (nominal value)			50 mV 	500 mV 
Analog value representation			Two's complement 	Absolute value and sign 
Sampling	Cyclic 	Selective 		
System frequency	50 Hz 	60 Hz 		
Wirebreak signal	Channel 0 to 3 	Channel 4 to 7 		
No wirebreak signal	Channel 0 to 3 	Channel 4 to 7 		

* Setting for PT 100: Measuring range 500 mV

10.3 460-7LA13 Analog Input Module

The 460-7LA13 analog input module has been developed from the 460-7LA12 analog input module. It offers the following advantages:

- Lower power consumption and heating
- Lower weight
- New PT 100 climatic measurement range (-100 °C to +100 °C) with high resolution (1/40 °C)

All functions of the 460-7LA12 module are also available on the 460-7LA13 module.

The following features are identical with the 460-7LA12 module:

- Transducer cabling
- Use of the 6ES5 498 range cards
- Assignment of the front connector
- System behaviour

Compared to the 460-7LA12 analog input module, the following features of the 460-7LA13 analog input module are new or different:

- New PT100 climatic measuring range (as alternative to the previous PT100 measuring range)
- Setting of the mode selector switches for the PT100 measuring ranges
- Setting of the mode selector switched for the measuring range 50 mV (e. g. for connection of thermocouples).

New PT100 Climatic Measuring Range

The same measuring range as with the 460-7LA12 analog input module exists also on the 460-7LA13 analog input module; i.e. the PT100 temperature range (-200 °C to +850 °C) is resolved in this measuring range to approximately 4000 units. This corresponds to a resolution of approx. 0.25 °C.

If the new PT100 climatic measuring range is selected via the mode selector switches, all eight analog inputs can be used in this measuring range only.

Do not use other than the 6ES5 498-1AA11 (50 mV/0.5 V) range card.

The following must be observed in conjunction with wire break monitoring in the PT 100 climatic measuring range:
If a line of the auxiliary circuit (IC+, IC-) is interrupted, the value "negative end value" is encoded for all inputs and the overflow bit is set to "1". In the case of transducer or measuring line break, the error bit for the corresponding channel is additionally set to "1".

For an exact representation of measured values in the PT100 climatic measuring range refer to Table 10.26.

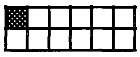
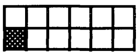
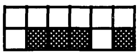
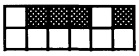
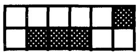
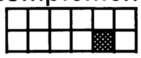
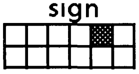
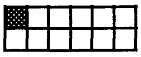
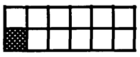
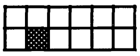
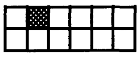
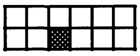
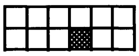
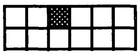
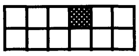
For setting of the "PT100" mode, the following markings are printed on the cover of the module:

Standard range:	"resistance thermometer uncompensated full range"
Climatic measuring range:	"resistance thermometer uncompensated low range"

Setting of Mode Selector Switches I and II

The mounting position and the setting of the mode selector switches corresponds to that of the 460-7LA12 module. The only difference is the setting of the PT100 measuring ranges (see Table 10.3).

Table 10.3 Setting Functions on the 6ES5 460-7LA13 Module

Function	Setting on Switch I		Setting on Switch II	
Reference junction compensation			Yes 	No 
Measuring range* (nominal value)			resistance thermometer compensated low range  500 mV, V ... mA resistance thermometer uncompensated full range  50 mV 	
Analog value representation			Two's complement 	Abs. value and sign 
Sampling	Cyclic 	Selective 		
System frequency	50 Hz 	60 Hz 		
Wirebreak signal	Channel 0 to 3 	Channel 4 to 7 		
No wirebreak signal	Channel 0 to 3 	Channel 4 to 7 		

Transducer Wiring

Transducers are wired in the same way as with the 460-7LA12 module. Unused inputs must be connected in parallel with switched inputs. An example is given in Fig. 10.11.

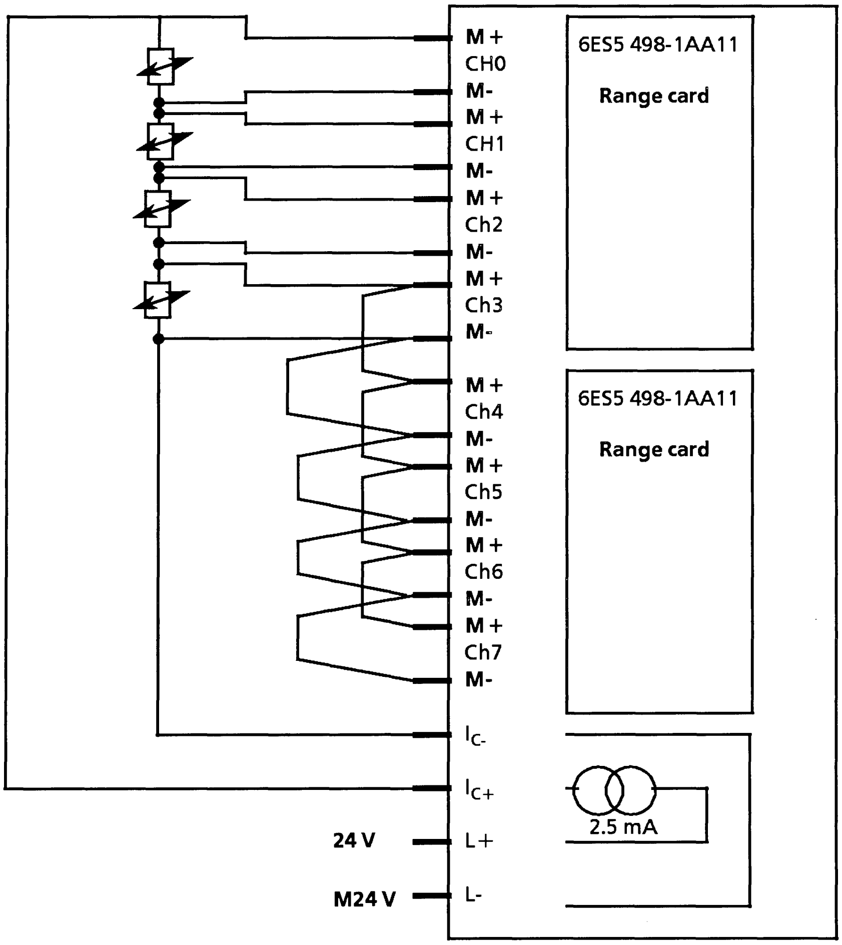
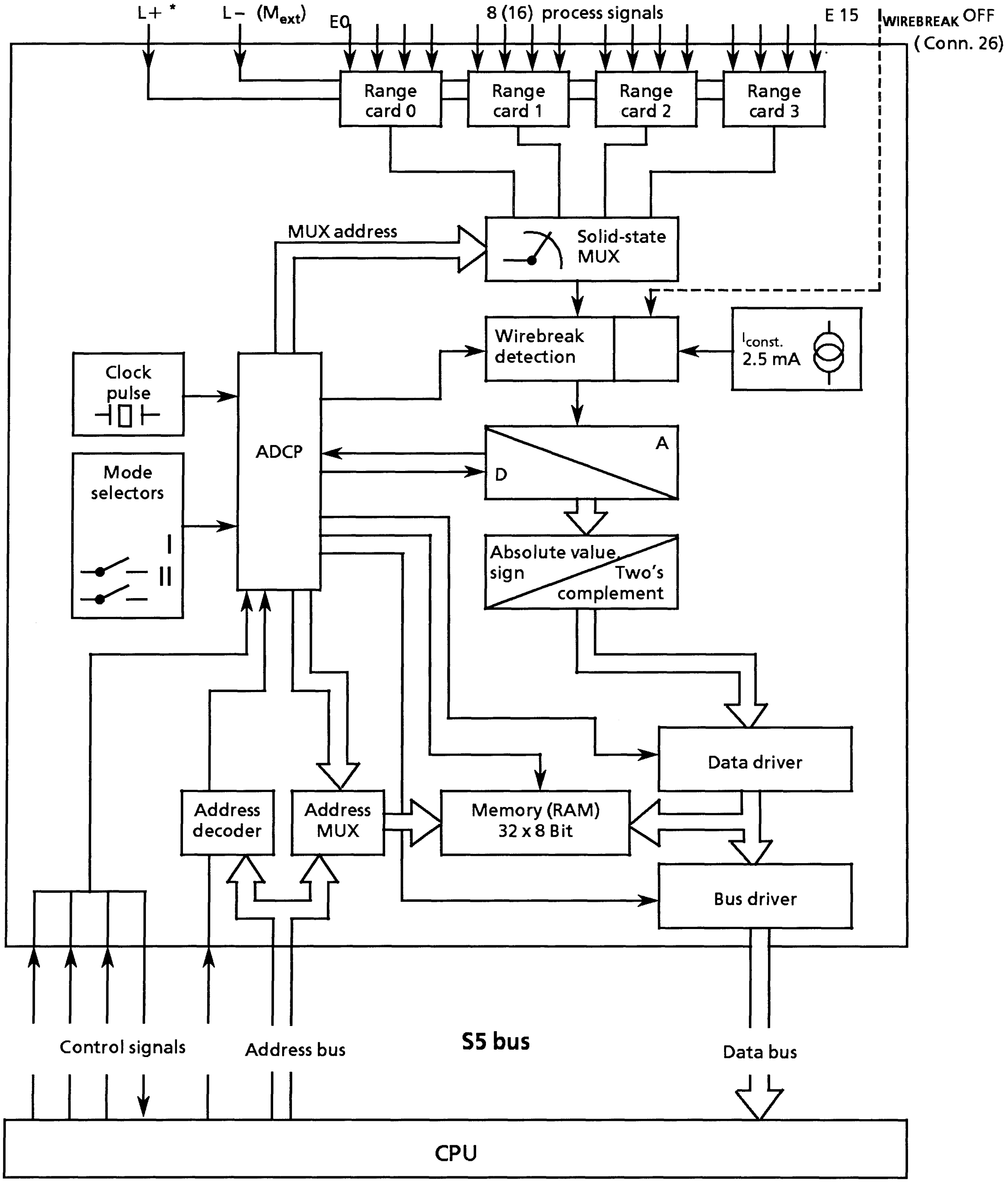


Figure 10.11 Wiring of Transducers on the 460-7LA13 Analog Input Module (for Climatic Measuring Range)

10.4 Analog Input Module 465-7LA13



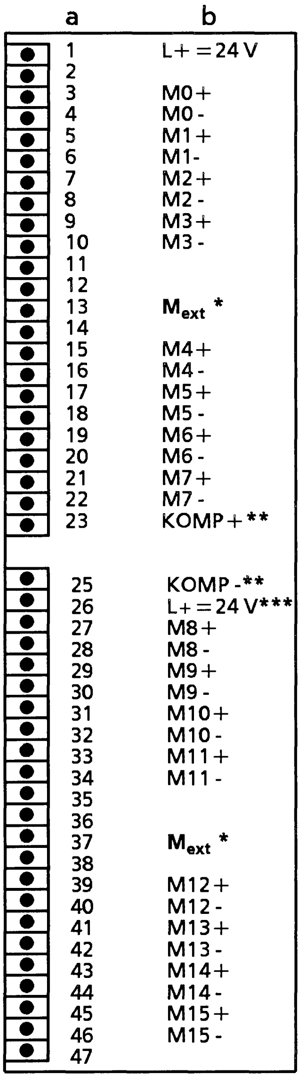
A/D	Analog-digital converter (ADC)
ADCP	ADC processor
MUX	Multiplexer

* Required only when using a -1AA51 module

Figure 10-12 Block Diagram with Signal Interchange between a 465 Non-Isolated Analog Input Module and the CPU

10.4.1 Connecting Transducers to the 465-7LA13 Analog Input Module

Pin assignments of the front connector



a	b
1	L+ = 24 V
2	
3	M0+
4	M0 -
5	M1+
6	M1-
7	M2+
8	M2 -
9	M3+
10	M3 -
11	
12	
13	M_{ext} *
14	
15	M4+
16	M4 -
17	M5+
18	M5 -
19	M6+
20	M6 -
21	M7+
22	M7 -
23	KOMP+**
25	KOMP -**
26	L+ = 24 V***
27	M8+
28	M8 -
29	M9+
30	M9 -
31	M10+
32	M10 -
33	M11+
34	M11 -
35	
36	
37	M_{ext} *
38	
39	M12+
40	M12 -
41	M13+
42	M13 -
43	M14+
44	M14 -
45	M15+
46	M15 -
47	

465-7LA13

a = Pin No.
b = Assignment

* Connection to the central grounding point of the controller
** Connection of the compensating box
*** Switching off the test current in the case of non-activated wirebreak signal

Figure 10.13 Pin Assignments for the 465 Analog Input Module

Note

Connection of transducers is described in detail in Section 10.2.1.

Note

Unused inputs must be short-circuited when using the 6ES5 498-1AA11 through-connection card.

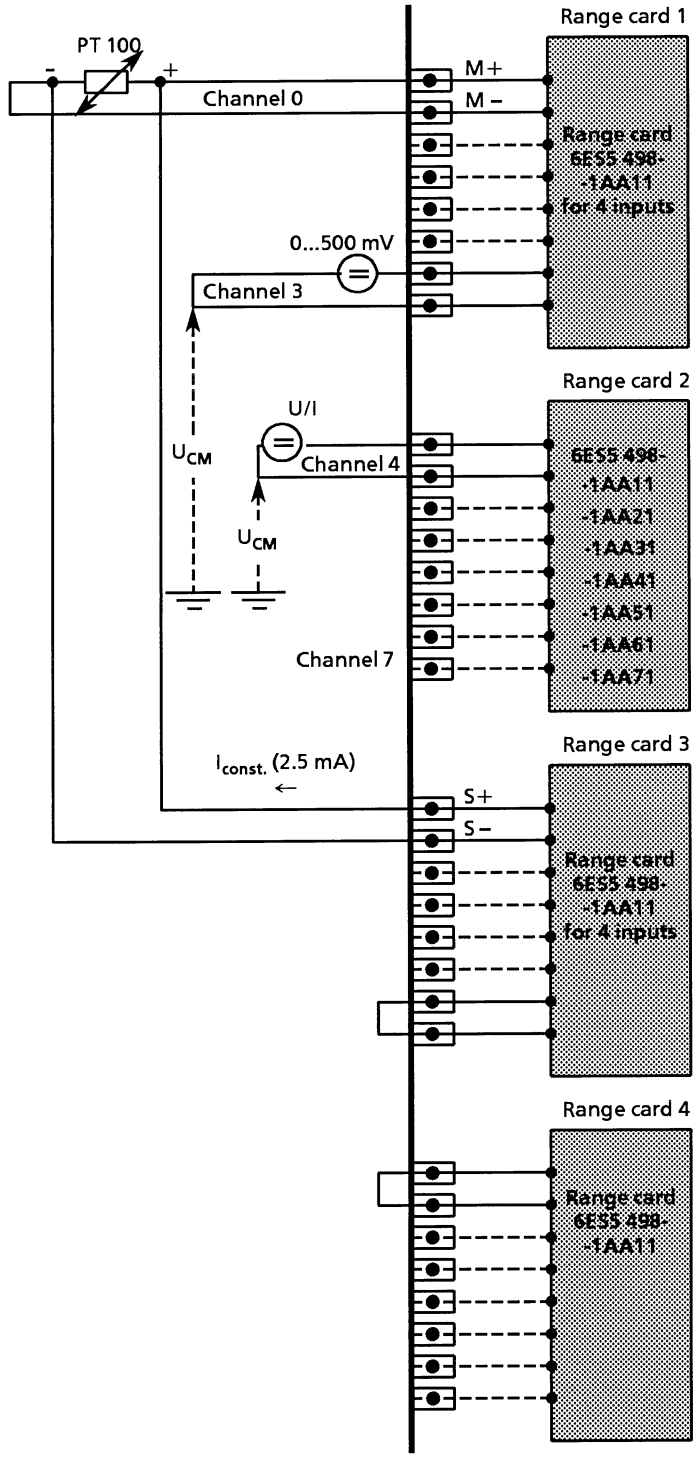
Note

Detailed information on address assignment for analog modules is presented in Chapter 6 (Addressing/Address Assignments). Please observe the information regarding wiring, electrical structure and conductor arrangement (see Sections 3.5.2 to 3.5.4).

Connecting Thermocouples with Compensation Boxes

Connection of thermocouples is the same as for the 460 module (see Section 10.2.1)

Connecting Resistance Thermometers (PT 100) to a 465-7LA13 Analog Module


Range card 1
PT 100
-
+
Channel 0
M+
M −
Range card 6ES5 498- -1AA11 for 4 inputs
0...500 mV
Channel 3
Range card 2
U/I
U_CM
Channel 4
U_CM
6ES5 498- -1AA11 -1AA21 -1AA31 -1AA41 -1AA51 -1AA61 -1AA71
Channel 7
Range card 3
I_const. (2.5 mA)
←
S+
S−
Range card 6ES5 498- -1AA11 for 4 inputs
Range card 4
Range card 6ES5 498- -1AA11

A constant-current generator supplies the relevant resistance thermometer with a current of 2.5 mA over pins "S+" and "S -" via a range card (6ES5 498-1AA11) (see Figure 10-14).

The voltage on the PT 100 is picked off over inputs "M+" and "M -".

Other potential-free voltage sensors (500 mV voltage range) can be connected to those inputs (M+/M -) not used for resistance thermometers.

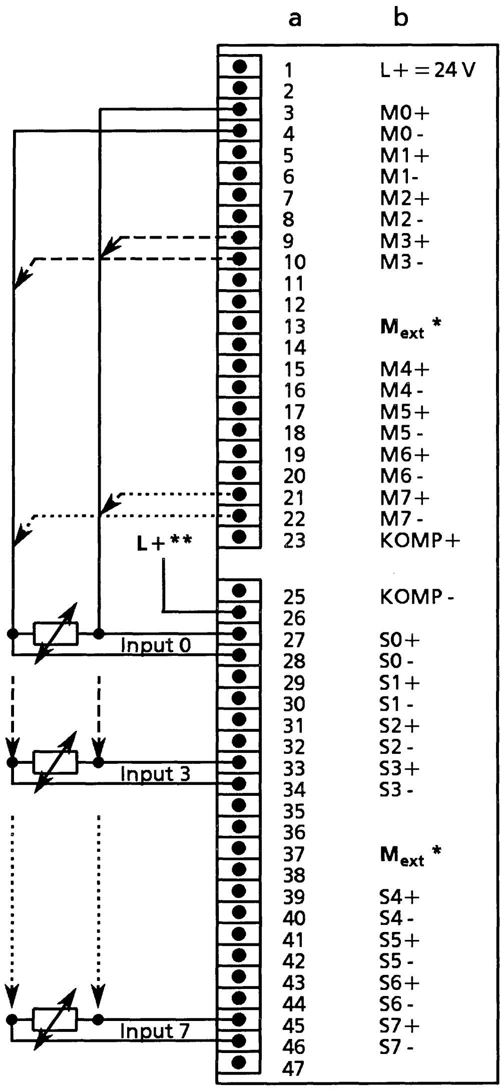
If no PT 100 is connected over input channels 4 to 7, other voltages and currents can be measured over these channels using a 498-1AA21, -1AA31, -1AA41, -1AA51, -1AA61 or -1AA71 range card (see Figure 10-14 range card 2). In this case, you must short-circuit the current outputs (S+, S-) belonging to the relevant card with a jumper. Should you fail to do so, the error bit would be set for the relevant channel and the value "0" decoded (see Figure 10-14 range card 4).

If you use a -1AA21, -1AA31 or -1AA61 range card for a channel group, no wirebreak signal may be enabled for that channel group.

A correction of 100 ohms (100 ohms = 0°C) must be made via the control program by specifying the appropriate upper and lower limiting values in FB250 (see Section 10.10.1).

Figure 10-14 Connecting Resistance Thermometers (PT 100s) to a 465 Analog Input Module

The following figure shows the pin assignments of the 465-7LA13 module for resistance thermometers.



a	b
1	L+ = 24 V
2	
3	M0+
4	M0 -
5	M1+
6	M1-
7	M2+
8	M2 -
9	M3+
10	M3 -
11	
12	
13	M_{ext} *
14	
15	M4+
16	M4 -
17	M5+
18	M5 -
19	M6+
20	M6 -
21	M7+
22	M7 -
23	KOMP+
25	KOMP -
26	
27	S0+
28	S0 -
29	S1+
30	S1 -
31	S2+
32	S2 -
33	S3+
34	S3 -
35	
36	
37	M_{ext} *
38	
39	S4+
40	S4 -
41	S5+
42	S5 -
43	S6+
44	S6 -
45	S7+
46	S7 -
47	

6ES5 465-7LA13

a = Pin No.
b = Assignment

* Connection to the central grounding point of the controller
** Required only for disconnecting the test current when the wirebreak signal is not activated

Figure 10.15 Pin Assignments for Analog Input Module 465

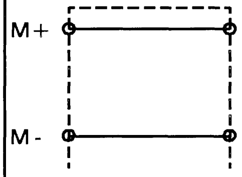
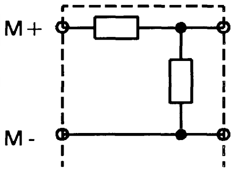
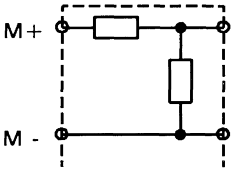
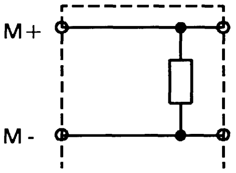
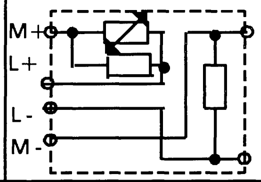
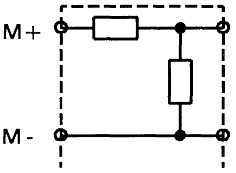
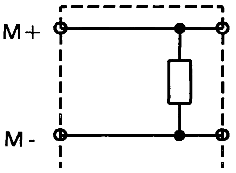
Connecting Transducers

Transducers are connected as in the case of the 460 module (see Section 10.2.1).

10.4.2 Startup of the 465-7LA13 Analog Input Module

Voltage dividers and shunts can be plugged in as range cards (see Table 10-4). They match the process signals to the input level of the module.
In this way, various measuring ranges can be set.

Table 10.4 Range Cards

Range card 6ES5 498-	Circuitry (4 times each)	Function 500 mV/mA/PT100	Function 50 mV
- 1AA11		± 500 mV; PT 100	± 50 mV
- 1AA21		± 1 V	± 100 mV *
- 1AA31		± 10 V	± 1 V *
- 1AA41		± 20 mA	± 2 mA *
- 1AA51**		+ 4 to + 20 mA two-wire transducer	
- 1AA61		± 5 V	± 500 mV *
- 1AA71		+ 4 to + 20 mA four-wire transducer	

* Possible measuring range for "50 mV" setting, but with higher incidence of error

** When a -1AA51 range card is used, there is no longer any galvanic isolation between analog inputs and L+/L-!

Note

In the case of the 1AA11 through-connection card, unused inputs must be short-circuited. In the case of a voltage divider or shunt card, unused inuts must ***not*** be short-circuited.

Function select switches for setting various functions are located on the back of the 465 module.

For this purpose, the switches must be set to the positions shown (see Table 10-5).

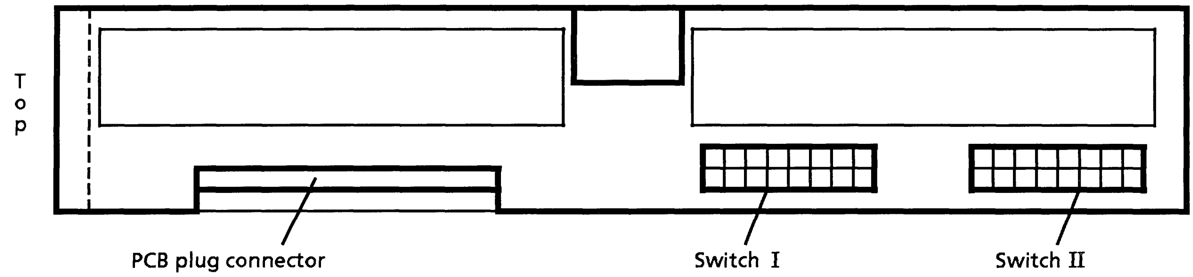
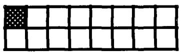
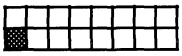
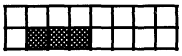
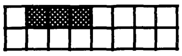
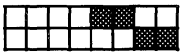
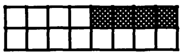
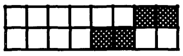
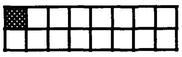
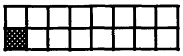
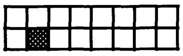
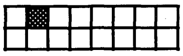
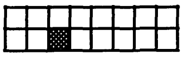
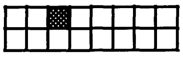
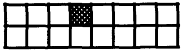
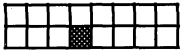
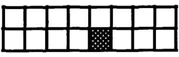
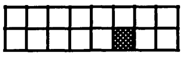
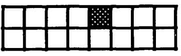
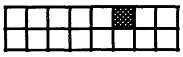
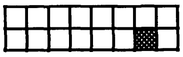
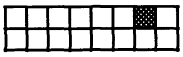


Figure 10.16 Position of the Function Select Switches of the 465-7LA13 Analog Input Module (Rear of the Module)

Table 10.5 Setting Functions on the 6ES5 465-7LA13 Module

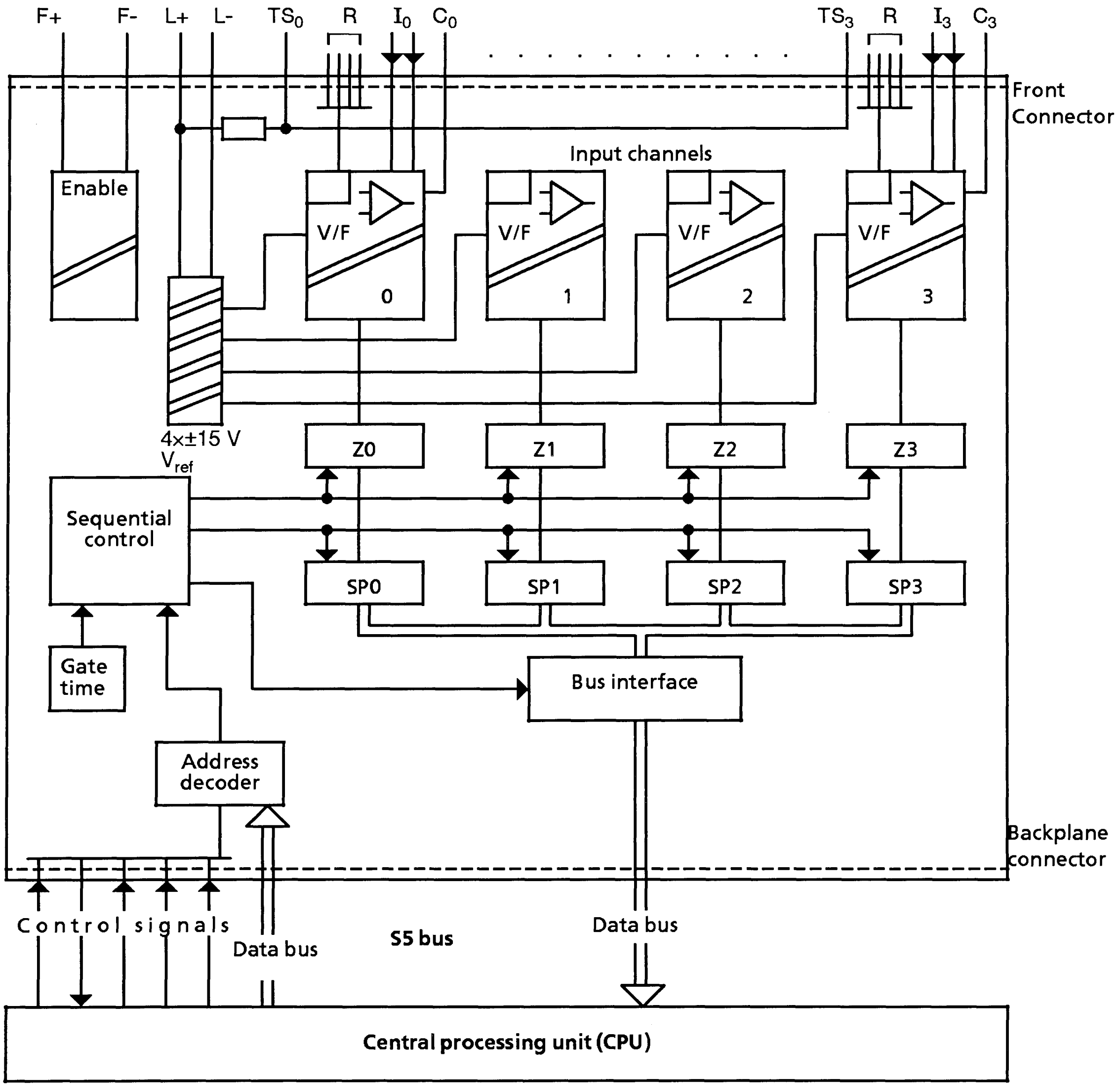
Function	Setting on Switch I		Setting on Switch II	
Reference junction compensation			Yes 	No 
Measuring range* (nominal value)			50 mV 	500 mV 
Measure with resistance therm., 4-wire/ 8-channel**				
Measure current or voltage			8 Channels 	16 Channels 
Sampling	Cyclic 	Selective 		
System frequency	50 Hz 	60 Hz 		
Channel operation	8 channels 	16 channels 		
Analog value representation	Abs. value and sign 	Two's complement 		
Wirebreak signal for 8 channels (16 channels)	Channel 0 to 3 (Channel 0 to 7) 	Channel 4 to 7 (Channel 8 to 15) 		
No wirebreak signal	Channel 0 to 3 (Channel 0 to 7) 	Channel 4 to 7 (Channel 8 to 15) 		
Monitor S+line to the PT 100 resistance therm. for wirebreak	... mV/... mA 	PT100 		

* Setting for PT 100: Measuring Range 500 mV

** Additional setting for PT 100: Reference junction compensation: No

10.5 463-4UA../-4UB.. Analog Input Module

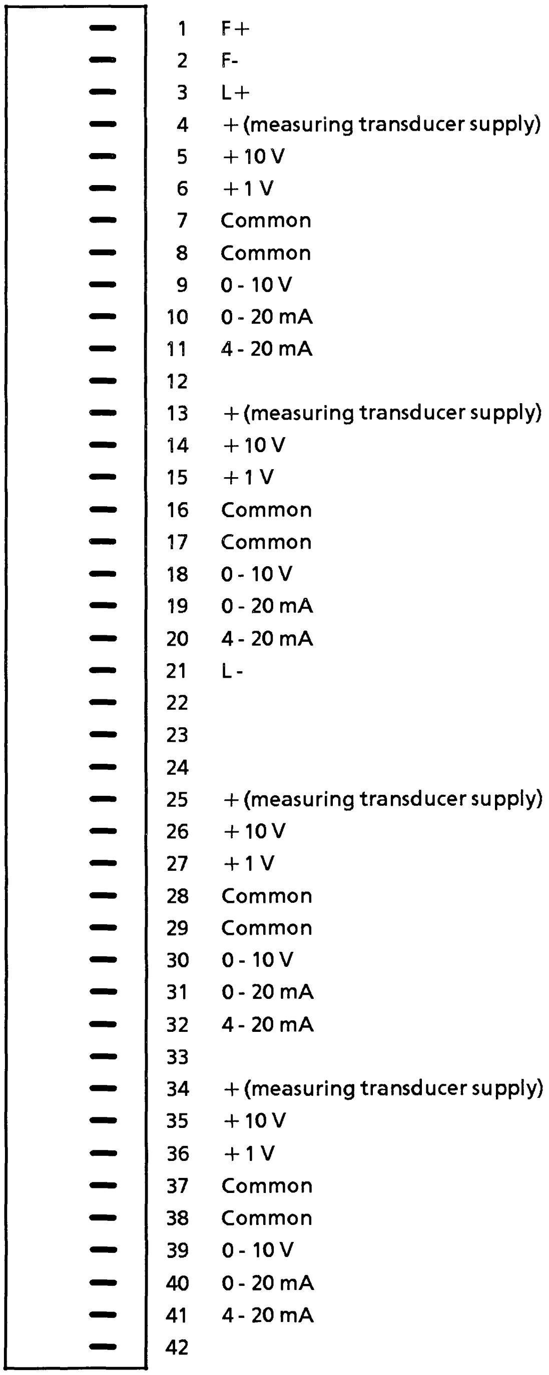
Figure 10.17 shows the block diagram of the 463-4U.. module.



C_0 to 3	Common Input (reference potential of inputs 0 to 3)	TS_0 to 3	Transducer Supply (supply for 2-wire MU of inputs 0 to 3)
I_0 to 3	Input (inputs 0 to 3)	V/F	Voltage-frequency converter
R	Range (measuring range)	Z	Counter
Sp	RAM		

Figure 10.17 Block Diagram with Signal Exchange between 463 Analog Input Module and CPU

10.5.1 Connection of Measuring Transducers to the 463-4UA../-4UB.. Analog Input Module



1	F+
2	F-
3	L+
4	+ (measuring transducer supply)
5	+ 10 V
6	+ 1 V
7	Common
8	Common
9	0 - 10 V
10	0 - 20 mA
11	4 - 20 mA
12	
13	+ (measuring transducer supply)
14	+ 10 V
15	+ 1 V
16	Common
17	Common
18	0 - 10 V
19	0 - 20 mA
20	4 - 20 mA
21	L -
22	
23	
24	
25	+ (measuring transducer supply)
26	+ 10 V
27	+ 1 V
28	Common
29	Common
30	0 - 10 V
31	0 - 20 mA
32	4 - 20 mA
33	
34	+ (measuring transducer supply)
35	+ 10 V
36	+ 1 V
37	Common
38	Common
39	0 - 10 V
40	0 - 20 mA
41	4 - 20 mA
42	

Figure 10.18 Pin Assignments of the 463 Analog Input Module

Pin Assignments of the Front Connector

The following figure shows

- the connection of the measuring transducers to the front connector
- the setting of the measuring range by plugging in jumpers at the front connector

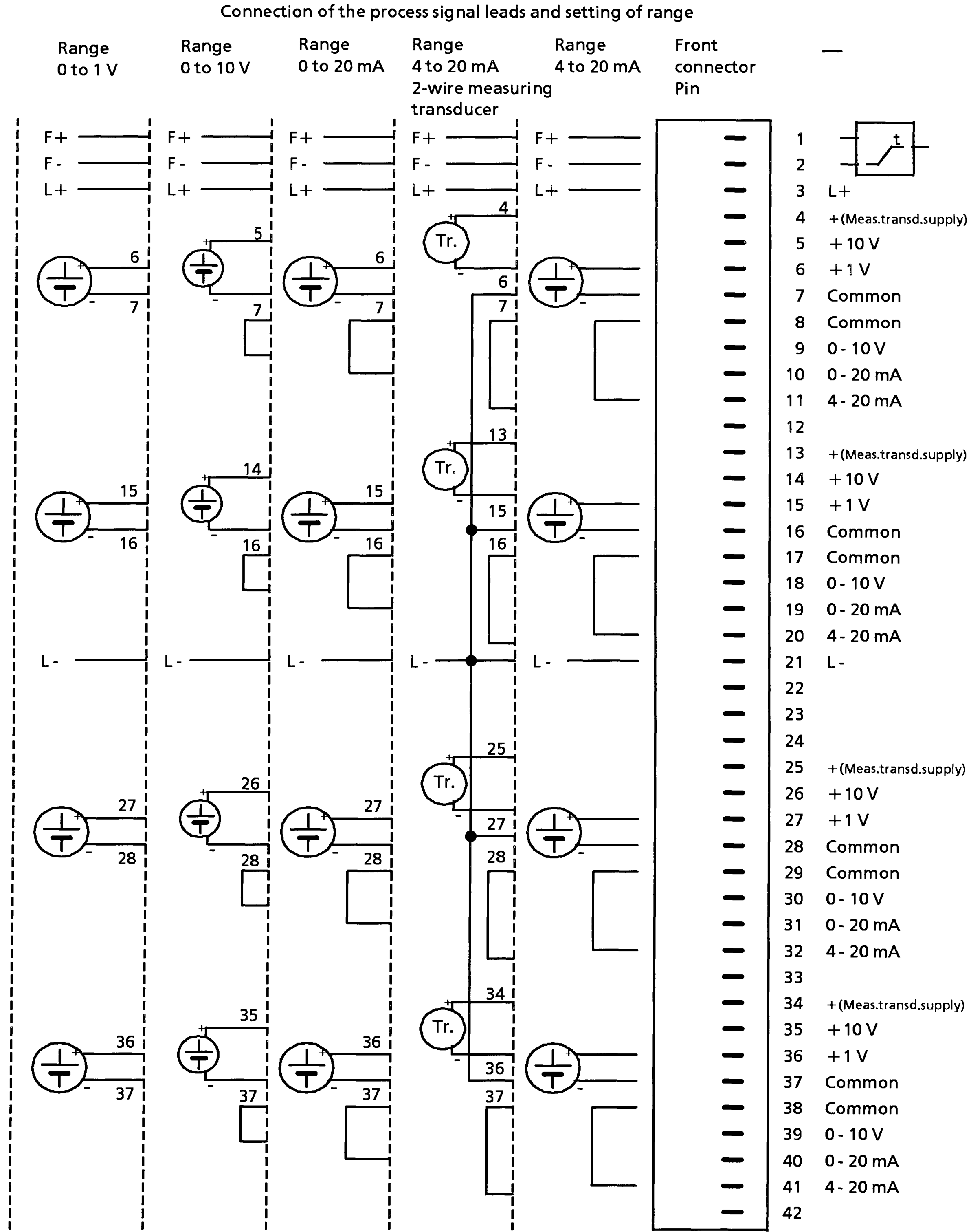


Figure 10.19 Connection of Measuring Transducers and Setting of Measuring Range on the 463 Analog Input Module

Connection of Measuring Transducers

The measuring transducers are connected to the analog input module via shielded cables of a maximum length of 200 m. Cable lengths of up to 500 m are possible, if the cables are not laid together with power cables.

Voltage sensors, current sensors, 2-wire and 4-wire measuring transducers can be connected in any configuration. Four short-circuit-proof supply connections are available on the front connector for 2-wire measuring transducers.

Note

If 2-wire measuring transducers are used, the reference potential (Common Input) of these channels has to be connected to L-. There is then no longer a galvanic isolation between the channels and the supply voltage L+/L-.

Please note that the module's bus interface is activated with 24 V via the F+ and F- enable lines at the front connector.

10.5.2 Startup of the 463-4UA../-4UB.. Analog Input Module

The individual measuring ranges are set by jumpering the inputs on the front connector (see Figure 10.19).

Two switches are provided on the module for setting

- the data format for the 4 to 20 mA range and
- the module addressing.

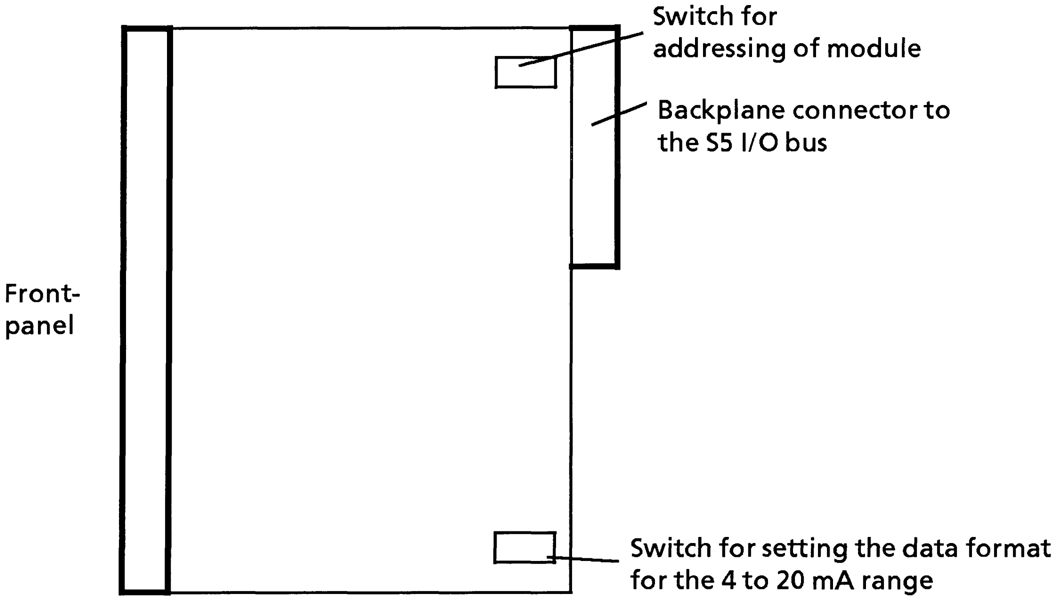

Switch for addressing of module
Backplane connector to the S5 I/O bus
Front-panel
Switch for setting the data format for the 4 to 20 mA range

Figure 10.20 Position of Switches on the 463 Analog Input Module

Note

An adapter casing (6ES5 491-0LB12) is required to use the 463 analog input module in the S5-115U programmable controller.
A 42-pin front connector is required as accessory;

- 6ES5 497-4UA22 for crimp connection
  or
- 6ES5 497-4UB12 for screw connection

Setting the Data Format for the 4 to 20 mA Range

If the 4 to 20 mA inputs are used, data representation from 0 to 1032 bits or 256 to 1279 bits can be selected by pressing the corresponding switch. All four input channels can have different data formats.

When the voltage or 0 to 20 mA inputs are used, the corresponding switches remain in the "OFF" position.

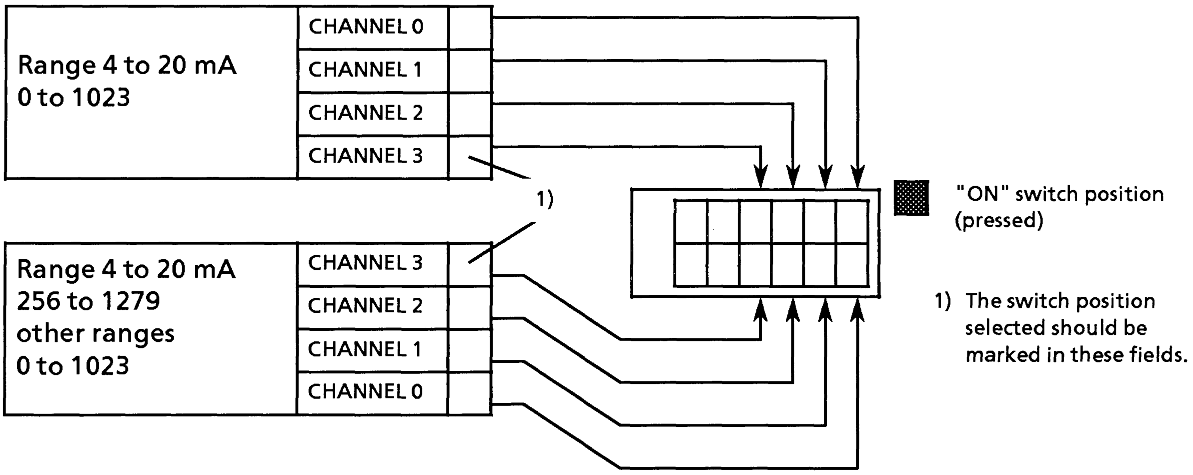


Figure 10.21 Labelling of Switch on the Cover of the 463 Analog Input Module

Addressing and Address Setting on the 463 Module

The analog input module uses an address space of 8 bytes.
The four 16-bit (2-byte) measured value memories can be scanned one after the other by the control program through word operations via the S5 bus. (L PW (module address + channel address x 2))

Table 10.6 shows the switch positions for the individual addresses selected. Whether address space is used in the O peripheral area or in the P area is set on the IM 314 (see section 3.3.2). "■" means switch position "ON" (pressed).

Table 10.6 Address Setting on the 463 Analog Input Module

Channel number						0	1	2	3
						Starting address			
ADB	7	6	5	4	3	PW	PW	PW	PW
Value	128	64	32	16	8				
Addressing switch	■					128	130	132	134
	■				■	136	138	140	142
	■			■		144	146	148	150
	■			■	■	152	154	156	158
	■		■			160	162	164	166
	■		■		■	168	170	172	174
	■		■	■		176	178	180	182
	■		■	■	■	184	186	188	190
	■	■				192	194	196	198
	■	■			■	200	202	204	206
	■	■		■		208	210	212	214
	■	■		■	■	216	218	220	222
	■	■	■			224	226	228	230
	■	■	■		■	232	234	236	238
	■	■	■	■		240	242	244	246
	■	■	■	■	■	248	250	252	254

10.6 466-3LA11 Analog Input Module

Figure 10-22 shows the block diagram of the 466-3LA11 module.

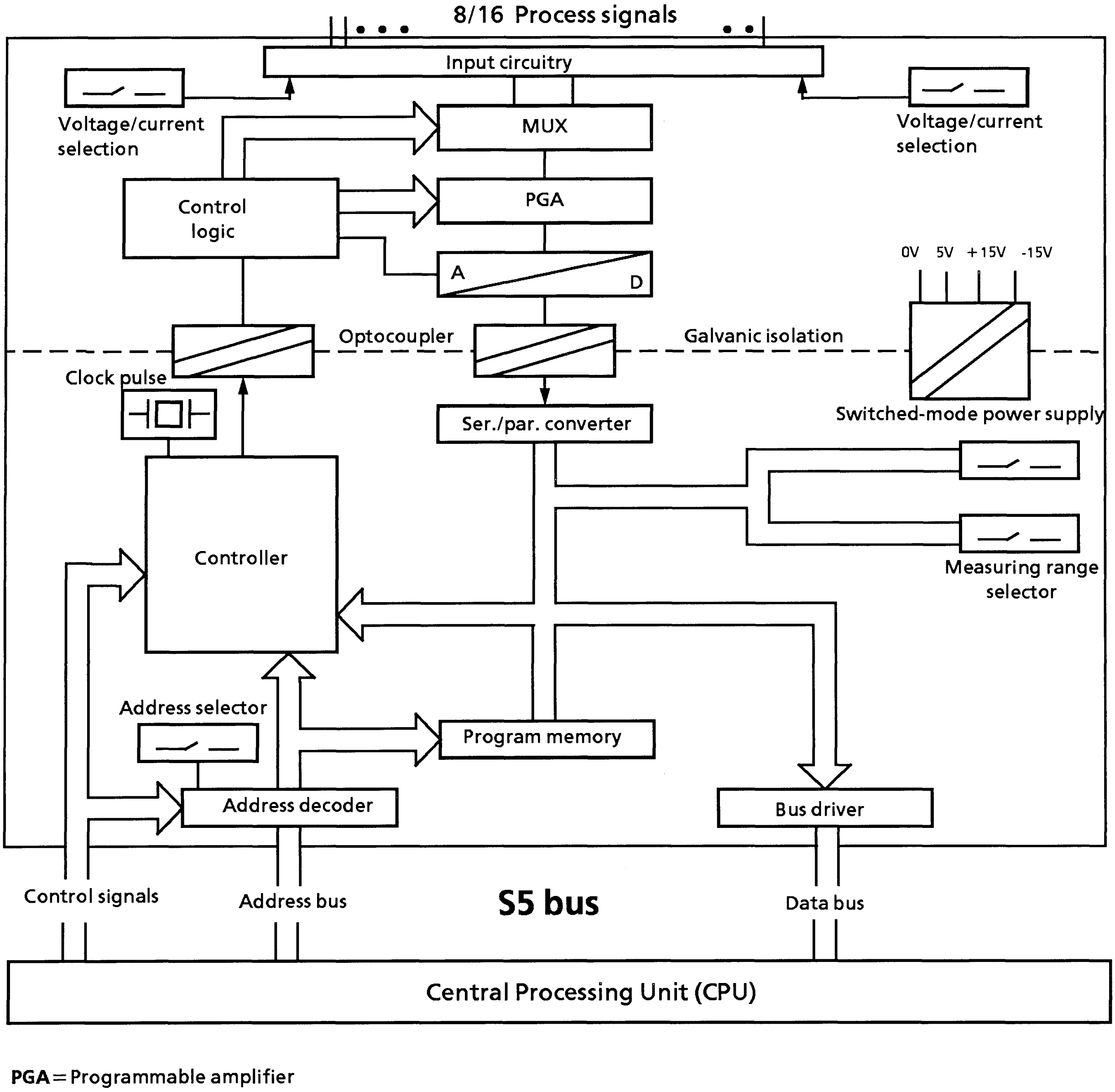


Figure 10-22 Block Diagram of the 466-3LA11 Analog Input Module

Note

Please note that the 466 module has very fast processing times. Since it is very fast, it is more suitable for closed-loop control tasks than for the connection of thermocouples and resistance thermometers.

10.6.1 Connecting Transducers to the 466-3LA11 Analog Input Module

The pin assignments of the 466-3LA11 analog input module depend on the type of measurement (common-reference measurement or differential measurement).

Common-reference Measurement

1	
2	M0+
3	M0-
4	M8-
5	M8+
6	
7	M1+
8	M1-
9	M9-
10	M9+
11	
12	M2+
13	M2-
14	M10-
15	M10+
16	
17	M3+
18	M3-
19	M11-
20	M11+
21	
22	
23	
24	M4+
25	M4-
26	M12-
27	M12+
28	
29	M5+
30	M5-
31	M13-
32	M13+
33	
34	M6+
35	M6-
36	M14-
37	M14+
38	
39	M7+
40	M7-
41	M15-
42	M15+
43	

In the case of common-reference measurement, all signal lines have a common reference point. The reference point is produced by running all M - inputs to one point (see Figure 10.24).
Since this type of measurement is susceptible to noise, the signal sources should be located close to the 466 analog input module.

There are 16 channels available; unused channels must be short-circuited (jumper between M+ and M -).

Labelling and Grouping of Channels
The channels are labelled as follows on the module:

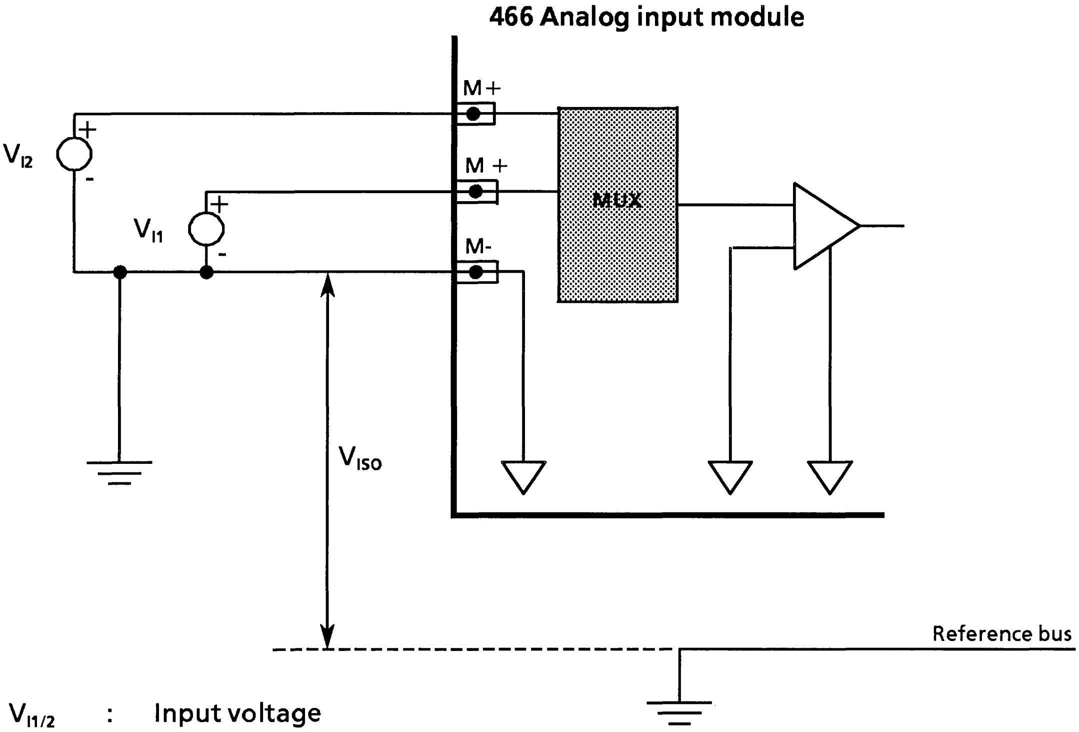
Channel 0:	M0+ M0 -
Channel 1:	M1+ M1 -
:	:
Channel 15:	M15+ M15 -

The channels are arranged in groups of four, for which separate measuring ranges can be set:

Channel group I:	Channel 0 to 3
Channel group II:	Channel 4 to 7
Channel group III:	Channel 8 to 11
Channel group IV:	Channel 12 to 15

Figure 10-23 Pin Assignments of the 466 Analog Input Module in the Case of Common-Reference Measurement

Figure 10.24 shows the connection of transducers to the 466 analog input module. All "M-" connection points are linked to each other internally on the module (this applies ***only*** to common-reference measurement!).



$V_{I1/2}$: Input voltage
V_{ISO} : Isolating voltage

▽ : Equipotential: this potential is determined by the sensor reference potential (external reference potential)

Figure 10.24 ***Connecting Transducers to the 466 Analog Input Module (Common-Reference Measurement)***

Note

See Sections 3.5.2 to 3.5.4 for further information on wiring, electrical structure and conductor arrangement.

Differential Measurement

Differential measurement is a method of measuring which compensates for noise on the line.

Each signal line is assigned its own signal reference line. By measuring the difference between the signal line and the signal reference line, noise on both lines is compensated for.

Unused channels also must be short-circuited when using this method of measuring (jumper between M+and M -).

Differential measurement is required in the following cases:

- When the sensors are connected to different supplies
- When different signal sources are physically separate
- When signals must be captured with high accuracy
- When a high level of noise is expected.

Labelling and Grouping of Channels

The channels are labelled as follows on the module:

Channel 0:	M0+
	M0-
Channel 1:	M1+
	M1-
:	:
Channel 7:	M7+
	M7-

The channels are arranged in groups of four, for which separate measuring ranges can be set:

Channel group I:	Group 0 to 3
Channel group II:	Group 4 to 7

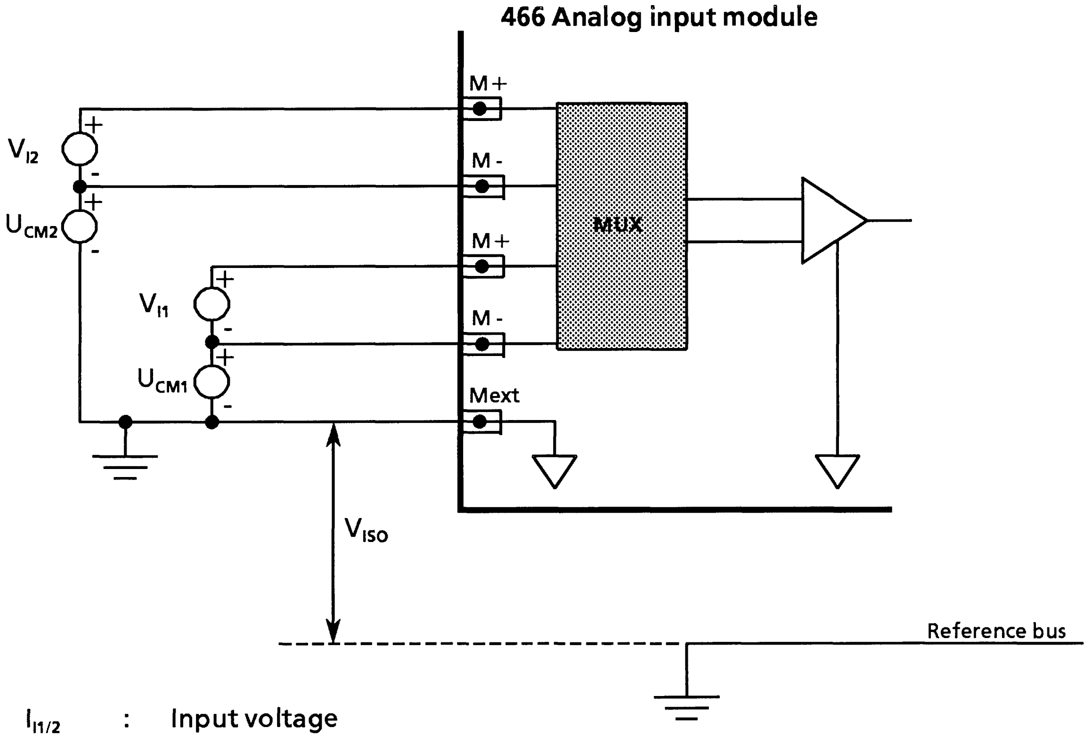
1	
2	M0+
3	M ext
4	M ext
5	M0-
6	
7	M1+
8	M ext
9	M ext
10	M1-
11	
12	M2+
13	M ext
14	M ext
15	M2-
16	
17	M3+
18	M ext
19	M ext
20	M3-
21	
22	
23	
24	M4+
25	M ext
26	M ext
27	M4-
28	
29	M5+
30	M ext
31	M ext
32	M5-
33	
34	M6+
35	M ext
36	M ext
37	M6-
38	
39	M7+
40	M ext
41	M ext
42	M7-
43	

Figure 10.25 Pin Assignments of the 466 Analog Input Module in the Case of Differential Measurement

Figure 10.26 shows the connection of transducers to the 466 analog input module.

When connecting transducers, you must take account of the following conditions:

$V_I + V_{CM} < 12V$ (i.e. the sum of the voltage measuring set and the common mode must be less than 12 V; current measuring ranges correspond to a voltage of 2.5 V)



$I_{I1/2}$: Input voltage

$I_{CM1/2}$: Common mode

I_{ISO} : Isolating voltage

▽ : Equipotential: this potential is determined by the sensor reference potential (external reference potential)

Figure 10.26 Connecting Transducers to the 466 Analog Input Module (Differential Measurement)

Note

See Sections 3.5.2 to 3.5.4 for further information on wiring, electrical structure and conductor arrangement.

10.6.2 Startup of the 466-3LA11 Analog Input Module

The operating mode of the 466 analog input module is set exclusively via switches on the the printed circuit board. Figure 10.27 shows the labelling and locations of the switches on the PCB.

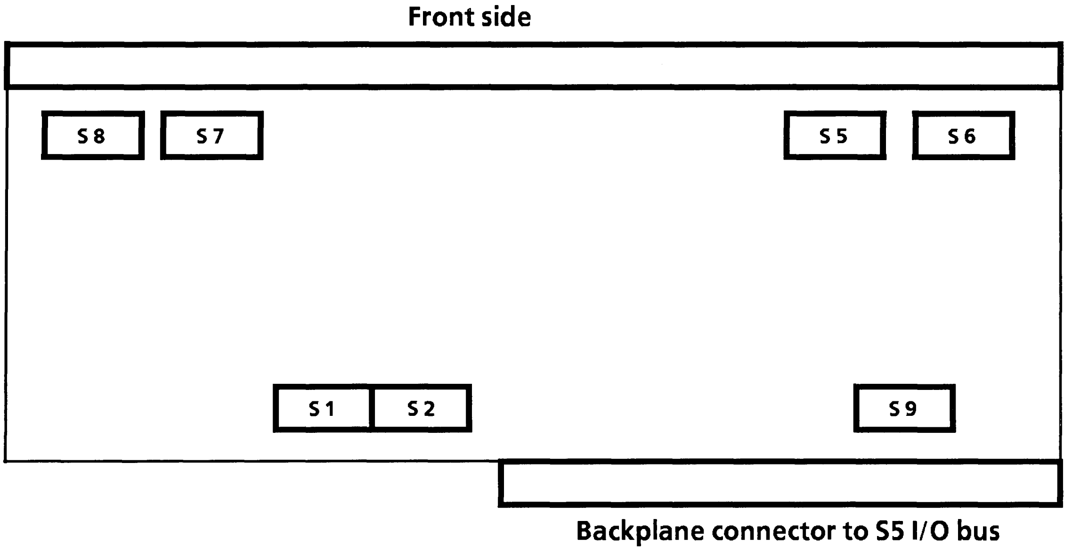


Figure 10.27 Locations of the Mode Selectors on the 466-3LA11 Analog Input Module

Note

An adapter casing (e.g. 6ES5 491-0LB12) is required for using the 466 analog input module in the S5-115U.
You also require a 43-pin front connector K;

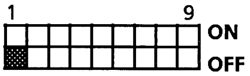
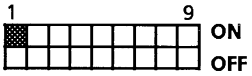
- 6XX3 068 for crimp connections
  or
- 6XX3 081 for screw connections.

Setting the Type of Measurement

Common-reference Measurement/Differential Measurement

Set switch **S 9** to the type of measurement (common-reference or differential). The switch positions refer to the module as represented in Figure 10.27:

Table 10.7 Setting the Type of Measurement (Common-Reference or Differential)

Type of Measurement	Switch Position S 9
Common-reference measurement	
Differential measurement	

Current/Voltage Measurement for Individual Channel Groups

If you have set ***differential measurement*** at Switch ***S 9***, there are two channel groups available to you, each with four channels. You can configure each channel group separately for current or voltage measurement. For this purpose, you must set the switches **S 5**, **S 6**, **S 7** and **S 8** (see Table 10.8 and 10.9). Switches S 5 and S 7 permit three settings (Left, Middle, Right); switches S 6 and S 8 permit two settings (Left, Right). The switch positions refer to the module as represented in Figure 10.27:

Table 10.8 Setting Current/Voltage Measurement for Channel Group I

Channel Group I (Channel 0 to 3)	Switch S 5	Switch S 6
Current	■□□	□■
Voltage	□■□	□■

Table 10.9 Setting Current/Voltage Measurement for Channel Group II

Channel Group II (Channel 4 to 7)	Switch S 7	Switch S 8
Current	■□□	□■
Voltage	□■□	□■

If you have set ***common-reference measurement*** at Switch **S 9**, there are four channel groups available to you, each with four channels. You can configure each channel group separately for current or voltage measurement. For this purpose, you must set the switches **S 5, S 6, S 7** and **S 8** **(see Table 10.10 to 10.13). Switches S 5 and S 7 permit three settings (Left, Middle, Right); switches S 6 and S 8 permit two settings (Left, Right). The switch positions refer to the module as represented in Figure 10.27:**

Table 10.10 Setting

Channel Group I (Channel 0 to 3)	Switch S 5
Current	□□■
Voltage	□■□

Table 10.11 Setting Current/Voltage Measurement for Channel Group II

Channel Group II (Channel 4 to 7)	Switch S 7
Current	□□■
Voltage	□■□

Table 10.12 Setting Current/Voltage Measurement for Channel Group III

Channel Group III (Channel 8 to 11)	Switch S 6
Current	■□
Voltage	□■

Table 10.13 Setting Current/Voltage Measurement for Channel Group IV

Channel Group IV (Channel 12 to 15)	Switch S 8
Current	■□
Voltage	□■

Setting the Measuring Range

The 466 analog input module has 12 measuring ranges. One measuring range can be selected for each channel group (i.e. for four inputs each), independently of the other channel groups.
Set the measuring ranges with switches S 1 and S 2. See Figure 10.28 for the assignment of switches to channel group.

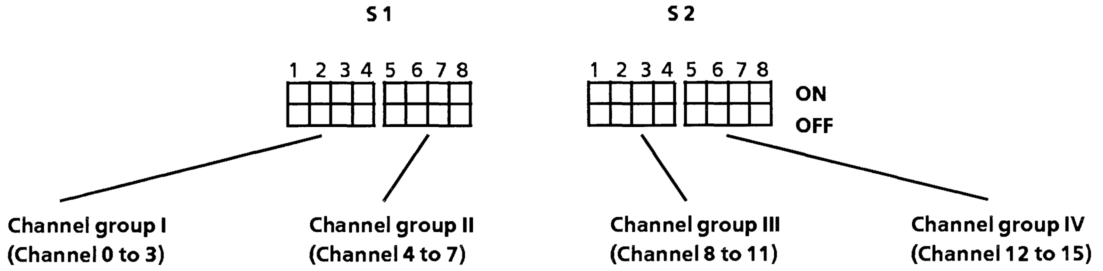
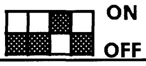
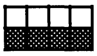
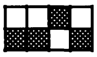
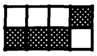
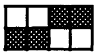
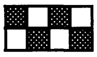
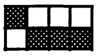
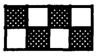
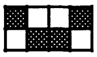
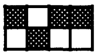
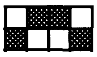
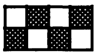


Figure 10.28 Assignment of Switches S 1/S 2 to Channel Group

The same measuring range coding applies to all channel groups. For this reason, the following table (see Table 10.14) contains only the measuring range setting for one channel group. The switch positions refer to the module as represented in Figure 10.27.
Please note that the type of measurement (current/voltage) must be set additionally with switches S 5 to S 8!

Table 10.14 Setting the Measuring Range for One Channel Group (4 Channels per Group)

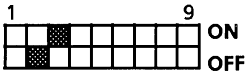
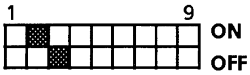
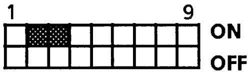
Measuring Range	Switch Positions
0 - 20 mA	
0 - 1.25 V	
0 - 2.5 V	
0 - 5 V	
0 - 10 V	
±20 mA	
±1.25 V	
±2.5 V	
±5 V	
±10 V	
4 - 20 mA	
1-5 V	

Setting the Data Format

The data format must be set with switch S 9:

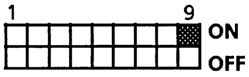
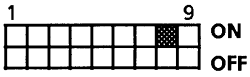
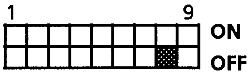
- Two's complement - 12-bit two's complement representation (range: 0 to 4095 units unipolar, or - 2048 to + 2047 units bipolar)
- Number with sign - 11-bit number and 1-bit sign (range: 0 to 4095 units unipolar, or - 2048 to + 2047 units bipolar)
- Binary - 12-bit binary number (range 0 to 4095 both for unipolar and bipolar variables)

Table 10.15 Setting the Data Format

Data Format	Switch Position S 9
Two's complement	
Number with sign	
Binary	

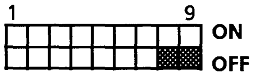
Setting the Connection Type and the Module Starting Address

Table 10.16 Setting the Connection Type and Address Range

466-3LA11 Module	Switch Position S 9
When operating in CC or EU over distributed connections with IM 304/314, 307/317, 308/318-3	
Address in I/O area (P area)	
Address in extended I/O area (Q area)*	

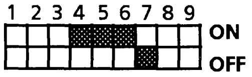
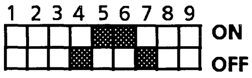
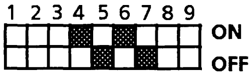
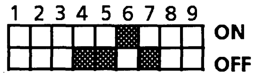
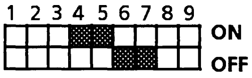
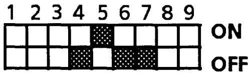
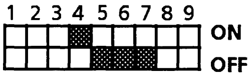
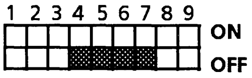
* Possible in expansion unit with distributed connections only

Table 10.16 Setting the Connection Type and Address Range (continued)

466-3LA11 Module	Switch Position S 9
When operating in distributed EU 701-2/3 with AS 301/310, only in I/O area (P area)	

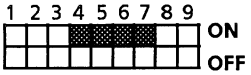
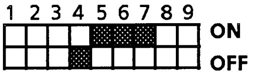
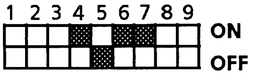
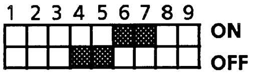
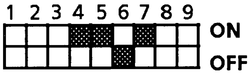
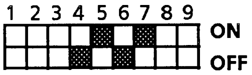
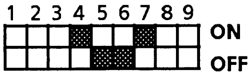
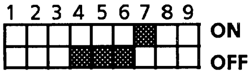
See Table 10.17 for the precise setting of the module starting addresses.

Table 10.17 Setting the Module Starting Addresses (P area)

Module Starting Address (P area)	Switch Position S 9
128 (F080$_H$)	
144* (F090$_H$)	
160 (F0A0$_H$)	
176* (F0B0$_H$)	
192 (F0C0$_H$)	
208* (F0D0$_H$)	
224 (F0E0$_H$)	
240* (F0F0$_H$)	

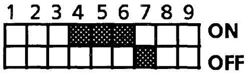
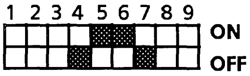
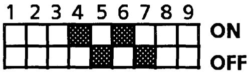
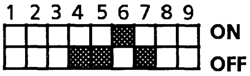
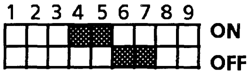
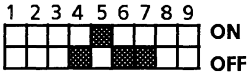
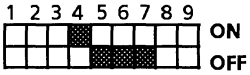
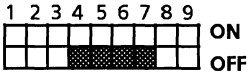
* Can only be set in the case of differential measurement

Table 10.18 Setting the Module Starting Addresses (Q area)

Module Address (Q area)	Switch position S 9
000 (F100$_H$)	
016* (F110$_H$)	
032 (F120$_H$)	
048* (F130$_H$)	
064 (F140$_H$)	
080* (F150$_H$)	
096 (F160$_H$)	
112* (F170$_H$)	

* Can only be set in the case of differential measurement

Table 10.18 Setting the Module Starting Addresses (Q area) (continued)

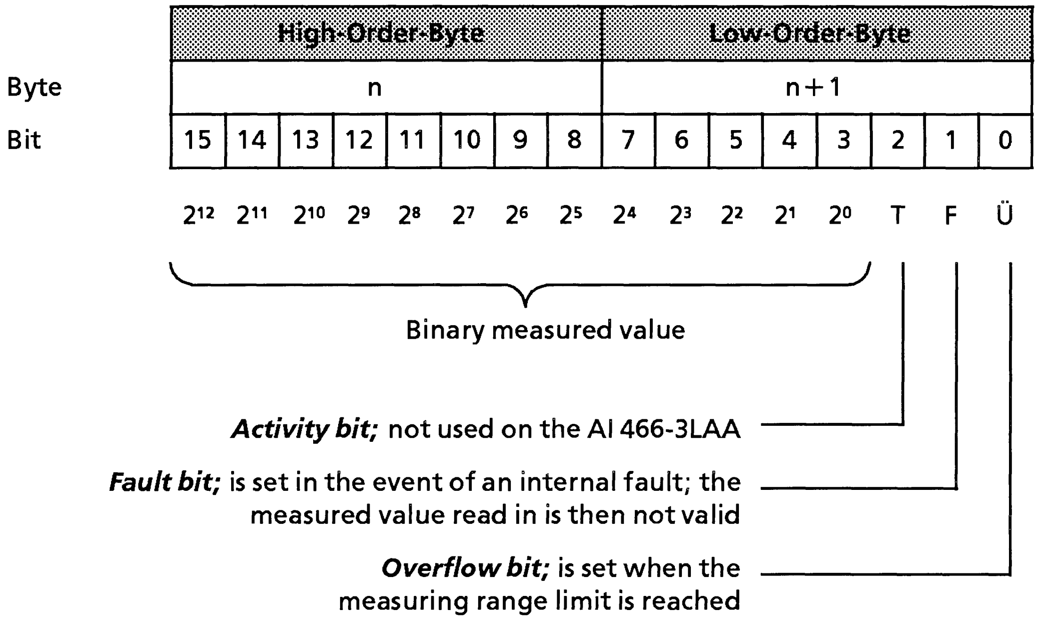
Module Address (Q area)	Switch Position S 9
128 (F180$_H$)	
144* (F190$_H$)	
160 (F1A0$_H$)	
176* (F1B0$_H$)	
192 (F1C0$_H$)	
208* (F1D0$_H$)	
224 (F1E0$_H$)	
240* (F1F0$_H$)	

* Can only be set in the case of differential measurement

10.7 Representation of the Digital Input Value

The analog value has the same representation in the three analog input modules.
However, there are differences in the case of analog value evaluation where the individual analog input modules are concerned, especially bits 0 to 2 (see Figure 10.29).

After an analog signal is converted, the digital result is stored in the module's RAM. Figure 10.29 explains the individual bits of the two bytes.



	High-Order-Byte								Low-Order-Byte							
Byte	n								n+1							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	2^{12}	2^{11}	2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	T	F	Ü

Binary measured value

Activity bit; not used on the AI 466-3LAA

Fault bit; is set in the event of an internal fault; the measured value read in is then not valid

Overflow bit; is set when the measuring range limit is reached

Figure 10.29 Representation of the Digitized Measured Value

Bits 0 to 2 are irrelevant for the measured value. They provide information on the measured value representation. Table 10.19 describes these bits in detail.

Table 10.19 Meaning of Bits 0 to 2 for Analog Input Modules

Bit	Meaning	Signal State	Meaning of the Signal State
Ü	Overflow bit	1	Range exceeded*
F	Fault bit	1	Wire break
T	Activity bit	0	Cyclic scan or "Not active" (for single scan)
		1	Coding procedure for single scan not yet terminated

* An overflow at one measuring point has no effect on the overflow bits of the other channels, i.e. the values on the other channels are correct and may be evaluated.

Special Features of the 466 Module

- Bit 15 (2^{12}) indicates the sign in the case of bipolar measured value representation (two's complement and number with sign).
- Bit 14 (2^{11}) is not used in the case of bipolar measured value representation (no overrange!).
- The 466 module has no overrange.
- Selective sampling is not possible on the 466 module (activity bit is not set).

10.7.1 Types of Representation of the Digital Input Value for the 460 and 465 Analog Input Modules

The way in which the analog value is represented depends on the type of module (see Tables 10.20 to 10.25).

Table 10.20 Representation of Digitized Measured Values of the 460 and 465 AI (Two's Complement; Measuring Range ± 50 mV, ± 500 mV, ± 1000 mV)

Meas. Value in mV (±50)	Meas. Value in mV (±500)	Meas. Value in mV (±1000)	Units	Digitized Measured Value													T	F	OV	Range
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
100.000	1000.00	2000.00	4095+OV	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	Overflow
99.976	999.76	1999.52	4095	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	Overrange
:	:	:	:							:										
50.024	500.24	1000.48	2049	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	
50.000	500.00	1000.00	2048	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Nominal range
49.976	499.76	999.52	2047	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	
:	:	:	:							:										
25.000	250.00	500.00	1024	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
24.976	249.76	499.52	1023	0	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	
:	:	:	:							:										
0.024	0.24	0.48	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	
0.000	0.00	0.00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
-0.024	-0.24	-0.48	-1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	
:	:	:	:							:										
-24.976	-249.76	-499.52	-1023	1	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	
-25.000	-250.00	-500.00	-1024	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
:	:	:	:							:										
-49.976	-499.76	-999.52	-2047	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	
-50.000	-500.00	-1000.00	-2048	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
-50.024	-500.24	-1000.48	-2049	1	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	Overrange
:	:	:	:							:										
-99.976	-999.76	-1999.52	-4095	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	
-100.000	-1000.00	-2000.00	-4095+OV	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	Overflow

Table 10.21 Representation of Digitized Measured Values of the AI 460 and 465 (Two's Complement; Measuring Range ± 5 V, ± 10 V, ± 20 mA)

Meas. Value	Meas. Value	Meas. Value	Units	Digitized Measured Value													T	F	OV	Range
in V (± 5)	in V (± 50)	in mA (± 20)		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
10.0000	20.0000	40.0000	4095+OV	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	Overflow
9.9976	19.9952	39.9902	4095	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	Overrange
:	:	:	:							:										
5.0024	10.0048	20.0098	2049	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	
5.0000	10.0000	20.0000	2048	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Nominal range
4.9976	9.9952	19.9902	2047	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	
:	:	:	:							:										
2.5000	5.0000	10.0000	1024	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
2.4976	4.9952	9.9902	1023	0	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	
:	:	:	:							:										
0.0024	0.0048	0.0098	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	
0.0000	0.0000	0.0000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
-0.0024	-0.0048	-0.0098	-1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	
:	:	:	:							:										
-2.4976	-4.9952	-9.9902	-1023	1	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	
-2.5000	-5.0000	-10.0000	-1024	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
:	:	:	:							:										
-4.9976	-9.9952	-19.9902	-2047	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	
-5.0000	-10.0000	-20.0000	-2048	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
-5.0024	-10.0048	-20.0098	-2049	1	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	Overrange
:	:	:	:							:										
-9.9976	-19.9952	-39.9902	-4095	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	
-10.0000	-20.0000	-40.0000	-4095+OV	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	Overflow

Table 10.22 Representation of Digitized Measured Values of the 460 and 465 AI (Number and Sign; Mesuring Range ± 50 mV, ± 500 mV, ± 1000 mV)

Meas. Value in mV (±50)	Meas. Value in mV (±500)	Meas. Value in mV (±1000)	Units	Digitized Measured Value													T	F	OV	Range
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
100.000	1000.00	2000.00	4095+OV	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	Overflow
99.976	999.76	1999.52	4095	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	Overrange
:	:	:	:							:										
50.024	500.24	1000.48	2049	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	
50.000	500.00	1000.00	2048	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Nominal range
49.976	499.76	999.52	2047	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	
:	:	:	:							:										
25.000	250.00	500.00	1024	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
24.976	249.76	499.52	1023	0	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	
:	:	:	:							:										
0.0024	0.24	0.48	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	
0.0000	0.00	0.00	+0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0.0000	0.00	0.00	-0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
-0.0024	-0.24	-0.48	-1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	
:	:	:	:							:										
-24.976	-249.76	-499.52	-1023	1	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	
-25.000	-250.00	-500.00	-1024	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
:	:	:	:							:										
-49.976	-499.76	-999.52	-2047	1	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	
-50.000	-500.00	-1000.00	-2048	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
-50.024	-500.24	-1000.48	-2049	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	Overrange
:	:	:	:							:										
-99.976	-999.76	-1999.52	-4095	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	
-100.000	-1000.00	-2000.00	-4095+OV	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	Overflow

Note

Bit 7 in the high-order byte is the sign (S).
If S is 0, the value is positive. If S is 1, the value is negative.

Table 10.23 Representation of Digitized Measured Values of the 460 and 465 AI (Number and Sign; Measuring Range ± 5 V, ± 10 V, ± 20 mA)

Meas. Value	Meas. Value	Meas. Value	Units	Digitized Measured Value													T	F	OV	Range
in V (±5)	in V (±20)	in V (±10)		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
10.0000	20.0000	40.0000	4095+OV	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	Overflow
9.9976	19.9952	39.9902	4095	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	Overrange
:	:	:	:							:										
5.0024	10.0048	20.0098	2049	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	
5.0000	10.0000	20.0000	2048	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Nominal range
4.9976	9.9952	19.9902	2047	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	
:	:	:	:							:										
2.5000	5.0000	10.0000	1024	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
2.4976	4.9952	9.9902	1023	0	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	
:	:	:	:							:										
0.0024	0.0048	0.0098	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	
0.0000	0.0000	0.0000	+0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0.0000	0.0000	0.0000	-0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
-0.0024	-0.0048	-0.0098	-1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	
:	:	:	:							:										
-2.4976	-4.9952	-9.9902	-1023	1	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	
-2.5000	-5.0000	-10.0000	-1024	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
:	:	:	:							:										
-4.9976	-9.9952	-19.9902	-2047	1	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	
-5.0000	-10.0000	-20.0000	-2048	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
-5.0024	-10.0048	-20.0098	-2049	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	Overrange
:	:	:	:							:										
-9.9976	-19.9952	-39.9902	-4095	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	
-10.0000	-20.0000	-40.0000	-4095+OV	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	Overflow

Set the measuring range of the module to 500 mV and plug in a 6ES5 498-1AA 71 module.

The measuring range 4 to 20 mA is resolved into 2048 units from 512 to 2560. For representation in the range 0 to 2048, 512 units must be subtracted at the software level.

Table 10.24 Representation of Digitized Measured Values of the 460 and 465 AI (Current Measuring Range 4 to 20 mA)

I_E in mA	Units	V_I in mV	Digitized Measured Value													T	F	OV	Range
			15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
32.796	4096+	1024.00	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	Overflow
31.992	4095	1023.76	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	Overrange*
24.000	3072	750.00	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
23.992	3071	749.76	0	1	0	1	1	1	1	1	1	1	1	1	1	0	0	0	
20.008	2561	625.24	0	1	0	1	0	0	0	0	0	0	0	0	1	0	0	0	
20.000	2560	625.00	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	Nominal range
16.000	2048	500.00	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
4.000	512	125.00	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	
3.992	511	124.76	0	0	0	0	1	1	1	1	1	1	1	1	1	0	0	0	Range violation
3.000	384	93.75	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	
2.992	383	93.50	0	0	0	0	1	0	1	1	1	1	1	1	1	0	0	0	
0.000	0	0.00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Overflow

* Short-circuit of the two-wire transducer

Note

The 31.25 Ω shunt resistor integrated in the 498-1AA71 suppresses the wire break signal (the F bit is not set). You can thus detect a wire break only by comparing the measured value with a lower limiting value in the user program. A measured value lower than, for example, 1 mA (= 128 units) would then be interpreted as a wire break.

Table 10.25 Representation of Digitized Measured Values of the 460 and 465 AI for Resistance-Type Sensors

Sensor Resistance (Ω)	Units	Digitized Measured Value													T	F	OV	Range
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
400.000	4095	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	Overflow
399.900	4095	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	Overrange
200.098	2049	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	
200.000	2048	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Nominal range
199.900	2047	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	
100.000	1024	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
99.900	1023	0	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	
0.098	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	
0.000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

The resolution in the case of the PT 100 is approximately 1/3 °C 10 units correspond to approximately 1 Ω.

You can use the assignment in Figure 10.30 for the PT 100 resistance sensor.

Linearization of the digital input values is not carried out via the modules. You can linearize the input values only via the relevant software solution.

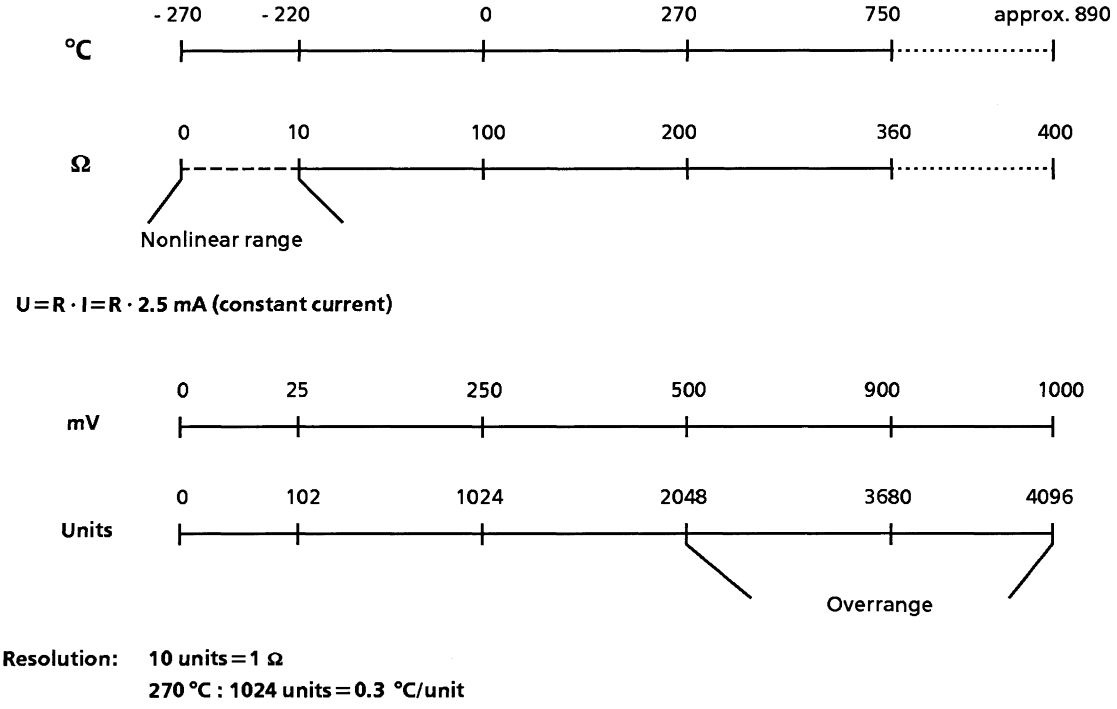


Figure 10.30 PT 100 on SIMATIC Analog Input Modules

Representation of Measured Value for the New PT100 Climatic Measuring Range of the 460-7LA13 AI

Table 10.26 Representation of Digitized Measured Values for the PT100 Climatic Measuring Range of the 460-7LA13 AI

Units	PT100/ ohm	°C	mV at PT100	mV compensated	Digitized Measured Value													T	F	OV	Range
					15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
>4095	140		350	100	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	Overflow
+4095	139.99	103.74	349.976	99.976	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	Overrange
+2049	120.01	51.61	300.024	50.024	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	
+2048	120	51.580	300	50	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Nominal range
+1	100.01	0.026	250.024	0.024	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	
0	100	0	250	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
-1	99.99	-0.026	249.976	-0.024	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	
-2048	80	-50.780	200	-50	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
-2049	79.99	-50.81	199.976	-50.024	1	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	Overrange
-4095	60.01	-100.60	150.024	-99.976	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	
<-4095	60		150	-100	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	Overflow
Wire break																					
-4095	x	x	0	-250	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	Wire break I_{C+}/I_{C-} 2)
-4095	x	x	0	-250	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	Wire break transducer measuring circuit 1)

T: Active F: Error bit OV: Overflow X: Any

1) Only with wire break detection activated
Error bit = 1 only in case of faulty channel;
In case of transducer break overflow bit = 1 for all channels

2) Through the PT100 series connection, this bit combination is set for all channels in the case of a break in the supply circuit.

10.7.2 Types of Representation of the Digital Input Value for the 463 Analog Input Module

With the analog input, the analog value is represented in the two's complement with the following data format.

	High-Order Byte								Low-Order Byte							
Byte	n								n+ 1							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	2^{11}	2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	-	-	-	OV

Binary measured value

Overflow bit (is set when the measuring range limit is reached)

Figure 10.31 Representation of Digitized Measured Values of the 463 Analog Input Module

Table 10.27 Representation of Digitized Measured Values of the 463 AI (Two's Complement; Measuring Range 0 to 10 V, 0 to 1 V, 0 to 20 mA, 4 to 20 mA)

Nominal Range				Digitized Measured Value												n. b.			ov	Range
0 to 10 V	0 to 1000 m V	0 to 20 mA	Units	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
19.99	1999	39.98	2047	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1	Overflow
15.00	1500	30.00	1536	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	
14.99	1499	29.98	1535	0	1	0	1	1	1	1	1	1	1	1	1	0	0	0	0	Overrange
12.50	1250	25.00	1280	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	
10.01	1001	20.02	1025	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	
10.00	1000	20.00	1024	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Nominal Range
9.99	999.02	19.98	1023	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0	
5.00	500.00	10.00	512	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
4.99	499.00	9.98	511	0	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0	
2.5	250.00	5.00	256	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	
0.098	0.976	0.02	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
-0.098	-0.976	-0.02	-1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	
-0.5	-50	-1	-51	1	1	1	1	1	1	0	0	1	1	0	1	0	0	0	0	

n. b. Unused

Table 10.27 Representation of Digitized Measured Values of the 463 AI (Two's Complement; Measuring Range 0 to 10 V, 0 to 1 V, 0 to 20 mA, 4 to 20 mA) (continued)

Nominal Range			Digitized Measured Value												n.b.			ov	Range
4 to 20 mA*	4 to 20 mA**	Units	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
35.98	31.98	2047	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1	Overflow
28.00	24.00	1536	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	
27.98	23.98	1535	0	1	0	1	1	1	1	1	1	1	1	1	0	0	0	0	Overrange
24.00	20.00	1280	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	
20.02	16.02	1025	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	
20.00	16.00	1024	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Nominal Range
19.98	15.98	1023	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0	
12.00	8.00	512	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
11.98	7.98	511	0	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0	
	4.00	256	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	
4.016	0.016	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
3.984	-0.016	-1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	
3.184	-0.816	-51	1	1	1	1	1	1	0	0	1	1	0	1	0	0	0	0	

n.b. Unused

* With data format setting 0 to 1023 (via switch on the module)

** With data format setting 256 to 1279 (switch on the module)

A shunt resistance of 50 ohms is used for the 0 to 20 mA measuring range; for the 4 to 20 mA range, the resistance is 62.5 ohms.

The shunt resistors are permanently installed on the 463 analog input module. Wire break detection is in principle not possible; for the 4 to 20 mA current measuring range, a wire break can be detected for currents < 3 mA.

If 2-wire measuring transducers (4 to 20 mA) are used and their plus and minus terminals are short-circuited, the current is limited to approximately 28 mA. Until activation of the thermal current limiter (approx. 3 s), a short-circuit current of approx. 250 mA flows, which sets the overflow bit on the short-circuited channel during the 3-s period.

10.7.3 Forms of Representation of the Digital Input Values for the 466 Analog Input Module

Tables 10.28 to 10.36 give information on the representation of the digitized measured value depending on the measuring range selected.
The 466 analog input module has no overrange.

Table 10.28 Representation of Digitized Measured Values of the 466 AI (Measuring Range 0 to 20 mA; 0 to 5 V and 0 to 10 V; unipolar)

Meas. Value in V (0 to 5 V)	Meas. Value in V (0 to 10 V)	Meas. Value in mA (0 to 20 mA)	Units	Digitized Measured Value*													T	F	OV
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
4.9988	9.9976	19.9951	4095	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1
4.9976	9.9951	19.9902	4094	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
:	:	:	:						:									:	
0.0012	0.0024	0.00488	0001	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0.0000	0.0000	0.00000	0000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

* Same representation as for two's complement, number and sign and binary representation

Table10.29 Representation of Digitized Measured Values (Two's Complement; Measuring Range ± 5 V, ± 20 mA and ± 10 V; bipolar)

Meas. Value in V (±5 V)	Meas. Value in V (±10 V)	Meas. Value in mA (±20 mA)	Units	Digitized Measured Value													T	F	OV
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
4.9976	9.9951	19.99020	2047	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	1
4.9951	9.9902	19.98040	2046	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0
:	:	:	:						:									:	
0.0024	0.0049	0.00976	0001	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0.0000	0.0000	0.00000	0000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-0.0024	-0.0049	-0.00976	-0001	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
:	:	:	:						:									:	
-4.9976	-9.9951	-19.99020	-2047	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0
-5.0000	-10.0000	-20.00000	-2048	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Table 10.30 Representation of Digitized Measured Values (Number and Sign; Measuring Range ± 5 V, ± 20 mA and ± 10 V; bipolar)

Meas. Value in V (±5 V)	Meas. Value in V (±10 V)	Meas. Value in mA (±20 mA)	Units	Digitized Measured Value													T	F	OV
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
4.9976	9.9951	19.99020	2047	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	1
4.9951	9.9902	19.98040	2046	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0
:	:	:	:						:									:	
0.0024	0.0049	0.00976	0001	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0.0000	0.0000	0.00000	0000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-0.0024	-0.0049	-0.00976	-0001	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
:	:	:	:						:									:	
-4.9976	-9.9951	-19.99020	-2047	1	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0
-5.0000	-10.0000	-20.00000	-2048	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Table 10.31 Representation of Digitized Measured Values (*Binary*; Measuring Range ± 5 V, ± 20 mA and ± 10 V; bipolar)

Meas. Value in V (±5 V)	Meas. Value in V (±10 V)	Meas. Value in mA (±20 mA)	Units	Digitized Measured Value													T	F	OV
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
4.9976	9.9951	19.99020	4095	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1
4.9951	9.9902	19.98040	4094	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
:	:	:	:							:								:	
0.0024	0.0049	0.00976	2049	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0.0000	0.0000	0.00000	2048	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-0.0024	-0.0049	-0.00976	2047	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0
:	:	:	:							:								:	
-4.9976	-9.9951	-19.99020	0001	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
-5.0000	-10.0000	-20.00000	0000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Table 10.32 Representation of Digitized Measured Values (Measuring Range 0 to 1.25 V, and 0 to 2.5 V; unipolar)

Meas. Value in V (0 to 1.25 V)	Meas. Value in V (0 to 2.5 V)	Units	Digitized Measured Value*													T	F	OV
			15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1.2497	2.4994	4095	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1
1.2494	2.4988	4094	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
:	:	:							:								:	
0.0003	0.0006	0001	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0.0000	0.0000	0000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

* Same representation as for two's complement data format, number and sign and binary representation

Table 10.33 Representation of Digitized Measured Values (Two's Complement; Measuring Range ± 1.25 V, and ± 2.5 V; bipolar)

Meas. Value in V (±1.25 V)	Meas. Value in V (±2.5 V)	Units	Digitized Measured Value													T	F	OV
			15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1.2494	2.4988	2047	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	1
1.2488	2.4975	2046	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0
:	:	:							:								:	
0.0006	0.0012	0001	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0.0000	0.0000	0000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
-0.0006	-0.0012	-0001	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
:	:	:							:								:	
-1.2494	-2.4988	-2047	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0
-1.2500	-2.5000	-2048	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Table 10.34 Representation of Digitized Measured Values (Number and Sign; Measuring Range ± 1.25 V, and ± 2.5 V; bipolar)

Meas. Value in V (±1.25 V)	Meas. Value in V (±2.5 V)	Units	Digitized Measured Value													T	F	OV
			15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1.2494	2.4988	2047	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	1
1.2488	2.4975	2046	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0
:	:	:							:								:	
0.0006	0.0012	0001	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0.0000	0.0000	0000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-0.0006	-0.0012	-0001	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
:	:	:							:								:	
-1.2494	-2.4988	-2047	1	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0
-1.2500	-2.5000	-2048	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Table 10.35 Representation of Digitized Measured Values (Binary; Measuring Range ± 1.25 V, and ± 2.5 V; bipolar)

Meas. Value in V (±1.25 V)	Meas. Value in V (±2.5 V)	Units	Digitized Measured Value													T	F	OV
			15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1.2494	2.4988	4095	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1
1.2488	2.4975	4094	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
:	:	:							:								:	
0.0006	0.0012	2049	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0.0000	0.0000	2048	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-0.0006	-0.0012	2047	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0
:	:	:							:								:	
-1.2494	-2.4988	0001	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
-1.2500	-2.5000	0000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Table 10.36 Representation of Digitized Measured Values (Measuring Range 4 to 20 mA and 1 to 5 V)

Meas. Value in V (1 to 5 V)	Meas. Value in mA (4 to 20 mA)	Units	Digitized Measured Value*													T	F	OV
			15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
4.998	19.992	2559	0	1	0	0	1	1	1	1	1	1	1	1	1	0	0	1
4.000	16.000	2048	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1.000	4.000	512	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0.998	3.992	511	0	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0
0.750	3.000	384	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0
0.748	2.992	383	0	0	0	0	1	0	1	1	1	1	1	1	1	0	0	0
0.000	0.000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

* Same representation as for two's complement data format, number and sign and binary representation

The measuring ranges 4 to 20 mA and 1 to 5 V (see Table 10-31) are resolved to 2048 units in the interval 512 to 2560. For representation in the range 0 to 2048, 512 units must be subtracted per software.

A wirebreak signal is not provided. You can scan the measured value in the user program for a lower limit and interpret values below this limit as wirebreak.

10.8 Wirebreak Signal and Sampling for Analog Input Modules

Wirebreak Signal

Wirebreak is signalled **only** in the case of the **460** and **465** analog input modules.

If a 6ES5 498-1AA11 range card (through-connection card) is used, you can select the "Wirebreak signal" function to monitor the sensors connected to the inputs (see Tables 10.1 to 10.5). You can select wirebreak detection for 8 or 16 inputs for 16-channel operation or for 4/8 inputs for 8-channel operation.

The wirebreak signal is issued under the following conditions:
Before each input value is decoded, a constant current is applied briefly (1.6 ms) to the input terminals and the resulting voltage compared with a limiting value. If the sensor circuit or supply lead is interrupted, the voltage exceeds the limiting value and a wirebreak signal is generated (bit 1 is set in data byte 1; refer to Section 10.5.1). The ADC decodes the value "0".

When the signal at the input is measured with a digital voltmeter, the constant-current pulses may cause apparent fluctuations in the signal. When the input circuit that supplies the analog value has capacitive characteristics, the constant current falsifies the measured value.

Should these apparent fluctuations in the signal prove annoying, e.g. on startup, the test current can be deactivated on the 460-7LA12 and 465-7LA12 analog input modules by applying + 24 V to pin 26 in the front connector and 0 V to pin 47 (L-) of the 460-7LA12 module or to pin 37 (M_{ext}) of the 465-7LA13 module. In addition, mode selector I must be set to "No wirebreak signal".

A wirebreak signal serves a practical purpose only in conjunction with a 6ES5 498-1AA11 through connection card. It is not possible to detect a wirebreak on the 6ES5 498-1AA41, -1AA51 or -1AA71 range cards, as the measuring inputs are terminated with low-resistance shunts. On all other range cards, a wirebreak signal results in an undefined reaction.

Wirebreak Signal in Conjunction with Resistance Thermometers

An interruption in the supply leads to a resistance thermometer is reported as follows:

Table 10.37 Wirebreak Signal in Conjunction with Resistance Thermometers

Wirebreak on	Digitized Analog Value (460/465 Module)	Status of the Error Bit (460 Module)	Status of the Error Bit (465 Module)
M+	0/0	1	1
M -	0/0	1	1
PT100 (resist.-type sensor)	0*/0	0*	1
S+	0/0	0	1
S -	0/0	0	1

* On the 460 analog input module, the value "0" is also decoded for the unbroken PT 100 resistors and error bit F set to 0.

The overflow bit is set separately for each channel in the case of the 460/465-7LA12 modules.

The S+ lines to the resistance thermometer can be monitored for a wirebreak on the 465 -7LA12 analog input module by setting switch 7 of mode selector I to "PT 100" (PT 100 constant power supply). The error bit is also set to flag a wirebreak in this line.

Unused channels can be used to measure voltages or currents when the current sourcing outputs (S+, S -) associated with the relevant measuring channel are short-circuited with a jumper. Without this jumper, the error bit would be set for this channel and the value "0" decoded.

The S+ lines are not monitored for wirebreak when mode selector II is in the "Current or voltage measurement" position. In this case, the error bit is not set when a wirebreak occurs. This switch setting should be selected when only voltages or currents are to be measured (see Figure 10.7).

The following general rule applies: When the wirebreak signal is to be issued, the measuring circuit must have a low resistance (<1 kΩ).

Sampling

The **460** and **465** modules offer two methods of sampling the analog value:

- Cyclic sampling and
- Selective sampling

The **463** module implements only cyclic sampling.
The **466** module implements only cyclic sampling because of its high speed.

Cyclic Sampling

The modules's processor decodes all inputs.
However, there are differences between the individual modules.
For example, the amount of time that elapses before a measured value is updated depends on the number of input channels. The time required for decoding depends on the input value. In the case of the 460 analog input module, when $V_I = 0$ V, decoding takes 40 ms; when V_I = nominal value, decoding takes 60 ms.

Table 10.38 Scan Times

Module	460	463	465		466	
Channels	8	4	8	16	8	16
Scan Time*	480 ms	20 ms	480 ms	960 ms	2 ms	4 ms

* Nominal value applied to all inputs

In the case of the 460/465 modules, the digitized measured values are stored in the circulating buffer under the channel address (the high-order byte under address n, the low-order byte under address n + 1), and can be read out from the buffer whenever required.

Selective Sampling

Selective sampling is ***not*** possible on the **463** and **466** modules.

Double addressing cannot be used for selective sampling, i.e. an address cannot be assigned to an analog output module and an analog input module.

In the case of the 460 and 465 modules, the initiative for decoding a measured value comes from the CPU when this function is used. The module must be accessed once with a Write command (T PW) under the relevant channel address; the data itself is of no relevance. In this way, only the measured value of the activated channel is decoded and the other channels are ignored. During decoding, an activity bit is set on the data bus (**A = 1**, see also Section 10.7). The module sets the activity bit independently, i.e. if several channels are to be decoded using selective sampling, the activity bit cannot be assigned to one channel! The valid digitized measured value can be read out from two bytes once the activity bit has been reset (A = 0, negative-going edge).

Repeated scanning of the activity bit loads both the bus and the CPU. This results in non-periodic measured value acquisition when different measured values are involved, and is therefore not desirable for PID control tasks.

A better method is time-controlled program execution, in which certain program sections, for instance FB13, are automatically inserted into the program every 100 ms by a time-controlled block (OB13), thus producing a constant time base while offloading the bus and the CPU.

The associated sample program is written as follows:

FB13 STL	Description
NAME:SEL-SAMP	EXAMPLE FOR SELECTIVE SAMPLING
:L PW128	READ ANALOG VALUE
:T FW128	TRANSFER TO AUXILIARY FLAG
:A F 129.2	SCAN ACTIVITY BIT
:JC =END	IF = 1, JUMP TO END
:T FW10	IF = 0, TRANSFER MEASURED VAL. TO FW 10
:T PB128	INITIATE SAMPLING
END:	(1ST VAL. IS INVALID FOLLOWING RESTART)
:BE	

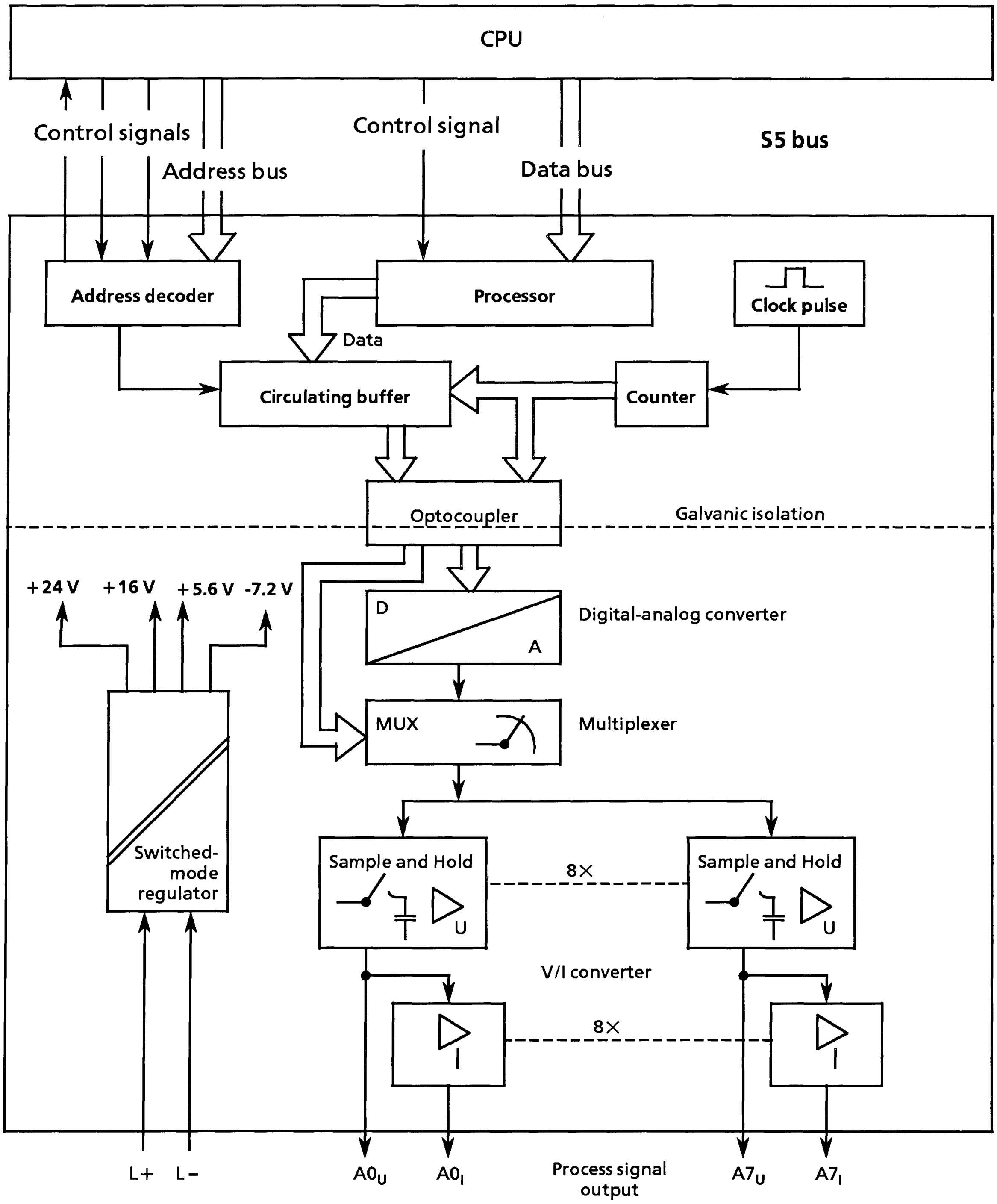
10.9 Analog Output Modules

The CPU processes the digital values that the analog output modules convert to the required voltages or currents. Various floating modules cover individual voltage and current ranges.

Signal Interchange between CPU and Module

The CPU transfers a digital value to the module's memory under a specified address.The user starts the transfer via FB251 or "TPY" or "TPW" operation.

Block diagram10.32 illustrates the principle of operation of the 470 analog output module.



MUX	Multiplexer
D/A	Digital-analog converter

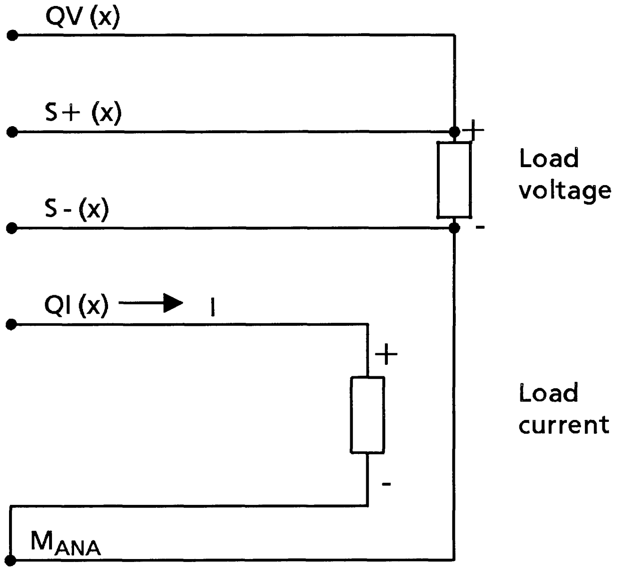
Figure 10.32 Block Diagram with Signal Interchange between CPU and a 470 Analog Output Module

10.9.1 Connecting Loads to Analog Output Modules

When loads are connected to analog output modules, the voltage is measured directly across the load via high-resistance sensing lines (S+/S -). The output voltage is then corrected so that the load voltage is not falsified by voltage drops on the lines.

In this way it is possible to compensate voltage drops of up to 3 V per line.

Figure 10.33 shows the design of this circuit.

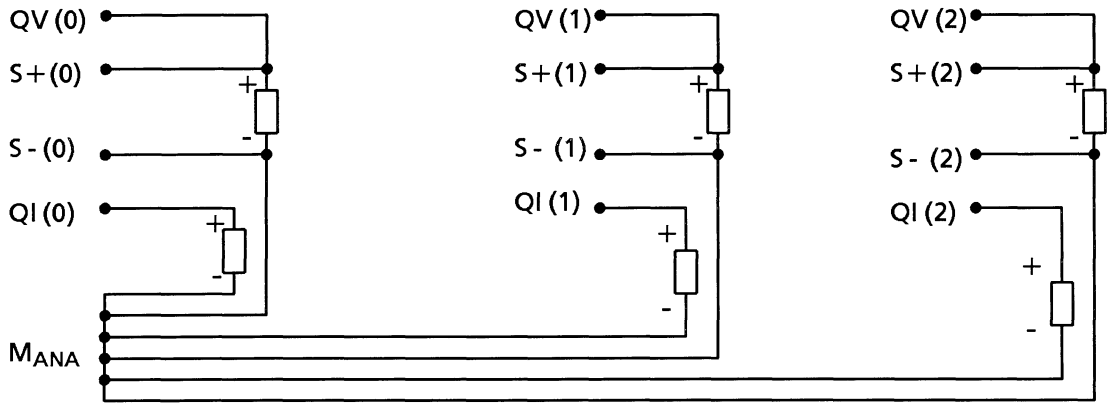


QV	(x)	=	Analog output voltage (QV = Output voltage)
QI	(x)	=	Analog output current (QI = Output current)
S+	(x)	=	Sensing line+ (S+ = Sensing line+)
S -	(x)	=	Sensing line- (S - = Sense line -)
M_{ANA}		=	Ground terminal of the analog component
	x	=	Channel no. (0 to 7)

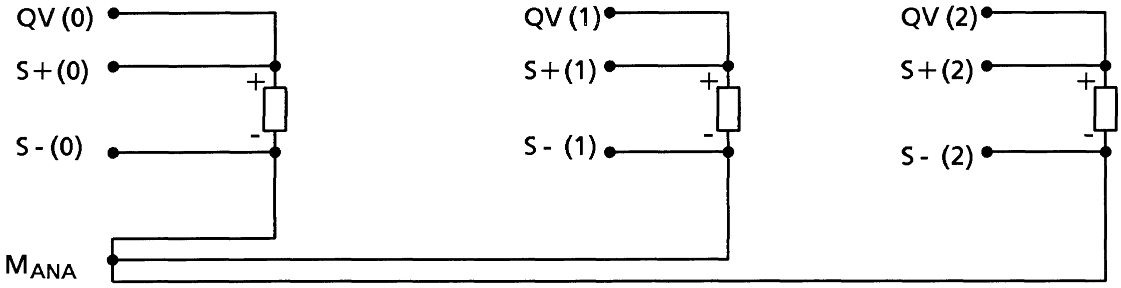
Figure 10.33 Connecting Loads

Connecting Loads to Current and Voltage Outputs

Figure 10.34 shows how to wire the analog output module.



6ES5 470-7LAxx
6ES5 470-7LCxx



6ES5 470-7LBxx

Figure 10.34 Connecting Loads to Current and Voltage Outputs

Note

If voltage outputs are not used, or if only current outputs are connected, jumpers must be inserted in the front connector for the unused voltage outputs. To do this, connect QV (x) to S+ (x) and S - (x) to M_{ANA}.
Unused current outputs remain open.

10.9.2 Digital Representation of an Analog Value

The CPU uses two bytes to represent the value of an output channel. Figure 10.35 explains the individual bits:

	High-Order Byte								Low-Order Byte							
Byte No.	n								n+ 1							
Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	2^{11}	2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	x	x	x	x

$$\underbrace{2^{11}\ 2^{10}\ 2^9\ 2^8\ 2^7\ 2^6\ 2^5\ 2^4\ 2^3\ 2^2\ 2^1\ 2^0}_{\text{Binary signal}}$$

x represents an irrelevant bit

Figure 10.35 Representation of an Analog Output Signal in Digital Form

Note

For the two's complement, bit 2^{11} indicates the sign (0 equals a positive value, 1 a negative value).

Table 10.39 lists the output voltages or currents of the individual 470-... analog output modules.

Table 10.39 Analog Output Signals

Units	Output Voltages and Currents of the Modules				Digitized Output Value*												Range
	-7LA/B12 in V	-7LA12 in mA	-7LC12 in V	-7LC12 in mA	2^{11}	2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	
+1280	+12.5	25.0	6.0	24.0	0	1	0	1	0	0	0	0	0	0	0	0	Overrange
+1025	+10.0098	20.0195	5.004	20.016	0	1	0	0	0	0	0	0	0	0	0	1	
+1024	+10.0	20.0	5.0	20.0	0	1	0	0	0	0	0	0	0	0	0	0	Nominal value
+1023	+9.99	19.98	4.995	19.98	0	0	1	1	1	1	1	1	1	1	1	1	
+512	+5.0	10.0	3.0	12.0	0	0	1	0	0	0	0	0	0	0	0	0	
+256	+2.5	5.0	2.0	8.0	0	0	0	1	0	0	0	0	0	0	0	0	
+128	+1.25	2.5	1.5	6.0	0	0	0	0	1	0	0	0	0	0	0	0	
+64	+0.625	1.25	1.25	5.0	0	0	0	0	0	1	0	0	0	0	0	0	
+1	+0.0098	0.0195	1.004	4.016	0	0	0	0	0	0	0	0	0	0	0	0	
0	+0.0	0.0	1.0	4.0	0	0	0	0	0	0	0	0	0	0	0	0	
-1	-0.0098	0.0	0.996	3.984	1	1	1	1	1	1	1	1	1	1	1	1	
-64	-0.625	0.0	0.75	3.0	1	1	1	1	1	1	0	0	0	0	0	0	
-128	-1.25	0.0	0.5	2.0	1	1	1	1	1	0	0	0	0	0	0	0	
-256	-2.5	0.0	0.0	0.0	1	1	1	1	0	0	0	0	0	0	0	0	
-512	-5.0	0.0	-1.0	0.0	1	1	1	0	0	0	0	0	0	0	0	0	
-1024	-10.0	0.0	-3.0	0.0	1	1	0	0	0	0	0	0	0	0	0	0	
-1025	-10.0098	0.0	-3.004	0.0	1	0	1	1	1	1	1	1	1	1	1	1	Overrange
-1280	-12.5	0.0	-5.0	0.0	1	0	1	1	0	0	0	0	0	0	0	0	

* The insignificant bits have been omitted

10.10 Analog Value Matching Blocks

These blocks match the nominal range of an analog module to a normalized range that you can specify.

The CPU 945 has four analog value matching blocks which are assigned to the various analog input modules

● For the 460 and 465 analog input modules:	FB250
● For the 463 analog input module:	FB241
● For the 464-8 (ET 100/ET 200) analog input module:	FB242
● For the 466 analog input module:	FB243

The FBs read an analog value from the analog input module assigned to them and output a value XA as a 32-bit floating-point number in the scaled range specified. Define the desired range using the "upper limit" (OGR) and "lower limit" (UGR) parameters.

Specify the type of analog value representation (channel type) in the KNKT parameter.

The BU parameter is set when the analog value exceeds the nominal range.

10.10.1 FB250-Reading and Scaling Analog Values of the 460 and 465 Analog Input Modules

Call and Parameter Assignments

Parameter	Description	Type	Data Type	Assignment	STL
BG	I/O area and module address	D	KY	KY=0.128 to 0.224 (0.240)* (P area) KY=1.128 to 1.224 (1.240)* (Q area) KY=2.128 to 2.224 (2.240)* (IM3 area) KY=3.128 to 3.224 (3.240)* (IM4 area)	: JU FB 250 NAME : AI:460 BG : KNKD : OGR : UGR : EINZ : XA : FB : BU : TBIT :
KNKT	KN= Channel number KT= Channel type	D	KY	KY=x, Y x=0 to 15 (channel number) y=3 to 6 (type of channel) 3: fixed-point represent., unipolar (nom. range +512 to +2560) 4: fixed-point represent., unipolar (nominal range 0 to +2048) 5: abs. value represent., bipolar (nom. range -2048 to +2048) 6: fixed-point represent., bipolar (nom. range -2048 to +2048)	
OGR	Upper limit of the output value	D	KG	KG= $-1701412\times10^{+39}$ to $+1701412\times10^{+39}$	
UGR	Lower limit of the output value	D	KG	KG= $-1701412\times10^{+39}$ to $+1701412\times10^{+39}$	
EINZ	Selective sampling	I	BI	Flag or input bit	
XA	Output value as floating-point number	Q	BI	Flag, output or data double word (format KG)	
FB	Error bit	Q	BI	Flag or output bit	
BU	Range violation	Q	BI		
TBIT	Activity bit	Q	BI		

* 224 for 16 channels
240 for 8 channels

Selective Sampling

FB 250 permits reading of an analog value with selective sampling. Setting the "EINZ" parameter to "1" causes the analog input module to convert the analog value of the selected channel to a digital value immediately. During conversion (approximately 60 msec.), no further sampling operations involving this module may be initiated. Consequently, the function block that is presently active sets the TBIT to "1" until the converted value is read in. The TBIT is reset upon completion of selective sampling is terminated.

Scaling:
Function block FB250 converts the value read linearly to accord with the upper and lower limiting values using the following formula:

For channel type 3 (absolute value 4 to 20 mA):

$$XA = \frac{UGR \cdot (2560 - xe) + OGR \cdot (xe - 512)}{2048}$$

For channel type 4 (unipolar representation):

$$XA = \frac{UGR \cdot (2048 - xe) + OGR \cdot xe}{2048}$$

For channel type 5 and 6 (bipolar representation):

$$XA = \frac{UGR \cdot (2048 - xe) + OGR \cdot (xe + 2048)}{4096}$$

Where XA is the value output by the FB and
xe is the analog value read from the module.

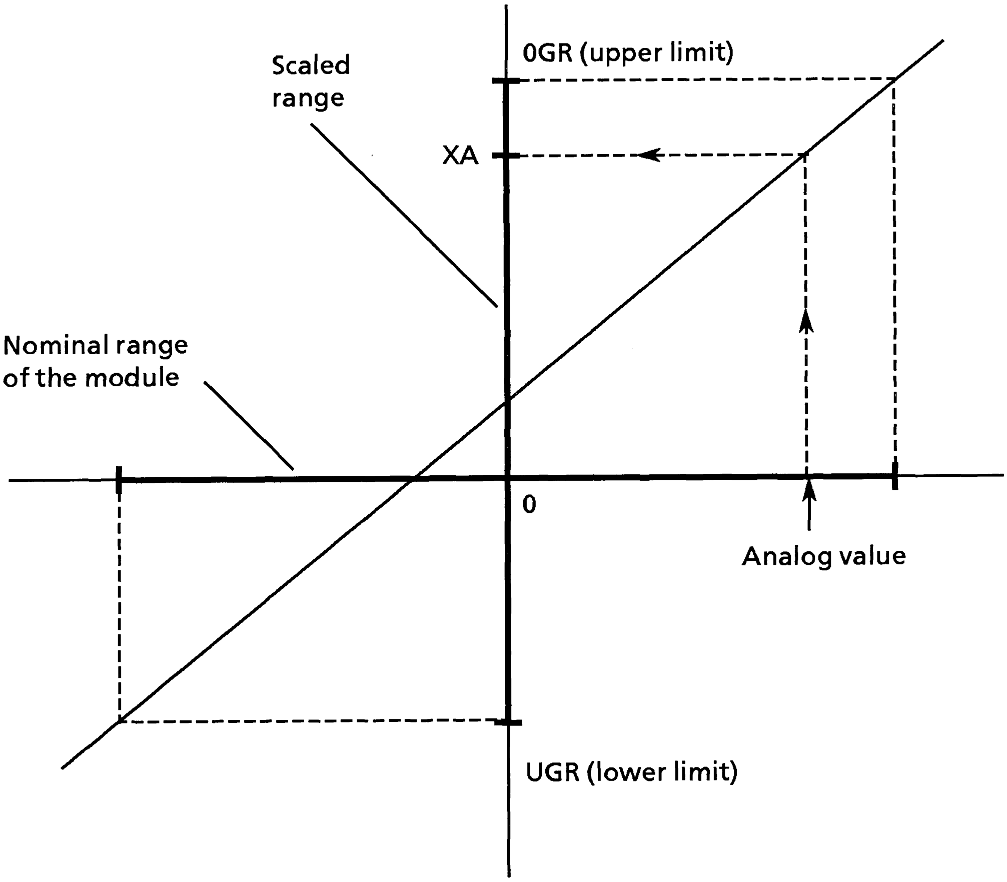


Figure 10.36 Schematic Representation of Conversion

10.10.2 FB241-Reading and Scaling Analog Values of the 463 Analog Input Module

Call and Parameter Assignments

Parameter	Meaning	Type	Data Type	Assignment	STL
BG	I/O area and module address	D	KY	KY=0.128 to 0.248 (P area) KY=1.0 to 1.248 (Q area) KY=2.0 to 2.248 (IM3 area) KY=3.0 to 3.248 (IM4 area)	: JU FB 241 NAME : AI:463 BG : KNKT : OGR : UGR : XA : FB : BU :
KNKT	KN= Channel number KD= Channel type	D	KY	KY=x, y x=0 to 3 (channel number) y=20 to 21 (channel type) 20: fixed-point represent., unipolar (nom. range 0 to +1024) 21: fixed-point represent. unipolar (nom. range +256 to +1280)	
OGR	Upper limit of the output value	D	KG	KG= $-1701412 \times 10^{+39}$ to $+1701412 \times 10^{+39}$	
UGR	Lower limit of the output value	D	KG	KG= $-1701412 \times 10^{+39}$ to $+1701412 \times 10^{+39}$	
XA	Output value as floating-point number	Q	D	Flag, output or data double word (format KG)	
FB	Error bit	Q	BI	Flag or output bit	
BU	Range violation	Q	BI		

10.10.3 FB242-Reading and Scaling Analog Values of the 464-8Mxxx Analog Input Module

Call and Parameter Assignments

Parameter	Meaning	Type	Data Type	Assignment	STL
BG	I/O area and module address	D	KY	KY=0.128 to 0.254 (P area) KY=1.0 to 1.254 (Q area)	: JU FB 242 NAME : AI:464 BG : KNKT : OGR : UGR : XA : FB : BU :
KNKT	KN= Channel number KD= Channel type	D	KY	KY=x, y x=0 to 3 (channel number) y=7 to 13 (channel type) 7: fixed-point represent., bipolar (nominal range ±2048) 8: fixed-point represent., bipolar (nom. range -200 to +1369) 9: fixed-point represent., bipolar (nom. range -200 to +1200) 10: fixed-point represent., bipolar (nom. range -199 to +900) 11: fixed-point represent., bipolar (nom. range -200 to +1700) 12: fixed-point repres., unipolar (nom. range +512 to +2560) 13: fixed-point repres., unipolar (nom. range 0 to +2048)	
OGR	Upper limit of the output value	D	KG	KG= - $1701412 \times 10^{+39}$ to $+1701412 \times 10^{+39}$	
UGR	Lower limit of the output value	D	KG	KG= - $1701412 \times 10^{+39}$ to $+1701412 \times 10^{+39}$	
XA	Output value as floating-point number	Q	D	Flag, output or data double word (format KG)	
FB	Error bit	Q	BI	Flag or output bit	
BU	Range violation	Q	BI		

10.10.4 FB243-Reading and Scaling Analog Values of the 466 Analog Input Module

Call and Parameter Assignment

Parameter	Meaning	Type	Data Type	Assignment	STL
BG	I/O area and module address	D	KY	KY=0.128 to 0.224 (P area) KY= 1.0 to 1.224 (Q area)	: JU FB 243 NAME : AI:466 BG : KNKT : OGR : UGR : XA : FB : BU :
KNKT	KN= Channel number KD= Channel type	D	KY	KY=x, Y x=0 to 15 (channel number) y=22 to 25 (channel type) 22: fixed-point number representation, bipolar (nom. range -2048 to +2048) 23: abs.-value represent., bipolar (nom. range -2048 to +2048) 24: binary represent., unipolar (nominal range 0 to +4095) 25: fixed-point repres., unipolar (nom. range +512 to +2559)	
OGR	Upper limit of the output value	D	KG	KG= - 1701412$\times 10^{+39}$ to +1701412$\times 10^{+39}$	
UGR	Lower limit of the output value	D	KG	KG= - 1701412$\times 10^{+39}$ to +1701412$\times 10^{+39}$	
XA	Output value as floating-point number	Q	D	Flag, output or data double word (format KG)	
FB	Error bit	Q	BI	Flag or output bit	
BU	Range violation	Q	BI		

10.10.5 Outputting an Analog Value -FB251-

Use function block FB251 to output analog values to analog output modules. Specify the module's type of analog representation (channel type) in the KNKT parameter. Values from the range between the "lower limit" (UGR) and the "upper limit" (OGR) parameters are converted to the nominal range of the relevant module using the following formula:

For channel type 0 (unipolar representation):

$$xa = \frac{1024 \cdot (XE-UGR)}{OGR-UGR}$$

For channel type 1 (bipolar representation):

$$xa = \frac{1024 \cdot (2 \cdot XE-OGR-UGR)}{OGR-UGR}$$

Where XE is the digital value specified in the function block and
xa is the value output to the module.

Calling and Initializing -FB251-

Parameter	Meaning	Type	Data Type	Assignment	STL
XE	Analog value to be output as fixed-point number	I	D	Input value (fixed-point) in the UGR to OGR range (flag, data double word; format KG)	: JU FB 251 NAME : RLG:AA XE : BG : KNKT : OGR : UGR : FB : BU :
BG	I/O range and module address	D	KY	0.128 to 0.240 (P area) 1.128 to 1.240 (Q area) 2.128 to 2.240 (IM3 area) 3.128 to 3.240 (IM4 area)	
KNKT	KN= Channel number KD= Channel type	D	KY	KY=x, y x=0 to 7 y=0;1 0: unipolar representation 1: bipolar fixed-point number	
OGR	Upper limit of the output value	D	KG	KG= - 1701412×10^{+39} to +1701412×10^{+39}	
UGR	Lower limit of the output value	D	KF	KG= - 1701412×10^{+39} to +1701412×10^{+39}	
FB	Error when setting the limit value	Q	BI	Flag or output bit "1" if UGR≥OGR, for illegal channel or slot number, or illegal channel type and in case of NAK of module	
BU	Analog value to be output exceeds UGR or OGR	Q	BI	Flag or output bit at "1, "XE is outside the range (UGR;OGR). XE assumes the limit value	

10.10.6 Extended Error Diagnostics with the Analog Value Matching Blocks

Besides the FB and BU bits, the analog value matching blocks offer further diagnostics facilities. As additional error diagnostics, the analog value matching blocks provide additional information in ACCU 1-L and in the RLO when exiting the block:

- RLO is set, when the FB or BU bit = "1"
- When FB = "0", the digital value read by the analog input module or written to the analog output module is available in ACCU 1-L.
- When FB = "1", an error number describing the error found is available in ACCU 1-L (see Table 10.40).

Table 10.40 Error Diagnostics with the Analog Value Matching Blocks

FB / Error no.	FB241	FB242	FB243	FB250	FB251
0001_H	Parameter BG<128 in P area	Parameter BG<128 in P area	Parameter BG<128 in P area	Parameter BG<128 in P, Q, IM3, IM4 area	Parameter BG<128 in P, Q, IM3, IM4 area
0002_H	Parameter KN>3	Parameter KN>3	Parameter KN>15	Parameter KN>15	Parameter KN>7
0003_H	Sum of parameter BG and 2×parameter KN>255	Sum of parameter BG and 2×parameter KN>255	Sum of parameter BG and 2×parameter KN>255	Sum of parameter BG and 2×parameter KN>255	Sum of parameter BG and 2×parameter KN>255
0004_H	KD not 20 or 21	KD<7 or KD>13	KD<22 or KD>25	KD<3 or KD>6	KD>1
0005_H	OGR≤UGR	OGR≤UGR	OGR≤UGR	OGR≤UGR	OGR≤UGR
0006_H	Parameter BG:x,y with x>3	Parameter BG:x,y with x>3	Parameter BG:x,y with x>1	Parameter BG:x,y with x>3	Parameter BG:x,y with x>3
$00FF_H$	NAK	NAK	NAK	NAK	NAK

10.11 Example of Analog Value Processing

Problem Definition:

A closed container contains a liquid. It should be possible to read the current liquid level on an indicating instrument whenever required. A flag is to be set when the liquid level reaches a specified limiting value.

- A 0 - 20 mA transducer transmits the liquid level signal (between 0 and 10 m) to a 6ES5 460-7LA12 (460 AI) analog input module.
- The analog input module converts the analog current values into digital units (0 - 2048 units), which can be postprocessed by the S5-115U's application program.
- The application program compares the values with a limiting value (max. permissible liquid level), sets a flag if necessary, and sends these values to a 6ES5 470-7LB11 (AO 470) analog output module.
- The analog output module reconverts the values into voltages (0 - 10 V). In response to these voltages, the needle on the analog display swings proportionally to the liquid level.

Figure 10.37 shows the system configuration.

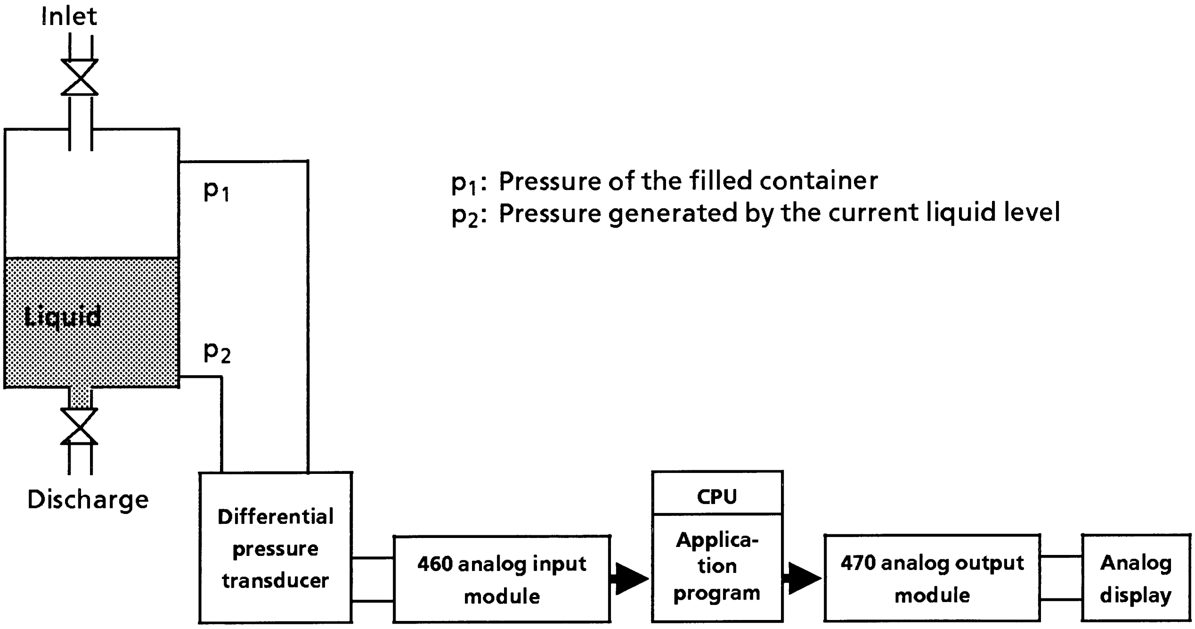

Inlet
Liquid
p_1
p_2
Discharge
p_1: Pressure of the filled container
p_2: Pressure generated by the current liquid level
Differential pressure transducer
460 analog input module
CPU
Application program
470 analog output module
Analog display

Figure 10.37 Example of Analog Value Processing

Startup Procedures

460 Analog Input Module:

- Connect the transducer directly to the front connector on the AI 460 (Terminals: MO+, MO -). The transducer supplies values between 0 and 20 mA, 0 mA corresponding to a liquid level of 0.00 meters and 20 mA to the maximum liquid level, which is 10.00 meters.
- Plug a ± 20 mA range card (6ES5 498-1AA41) into the AI 460.
  A digital value between 0 and 2048 units, which is subsequently processed by the application program, is then present at the output of the analog input modules's internal ADC (see Figure 10.38).

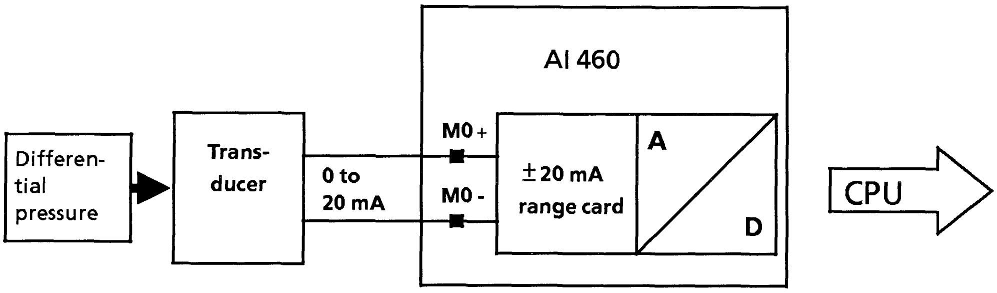


Figure 10.38 Function of the 460 Analog Input Module

- Set the mode selectors at the rear of the module as follows (Figure 10.39):

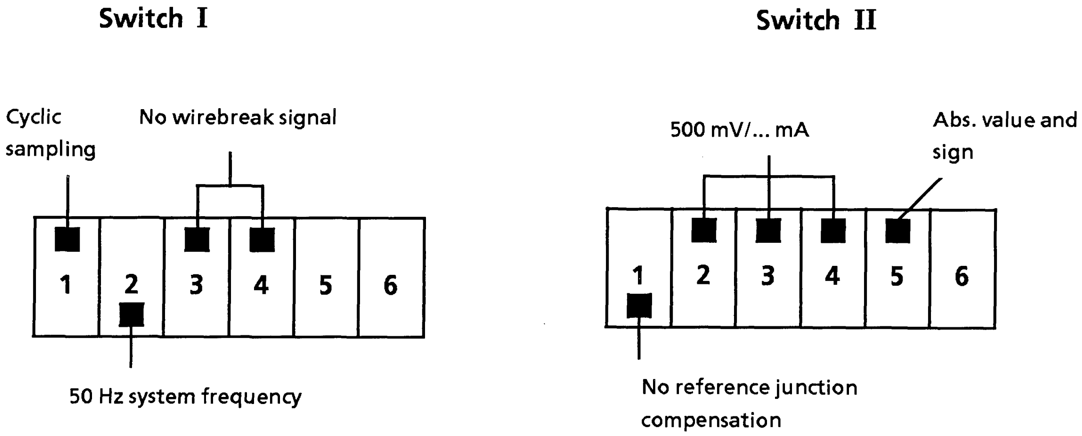


Figure 10.39 Setting Mode Selectors I and II

470 Analog Output Module:

- Connect the indicating instrument directly via the module's front connector (pins: QV0, S + 0, S - 0, M_{ANA}).
  The analog output modules outputs a voltage between 0 and 10 V to the indicating instrument, thus making it possible to read the liquid level as an analog value (Figure 10.40).

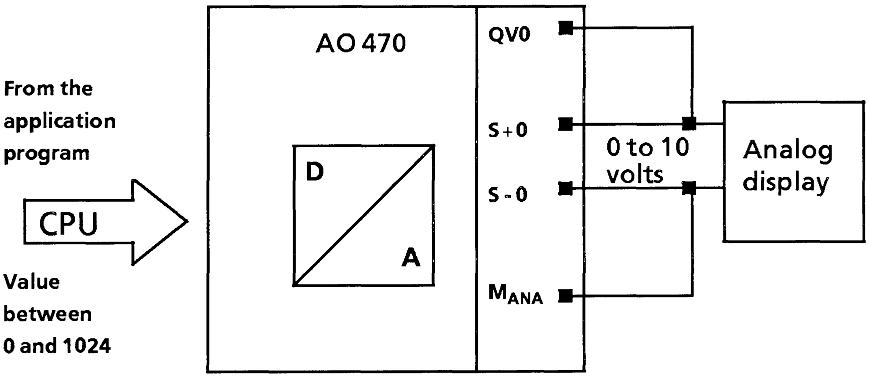


Figure 10.40 Function of the 470 Analog Output Module

Program Structure

- Call and parameterize "Read analog value" function block FB250 (for conversion to a range of from 0 to 1000 cm [XA parameter]).
- Generate the limiting value (PB9).
  A flag (F 14.6) is set when the liquid level exceeds 900 cm.
- Call and parameterize "Output analog value" function block FB251 (for conversion of a value in the range from 0 to 1000 cm [XE parameter] into a value between 0 and 1024 units for the AO 470).

Integral function blocks FB250 and FB251 are discussed in detail in Sections 10.10.1 and 10.10.5.

PB1 STL	Description
:JU FB 250	
NAME :AI:460	
BG :KY 0,128	MODULE STARTING ADDR: 128 (WHEN SLOT ADDRESSING IS FIXED: SLOT 0)
KNKT :KY 0,4	CHANNEL NO.: 0; UNIPOLAR REPRESENTATION: 4
OGR :KG +1000000+04	PHYSICAL MEASURING RANGE:
UGR :KG +0000000+00	0<XA<1000CM
	RELEVANT ONLY FOR SELECTIVE SAMPLING
EINZ :F 14.0	(SET IN EXAMPLE FOR: CYCLIC SAMPLING)
	IN FD 10: XA VALUE 0<XA<1000CM
XA :FD 10	RELEVANT ONLY WHEN SET FOR WIREBREAK SIGNAL
FB :F 14.1	IF LEVEL > 1000CM, BU = 1.
BU :F 14.2	RELEVANT ONLY FOR SELECTIVE SAMPLING.
TBIT :F 14.3	

PB1 STL (cont.)	Description
:JU PB 9	GENERATE LIMITING VALUE
:JU FB 251	OUTPUT ANALOG VALUE
NAME :RLG:AA	
XE :FD 10	XA (FB 250) = XE (FB 251)
BG :KY 0,160	MODULE STARTING ADDR.; 160 (FIXED SLOT ADDRESSING: SLOT 1)
KNKD :KY 0,0	CHANNEL NO.: 0; UNIPOLAR REPRESENTATION: 0
OGR :KG +10000000+04	PHYSICAL MEASURING RANGE:
UGR :KG +0000000+00	0<XA<1000CM
	FB = 1 in case of parameterization error or NAK.
FB :F 14.4	WHEN XA<UGR OR XA>OGR, BU = 1.
BU :F 14.5	
:BE	

PB9 STL	Description
:L KG +9000000+03	MAX. VAL. FOR LIQUID LEVEL
	MEASURED VALUE
:L FD 10	MEASURED VALUE > 900 ?
:<=G	IF YES, F 14.6 = INITIATE REACTION
:=F 14.6	IN SAME PROGRAM CYCLE.
:BE	

11 Parameterization of CPU 945 with DB1

Figures

Tables

11 Parameterization of CPU 945 with DB1

The CPU has functions which you can set to your own requirement. For example, you can initialize the following:

- Integral hardware clock
- Data interchange over SINEC L1
- Data interchange via 2nd interface (interface submodule)
  - ASCII driver
  - 3964 computer link
- Call interval for time-controlled program execution (OB10 to 13)
- System characteristics (e.g. scan time monitoring)
- Address for parameter error code.

You can initialize these functions in data block DB1.

11.1 Configuration and Default Settings for DB1

To make it easier for you to assign parameters, data block 1 is already integrated in the programmable controller with preset values (default parameters). After performing an overall reset, you can load the default DB1 from the programmable controller into your programmer and display it on the screen:

```
 0:      KS  ='DB1 TFB: OB13 100 ; SDP: ';
12:      KS  =' WD 500 ; END ';
19:
```

▨ Parameter block identifiers

This default DB1 contains one ***parameter block*** each for the following functions:

- Specifying the call interval for OB13; parameter block "TFB:".
- System characteristics (scan time monitoring); parameter block "SDP:".

What Typifies a Parameter Block?

A parameter block contains all the parameters of one function; it always starts with a block identifier followed by a colon. The colon must be followed by at least one space. The semicolon (;) indicates the end of a parameter block. The ***parameters*** are contained between the block identifier and the semicolon (;).

11.2 Setting the Addresses for the Parameter Error Code in DB1 (An example of how to set the parameters correctly)

We recommend that you use this example when you start setting your parameters. The following two reasons explain why.

1. There are no default parameters in DB1 for parameter block "ERT:". You must therefore enter this block complete. We will explain the entries step by step. In doing so, you will quickly learn the rules for initializing.
2. Properly entered, parameter block "ERT:" makes it easy for you to correct parameter errors. For this reason, you should complete this block in DB1 before changing or entering other parameters.

To help find parameter errors easier and to help correct them, you can ask the programmable controller to output error messages in a coded form. All you have to do is to tell the programmable controller where it should store the error code. Make this input in parameter block "ERT:" of DB1.
The error code can be stored in:

- flags/S flags
  or
- data blocks (DB/DX).

The entire error code consists of 20 flag bytes or 10 data words. You only need to indicate the start address for the error code in parameter block "ERT:".

Procedure:

Overall Reset has been performed on the CPU and the CPU is in the STOP state.

- Display default DB1 on the programmer
- Position the cursor on the E of the end identifier "END" at the end of default DB1
- Now enter the shaded characters:

DB 1	Explanation

`0:   KS  ='DB1 TFB: OB13 100 ; SDP:';`
`12:  KS  =' WD 500 ; ERT: ERR FY1 ;';`
`24:  KS  =' END';`
`26:`
	The parameter error code is stored from flag byte FY 1 after cold restart (automatically after power restore or manually).

- Use the following check list to make sure your entries are correct.
  - Is the block ID "ERT:" terminated by a colon?
  - Is at least 1 filler (a blank space) added after the colon?
  - Is the parameter name ("ERR") entered correctly?
  - Does at least 1 filler (a blank space) follow the parameter name?
  - Is the argument (for example "FY1") entered correctly?
  - Does at least 1 filler (a blank space) follow the argument?
  - Does a semicolon (;) indicate the block end?
  - The end ID "END" concludes DB1

- Transfer the changed DB1 to the programmable controller.
- Switch the programmable controller from STOP to RUN.

Changed DB1 parameters are accepted.

If you did not store the parameter block "ERT:" in DB1, you can localize the error in the ISTACK if there was an incorrect parameter setting. However, you will not know what type of error is present. The same thing applies if you made an error when you input the parameter block "ERT:".

11.3 How to Assign Parameters in DB1

As illustrated in Section 11.2, you use the following steps to change or expand the preset values of DB1:

- Display the default DB1, with its parameter block "ERT:" on the programmer.
- Position the cursor on the desired parameter block.
- Change or expand the parameters.
- Transfer the changed DB1 to the programmable controller.
- Switch the programmable controller from STOP to RUN.

Changed DB1 parameters are accepted.

The following applies when initializing in DB1:

- ***Not*** all parameters of a parameter block must be defined in DB1. If some parameters are not defined in DB1, the default setting of the relevant system data word automatically applies!
- If you delete a DB1 in the PLC, the integral DB1 is retrieved by an overall reset of the CPU.
- In addition to DBxDWy arguments, DXaDWa arguments can be used, if you store the parameters in DX blocks.
- You can set parameters in either uppercase or lowercase.

Note

If the CPU detects a parameter error in DB1, the restart is interrupted on transition from STOP → RUN or after POWER ON and the CPU goes into STOP with the identifier SYSFE in the ISTACK.

11.4 Rules for Setting Parameters in DB1

DB1 consists of the following:

- A start block ID .... : DB1
- One or more parameter blocks .... e. g.: TFB: OB13 100

  A parameter block consists of:

  - A block ID .... e. g.: TFB:
  - One or more parameters .... e. g.: OB13 100

    A parameter consists of:

    - A parameter name .... e. g.: OB13
    - One or more arguments .... e. g.: 100

  - A block end ID .... : ; (Semicolon)

- A block end symbol: .... END

The following is a list of all the rules you must observe when changing parameters in DB1 or when completing whole parameter blocks. If you do not observe these rules, the CPU cannot interpret your entries. The structure of this DB1 depends on whether interprocessor communication flags must be defined or not!

1. **If interprocessor communication flags must be defined:**
   DB1 begins with the definition of the interprocessor communication flags as described in Section 12.2.1. The "DB1" start identifier for the other DB1 parameters follows the interprocessor communication flag end identifier ($EEEE_H$). The three characters must not be separated by spaces. The "DB1" start identifier must be followed by at least one space.
   If no interprocessor communication flags must be defined:
   DB1 begins with the "DB1" start identifier. The three characters must not be separated by spaces. The "DB1" start identifier must be followed by at least one space.

2. The start identifier (including space) is followed by the block identifier of a parameter block. The parameter blocks can follow any order in DB1. The block identifier indicates a block of related parameters. The block identifier "TFB" stands for "*T*imer *F*unction *B*lock" (time-controlled execution). The block identifier must be followed immediately by a colon (:). If the colon is missing, the CPU skips this block and outputs an error message. The block identifier and its colon must be followed by at least one space.

3. The parameter name comes next. Parameter names are names for single parameters within a parameter block. Within a block, the first four characters of a parameter name must be different from each other. After the parameter name, you must add at least one filler.

4. At least one argument is attached to each parameter name. An argument is either a number or a STEP-5 operand that you must enter. If several arguments belong to a parameter name, then every argument must be followed by at least one filler (even the last one).
5. Use a semicolon (;) to identify a block end. After the semicolon, you must enter at least one filler. Leaving out the semicolon leads to misinterpretation in the programmable controller.
6. After the semicolon, additional parameter blocks can follow. (Use steps 2 through 5 to create additional parameter blocks.)
7. After the end of the last parameter block, you must enter the end ID "END". This identifies the end of DB1. If you forget to enter an end ID, this leads to errors in the programmable controller.

Points 1 through 7 present the minimal requirements for setting the parameters. Beyond that, there are additional rules that make it easier for you to assign parameters.
For example:

- you have the ability to add comments
- you can expand the mnemonics used as parameter names in plain text.

Comments can be added anywhere a filler is allowed. The comment symbol is the pound (#) sign. The comment symbol must be placed at the beginning and at the end of your comment. The text between two comment symbols may not contain an additional #.
Example: #Comment# . At least one filler must follow the # sign.

In order to make it easier to read parameter names, you can add as many characters as you wish if you add an underscore (__) after the abbreviated parameter name.
Example: SF becomes SF__SENDMAILBOX .
At the end of the input, you must add at least one filler.

There is a rule of thumb that will help you to check DB1. You should include at least one filler in the following instances:

- ***after*** the start ID
- ***before and after*** the block ID, parameter name, argument, and semicolon

11.5 How to Recognize and Correct Parameter Errors

Should an error occur while assigning parameters and the programmable controller does not go to the RUN mode, you have two possibilities for recognizing errors:

- by using a parameter error code
- by using the analysis function "ISTACK"

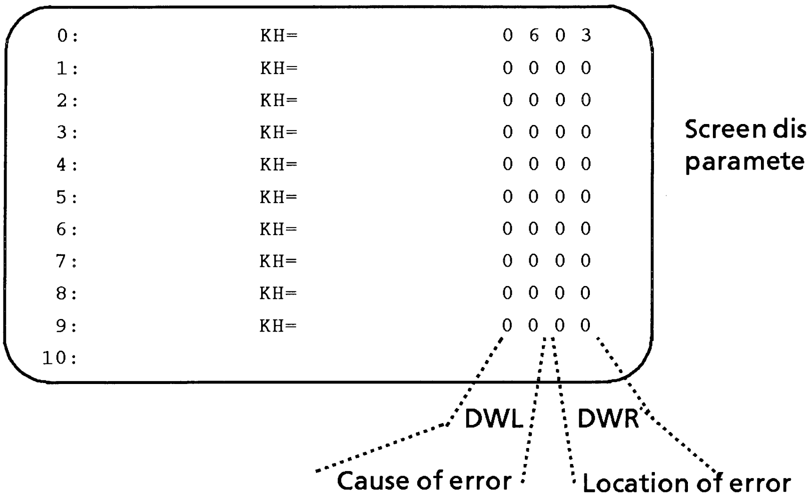
Both possibilities are described below.

Scanning the Parameter Error Code

If you have entered a start address for the parameter error code in parameter block "ERT:" of DB1, then you can retrieve the cause of the error, and the error location information at this address.

The entire error code occupies 10 data words or 20 flag bytes. In the following examples and tables, we assume that the error code is stored in a data block starting with data word 0. The error code occupies DW 0 through DW 9. In the "Flag" operand area, this corresponds to FW 0 through FW 18.

Example:
You entered the start address DB3 DW 0 in parameter block "ERT:". The parameters set in DB1 have already been transferred to the programmable controller. Then you continue to set parameters in DB1. While attempting to transfer the changed DB1 parameters to the programmable controller, you find out that the programmable controller remains in the STOP mode. You suspect that the reason the programmable controller remains in the STOP mode is that there is a parameter error. To find the error, display DB3 on the programmer. The entire contents of DB3 appear on the screen. DW 0 through DW 9 contain the code for the parameter error. In the following figure, you see how your screen could look. The error code containing the cause or location of the error is listed in Tables 11.1 and 11.2.



```
 0:        KH=        0 6 0 3
 1:        KH=        0 0 0 0
 2:        KH=        0 0 0 0
 3:        KH=        0 0 0 0
 4:        KH=        0 0 0 0
 5:        KH=        0 0 0 0
 6:        KH=        0 0 0 0
 7:        KH=        0 0 0 0
 8:        KH=        0 0 0 0
 9:        KH=        0 0 0 0
10:
```

Screen display with parameter error codes

Figure 11.1 Parameter Error Code

Table 11.1 DB1 Parameter Error Code: Cause of Error - Left Byte in Data Word

Error Code (left byte in DW)	Cause of Error (which error has occurred?)
00	No error
01	End identifier (END) missing
02	Unterminated comment before END or semicolon missing before END or END identifier missing
03	Block identifier syntax error or identifier unknown
04	Parameter syntax error
05	Argument syntax error or overrange
06	Overrange in an argument
10	DB does not exist
11	Insufficient space (DSB/DX/MB/SY)
12	Error in specifying weekday
13	Error in the date
14	Error in time format
15	Error in hour format
70	Header or footer of ASCII driver too long
71	Control character of ASCII driver faulty
73	Parameter DB does not exist (computer link and ASCII driver)
75	ME/ML only alternatively possible (ASCII driver)

Table 11.2 DB1 Parameter Error Code: Error Location - Right Byte in Data Word

Error Code (right byte in DW)	Error Location (in which parameter block has the error occurred?)
03	SL1: SINEC L1
05	Deactivation of ASCII driver or computer link 3964 driver
06	CLP: Clock parameters
09	TFB: Timer function block
11	SDP: System characteristics
13	RKT: Computer link (2nd interface)
14	ASC: ASCII driver (2nd interface)
99	ERT: Error return
F0	Error cannot be attributed to a block
FF	Error cannot be attributed to a block

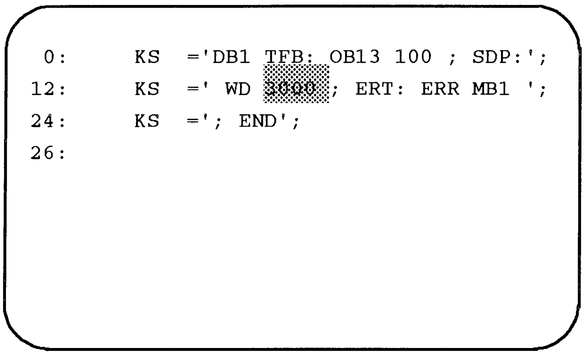
Locating Parameter Errors in the ISTACK

If the PLC detects a parameter error in DB1 during restart, the PLC remains in the STOP state and stores the asolute (error) address as well as the relative (error) address in the ISTACK. The STEP address counter (SAC) in the ISTACK then points either

- **to** the address that contains the wrong input or directly
- **in front of** the address that contains the wrong input.

The addresses are byte addresses.

Example:
You have entered DB1 as follows; the shaded area represents an error.



```
 0:      KS  ='DB1 TFB: OB13 100 ; SDP:';
12:      KS  =' WD 3000 ; ERT: ERR MB1 ';
24:      KS  ='; END';
26:
```

The decimal number at the beginning of each input line is the word address of the first user-programmable character in that line. Each word consists of two characters (two bytes)

Figure 11.2 DB1 with Parameter Error

The ISTACK indicates the following as a result of this error:

- The absolute (error) address: $\mathrm{AB14C_H}$ (absolute SAC)
- The relative (error) address: $\mathrm{001C_H}$ (relative SAC)

To locate the error accurately in your DB1, you must convert the relative byte address given as a hexadecimal number into a decimal word address.

The reason: The programmer counts the contents of a DB decimally and in words.
The SAC counts the contents of a DB hexadecimally and in bytes.

$\mathrm{001C_H}$	=	28_D	28_D	:	2_D	=	14_D
Hexadecimal byte address		Decimal byte address					Decimal word address

It follows that:
The error is in word address 14. In our example, address 14 (data word 14 and 15) is occupied by argument "3000". The entry "3000" is an error; reason: overrange.

11.6 Transferring the DB1 Parameters to the PLC

The CPU processes DB1 only after manual cold restart or after automatic cold restart after power restore.

You must perform a cold restart anytime you make changes to DB1. You can perform a cold restart by switching from:

- POWER OFF to POWER ON
  or from
- STOP to RUN

The programmable controller accepts the parameters from DB1 and stores them in the system data area.

Note

The programmable controller remains in the STOP mode if a parameter assignment error is found during start-up. The red LED lights up on the operator panel and ISTACK displays a DB1 addressing error.

11.7 Reference Table for Initializing DB1

Parameter	Argument	Permissible Range	Meaning
Blockkennung: SL1:			**SINEC L1**
SLN	p	p=0, 1 to 30, 255	"***SL***ave-***N***ummer" "0" = Master function for point-to-point link (at 2nd interface); These parameters apply for the 2nd interface, if no computer link or ASCII driver has been activated.
SM	DBxDWy or	x=0 to 255	Position of the ***S***end ***M***ailbox (start of SM)
RM	DXaDWy or	y=0 to 255.	Position of the ***R***eceive ***M***ailbox (start of RM)
CBR	FYy	a=0 to 255	Position of the ***C***oordination ***B***yte ***R***eceive
CBS	SYz	z=0 to 4095	Position of the ***C***oordination ***B***yte ***S***end
PGN	p	p=1 to 30	***PG*** bus ***N***umber Note: CBS and CBR are in a flag byte or in the high-order byte of the specified data word (DL)!
Block Identifier: SDP:			**System-Dependent -Parameters**
WD	p	p=0 to 2550	"***W***atch-***D***og" (scan time monitoring) can be set in milliseconds but only in steps of 10 msec.
RDLY	r	r=0 to 65535	"***R***un ***D***e***L***a***Y***" restart delay after POWER ON in msec.
RT	Y/N	-	"***R***esident ***T***imers" if "Y", all timers are retentive, if "N", only T 0 to 63*.
RC	Y/N	-	"***R***esident ***C***ounters" if "Y", all counters are retentive, if "N", C 0 to 63*.
RF	Y/N	-	"***R***esident ***F***lags" (if "Y", all flags/S flags are retentive, if "N", only the first half are retentive)*
PROT	Y/N	-	"***PROT***ection" activate software protection? (input/output of program no longer possible)
PIO	Y/N	-	"***P***rocess ***I***mage ***O***utput" disable output of process image?
PII	Y/N	-	"***P***rocess ***I***mage ***I***nput" disable read in of process image?
RPIC	s	s=0 to 255	"***R***educed ***P***rocess ***I***mage ***O***utput ***C***ounter" counter for cycles with reduced PIQ transfer
PPIT	Y/N	-	"***P***arallel ***P***rocess ***I***mage ***T***ransfer" If "N", the sequential PI transfer is set, if "Y", the parallel PI transfer is set.

* Additionally, set switch for Default/Overall Reset on the control panel of the CPU to "RE"

Parameter	Argument	Permissible Range	Meaning
Block Identifier: TFB:			**Timer-Function Block**
OB10 OB11 OB12 OB13	p	p=0 to 65535	Interval during which OB10 to 13 is called and processed (time in milliseconds)
Block Identifier: ASC:			**ASCII Driver at 2nd Interface**
			Note: Driver number "1" for the 2nd interface is set automatically.
SM	DBxDWy or	x=0 to 255	Position of the ***S***end ***M***ailbox (start of SM)
RM	DXaDWy or	y=0 to 255	Position of the ***R***eceive Mailbox (start to RM)
CBR	FBy or	a=0 to 255	Position of the ***C***oordination ***B***yte ***R***eceive
CBS	SYz	z=0to4095	Position of the ***C***oordination ***B***yte ***S***end
PAR			Position of the ***PAR***ameter set
MOD	n	n=1 to 8	"***MOD***e number" mode number of ASCII driver
BDR	m	m= 150, 200, 300, 600, 1200, 2400, 4800, 9600, 19200	"***B***au***D*** ***R***ate"
PRTY	p	p=	"***P***a***R***i***TY***"
		E	***E***ven
		O	***O***dd
		M	***M***ark ("1")
		S	***S***pace ("0")
		N	***N***one =no check
DF	q	q=0 to 5, 7, 8	"***D***ata ***F***ormat"
WCR	r	r=0 to 2550	"***W***ait time after ***C***arriage ***R***eturn" (in milliseconds)
WLF	r	r=0 to 2550	"***W***ait time after ***L***ine ***F***eed" (in milliseconds)
WFF	r	r=0 to 2550	"***W***ait time after ***F***orm ***F***eed" (in milliseconds)
DT	s	s=10 to 655350	"***D***elay ***T***ime" character delay time in millliseconds
ML	t	t=0 to 1024	"***M***ail ***L***ength" in bytes; for mode 1, 8

Parameter	Argument	Permissible Range	Meaning
Block identifier: ASC: (continued)			ASCII Driver at 2nd Interface

ME	u v	u, v=0 to 255	"***M***ail ***E***nd" end identifier; for mode 2, 3, 4, 5, 6
			RUB OUT = $7F_H$ = 127_D
			XON = 11_H = 17_D
			XOFF = 13_H = 19_D
			CR = $0D_H$ = 13_D
			LF = $0A_H$ = 10_D
			FF = $0C_H$ = 12_D
			EOT = 04_H = 4_D
			ETX = 03_H = 3_D

SLF	J/N	-	"***S***kip ***LF***"
LPP	w	w= 1 to 255	"***L***ines ***P***er ***P***age"
LM	w	w=0 to 255	"***L***eft ***M***argin"
PN	o/u	-	"***P***age ***N***umber" o=top u=bottom
HD1	"<string>"	max. 119 alpha-numeric characters	"***H***ea***D***er 1" *
HD2	"<string>"		"***H***ea***D***er 2" *
FT1	"<string>"		"***F***oo***T***er 1" *
FT2	"<string>"		"***F***oo***T***er 2" * Printer control characters are entered as hexadecimal character as follows: $xx Example: "TEXT $1B$38 FURTHER TEXT" (1B38=expanded print on)

* - CR (Carriage Return) is automatically generated in the parameter DB at the end of the ASCII string (separation of lines).
If HD 1/2, FT 1/2 are missing, only CR is stored (see Section 12.5)
- To send the $ character, $$ must be entered.
- To send the " character, $" must be entered.

Parameter	Argument	Permissible Range	Meaning
Block identifier: RKT:			**Computer Link at 2nd Interface**
			Note: Driver number "2" for the 2nd interface is set automatically.
SF	DBxDWy or	x=0 to 255	Position of the ***S***end ***M***ailbox (start of SM)
EF	DXaDWy or	y=0 to 255	Position of the ***R***eceive Mailbox (start to RM)
CBR	MBy or	a=0 to 255	Position of the ***C***oordination ***B***yte ***R***eceive
CBS	SYz	z=0 to 4095	Position of the ***C***oordination ***B***yte ***S***end
PAR			Position of the ***PAR***ameter set
MOD	n	n=1, 2	"***MOD***e number" 1=no block check character 2=send with block check character
BDR	m	m= 150, 200, 300, 600, 1200, 2400, 4800, 9600, 19200	"***B***au***D*** ***R***ate"
PRTY	p	p= E O M S N	"***P***a***R***i***TY***" ***E***ven ***O***dd ***M***ark ("1") ***S***pace ("0") ***N***one
DF	q	q=0 to 5, 7, 8	"***D***ata ***F***ormat"

PRI	l/h		"***PRI***ority"
			l=low
			h=high

DT*	s	s=10 to 655350	"***D***elay ***T***ime" (in milliseconds)
TIO*	s	s=10 to 655350	"***TI***me ***O***ut" (in milliseconds)
BWT*	s	s=10 to 655350	"***B***lock ***W***ait ***T***ime" (in milliseconds)
TTE	r	r =1 to 255	"***T***ries ***T***o ***E***rect"
TTS	r	r =1 to 255	"***T***ries ***T***o ***S***end"
Block identifier: S2T:			**Deactivate Driver**
DEAC	Y/N	-	Deactivate driver or computer link. If DEAC = "Y", driver number "0" is entered (BS46) and previously defined parameters for RKT or ASC are not transferred.

* DT<TIO<BWT

Parameter	Argument	Permissible Range	Meaning
Block identifier: CLP:			**Clock Parameter**
CLK	DBxDWy or	x=0 to 255	"***CLocK*** Data" start of clock data area
	DXxDWy or	y=0 to 255	
STW	FWb or	b=0 to 254	"***ST***atus ***W***ord" position of the status word
	SWz	z=0 to 4094	
SET	wd dd.mm.yy		Set clock time, date
	hh:mm:ss	wd =1 to 7	Weekday =Su to Sa
	AM/PM[1]	dd =01 to 31	Day
		mm =01 to 12	Month
		yy =00 to 99	Year
		hh =1 to 12	AM/PM
		00 to 23	24 hour mode
		ss =00 to 59	Seconds
TIS	wd dd.mm. hh:mm:ss AM/PM[1]		"***T***imer ***I***nterrupt ***S***et"
OHS	hhhhhh:mm:ss[2]		"***O***peration ***H***our counter ***S***et"
		hhhhhh =000000 to 999999	Hours
		mn =00 to 59	Minutes
		ss =00 to 59	Seconds
OHE	Y/N	-	"***O***peration ***H***our counter ***E***nable"
STP	Y/N	-	"***SToP***" Update clock in STOP state
SAV	Y/N	-	"***SAV***e" Save clock time after last RUN→ STOP or POWER OFF
CF	p	-400 to +400	"***C***orrection ***F***actor" Enter correction factor
Block identifier: ERT:			**Error Return**

ERR	DBxDWy or
	DXaDWy or
	MBz or
	SYp
		x=0, 2 to 255
		y=0 to 255
		a=0 to 255
		z=0 to 236
		p=0 to 4076
			"***ERR***ors" Position of the error code

[1] If an argument (e.g. Weekday) is not to be transferred, enter XX! The clock will then continue with the current value. If you specify AM or PM after the clock time, the clock will operate in the relevant 12-hour mode. If you omit this argument, the clock will operate in 24-hr mode.

[2] If an argument (e.g. minute) is not to be transferred, enter XX! The clock will then continue with the current value.

There exists no parameter block for the definiton of communication flags. If you require communication flags for the use of certain CPs, please proceed as described in Section 12.2.1. Only after the definition of the communication flags, you start to parameterize the functions described here.

11.8 DB1 Programming Example

The following example of a DB1 program shows you the complete DB1 parameterization once again.
The following have been parameterized:

- System characteristics
- Data interchange over SINEC L1
- Time-driven processing
- ASCII driver
- RK 3964 computer link (is not activated in the example)
- Integral clock
- Address for parameter error code

STL	Explanation
0: KC ='DB1 ';	DB1 header
12: KC ='# Sys. characteristics # ';	Comment
24: KC ='SDP: WD_scan monit. 500 ';	Block identifierand parameter cycle monitoring
36: KC ='RDLY_run delay 1000 ';	Restart delay
48: KC ='RT_resident_timers n ';	Retentive feature of the timers (half or all)
60: KC ='RC_resident_counters n ';	Retentive feature of the counters (")
72: KC ='RF_resident_flags n ';	Retentive feature of the flags/S flags (")
84: KC ='PROT_software protection n ';	Software protection active or not
96: KC ='PIO_inhibit n ';	Disable process image of the outputs
108: KC ='PII_inhibit n ';	Disable process image of the inputs
120: KC ='RPIC_red_PAA-Transf 10 ';	Reduced PIQ transfer (10 times)
132: KC ='PPIT_parallel_PA n ';	Parallel PI transfer
144: KC ='; ';	Block end identifier
156: KC ='# Sinec L1 parameters # ';	Comment
168: KC ='SL1: SLN_slave number 2 ';	Sinec L1 parameter; slave number
180: KC ='PGN_PG_bus number 2 ';	Programmer bus number
192: KC ='SM_send mlbx. DB60DW40 ';	Position of the send mailbox
204: KC ='RM_rec. mlbx. DB60DW0 ';	Position of the receive mailbox
216: KC ='CBS_CB_send MB61 ';	Coordination byte send
228: KC ='CBR_CB_rec. MB60 ';	Coordination byte receive
240: KC ='; ';	Block end identifier
252: KC ='# Time-driven proc. # ';	Comment

264: KC ='TFB: ';
';
	Block identifier for time-controlled processing

276: KC ='OB10_interval 400 ';	Call interval OB 10
288: KC ='OB11_interval 300 ';	Call interval OB 11
300: KC ='OB12_interval 200 ';	Call interval OB 12
312: KC ='OB13_interval 100 ';	Call interval OB 13
324: KC ='; ';	Block end identifier
336: KC ='# ASCII driver # ';	Comment
348: KC ='ASC: ';	Block identifier for ASCII driver
360: KC ='SM Send mailbox DB203DWO ';	Position of the send mailbox
372: KC ='RM rec. mailbox DB204DWO ';	Position of the receive mailbox
384: KC ='CBS CB send MB200 ';	Coordination byte send
396: KC ='CBR CB Receive MB201 ';	Coordination byte receive
408: KC ='PAR_Parameter DB202DWO ';	Position of the parameter DB
420: KC ='MOD Mode No. 6 ';	Mode number

STL (Continued)	Explanation
432: KC ='BDR_Baudrate 9600 ';	Baud rate
444: KC ='PRTY_Parity E ';	Parity (even)
456: KC ='DF_Data format 0 ';	Data format
468: KC ='WCR_Wait_after_CR 0 ';	Wait after CR
480: KC ='WLF_Wait_after_LF 0 ';	Wait after LF
492: KC ='WFF_Wait_after_FF 0 ';	Wait after FF
504: KC ='DT_Char delay_time 100 ';	Character delay time
516: KC ='ME_End charac. 0 4 ';	End character 00H 04H(=EOT)
528: KC ='SLF_Skip_LF n ';	No LF skip
540: KC ='LPP_Lines_per_Page 66 ';	Lines per page
552: KC ='LM_Left_Margin ';	Left margin
564: KC ='PN_Page_No u ';	Page number bottom
576: KC ='HD1_Header1 ';	Header 1
588: KC ='"$1B$38 LISTING C ';	ASCII string including control character
600: KC ='PU945-ASCII driver $1B';	for printer
612: KC ='$3C" ';	
624: KC ='HD2_Header2 ';	Header 2
636: KC ='"$0A====================';	ASCII string including control character
648: KC ='========================';	(here LF)
660: KC ='========================';	
672: KC ='============" ';	
684: KC ='FT1_Footer1 ';	Footer 1
696: KC ='"$0A********************';	ASCII string with control character
708: KC ='**************<Page>****';	
720: KC ='************************';	
732: KC ='************" ';	
744: KC ='FT2_Footer2 ';	Footer2
756: KC ='"$0A Exa';	ASCII string with control character
768: KC ='mple CPU945-ASCII driver';	
780: KC ='interface " ';	
792: KC ='; ';	Block end identifier
804: KC ='# Parametr. RK3964-TR # ';	Comment
816: KC ='# RKT: ';	Block identifier RK driver (is not
828: KC =' ';	activated because of # .. #)
840: KC ='SM DB223DW0 RM DB224DW0';	Send and receive mailbox
852: KC =' PAR DB222DW0 CBR MB100';	Parameter DB and CBR
864: KC =' CBS MB101 MOD 2 ';	CBS and mode number
876: KC ='BDR 9600 PRTY E ';	Baud rate and parity
888: KC ='DF 1 PRI h DT 220 ';	Data format, priority and CUC
900: KC ='TIO 2000 BWT 4000 ';	Acknowl. delay and block wait time
912: KC ='TTE 6 TTS 6 ';	Tries to erect and tries to send
924: KC ='; # ';	Block end identifier
936: KC ='# SI2-Tr. deactivate # ';	Comment
948: KC ='S2T: DEAC n ; ';	Deactivate block SI2 driver

STL (Continued)		Explanation
960:	KC ='# HW clock params. # ';	Comment
972:	KC ='CLP: ';	Block identifier HW CLOCK
984:	KC ='CLK_clock data DB2DW0 ';	Start of the clock data area
996:	KC ='STW_status word MW190 ';	Status word for clock
1008:	KC ='CF_correct._factor-12 ';	Correction factor
1020:	KC ='STP_Update_in_Stop y ';	Update in STOP
1032:	KC ='SAV_Save_time y ';	Save clock time in RN/STOP
1044:	KC ='OHE_op._hrs._counter y ';	Enable operating hours counter
1056:	KC ='SET_set_clock_time ';	Set clock time
1068:	KC ='02 05.04.93 02:00:00 pm ';	
1080:	KC ='TIS_set_time_int. ';	Timer interrupt
1092:	KC ='02 05.04. 03:XX:XX pm ';	
1104:	KC ='OHS_set_op._hrs._count. ';	Set operating hours counter
1116:	KC ='000010:10:00 ';	
1128:	KC ='; ';	Block end identifier
1140:	KC ='# DB for DB1 error # ';	Comment
1152:	KC ='ERT: ';	Block identifier for position of error DB
1164:	KC ='ERR_error code DB DW0 ';	Position of error code area
1176:	KC ='; ';	Block end identifier
1188:	KC ='END ';	DB1 end identifier
1200:		

12 Communications Capabilities

Figures

Tables

12 Communications Capabilities

This chapter informs you of

- the various communications capabilities of the CPU 945,
- the characteristics of the various communications capabilities,
- the parameterization of the CPU for communications and "initiation" of communciations,
- special requirements to be observed when using two interfaces, and
- the various interface modules that can be used in connection with the CPU 945.

12.1 Overview of the Communications Capabilities Offered by the CPU 945

The CPU 945 offers various possibilities for exchanging data with other modules/communications partners.

Data is exchanged in two different ways:

- via the S5 backplane bus
- via the CPU interfaces (SI 1 and SI 2)

Additional intelligent I/O modules (CPs/IPs) can be used in the S5-115U programmable controller. The CPU 945 communicates with these modules over the S5 backplane bus. The CPs/IPs can be used for a wide range of applications, such as communications via different LAN types (SINEC L1, L2, H1, etc.), visualization, signal preprocessing. Depending on the IPs/CPs used, the modules communicate in one of the following manners:

- via interprocessor communication flags
- via data handling blocks (page addressing/dual-port RAM)
- via the I/O area

The serial interfaces of the CPU 945 provide additional communications facilities (see Table 12.1).

In contrast to CPUs 941 to 944, the CPU 945 permits the connection of various interface modules as a second serial interface.

Interface modules available:

- Programmer interface module (15-pole)
- TTY interface module (25-pole)
- V.24 interface module (25-pole)
- RS 422-A/485 interface module (15-pole)
- SINEC L1 interface module (15-pole)
  The SINEC L1 interface module is essential for a SINEC L1 or point-to-point link with SI 2.

Table 12.1 lists the various communications capabilities permitted by the two serial interfaces of the CPU 945.

Table 12.1 Communications Capabilities with the Serial Interfaces of the CPU 945

Communications Capabilities	SI 1 Serial Interface 1	SI 2 Interface Module	
Programmer functions	Yes	Yes	Programmer submodule V.24 submodule TTY submodule RS 422-A/485 submodule SINEC L1 submodule
Communications via SINEC L1 LAN (as slave or as a slave partner in a point-to-point link)	Yes	Yes	SINEC L1 submodule
Point-to-point connection (SINEC L1 protocol; master)	No	Yes	SINEC L1 submodule
Computer link using the 3964(R) transmission protocol	No	Yes	Programmer submodule V.24 submodule TTY submodule RS 422-A/485 submodule SINEC L1 submodule
ASCII driver			

The communications capabilities

- communications via SINEC L1 LAN,
- point-to-point link (with SINEC L1 protocol),
- computer link with 3964(R) transmission protocol,
- ASCII driver

can be activated through

- parameterization in the DB1 or
- the respective defaults in the system data area.

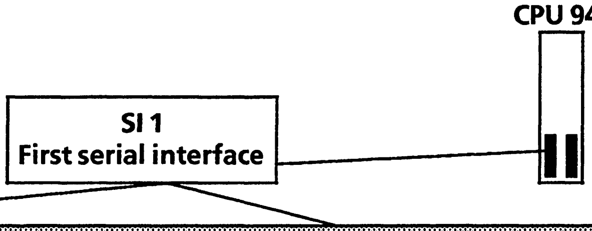
Advantages of using a second serial interface:

- Programmer/operator panel and Programmer/operator panel, also connected to CPU 945, can be operated in parallel;
- Programmer/operator panel and SINEC L1 (slave) can be connected to the CPU 945 in parallel;
- Parallel operation of programmer/operator panel and link with another partner possible via
  - point-to-point link (SINEC L1 protocol),
  - computer link with 3964(R) transmission protocol,
  - ASCII driver;
- Low-cost links with other PLCs (SIMATIC S5) via point-to-point link, e.g. CPU 945 with S5-100U (CPU 102), no CP required;
- Simple connection to SIEMENS equipment via computer link, e.g. between CPU 945 and SICOMP PC;
- Simple connection to third-party devices via ASCII driver, e.g. connection of a barcode reader to the CPU 945.

Interfacing capabilities of the first serial interface of the CPU 945

The following table provides you with an overview of

- the communications methods available for the individual communications partners at the first serial interface of the CPU 945 and
- the devices/systems that can be linked with the CPU 945 via this interface.



Communications Methods	Devices/Systems that can be Connected (Examples)	
Programmer functions (see Chapters 4/5)	● Programmers for SIMATIC S5	- PG 710 to 770, S5-DOS V6.1 and higher
	● Operator control and monitoring for SIMATIC S5	- OP 393
SINEC L1 (see Section 12.3)	● SINEC L1 LAN CPU 945 can be connected as slave in a SINEC L1 LAN*	- S5-90U, S5-95U - S5-100U (CPU 102/103) - S5-115U (CPU 941/942/943/ 944/945) - CP 521 SI - CP 530

* A number of third-party devices feature interfaces for SINEC L1 thanks to the wide-spread use of SINEC L1

Figure 12.1 Interfacing Capabilities of Interface 1 of the CPU 945

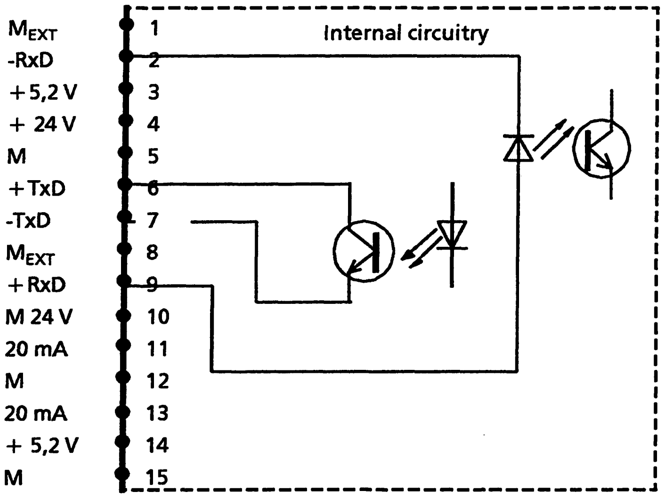
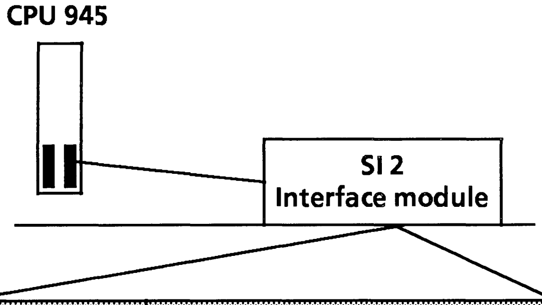


Figure 12.2 Assignments of SI 1

Interfacing capabilities of the second serial interface of the CPU 945

The following table provides you with an overview of

- the communications methods available for the individual communications partners at the second serial interface of the CPU 945 and
- the devices/systems that can be linked with the CPU 945 via this interface.



Communications Methods	Devices/Systems that can be Connected (Examples)	
Programmer functions (see Chapters 4/5)	● Programmers for SIMATIC S5	- PG 710 to 770, S5-DOS V6.1 and higher
	● Operator control and monitoring systems for SIMATIC S5	- OP 393
SINEC L1 (see Section 12.3)	● SINEC L1 LAN CPU 945 can be connected as master or slave in a SINEC L1 LAN*	- S5-90U, S5-95U - S5-100U (CPU 102/103) - S5-115U (CPU 941/942/943/944/945) - CP 521 SI - CP 530
Point-to point connection (see Section 12.4)	● Point-to-point link CPU 945 can be master or slave*	
ASCII driver (see Section 12.5)	● Devices that can receive/send data without using a transmission protocol	- Printers - Terminals - Barcode readers - Modems
3964(R) computer link (see Section 12.6)	● Devices capable of using the 3964(R) transmission protocol*	- SICOMP PC - SICOMP M - Teleperm M - Moby 1 - CP 523 - CP 524 (with special driver) - CP 525 (with special driver)

* A number of third-party devices feature interfaces for SINEC L1 and the 3964(R) transmission protocol thanks to the wide-spread use of SINEC L1.

Figure 12.3 Interfacing Capabilities of the Second Interface of the CPU 945

12.2 Data Interchange over the S5 Backplane Bus of the Programmable Controller

There are (in principle) three ways of organizing data interchange between the CPU 945 and the CPs/IPs via the backplane bus of an S5-115U programmable controller.

- Data interchange over interprocessor communication flags
- Data interchange over the I/O area
- Data interchange over data handling blocks (page addressing)

These options are described in the following sections.

12.2.1 Data Interchange over Interprocessor Communication Flags

Interprocessor communication flags are used for transmitting binary signals between the CPU 945 and some types of communications processors (e. g. CP 526).

Interprocessor communication flags are flag bytes that are read cyclically (input flags) or output (output flags) by the CPU.

Unlike other flags, however, interprocessor communication flags are stored in a special memory area on one or more CPs. This memory area comprises 256 bytes between the addresses $0F200_H$ and $0F2FF_H$.

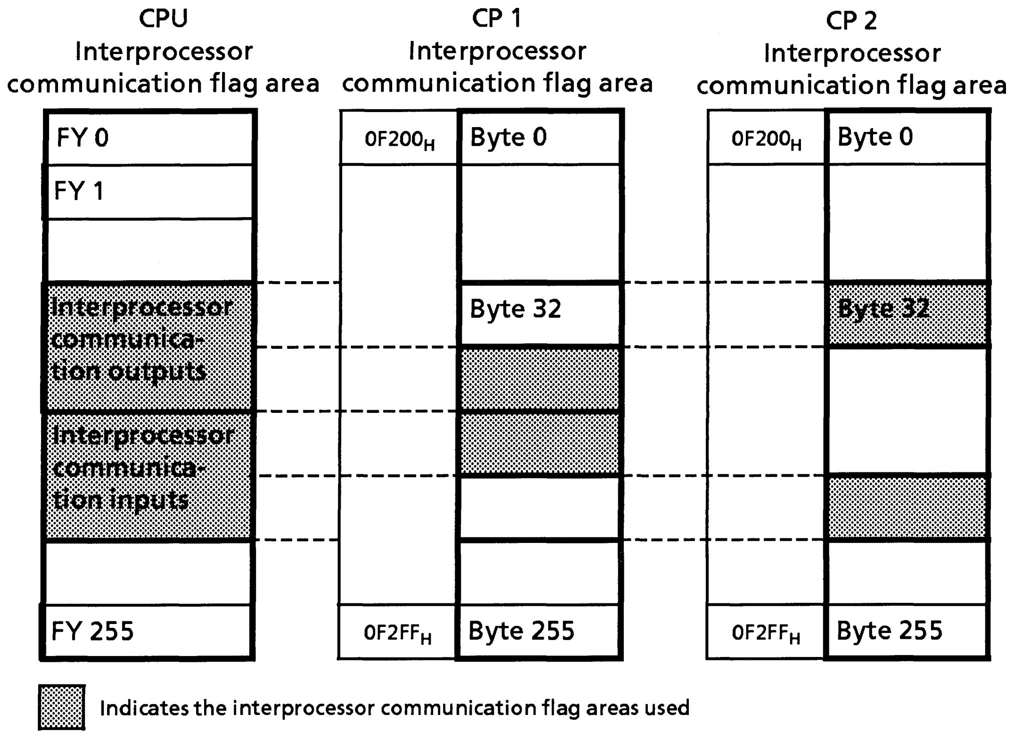


Figure 12.4 Principle of Data Interchange between CPU and CPs

The transfer of interprocessor communication flags is similar to the transfer of inputs and outputs to and from the process images. The procedure is as follows:

- The interprocessor communication input flags are read in and stored in the appropriate flag bytes prior to cyclic program execution.
- Interprocessor communication output flags are read in the relevant flag bytes and transferred to the appropriate CPs at the end of program execution.

Interprocessor communication output flags can be treated like normal flags.
Interprocessor communication input flags should be scanned only, since the setting or resetting of bits can be canceled during the next data transfer.

The control program must identify interprocessor communication flags byte by byte in data block DB1 as input flags or output flags.

Definition of the Interprocessor Communication Flags in DB1

You can program DB1 in the following two ways:

- with the help of a screen form on a programmer
- through direct input of data words

Note

If you are using interprocessor communication flags and you use DB1 as parameter DB for internal functions (see Chapter 11), then proceed as follows:

- ▶ Overall reset
- ▶ Transfer integrated DB1 from the CPU 945 to the programmer
- ▶ Insert interprocessor communication flag agreements (as described below) ***before the DB1 parameters awaiting interpretation (see Chapter 11)***
- ▶ ***Modify and expand the other DB1 parameters (see Chapter 11)***
- ▶ ***Transfer the modified and expanded DB1 to the PLC***

The first three data words form the header ID. Always program them as follows:

DW 0	:	KH = 4D41	or	KS = MA
DW 1	:	KH = 534B	or	KS = SK
DW 2	:	KH = 3031	or	KS = 01

After specifying an ID for the operand area, enter the numbers of all flag bytes used. Conclude the interprocessor communication flag list with an end ID. The IDs are as follows:

KH = CE00	for	interprocessor communication input flags
KH = CA00	for	interprocessor communication output flags
KH = EEEE	for	end

You can use a total of 256 bytes as interprocessor communication flags. Number the bytes in relation to the start address of the interprocessor communication flag area (FY 0 to FY 255). The end identifier can be followed by the DB1 section in which internal functions are parameterized (see Chapter 11).

Example: Define flags bytes FY10, FY20, and FY30 as interprocessor communication input flags. Define flag bytes FY11 and FY22 as interprocessor communication output flags.

Assign DB1 as follows:

```
DW  0   :   KH  =  4D41
    1   :   KH  =  534B     Header ID
    2   :   KH  =  3031     (KS = MASK 01)

DW  3   :   KH  =  CE00
    4   :   KF  =  +10      Interprocessor communication
    5   :   KF  =  +20      input flags
    6   :   KF  =  +30

DW  7   :   KH  =  CA00
    8   :   KF  =  +11      Interprocessor communication
    9   :   KF  =  +22      output flags

DW  10  :   KH  =  EEEE     End ID
```

The following points apply to the assignment of DB1:

- The interprocessor communication flag definitions must always start from DW 0 in DB1, i. e. they must ***precede*** the parameter data to be interpreted.
- You can enter interprocessor communication flag areas in any order.
- You can enter the byte numbers for an area in any order.
- The CPU accepts the entries in DB1 only during Manual or Automatic Restart. You must therefore execute a program restart each time you modify DB1.

Signal Exchange with a CP

Set jumpers on the CP to enable the area as required by the interprocessor communication flag bytes. The jumpers divide the area between bytes 0 (0F200) and 255 (0F2FF) into eight blocks of 32 bytes each.

Normally the entire interprocessor communication flag area is enabled. Setting is necessary only when you use several CPs with interprocessor communication flags.

Specify the desired interprocessor communication flags in DB1. The bytes must be in the set area. You can choose any bytes from this area. However, use only as many bytes as necessary to keep the transfer time as short as possible.

Example: 20 interprocessor communication flag bytes are needed for a signal exchange:

- 14 bytes to transfer information to the CP
- 6 bytes to fetch information from the CP

The jumper setting on the CP enables the area between byte 128 (0F280) and byte 159 (0F29F).

The interprocessor communication flags are defined in DB1 as follows:

Outputs: FY 128 ... 141
Inputs: FY 142 ... 147

The words in the DB are assigned as follows:

```
DW  0   :   KH  =  4D41
    1   :   KH  =  534B     Header ID
    2   :   KH  =  3031

DW  3   :   KH  =  CE00
    4   :   KF  =  +142
    5   :   KF  =  +143
    6   :   KF  =  +144     Interprocessor communication input
    .                       flags
    .
    .
DW  9   :   KF  =  +147

DW  10  :   KH  =  CA00
    11  :   KF  =  +128
    12  :   KF  =  +129     Interprocessor communication output
    .       .         .     flags
    .       .         .
    .       .         .
DW 24       KF  =  +141

DW  25  :   KH  =  EEEE     End ID
```

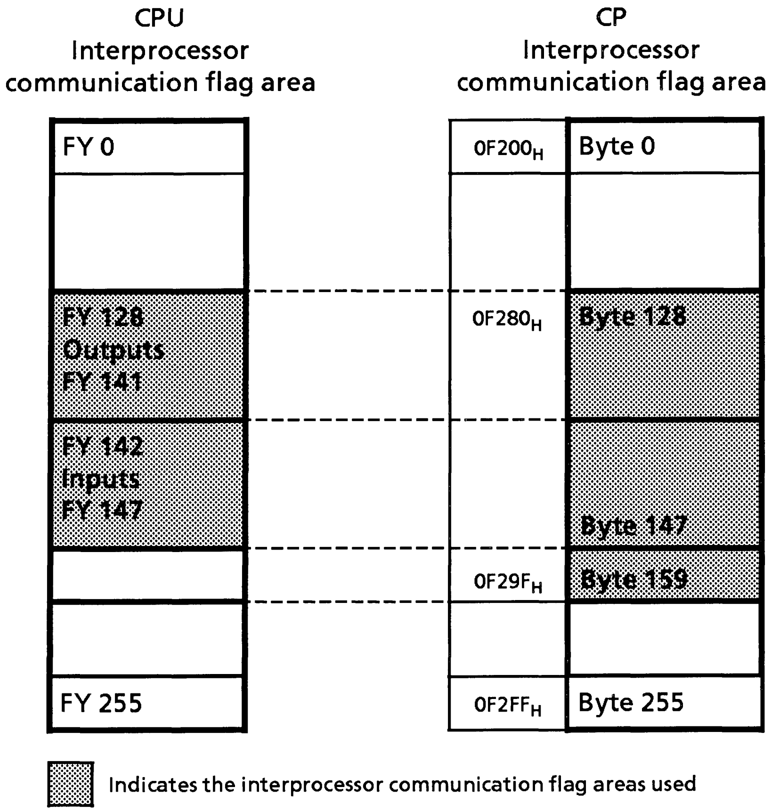


Figure 12.5 Interprocessor Communication Flag Areas Used for Signal Exchange with a CP (Example)

Special Points to Observe when Using the CP 525 and CP 526 in RESTART Mode

Note

If the CP 525 and CP 526 are used in the S5-115U, the interprocessor communication flag area enabled on the CPs should be reset on restart in connection with the following CP functions:

CP 525 (6ES5 525-3UA11):

- Component: Event printer if group disable bits are used
- Component: Operator-process communication and visualization with the 3975 display unit if bit set and reset commands are used

  general: Group disable bits should always be located in the interprocessor communication flag area enabled per jumper setting.

CP 526 (6ES5 526-3Lxxx):

- Basic board: If bit set and reset commands are used

Before synchronizing the CPs, an FB should be called in OB21/22. This FB should be programmed as shown in the following example:

Example: Function block FBxxx (e.g. FB11) for resetting the interprocessor communication flag area on a CP.
The communication flag areas enabled by jumpers on the CP can be reset with the following block. This FB must be specified with its starting flag byte (V-FY) and end flag byte (B-FY) for each contiguous communication flag area.

If a flag byte that does not define an area boundary is specified here, the entire area is still reset.

V-FY	:	FY 35	(from)
B-FY	:	FY 165	(to)

This resets the communication flag area from flag byte FY 32 to flag FY 191. This area must naturally have been enabled on the CP.

FB11	STL	Description
NAME : K-FY	LOE	FB for resetting IPC flags
DECL : V-FY	I/Q/D/B/T/C: I BI/BY/W/D: BY	
DECL : B-FY	I/Q/D/B/T/C: I BI/BY/W/D: BY	
:	LW =B-FY	Load relative end address
:	L KH 00E0	Refer to 32-byte area
:	AW	
:	ADD DH 0000 F21C	Add to starting address
:	T FW 252	+ 28 for the IPC area
:		and buffer result
:	LW =V-FY	Relative end address
:	L KH 00E0	Refer to 32-byte area
:	AW	
:	ADD DH 0000 F200	Add to starting address
:		of IPC area
:	JU =M002	
M001 :	TAK	Loop for deleting the IPC flag
:	ADD KF +4	
M002 :	L KB 0	
:	TAK	
:	TDI A2	
:	L FW 252	
:	<D	
:	JC =M001	
:	BE	

Signal Exchange with Several CPs

If one CPU addresses several CPs, one or more interprocessor communication flag areas must be enabled on each CP. When setting the jumpers on the CPs, please note the following points:

- The areas on the individual CPs must not overlap (to prevent duplicate address assignment).
- The areas on the individual CPs do not have to be assigned consecutively.

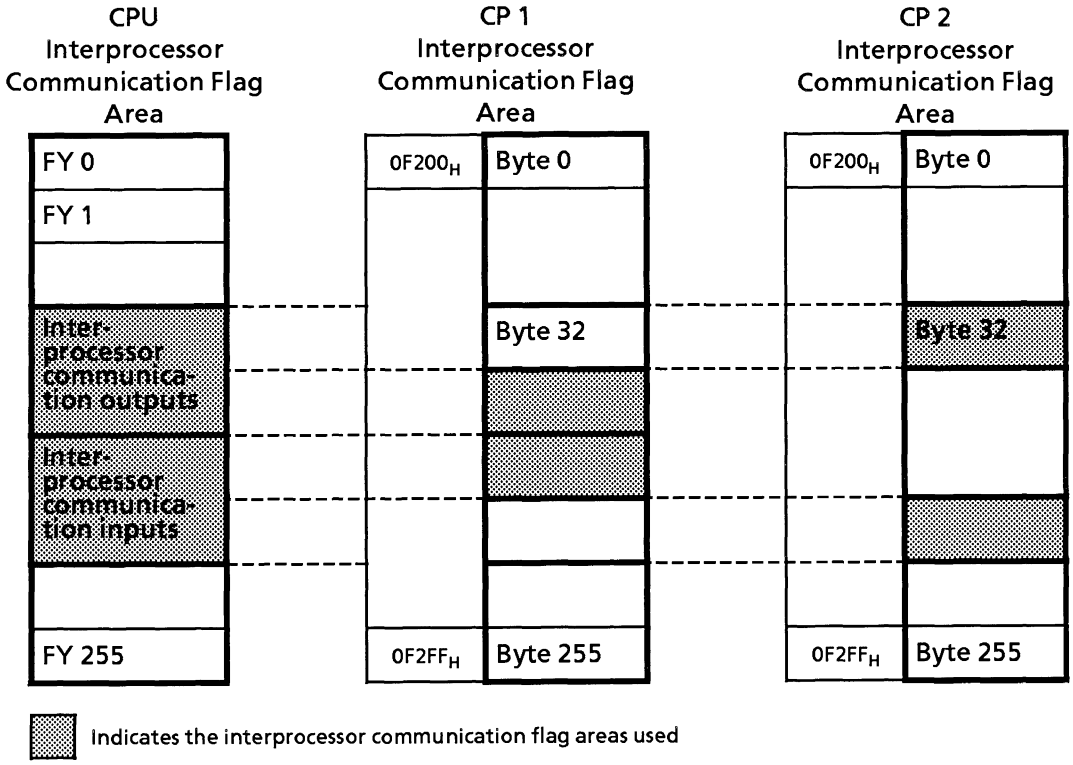


Figure 12.6 Interprocessor Communication Flag Areas when Several CPs Are Used

Define the interprocessor communication flag bytes in data block DB1 as described above.

Example: Use one CPU to address two CPs. Table12.2 shows the flag bytes needed and possible numbering.

Table 12.2 Definition of Interprocessor Communication Flags when Two CPs Are Used (Example)

CPs	Number of Control Bytes (Outputs)	Number of Scan Bytes (Inputs)	CP Flag Areas Set	CPU Interprocessor Communication Output Flags	CPU Interprocessor Communication Input Flags
CP 1	8	4	Byte 128 to 159	FY 128 to 135	FY 156 to 159
CP 2	6	10	Byte 160 to 191	FY 170 to 175	FY 160 to 169

12.2.2 Data Interchange over the I/O Area

Low-range signal-preprocessing IPs or CPs generally communicate via the standard I/O area for digital and analog I/O: 0F000$_H$ to 0F0FF$_H$ (also via the Q area). In these cases, the basic load and transfer operations, such as L PY, L PW, T PY, T PW are used for data interchange.

12.2.3 Data Interchange over Data Handling Blocks FB244 to FB249

Complex functions are handled by those modules of the SIMATIC S5 system that can be programmed and parameterized (CPs and IPs).
These modules incorporate a dual-port RAM (page) of either 1 × 2K or 2 × 2K bytes for data interchange between CPU and CP/IP.
In the CPU 945, this buffer area is assigned an address range (0000 F400$_H$ to 0000 F7FF$_H$ or 0000 F400$_H$ to 0000 FBFF$_H$) which can be accessed via a page.

Since all CPs/IPs use the same address range (0000 F400$_H$ to 0000 FBFF$_H$) for page addressing, the address range must be allocated to a specific CP/IP with the aid of a particular page number (= interface number):

- The CP/IP is assigned an "address" by setting the interface number (0 to 255).
- In the PLC program, the CP/IP is addressed via programmable "data handling blocks". The data handling blocks are used to enter the desired interface number in the page select register (address: 0000 FEFF$_H$) and then address the required CP/IP. They handle communication between the CPU and the CP/IP autonomously.

Blocks FB244 to FB249 permit the use of communications processors and intelligent I/O modules. These so-called data handling blocks coordinate data interchange between the CPU and the page area of these modules. Data handling blocks require neither program memory space nor flag, timer or counter areas.

Parameters

The data handling blocks use the parameters listed in Table 12.3. The individual parameters are explained on the following pages.

Table 12.3 List of Parameters Used

Designation		Meaning
SSNR	:	Interface number (page number)
A-NR	:	Job number
ANZW	:	Job status word (double word)
QTYP/ZTYP[1]	:	Type of data source or destination
DBNR[1]	:	Data block number
QANF/ZANF[1]	:	Relative start address for type
QLAE/ZLAE[1]	:	Number of source/destination
PAFE[2]	:	Parameter assignment error byte
BLGR	:	Frame size

[1] If these parameters are not required for a call (e. g. ALL function), they can be skipped by means of "CR" when initializing the block

[2] Only direct parameterization possible

Parameter Description

The formal operands that you must supply when using data handling blocks are explained below.

SSNR - Interface Number

The SSNR parameter specifies the logical number of the interface (page) to which a particular job refers.

12

Parameter		Assignment		
Type	Format			
Data (Byte)	KY	KY=	x, y	
			x=0	Direct parameterizing
			y=0 to 255	Interface number (page address)
			x≠0	Indirect parameterization
			y=0 to 255	Data word number. The SSNR, A-NR, and ANZW parameters are stored beginning with the next data word in the current DB (or DX).

A-NR - Job Number

The jobs for an interface are characterized by this number.

Parameter		Assignment	
Type	Format		
Date (byte)	KY	KY= x, y	Parameter x is ignored "y" represents the job number.
		y=0	ALL function [1]
		y=1 to 223	Direct function Number of the job to be executed. [2]

[1] The "ALL" function is not permitted for the FETCH block.

[2] Refer to the SINEC L1 Local Area Network manual for an explanation of the individual job numbers.

ANZW - Job Status Word

Use this parameter to specify the address of a double word (DW* n/DW n+1 or FW n and FW n+2) that indicates the processing status of a particular job.

Parameter		Assignment	
Type	Format		
Address (word)	W	0 to 255	Address of the job status word for direct parameterization Permissible areas: FW 0 to 252 DW 0 to 255

* DW refers to the current data block.

QTYP/ZTYP - Type of Data Source or Data Destination

Assign these parameters ASCII characters that specify the type of data source (for SEND) or data destination (for RECEIVE or FETCH).

Parameter		Assignment
Type	Format	
Data (character)	KS	KS= DB, DX, QB, IB, FY, TB, CB, AS, RS, PB (see Table 12.4) Direct parameterization: The specification of the data source (or data destination) is specified directly in the QTYP/ZTYP, DBNR, QANF/ZANF, QLAE, ZLAE parameters. KS=NN No parameter assignment: Data source (or data destination) specifications are located in the job on the CP. KS=RW, XX Indirect parameterization: Specifications for data source (or data destination) are located in a data area specified by the DBNR and QANF/ZANF parameters.

DBNR - Data Block Number

If DB, DX, RW or XX were assigned to the parameters QTYP/ZTYP, the DBNR parameter must specify the number of the required data block.

Parameter		Assignment
Type	Format	
Data (byte)	KY	KY= 0, y y=0 to 255 Number of the data block containing the data

QANF/ZANF - Start Address of the Source or Destination Data Block

When parameterization is indirect (QTYP/ZTYP=RW or XX), specify the number of the DW at which the parameter block begins.
When parameterization is direct, QANF/ZANF refers to the specified area.

Parameter		Assignment
Type	Format	
Data (fixed-point number)	KF	Permissible range (see Table 12.4)

QLAE/ZLAE - Length of the Source or Destination Data Frame

When parameterization is direct, the source or destination type specification is understood to be the number of either bytes or words.

Parameter		Assignment	
Type	Format		
Data (fixed-point number)	KF	Permissible range	(see Table 12.4)
		- 1 :	The "joker length" - 1 means the following: ● for RECEIVE: As much data as the transmitter sends or as much as area limitations permit. ● for SEND: Data is transmitted until a particular area boundary is reached.

Summary:

Table 12.4 QTYP/ZTYP Parameters

	QTYP/ZTYP Description	DBNR Meaning Area Permitted	QANF/ZANF Meaning Area Permitted	QLAE/ZLAE Meaning Area Permitted
NN	No source/ destination parameters in the block. Parameters have to be in the CP.	Irrelevant	Irrelevant	Irrelevant
XX	Indirect addressing: parameters are stored in the data block (specified with DBNR and QANF).	DB/DX in which the source/destination parameters are stored 0 to 255	DW number with which the parameters begin 0 to 2044	Irrelevant
RW	Indirect addressing without data exchange. Source or destination parameters are stored in a DB.[1]	DB/DX in which the source/destination parameters are stored 0 to 255	DW number with which the parameters begin 0 to 2040	Irrelevant
DB	Source/destination data from/to the data block in program memory	DB from which source data are taken or to which destination data are transferred 0 to 255	DW number beginning with which the data is to be read or written 0 to 2047	Length of the source/destination data in words 0 to 2048
DX	Source/destination data from/to the data block (DX) in program memory	DX from which source data are taken or to which destination data are transferred 0 to 255	DX number beginning with which the data is to be read or written 0 to 2047	Length of the source/destination data in words 0 to 2048
FY	Source/destination data from/to the flag area	Irrelevant	Flag byte number beginning with which the data is to be read or written 0 to 255	Length of the source/destination data in bytes 0 to 256
QB	Source/destination data from/to the process output image (PIQ)	Irrelevant	Output byte number beginning with which the data is to be read or written 0 to 127	Length of the source/destination data in bytes 0 to 128

[1] Assigning RW to ZTYP is not permitted for the RECEIVE block.

Table 12.4 QTYP/ZTYP Parameters (Continued)

	QTYP/ZTYP Description	DBNR Meaning Area Permitted	QANF/ZANF Meaning Area Permitted	QLAE/ZLAE Meaning Area Permitted
IB	Source/destination data from/to the process input image (PII)	Irrelevant	Input byte number beginning with which the data is to be read or written 0 to 127	Length of the source/destination data in bytes 0 to 128
PB	Source/destination data from/to I/O modules. Source data from input modules. Destination data to output modules	Irrelevant	Peripheral byte number beginning with which the data is to be read or written 0 to 127 digital I/Os 128 to 255 analog I/Os	Length of the source/destination data in bytes 0 to 256
ZB	Source/destination data from/to counter locations	Irrelevant	Number of the counter location beginning with which the data is to be read or written 0 to 255	Length of the source or destination data in words (counter location = 1 word) 0 to 256
TB	Source/destination data from/to timer locations	Irrelevant	Number of the timer location beginning with which the data is to be read or written 0 to 255	Length of the source or destination data in words (timer location = 1 word) 0 to 256
AS	Source/destination data from/to memory locations absolute-addressed	64K area address (page address) 0.0 to 0.14	Offset address in the 64K area beginning with which the data is to be read or written. (Must be specified in KF, interpreted as KH) 0 to +32767 - 32768*	Length of the source/destination data block in words 0 to 32767
BS	Source/destination data from/to system data area or extended system data area	Irrelevant	Number of the system data word beginning with which the data is to be read or written. 0 to 255≙BS 256 to 511≙BT	Length of the source/destination data in words 0 to 512

* Permissible ranges:
1: 00000_H to $0FFFF_H$ (S5 bus)
2: 20000_H to $205FF_H$ (C, T, F, PII, PIQ)
3: 80000_H to $DFFFF_H$ (program memory)
4: $E0000_H$ to $E13FF_H$ (S flags, BS and BT ranges)

BLGR - Frame Size

The BLGR parameter specifies the maximum size of the data frame that can be exchanged between a PLC and a CP during one pass of the data handling block (applies only to SYNCHRON).

Parameter		Assignment		
Type	Format			
Data (byte)	KY	KY =0,y	Frame size	Execution Time (SEND and RECEIVE)
		y =0	64 bytes *	Execution time with RLO=0 (empty pass) SEND/RECEIVE: $t \approx 54\mu s + 4 \times$ ready delay time ** Execution time including data transfer SEND: $t \approx 1070\mu s + 30 \times$ ready delay time ** $+ n \times (2.5\mu s +$ ready delay time **) RECEIVE: $t \approx 1280\mu s + 20 \times$ ready delay time ** $+ n \times (5\mu s +$ ready delay time **) with n being the minimum value for message frame length or frame size (in bytes)
		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 to 254	same as for y=0	

* The block uses the default parameter. (On the S5-115U, the frame size is set to 64 bytes).

** See Table 12.5.

Table 12.5 Ready delay times of the CPs and IPs

Communications Processor	Ready Delay Time in µs
CP 523	3 to 100
CP 524	1
CP 525	3
CP 526	3
CP 530	3 to 130
CP 535	3
CP 551	3
CP 552	3
IP 252	10
IP 246	1.5
IP 247	1.5
CP 527	3
CP 5430	1
CP 143	3

PAFE - Parameter Assignment Error Byte

For PAFE, specify a byte that is set if the block detects a parameter assignment error. The following can be parameter assignment errors:

- No such interface.
- The QTYP/ZTYP, QANF/ZANF or QLAE/ZLAE parameters were assigned incorrectly.

Parameter		Assignment
Type	Format	
Address (byte)	BY	FY 0 to 255

Direct and Indirect Initialization of SSNR, ANR, ANZW or BLGR

The high-order byte of the SSNR parameter is the selection criterion for direct or indirect parameterization:

- High-order byte of SSNR = 0 means direct parameterization
  SSNR, A-NR, ANZW or BLGR are specified directly in the block.
- High-order byte of SSNR ≠ 0 means indirect parameterization
  SSNR, A-NR and ANZW/BLGR are stored in the current data block (DB or DX), beginning with the data word specified in the low-order byte of the SSNR parameter.

SSNR and A-NR have the same data format (KY) in both cases. Representation formats are different for the job status word. While the address of the job status word is specified directly when parameterization is direct (e.g. FW 100), a double word is used for indirect parameterization. The first data word specifies the data area in the KS data format (ASCII code).

Meanings:		
	FW	Job status word is located in the flag area
	DB	Job status word is located in the DB data block
	DX	Job status word is located in the DX data block

The second data word includes the ANZW address in the KY data format and, if the job status word is located in a DB/DX, additionally the block number (in the first byte of the KY format).

Examples:

Direct parameter assignment of SSNR, A-NR and ANZW

- Job status word as flag

Parameter Assignments	Explanation
:JU FB 245	
NAME :RECEIVE	
SSNR : KY 0,3	Interface is assigned number 3
A-NR : KY 0,100	The job number is 100
ANZW : FW 240	Flag words 240 and 242 are used as job status words.

- Job status word in a data block

Parameter Assignments	Explanation
:C DB 47	DB47 is activated
:JU FB 247	
NAME :CONTROL	
SSNR : KY 0,3	The interface number is 3
A-NR : KY 0,100	The job number is 100
ANZW : DW 40	Data words 40 and 41 in DB47 are used as job status word.

Indirect Initialization of SSNR, A-NR and ANZW

- Job status word as flag

Parameter Assignments	Explanation
:C DB 44	Open DB44
:JU FB 244	
NAME :SEND	
SSNR : KY 255,1	ID for indirect parameterization
.	The data area for parameterization
.	begins with DW 1.
A-NR : KY 0,0	Irrelevant
ANZW : FW 0	Irrelevant
DB44	
DW 1: KY = 0,1	The interface number is 1
DW 2: KY = 0,31	The job number is 31.
DW 3: KS = FW	The job status word is in the flag area.
DW 4: KY = 0,200	The job status word is represented in flag words FW 200 and FW 202.

- Job status word in a data block

Parameter Assignments	Explanation
:C DB 24	Open DB24
:JU FB 244	
NAME :SEND	
SSNR : KY 255,1	ID for indirect parameterization
.	The data area for parameterization
.	begins with DW 1.
A-NR : KY 0,0	Irrelevant
ANZW : FW 0	Irrelevant
DB24	
DW 1: KY = 0,1	The interface number is 1.
DW 2: KY = 0,31	The job number is 31.
DW 3: KS = DB	The job status word is in a data block.
DW 4: KY = 222,10	Address of the ANZW (DW 10 and DW 11 in DB222)
DB222	
DW 10	Job status word
DW 11	

Indirect Initialization of SSNR and BLGR (SYNCHRON)

Parameter Assignments	Explanation
:C DB 49	Open DB49
:JU FB 249	
NAME :SYNCHRON	
SSNR : KY 255,100	ID for indirect parameterization The data area for parameterization begins with DW 100.
BLGR : KY 0,0	Irrelevant
DB49	
DW 100: KY = 0,10	The interface number is 10.
DW 101: KY = 0,6	The block size is set at 512 bytes.

Direct and indirect parameterization of QTYP/ZTYP, DBNR, QANF/ZANF and QLAE/ZLAE

When ***direct*** parameterization is selected, the data handling block processes the source and destination parameters (including QTYP/ZTYP, DBNR; QANF/ZANF and QLAE/ZLAE) stated in the block call directly.

In the case of ***indirect*** parameterization, the block parameters point at a parameter area in a data block which contains the actual source or destination parameters.

For indirect parameterization of the source and destination parameters, "XX or "RW" (KS data format) is to be used for the type of data source or data destination. For DBNR, the number of the DB or DX data block must be specified in the low-order byte. If the high-order byte is equal to 0, the data handling block expects to find the parameter list in a DB data block; otherwise (high-order byte not equal to 0), in a DX data block. The relative start address for a type (QANF/ZANF) contains the word number where the parameter list starts. QLAE/ZLAE are irrelevant.

For operations with type of data source "AS" (=absolute addressing of memory locations), the DBNR parameter is used to define the required 64K memory area via the four most significant address bits.

In this case, the QANF/ZANF parameter describes the least significant (16-bit) portion of the 20-bit address.

When XX is used for indirect parameterization, enter the following data in the data block specified by the formal operand "DBNR":

Address in the Data Block	Parameter Type	Assignment	Explanation
QANF +0	KS	DB, DX, QB, IB, FY, TB, CB, AS, NN, PB, RS	Type of source or destination
+1	KY	0,0 to 0,255*	Number of the DB for source or destination type DB (high-order byte=0)
+2	KF	0 to 2047*	Start address of the source or destination area QANF/ZANF
+3	KF	1 to 2048* (-1 =joker length)	Length of the source or destination area

* Only if "DB/DX" was previously assigned (otherwise according to type of data, see Table 12.4)

For indirect parameterization with RW, the data in the block with the "DBNR" number must contain the following information:

Address in the Data Block	Parameter Type	Assignment	Explanation
QANF +0	KS	DB, DX, QB, IB, FY, TB, CB, AS, RS, PB, NN	Source type specification
+1	KY	0,0 to 0,255*	Number of the DB for source type "DB" (high-order byte=0)
+2	KF	0 to 2047*	Start address of the source data block
+3	KF	1 to 2048* (-1 to joker length)	Source data block length
+4	KS	DB, DX, QB, IB, FY, TB, CB, AS, NN, PB, RS	Destination type specification
+5	KY	0,0 to 0,255*	Number of the DB for destination type "DB" (high-order byte=0)
+6	KF	0,0 to 2047*	Start address of the destination data block
+7	KF	1 to 2048* (-1 to joker length)	Destination data block length

* Only if "DB/DX" was previously assigned (otherwise according to type of data, see Table 12.4)

Examples:

FB244 STL			Explanation
JU		FB 244	
NAME:		SEND	
SSNR:	KY	0,21	
A-NR:	KY	0,33	
ANZW:	FW	100	
QTYP:	KS	XX	Indirect parameterization
DBNR:	KY	0,25	Parameters are in DB 25
QANF:	KF	11	from DW 11 onwards
QLAE:	KF	0	Irrelevant
PAFE:	FY	13	

DB 25		Explanation	
DB	25		
DW 11:	KS DB	Parameter QTYP = DB	(QDB=
DW 12:	KY 0,17	Parameter DBNR = 17	DB17
DW 13:	KF +3	Parameter QANF = 3	from DW3)
DW 14:	KF +5	Parameter QLAE = 5	

Absolute addressing, AS

Absolute address (hexadecimal)	`E 0200`
- 64K area address	`E (as decimal number: 14)`
- Least significant part of address (hexadecimal):	`0200`
or as fixed-point number:	`512`

FB244 STL	Explanations
`JU          FB 244`	
`NAME:       SEND`	
`SSNR:  KY   0.21`	
`A-NR:  KY   0,33`	
`ANZW:  FW   100`	
`QTYP:  KS   AS`	Operation with AS
`DBNR:  KY   0,14`	64K area: 14
`QANF:  KF   +512`	Least significant part of address 512 ≙ 200_H
`QLAE:  KF   +200`	
`PAFE:  FY   13`	

Format and Meaning of the Job Status Word

The job status word is used to store information on the status of jobs. Specify the address of the job status word when assigning parameters. Starting at this address, information can be read out and processed further.

Assign parameters to the ANZW such that a separate job status word is addressed for each job defined.

The job status word is part of a double word that is addressed by the ANZW parameter.

Table 12.6 Basic Format of the Double Word for the Job Status

Word No.	Meaning
n	Job status word
n+1	Length word

Job Status Word

The job status word is divided into four parts. Figure 12.7 explains the individual bits.

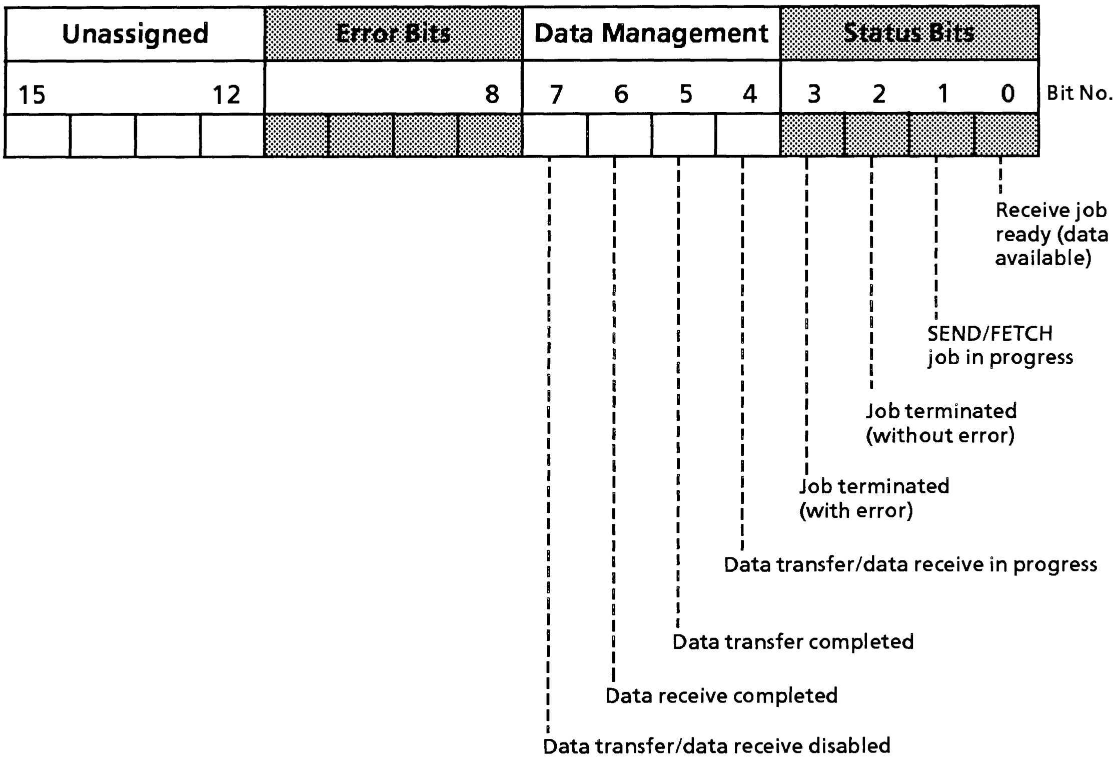

Unassigned
Error Bits
Data Management
Status Bits
15
12
8
7
6
5
4
3
2
1
0
Bit No.
Receive job ready (data available)
SEND/FETCH job in progress
Job terminated (without error)
Job terminated (with error)
Data transfer/data receive in progress
Data transfer completed
Data receive completed
Data transfer/data receive disabled

Figure 12.7 Format of the Job Status Word

Description of the Error Bits:

Error bits in the job status word are valid only if the "Job terminated with error" bit (bit 3) is set. Table 12.7 lists the possible errors.

Table 12.7 Description of the Error Bits

Value of the Error Tetrad	Error
0	No error If the "Job terminated with error" bit is set nonetheless, this means that the CP has set up the job again after a Cold Restart or a RESET.
1 to 5	PLC error, error code the same as in PAFE
6 to F	CP error CP-specific errors. Use the appropriate CP manual to determine the cause of the error.

Description of the Status and Data Management Bits

The status bits and the data management bits can be set/reset and evaluated both by the user and via data handling blocks.

The table below shows the situations in which these bits are set or reset.

Table 12.8 Meanings of bits 0 to 7 in the job status word

Bit No.	Set by	Reset/ Overwritten by	Evaluated by
0 Receive job ready (data available)	DHB	DHB	● RECEIVE block (When this bit is set, handshaking with the CP is initiated). ● User (Scan to see if there is a frame).
1 SEND/FETCH job running	DHB (as soon as the CP receives a job request)	DHB (when the CP serviced the job request)	● SEND/FETCH block (A new job is sent only after the old job has been processed). ● User (Scan to see if a new job is to be initiated).
2 Job completed (without error)	DHB (if the job was completed without error)	DHB (if the job is reinitiated)	User (Scan to see if the job was completed without error).
3 Job completed (with error)	DHB (if the job terminated with error). (The cause of the error is stored in the high-order byte of the job status word).	DHB (if the job is reinitiated)	User (Scan to see if the job was completed without error).

Table 12.8 Meanings of bits 0 to 7 in the job status word (continued)

Bit No.	Set by	Reset/ Overwritten by	Evaluated by
4 Data transfer/ reception running	DHB/SEND, RECEIVE (if data exchange has begun for a job Example: initiation with DIRECT function but exchange via ALL function	DHB/SEND, RECEIVE (if data exchange is completed for a job)	User (Scan to see if the data frame has just been transferred). [1]
5 Data transfer completed	SEND block (if data has been transferred for a job)	● SEND block (if data transfer has been started for a new job) ● User (if an evaluation was made)	User (Scan to see if the dataset for a job has already been transferred to the CP and when a new dataset can be made available for a current job).
6 Data reception completed	RECEIVE block (if data reception has been concluded for a job)	● RECEIVE block (if data transfer has been started for a new job) ● User (if an evaluation was made)	User (Scan to see if the data frame of a new job has already been transferred to the PLC and when a new data frame was transferred to the PLC for a job currently in progress).
7 Data transfer/ reception blocked	User (Access of the SEND and RECEIVE blocks to an area is prevented at the first data frame. Jobs already started are terminated).	User (The relevant data area is enabled).	SEND-RECEIVE block (If the bit is set, the blocks do not execute any data traffic. Instead, they report an error to the CP).

[1] During data transfer between the CP and PLC, you can no longer modify the data for a job. This fact is not critical for small data packets since, in this case, data exchange can be handled in one block pass. However, large amounts of data can be transferred in blocks only. Consequently, data exchange can stretch over several program scans, depending on the frame size specified in the SYNCHRON block.

Meaning of the job status word for SEND ALL and RECEIVE ALL

For the ALL functions, the SEND and RECEIVE blocks enter the job number for which they were active in the current pass in the low-order byte. Job number "0" (empty run) means that no job was processed.

Length Word:

In the length word, the SEND and RECEIVE data handling blocks enter the amount of data (in bytes) already transferred for a particular job. Table 12.9 shows how the length word is acted upon.

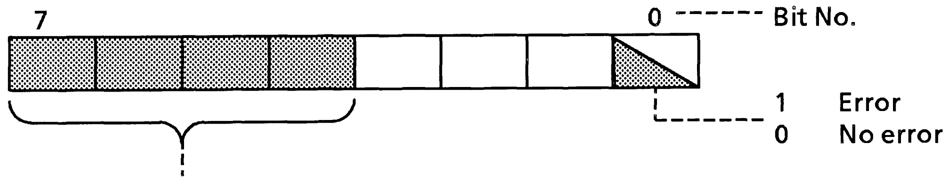
Table 12.9 Accessing the Length Word

Set by	Reset/ Overwritten by	Evaluated by
HTB/SEND, RECEIVE (during data exchange) The contents are calculated from the current number of transfers plus the quantity of (blocked) data already exchanged.	HTB/SEND,RECEIVE FETCH by over-writing during the next job.	User (If bit 2, 5, or 6 is set in the job status word, the current source or destination length is in the length word. If bit 3 is set, the length word contains the amount of data transferred prior to detection of an error).

"Parameter Assignment Error" (PAFE) Byte

Only a flag byte is suitable as a condition code byte.

Various parameter assignment errors are reported in the PAFE byte (in the high-order tetrad). When assigning parameters, specify the address under which this information can be accessed. Figure 12.8 describes the individual bits.



0	No error
1	Incorrect ORG format, QTYP/ZTYP illegal (PLC or CP)
2	Area not available (no DB or illegal DB)
3	Area too small
4	Time-out (QVZ)/no access possible
5	Wrong job status word
6	No source or destination parameters for SEND/RECEIVE ALL (system error)
7	No such interface
8	Interface not ready
9	Interface overloaded
A	Unassigned
B	Illegal A-NR
C	No acknowledgement or negative acknowledgement from the interface (CP)
D	Parameter/BLGR illegal (first byte)
E	Unassigned
F	Unassigned

Figure 12.8 Format of the "PAFE" Byte

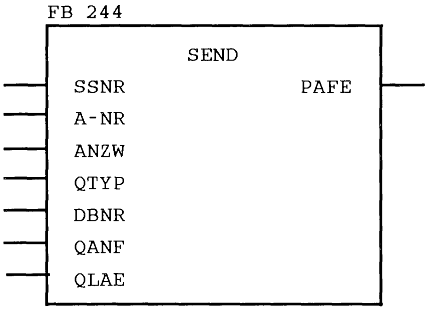
The SEND block - FB 244 -

The FB 244 requests that data be sent to a module with page addressing.

A distinction is made between two function modes:

- SEND ALL
  In this mode, the function block is used for so-called background communications and can transfer data to a CP/IP if requested by the same.
- SEND DIRECT
  A specific send request can be initiated and the required data transmitted at the same time.

Calling Function Block FB244 (Example: SEND DIRECT)

STL	CSF/LAD
:JU FB 244 NAME :SEND SSNR : KY 0,10 A-NR : KY 0,32 ANZW : FW 14 QTYP : KS DB DBNR : KY 0,10 QANF : KF +1 QLAE : KF +33 PAFE : FY 13 : ***	 FB 244 SEND SSNR PAFE A-NR ANZW QTYP DBNR QANF QLAE

Description of the SEND ALL Function

For the SEND ALL function, the block requires the following parameters:

- SSNR - interface number
- A-NR - job number (assign "0")
- ANZW - job status word
- PAFE - parameter assignment error byte

All other parameters are irrelevant for this job.

The CP uses the communications area to provide the following information:

- address of the job status word
- type of data
- amount of data
- start address of the data area

The following bits are evaluated or set/reset in the job status word for the pertinent job:

- data transfer disabled
- data transfer completed
- data transfer in progress

The SEND block enters the number of bytes transferred in the data word that follows the job status word of the SEND DIRECT job.

The SEND block must be called in the control program in "ALL" mode at least once per interface if

- the CP can request data from a PLC on its own initiative, e.g., the CP 526 for display output or the CP 143 with the job mode "READ PASSIVE".
- a CP job is initiated with SEND DIRECT, but the CP asks the PLC for the data for this job via "background communications".
- the amount of data to be transmitted to the CP with SEND DIRECT is greater than the specified frame size.

Description of the SEND DIRECT Function

The SEND DIRECT function works with the following parameters:

- SSNR - interface number
- A-NR - job number (assign "≠0")
- ANZW - job status word
- PAFE - parameter assignment error byte
- QTYP - source type
- DBNR - data block number
- QANF - source start address
- QLAE - amount of source data

Normally, the SEND DIRECT function is called in the cyclic part of the control program. The block can be invoked in an interrupt program (see Section 2.8.4), but the job status word would not be updated cyclically in this case. This task must then be performed by the CONTROL block.

The following two conditions must be met to transfer data or to activate a SEND job:

- RLO "1" was forwarded to the function block
- The CP enabled the job. (The "SEND/FETCH in progress" bit of the condition code word is "0").

If RLO "0" is forwarded (empty run), only the job status word is updated.
If "NN" is entered in the QTYP parameter, the source parameters have to be stored in the CP. If not, the job is aborted with error.

Data interchange can proceed as follows:

- The requested data is transferred directly to the CP.
- The CP asks only for the job parameters.
- The amount of data to be transmitted is too large. The block transfers the parameters and the first data block to the CP. Then the CP requests the remaining data or an additional data frame from the PLC via the SEND ALL function.

For the block user, the operator interface is the same in all initiation modes. However, in the last two cases, the instant of data transfer is postponed by at least one program cycle.

Description of the WRITE Function

If "RW" is entered in the QTYP parameter, the block transfers the indirectly specified source and destination parameters to the CP. Then the destination parameters are sent along with the useful data (requested via the SEND ALL function) to the communications partner (WRITE function).

The RECEIVE Block - FB245 -

FB245 requests reception of data from a module with page addressing. A distinction is made between two function modes:

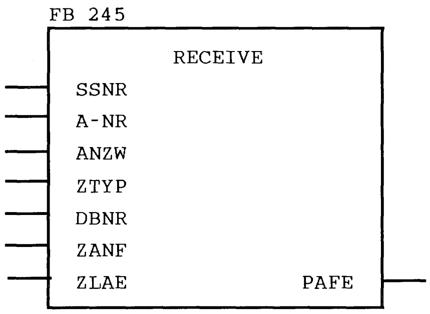
- RECEIVE ALL
  Data can be received for any job. In this mode, the function block is used for so-called background communications and can request any data a CP/IP is prepared to transfer to the CPU.
- RECEIVE DIRECT
  Data is received for a specific job.

Calling Function Block FB245 (Example)

STL

```
        :JC   FB 245
NAME    :RECEIVE
SSNR    :     KY 0,10
A-NR    :     KY 0,101
ANZW    :     FW 24
ZTYP    :     KS DB
DBNR    :     KY 0,10
ZANF    :     KF +100
ZLAE    :     KF -1
PAFE    :     FY 23
        : ***
```

CSF/LAD



Description of the RECEIVE ALL Function

The block needs the following parameters in RECEIVE ALL mode:

- SSNR - interface number
- A-NR - job number (assign "0")
- ANZW - job status word
- PAFE - parameter assignment error byte

All other parameters are irrelevant for this job.

The CP provides the following information via the communications area:

- address of the job status word
- type of data
- amount of data
- start address of the data area

The following bits are evaluated or set/reset in the status word for the pertinent job:

- data transfer disabled
- data transfer completed
- data transfer in progress

The block enters the amount of data transferred for a job in the data word that follows the job status word of the associated RECEIVE DIRECT job.

The RECEIVE block must be called in the control program in "ALL" mode at least once per interface if

- the CP wants to give data to the PLC on its own initiative.
- the amount of data to be received with RECEIVE DIRECT exceeds the specified frame size.
- the CP uses RECEIVE DIRECT only to enable receive data, and transfers data to the PLC via "background communications".

You can call FB245 in RECEIVE ALL mode in

- the cyclic program (e.g., in OB1)
- the service routine for timed interrupts (e.g. prompter block)
- the service routine for process interrupts (see section 2.8.4)

Description of the RECEIVE DIRECT Function

The RECEIVE DIRECT function works with the following parameters:

- SSNR - interface number
- A-NR - job number (assign "≠0")
- ANZW - job status word
- PAFE - parameter assignment error byte
- ZTYP - destination type
- DBNR - data block number
- ZANF - destination start address
- ZLAE - amount of destination data

Normally, the RECEIVE DIRECT function is called in the cyclic part of the control program. This block can also be called in an interrupt program (see Section 2.8.4), but the job status word is not updated cyclically in this case. The CONTROL block must then perform this task.

The RECEIVE block communicates with the CP on a handshaking basis under the following conditions only:

- RLO "1" has been forwarded to the function block.
- The CP has enabled the job. (The "RECEIVE ready" bit in the job status word is set).

12

When RLO "0" is forwarded (empty run), only the job status word is updated.

If "NN" is assigned to the ZTYP parameter, the CP must provide the destination parameters. Otherwise, the job is aborted with an error.

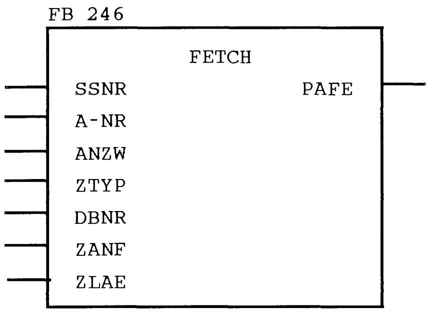
If the CP provides the destination parameters, when ZTYP is not "NN", only the parameter specifications in the block are noted.

Large amounts of data can be received in the form of frames only. Only one data frame can be received at a time with RECEIVE DIRECT. The remaining data or additional data frames must therefore be transferred to the PLC with RECEIVE ALL.

The FETCH Block - FB246 -

FB246 requests that data can be fetched from a communications partner over a CP. The data is received via function block FB245 in RECEIVE ALL mode. You can use the FETCH block only to fetch data for a specific job (FETCH DIRECT function).

Calling the FETCH Block (Example)

STL	CSF/LAD
:JU FB 246 NAME :FETCH SSNR : KY 0,10 A-NR : KY 0,101 ANZW : FW 9 ZTYP : KS DB DBNR : KY 0,46 ZANF : KF + 5 ZLAE : KF + 20 PAFE : FY 14 : ***	 FB 246 FETCH SSNR A-NR ANZW ZTYP DBNR ZANF ZLAE PAFE

Description of the FETCH Function

All parameters must be assigned for the FETCH function. The destination parameters (ANZW, ZTYP, DBNR, ZANF, ZLAE) are passed to the CP during handshaking. As soon as the requested data arrives, the CP provides the RECEIVE ALL block with both parameters and data. The FETCH block itself does not transfer or receive data.

The FETCH job is activated under the following conditions:

- RLO "1" has been forwarded to the function block.
- The CP has enabled the function. (The "SEND/FETCH in progress" bit is "0").

If RLO = 0 (empty run), only the job status word is updated.

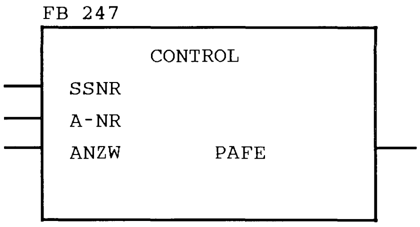
If "RW" is assigned to the ZTYP parameter, the FETCH block transfers the source and destination parameters and the address of the job status word to the CP.

FETCH can be invoked in the cyclic program or in an interrupt service routine (see Section 2.8.4). The FETCH or CONTROL block updates the job status word.

The CONTROL Block - FB247 -

FB247 updates the job status word for a specific job or indicates which job is currently in progress.

Calling FB247 (Example)

STL	CSF/LAD
:JU FB 247 NAME :CONTROL SSNR : KY 0,10 A-NR : KY 0,101 ANZW : FW 20 PAFE : FY 22 : ***	

Description of the CONTROL Function

The CONTROL function requires the following parameters:

- SSNR - interface number
- A-NR - number of the job to be monitored
- ANZW - job status word where the result is to be stored
- PAFE - parameter assignment error byte

The CONTROL block implements different functions depending on the job number.

A-NR="0"

The CP is asked which job is currently in progress. The CP writes the number of the current job in the low-order byte of the job status word when the CONTROL block is being processed.

A-NR ≠ "0"

The block executes in CONTROL DIRECT mode:

- The status of a specific job is interrogated.
- The job status word is updated.

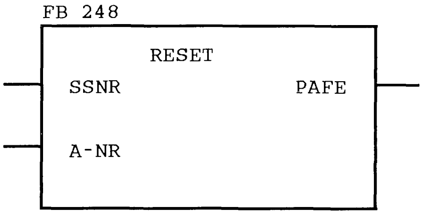
Processing of this block does not depend on the RLO. However, FB247 should be called in the cyclic part of the control program.

The RESET Block - FB248 -

FB248 resets a job executing over the specified interface. RESET can execute in two different modes:

- RESET ALL
  If you assign "0" as the job number, all jobs for the specified interface are reset.
- RESET DIRECT
  If you assign a number "≠0" as the job number, only the specified job is reset.

Calling FB 248 (Example)

STL	CSF/LAD
:JU FB 248 NAME :RESET SSNR : KY 0,1 A-NR : KY 0,0 PAFE : FY 111 : ***	

Parameter Description

FB248 requires the following parameters:

- SSNR - interface number
- A-NR - number of the job that is to be reset
- PAFE - parameter assignment error byte

RESET Function Description

In both modes, jobs are reset in the following way:

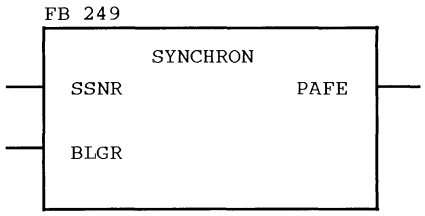
- the job data are deleted.
- active jobs are aborted.

FB248 executes dependent on the RLO, and can be invoked in both the cyclic program and in an interrupt service routine (see Section 2.8.4).

The SYNCHRON Block - FB249 -

Each time the PLC is restarted, FB249 parameterizes the interface on a module with page addressing for communication with the control program. This synchronization is essential for proper execution of the data handling blocks.

Calling FB249 (Example)

STL	CSF/LAD
:JU FB 249 NAME :SYNCHRON SSNR : KY 0,1 BLGR : KY 0,5 PAFE : FY 100 : ***	

Parameter Description

FB249 requires the following parameters:

- SSNR - interface number
- BLGR - frame size
- PAFE - parameter assignment error byte

SYNCHRON Function Description

After you enter the desired frame size for the BLGR parameter, the CP checks this value according to module-specific criteria and determines the final frame size.

In certain cases, this means that the frame size specified in the parameter is invalid.

The final frame size specifies how much data (bytes) can be transferred directly when the SEND and RECEIVE blocks are called. For larger amounts of data, continuation frames are generated and transferred with the ALL functions of these blocks.

FB249 synchronizes the PLC and the CP on each PLC restart. Consequently, FB249 should be called in RESTART blocks OB21 and OB22.

12.3 SINEC L1 Local Area Network

The SINEC L2 local area network is used for networking SIMATIC S5 programmable controllers of the U range; it works according to the master-slave principle. One master and up to 30 slaves can be connected to the SINEC L1 bus.

- The ***master*** is always a CP 530. It handles all communication and monitoring tasks in conjunction with data communications via the LAN.
- ***Slaves*** can either be
  - CP 530 communications processors or
  - CPUs with L1 slave interface (e.g. SI 1 of the CPU 945 can be directly connected; SI 2 via SINEC L1 interface module only).

"Data handling blocks" which support communications with the CP 530 are integrated in the central processing units of the S5-115U (see Section 12.2.3).

12.3.1 Connection of the S5-115U PLC to the L1 Bus Cable

Each node, master or slave, needs a BT 777 transceiver as level converter.
The BT 777 can be connected

- to the slave's programmer interface (CPU 945: SI 2 with SINEC L1 interface module only); in this case, the data is to be interchanged via send/receive mailboxes as described in the following or
- to the SINEC L1 interface of the CP 530 - either as master or slave. (In this case, proceed as described in the "SINEC L1 Local Area Network"; 6ES5 998-7LA21. Data is exchanged via data handling blocks.)

Data is transferred over a 4-wire shielded cable that interconnects the various transceivers.

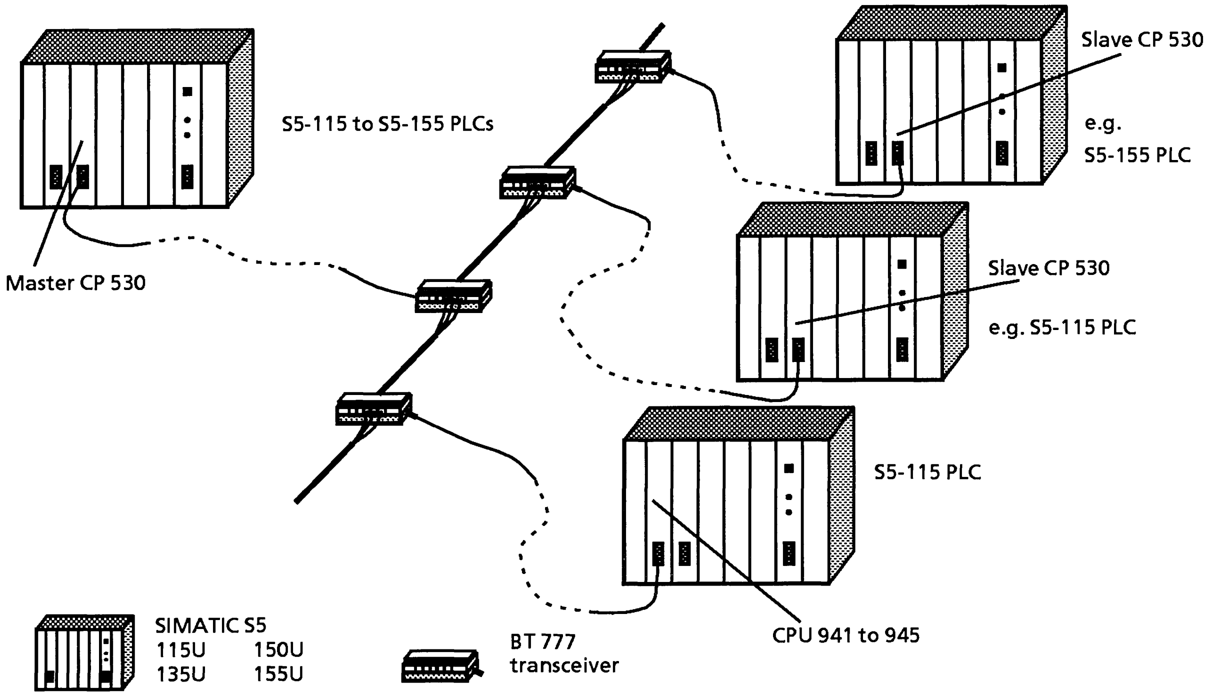


Figure 12.9 Programmable Controllers Linked via the SINEC L1 LAN

You can transfer data over the SINEC L1 local area network in the following two ways:

- from one node to another
  - master → slave
  - slave → master
  - slave → slave
- from one node to all other nodes simultaneously (broadcasting).

Besides data, you can also transmit programmer functions on the SINEC L1 local area network. A programmer that is connected to the master's CP 530 can address individual slaves (see SINEC L1 manual 6ES5 998-7LA21).

12.3.2 Coordinating Data Interchange by Connecting the CPU 945 to the SINEC L1 Bus via One of Its Serial Interfaces

A slave needs the following to interchange data:

- a slave number (1 to 30)
- a Send mailbox (SF)
- a Receive mailbox (EF)
- a coordination byte for "send"
- a coordination byte for "receive"

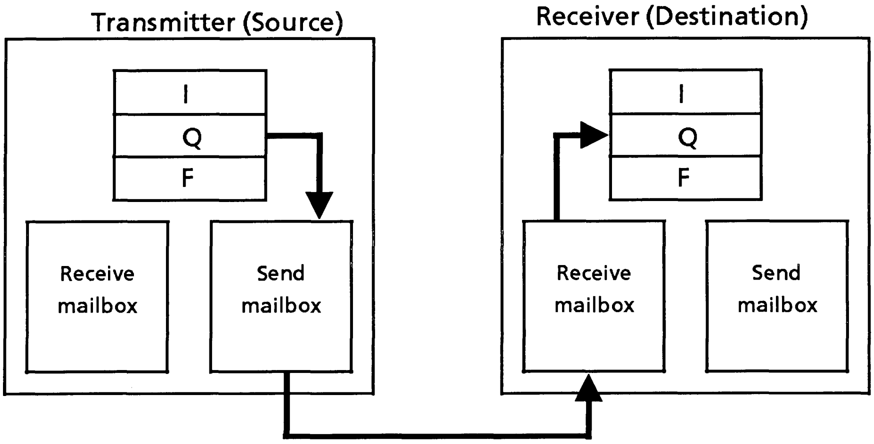
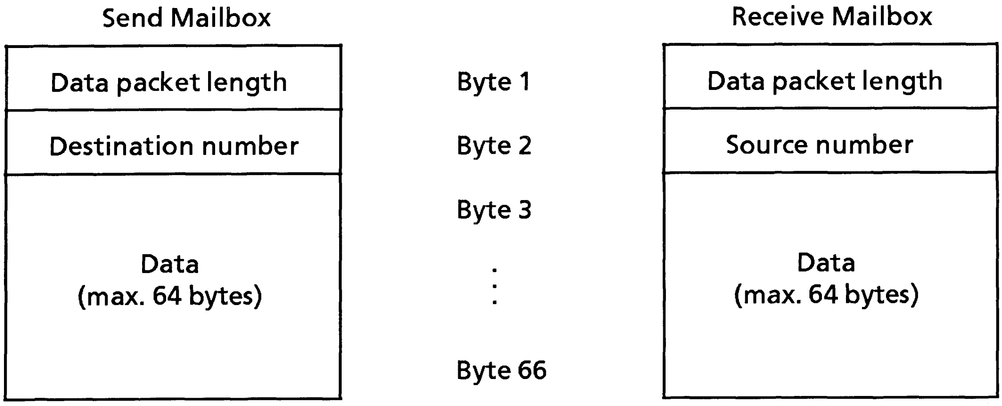


Figure 12.10 Data Transport via SINEC L1 (Example)

Send and Receive Mailboxes

The Send and Receive mailboxes contain send and receive data. They can hold up to 64 bytes of information. They also contain the following:

- Length of the data packet (1 to 64 bytes)
- L1 address of the sender (receive mailbox)
  - the receive mailbox contains the source number
- L1 address of the receiver (send mailbox)
  - the destination number is specified in the send mailbox.



Send Mailbox		Receive Mailbox
Data packet length	Byte 1	Data packet length
Destination number	Byte 2	Source number
Data (max. 64 bytes)	Byte 3 . : Byte 66	Data (max. 64 bytes)

Figure 12.11 Structure of the Send and Receive Mailboxes for SINEC L1

The source or destination number indicates the "L1 node" with which you want to communicate. Refer to Table 12.10 for the meaning of these numbers.

Table 12.10 Destination and Source Number Assignment

Assignment	Partner
0	Master
1 to 30	Slave
31	Broadcasting

Use the control program to access the mailboxes. The location of the mailboxes can be parameterized.

You can define the starting addresses of the mailboxes in either of the following ways:

- Specify a data block (DB/DX) and a data word.
- Specify a flag byte (FY, SY).

Coordination bytes for "Send" and "Receive"

The coordination bytes form the interface to the operating system of the CPU 945.
The control programs of the slaves use these bytes to track and manipulate the data traffic via the LAN.

The following table describes the meanings of the individual bits of the "Receive" coordination byte (CBR).

The "Receive" coordination byte can either be

- a flag byte (FY, SY)
  or
- the high-order byte of a data word (DL in DB/DX).

Table 12.11 Meanings of the Individual Bits of the "Receive" Coordination Byte (CBR) for SINEC L1

Bit	Information from the bus master	Meaning	Access
0	No	**Error** 0: No error 1: Error during last data transfer	• The control program may only read the bits. • The operating system sets and resets the bits.
1	Yes	**Slave OFF** 0: No slave failed 1: At least one slave failed	
2	Yes	**LAN RUN** 0: LAN is in the STOP mode 1: LAN is in the RUN mode	
3	Yes	**PG bit** 0: Programmer does not request LAN access 1: Programmer requests LAN access	
4	Yes	**Interrupt** 0: No signal 1: Data packet is sent as express transmission	
5	Insignificant		
6			
7	No	**REC-PERM** 0: Program can fetch data from the Receive mailbox. The operating system has no access to the data. 1: The data in the REC mailbox may not be evaluated by the program. The operating system can receive data from the LAN via the Receive mailbox. If REC-PERM = "1", the operating system fills the Receive mailbox with data. Then the operating system resets REC-PERM to "0".	• Bit is set by the control program. • Bit is reset by the operating system.

The following table describes the meanings of the individual bits of the "Send" coordination byte (CBS).

The "Send" coordination byte can either be

- a flag byte (FY, SY)
  or
- the high-order byte of a data word (DL in DB/DX).

Table 12.12 Meanings of the Individual Bits of the "Send" Coordination Byte (CBS) for SINEC L1

Bit	Information for the bus master	Meaning	Access
0	No	**Error during last data transfer** 0: No error 1: Error detected	The bit is set and reset by the operating system.
1	Insignificant		
2			
3			
4	Yes	**Interrupt** 0: Normal message 1: Slave wants to send an express transmission or overwrite the old message.	● The bit is set and reset by the control program. ● The bit is reset by the operating system.
5	Insignificant		
6			
7	No	**SEND-PERM** 0: Program can process the Send mailbox. Operating system has no access to data. 1: Program is not allowed to alter data in the Send mailbox. Send mailbox is enabled for transmission via the bus. SEND-PERM = "1" causes the operating system to transmit the contents of the Send mailbox. Then the operating system resets the SEND-PERM bit to "0".	● The bit is set by the control program. ● The bit is reset by the operating system.

Note

The bits in the coordination bytes can be set or reset by the operating system after any operation and irrespective of the PLC cycle. This means that a multiple scan of a coordination byte within one program cycle can have different results (This must be taken into consideration for an edge evaluation!).

12.3.3 Assigning Parameters to the S5-115U for Data Interchange via SINEC L1

When using SINEC L1, the following must always be specified:

- local slave number
- data or flag areas assigned for the Send *and* Receive mailboxes
- location of the coordination bytes (CBR and CBS)

If necessary, you can also specify

- your own programmer number for programmer bus functions.

The parameter assignments can be made either

- in DB 1 (see Chapter 11) or
- in a function block which is called by one of the two restart organization blocks (OB21 or OB22). Table 12.13 lists the parameters that are stored from system data word 57 onwards in the system data area of the CPU 945.

Table 12.13 SINEC L1 Parameter Block

System Data Word	High-Order Byte	Low-Order Byte	Absolute Address
RS 57	Programmer number * (1 to 30)	Slave number** (0 to 30)	0E 1072 0E 1073
RS 58	CBR Data ID ***	CBR DB/DX or flag byte number or high-order part of the S flag address	0E 1074 0E 1075
RS 59	CBR Data word number or low-order part of the S flag address	CBS Data ID ***	0E 1076 0E 1077
RS 60	CBS DB/DX or flag byte number or high-order part of the S flag address	CBS Data word number or low-order part of the S flag address	0E 1078 0E 1079
RS 61	SMB Data ID ***	SMB DB/DX or flag byte number or high-order part of the S flag address	0E 107A 0E 107B
RS 62	SMB Data word number or low-order part of the S flag address	RMB Data ID ***	0E 107C 0E 107D
RS 63	RMB DB/DX or flag byte number or high-order part of the S flag address	RMB Data word number or low-order part of the S flag address	0E 107E 0E 107F

* A programmer number is required if programmer functions are to be transmitted via the SINEC L1 LAN. Note carefully that slave number = 0 in the low-order byte indicates the master function. In this case, no programmer/OP functions are possible via the SI 2 interface of the CPU!
The programmer number is retained in the case of an overall reset of the CPU via the programmer bus.

** Slave number = "0": master function (see Section 12.4)

*** see Table 12.14

Specify the location of the coordination bytes and start addresses of the Send and Receive mailboxes with three bytes each.

Table 12.14 Data IDs of the SINEC L1 Parameter Block

Stored in	Data ID		Memory Area *	
	Hexa-decimal	ASCII		
Flag	$4D_H$	M	Flag byte number: 0 to 255	- - -
S flag**	53_H	S	S flag byte number: 0 to 4095	- - -
Data (DB)	44_H	D	DB No.: 0 to 255	Data word No.: 0 to 255
Data (DX)	58_H	X	DX No.: 0 to 255	Data word No.: 0 to 255

* Specifies the start addresses of the memory areas for the Send and Receive mailboxes

** High-order part of the S flag number: 00_H to $0F_H$; low-order part of the S flag number: 00_H to FF_H

Overflow

- when receiving

If the length of a data packet received exceeds the length of the Receive mailbox, the data received is rejected and the error bit (bit 0 in CBR) is set. The transmitting node receives a negative acknowledgement.

- when sending

If more data than the amount of data available in the Send mailbox are to be transmitted (Send mailbox too small), the send error bit (bit 0 in CBS) is set and the data is not transmitted.

The end of the receive area is marked by flag byte 255 in the flag area, 4095 in the S flag area or the last available data word (in the data block).

Example of SINEC L1 Parameter Assignment:

Set parameters in OB22 (OB21). FB 255 is used to handle parameter entries.

The formal operands indicate the type and number of the coordination bytes (CBR, CBS) and of the data mailboxes (RMB, SMB), e.g., TCBR is the type of "receive" coordination byte.

OB21/OB22 STL	Description
`      : JU   FB 255`	
`NAME  : L1   PARAM`	
`SLNO  :      KY 0,1`	`SLAVE 1`
`TCBR  :      KS FY`	`CBR   :   flag area`
`NCBR  :      KY 100,0`	`FY 100`
`TCBS  :      KS FY`	`CBS   :   flag area`
`NCBS  :      KY 101,0`	`FY 101`
`TSMB  :      KS DB`	`SMB   :   data block`
`NSMB  :      KY 2,1`	`DB2 beg. with DW1`
`TRMB  :      KS DB`	`RMB   :   data block`
`NRMB  :      KY 2,40`	`DB2 beg. with DW40`
`      : BE`	

Parameter description:

SLNO: PG bus address/data slave address (KY a, b):
a) PG bus address
b) Data slave number

TCBR/NCBR: Type of COOR byte for RECEIVE/SEND (KS):
Options are FY ≙ Flag byte
SY ≙ S flag byte
DW ≙ Data word (left) in DB
XW ≙ Data word (left) in DX

NCBR/NCBS: Number or address of COOR byte for RECEIVE/SEND (KY a, b):
a) For type FY ≙ Number of the flag byte
For type SY ≙ High-order part of the S flag address
For type DW ≙ Number of the data block (DB)
For type XW ≙ Number of the data block (DX)
b) For type FY ≙ "0"
For type SY ≙ Low-order part of the S flag address
For type DB ≙ Number of the data word (left; DB)
For type DX ≙ Number of the data word (left; DX)

TSMB/TRMB: Type of SEND/RECEIVE MAILBOX (KS):
Options are FY ≙ Flag byte
SY ≙ S flag byte
DB ≙ Data word (left; DB)
XB ≙ Data word (left; DX)

NSMB/NRMB: Number of SEND/RECEIVE MAILBOX (KY a, b):

a)	Type FY	≙	Number of the flag byte from which the send/receive mailbox starts
	Type SY	≙	High-order part of the S flag address
	Type DB	≙	Number of the DB data block
	Type DX	≙	Number of the DX data block
b)	Type FY	≙	"0"
	Type SY	≙	Low-order part of the S flag address
	Type DB	≙	Number of the data word in the DB from which the send/receive mailbox starts
	Type DX	≙	Number of the data word in the DX from which the send/receive mailbox starts

Example of type SY:

SY 258 (=1×256+2×1)
→NCBR/NCBS=1,2 and TCBR/TCBS or TSMB/TRMB=SY

FB255 STL	Description
NAME :L1 PARAM	
DECL :SLNO I/Q/D/B/T/C: D KM/KH/KY/KS/KF/KT/KC/KG: KY	
DECL :TCBR I/Q/D/B/T/C: D KM/KH/KY/KS/KF/KT/KC/KG: KS	
DECL :NCBR I/Q/D/B/T/C: D KM/KH/KY/KS/KF/KT/KC/KG: KY	
DECL :TCBS I/Q/D/B/T/C: D KM/KH/KY/KS/KF/KT/KC/KG: KS	
DECL :NCBS I/Q/D/B/T/C: D KM/KH/KY/KS/KF/KT/KC/KG: KY	
DECL :TSMB I/Q/D/B/T/C: D KM/KH/KY/KS/KF/KT/KC/KG: KS	
DECL :NSMB I/Q/D/B/T/C: D KM/KH/KY/KS/KF/KT/KC/KG: KY	
DECL :TRMB I/Q/D/B/T/C: D KM/KH/KY/KS/KF/KT/KC/KG: KS	
DECL :NRMB I/Q/D/B/T/C: D KM/KH/KY/KS/KF/KT/KC/KG: KY	
:LW =SLNO	L1-PG-bus/L1-data bus slave No.
:T FW 200	
:	
:	
:LW =TCBR	Type of coordination byte "R"
:T FW202	(Receive)
:	
:LW =NCBR	Address of the CBR
:T FW 203	
:	
:LW =TCBS	Type of coordination byte "S"
:T FW 205	(Send)
:	
:LW =NCBS	Address of the CBS
:T FW 206	
:	
:LW =TSMB	Type of send mailbox
:T FW 208	
:LW =NSMB	Address of send mailbox
:T FW 209	
:LW =TRMB	Type of receive mailbox
:T FW 211	
:LW =NRMB	Address of receive mailbox
:T FW 212	
:	
:L DH 000204D5	Transfer from flag area to
:L DH 000E107F	system data area
:TNB 14	
:	
:L KB 0	Reset scratch flag words
:T FD 200	
:T FD 204	
:T FD 208	
:T FW 212	
:	
:BE	

Use the following example to parameterize an S5-115U as PG bus node, i.e. without SINEC L1 parameter assignment:

Example: Initializing an CPU 945 connected only as programmer bus node to the SINEC L1 LAN. The function block for programmer address assignments (FB1) is invoked in the restart OBs (OB21 and OB22).

OB21/OB22 STL	Description
:	
:JU FB 1	Call FB 1 for programmer bus address assignments
NAME : PG-ADR	
PGAD : KY 1,0	Programmer bus address of the CPU is 1
:	(perm.values: 1 to 30)
:BE	

FB1 STL	Description
NAME :PG-ADR	
DECL :PGAD	I/Q/D/B/T/C : D KM/KH/KY/KS/KF/KT/KC/KG : KY
:L RS 57	Load system data word 57
:L KH 00FF	Delete old PG No.
:AW	
:LW =PGAD	Load PG bus address
:OW	RS 57 and PG bus address are ored,
:	The low-order byte of RS 57 remains unchanged
:T RS 57	write result of OR operation back to RS 57
:BE	

12.4 Point-To-Point Connection with SINEC L1 Protocol

A point-to-point connection is a SINEC L1 link between only two communications partners:

- CPU 945 (only with SINEC L1 interface module) as master
  and
- communications partner connected as slave (see Table 12.15).

Data, control information and data protection information can be transmitted via this link.

Table 12.15 Communications Partners (Slaves) for Point-to-Point Connection

Communications Partners	Connection
S5-100U with CPU 102/103	direct using the CPU interface
S5-90U/95U/101U	
S5-115U with CPU 941/942/943/944/945	direct using the CPU interface or CP 530
S5-135U/150U/155U	using CP 530

12.4.1 Point-To-Point Connection of a Communications Partner

You can establish connections in either of the two following ways:

- via a bus with transceivers (BT 777)
  or
- via a direct line (a direct line connection is possible only when the controllers are no further apart than 1000 m.) Use a four-wire, shielded cable with a cross-section of at least 0.14 mm^2 (26 AWG). SIMATIC cable 6ES5 707-1AA00 is recommended.

The transceiver or direct line is connected to the SINEC L1 interface module of SI 2.

For the connector pin assignments of the two 15-pole Cannon connectors for the direct line, refer to Section 12.7 "Interface Modules".

12.4.2 Parameter Assignment and Operation of the Point-To-Point Connection

Use the SINEC L1 parameter block to parameterize the interface on the CPU (see Section 12.3.3).

For a point-to-point connection, assign "0" as the slave number for the CPU 945 (master function only possible at SI 2). The communications partner is always addressed as slave 1.

A point-to-point connection can be established only if neither an ASCII driver nor any computer link is parameterized (see Sections 12.5 and 12.6).

As long as interface SI 2 has been parameterized for point-to-point connection, no programmer or OP can be operated over this connector.

Note

Neither broadcasting nor interrupt traffic is possible over point-to-point connections.

As with the SINEC L1 local area network, data is exchanged via Send and Receive mailboxes that the control program accesses with load and transfer operations.

The CPU operating system controls data transfer and stores the information in two coordination bytes. The control program can read and evaluate both bytes. The following tables explain the bits in the coordination bytes.

A coordination byte can either be

- a flag byte (FY/SY)
  or
- the high-order byte in a data word (DB/DX)

Table 12.16 Meanings of the Individual Bits of the Coordination Byte for "Receive" (CBR) in the Case of a Point-To-Point Connection

Bit	Information from the bus master	Meaning	Access
0	No	**Error** 0: No error 1: Receive error during last data transfer	• Control program may only read the bit. • Bit may only be set and reset by the operating system.
1	Yes	**Slave OFF** 0: Normal operation 1: Communications partner failure	
2	Yes	**LAN RUN** 0: SINEC L1 LAN is in the STOP mode 1: SINEC L1 LAN is in the RUN mode	
3	Insignificant		
4			
5			
6			
7	No	REC-PERM 0: Program can fetch data from the Receive mailbox The operating system has no access to the data. 1: Program is not allowed to evaluate receive data. Operating system may transfer data from the bus into the Receive mailbox.	• The bit is set by the control program. • The bit is reset by the operating system.

Table 12.17 Meanings of the Individual Bits of the Coordination Byte for "Send" (CBS) in the Case of a Point-To-Point Connection

Bit	Information for the bus master	Meaning	Access
0	No	**Error** 0: No error 1: Send error during last data transfer	Bit is set and reset by the operating system.
1	Insignificant		
2			
3			
4			
5			
6			
7	No	**SEND-PERM** 0: Program can process Send mailbox. Operating system has no access. 1: Operating system sends data from the Send mailbox to the bus. Program is not allowed to alter any send data.	• Bit is set by the control program. • Bit is reset by the operating system.

Note

The bits of the coordination bytes can be set or reset by the operating system after any operation and irrespective of the PLC cycle. This means that a multiple scan of a coordination byte within one program cycle can have different results (This must be taken into consideration for an edge evaluation!).

The location of the coordination bytes, and the Send and Receive mailboxes can be parameterized (as for SINEC L1) in a function block (see Section 12.3.3).

If the amount of data to be transferred exceeds a certain limit, the reaction is the same as for "Overflow" in the SINEC L1 LAN.

Differences between Point-to-Point Connection and CP 530

In a point-to-point connection, data is written directly into the CPU's program memory. The control program can therefore access this area only between the time the data has been received and the next frame enabled.

These steps must be coordinated over the control program. When using a CP 530, the data in a frame is initially stored in a CP 530 buffer.

The control program triggers data transfer to the appropriate DBs. Only one read operation is needed for this transfer. While the control program is processing the DBs, the CP 530 can receive the next frame.

12.5 ASCII Driver

The CPU 945 provides an ASCII driver for the second interface. The ASCII driver permits open communications with other communications partners, such as printers, terminals, etc.

The ASCII driver operates only under the following conditions:

- Initialization as described in Sections 12.5.4 and 12.5.5 and
- activation by setting RS 46 (E 105C$_H$) accordingly.

The ASCII driver can also be parameterized and activated by means of DB 1 (see Chapter 11).

Any error flags concerning CBR and CBS are entered in the low-order byte of RS 46 after activation of the ASCII driver.

Note

No other functions are possible at the second interface (e.g. programmer/OP, SINEC L1, point-to-point connection) when the ASCII driver is activated.

Table 12.18 Description of System Data Word 46 (ASCII Driver)

Byte	Contents	Description
High-order byte	00$_H$ *	Programmer/OP and SINEC L1 operation
	01$_H$	ASCII driver
Low-order byte		Error Check-Back Signals
	00$_H$	No error with regard to CBS and CBR
	01$_H$	Invalid driver number
	10$_H$	No CBS
	20$_H$	No CBR
	40$_H$	No CBS and no CBR

* Default value (after overall reset)

12.5.1 Data Traffic via the ASCII Driver

Figure 12.12 shows how the ASCII driver functions.

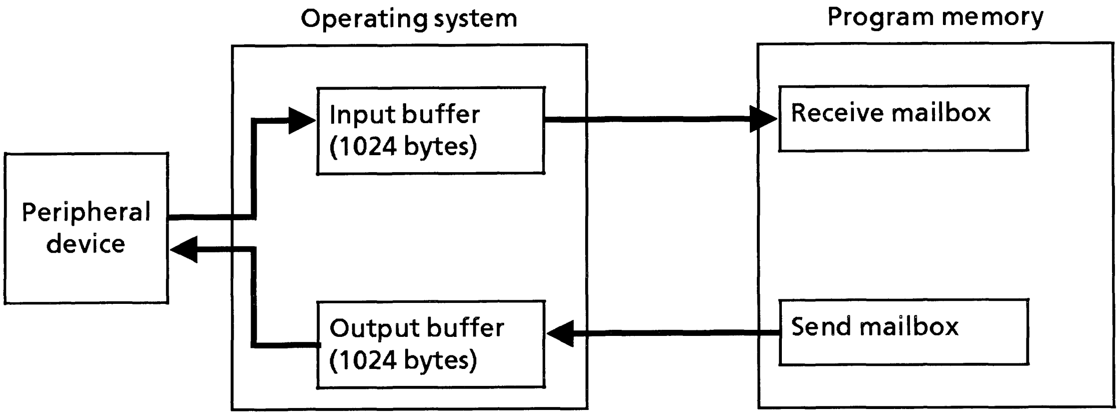


Figure 12.12 Example of Data Transport (ASCII driver)

Data traffic is bidirectional:

- Send
  The ASCII driver processes data in the program memory (e.g. the contents of a data block) and outputs it at the second interface.

- Receive
  A peripheral device sends data in ASCII code to the second interface. The ASCII driver processes the data and stores it in the program memory.

The two memory areas in internal RAM in which the send and receive data are stored are called the Send mailbox (SMB) and the Receive mailbox (RMB).

The data can be stored either in a data block (DB/DX) or in the flag area (FY/SY). You must enter the relevant information in the parameter block (see Table 12.24) or parameterize them in DB1 (see Chapter 11).

Send and Receive mailboxes have the following properties:

- 1024 bytes of input buffer are available in all modes. Up to 100 message frames can be stored.
- In modes where characters are interpreted when received (e.g. XON, XOFF), the ASCII driver can still receive data or message frames even if it has already sent XOFF to the communications partner. In this case, the ASCII driver receives data until the input buffer is full, or it receives message frames until the maximum possible number of frames has been reached.

  Example of "borderline case":
  If a received message frame is 1024 bytes long and the ASCII driver then sends XOFF, there is no way of buffering characters received from the communications partner after sending XOFF.
- In mode 1, 7 or 8 (see Section 12.5.3), you must enter in the first word of the send mailbox the number of bytes of data to be sent.

12.5.2 Coordination Bytes of the ASCII Driver

The ASCII driver monitors data communications by means of

- the coordination byte for "Send" (CBS)
  and
- the coordination byte for "Receive" (CBR).

The ASCII driver sets status and error flags in these bytes.

Coordination bytes can either be

- flag bytes (FY/SY)
  or
- high-order bytes in the data words (DB/DX).

Tables 12.19 and 12.20 describe the meanings of the coordination bytes.

Note

The bits of the coordination bytes can be set or reset by the operating system after any operation and irrespective of the PLC cycle. This means that a multiple scan of a coordination byte within one program cycle can have different results (This must be taken into consideration for an edge evaluation!).

Table 12.19 Meanings of the Individual Bits of the Coordination Byte for "Send" (CBS) in the Case of the ASCII Driver

Bit	Meaning		
	Hex code	Description	Reaction
0 to 6	07_H	Output buffer full	Data is rejected
	$0D_H$	Parameter assignment error	
	11_H	No Send mailbox	
	13_H	Frame is too long	
	$1B_H$	Break (in modes 6 and 7 only)	
7		**Send permitted** This bit is set by the user and reset by the ASCII driver when the send procedure is terminated, no matter if with or without error. The send procedure is initiated by a positive-going edge at bit 7. The send data and the location of the Send mailbox must not be modified as long as this bit is "1".	

Table 12.20 Meanings of the Individual Bits of the Coordination Byte for "Receive" (CBR) in the Case of the ASCII Driver

Bit	Meaning		
	Hex code	Description	Reaction
0 to 6	01_H	Character delay time exceeded	Data is valid until time is exceeded
	03_H	Parity error	Data is rejected
	07_H	Input buffer full	
	09_H	Too many frames received	Data is valid, subsequent frames are rejected
	$0B_H$	Frame larger than Receive mailbox	Data is rejected
	$0F_H$	No Receive mailbox	
	19_H	Framing error	
	$1B_H$	Break	
7		**Receive permitted** 1: The bit is set by the user and reset by the ASCII driver after transferring a message frame received from the input buffer to the Receive mailbox or setting an error flag in the CBR. The receive data and the location of the Receive mailbox must not be modified as long as this bit is "1". 0: As long as bit 7 is "0", the user has access to the Receive mailbox. When the input buffer is not yet full, message frames are entered there instead of being written into the Receive mailbox. Up to 100 message frames can be stored in the input buffer.	

12.5.3 Specifying the Type of Data Traffic by Means of Mode Numbers

You can use mode numbers (1 to 8) to specify the type of data traffic.

There are two types of protocol:

- Non-interpreting mode (mode numbers 1, 2, 3)
  No control characters (XON, XOFF) are used in the send and receive modes. However, modes 2 and 3 interpret end-of-frame characters.

- Interpreting mode (mode numbers 4 and 8)
  An XON/XOFF protocol is used for data interchange. When the signal state of the "receive permitted" bit changes, the ASCII driver sends the following control characters:
  - XOFF for negative-going edge
  - XON for positive-going edge.

Mode number

The table below explains the mode numbers. The default refers to word 7 in the ASCII parameter set (see Table 12.22). The mode numbers must be defined in system data word 55 (see Section 12.5.5) or parameterized in DB1 (see Chapter 11).

Table 12.21 Description of the Mode Numbers (ASCII Driver)

Mode	Description	Default in Word 7
1	Send n bytes; n must be specified in the first word of the Send mailbox (n is variable). Receive m bytes; m is specified in the ASCII parameter set and is then permanently assigned.	64 (for receive)
2	Send or receive data until the end-of-text character (low-order byte) defined in the parameter set is sent or received. The end-of-text character is also received or transmitted.	<CR>
3	Send or receive data until the two end-of-text characters defined in the parameter set are sent or received. Both end-of-text characters are received or transmitted. End of text is recognized only when the character defined in the high-order byte is sent or received before the character defined in the low-order byte.	<CR> <LF>
4	Same as mode 2. The following ASCII characters are interpreted upon reception: <RUB OUT> : delete last character <XON> : forward <XOFF> : interrupt send and wait for XON	<CR>
5	Same as mode 3. The following ASCII characters are interpreted upon reception: <RUB OUT> : delete last character <XON> : forward <XOFF> : interrupt send and wait for XON	<CR> <LF>
6	Printer Output Transmit send mailbox until an end-of-transmission character defined in the parameter set (low byte) is reached. The end-of-transmission character is not transmitted. Only XON and XOFF can be received. These characters are also interpreted.	<EOT>
7	Printer Output Send n bytes; n must be specified in the first word of the Send mailbox. (n is not transmitted.) Only XON and XOFF can be received. These characters are also interpreted.	none
8	As in mode 1; the following ASCII characters are also interpreted when received <RUB OUT> * : delete last character <XON> : continue to send <XOFF> : abort send and wait for XON	see mode 1

* If a received message frame of m bytes contains a RUB OUT, correspondingly less data is entered in the Receive mailbox and the character delay time responds → error 01 in CBR.

ASCII Codes and Corresponding Hexadecimal Numbers:

RUB OUT	→	$7F_H$
XON	→	11_H
XOFF	→	13_H
CR	→	$0D_H$
LF	→	$0A_H$
FF	→	$0C_H$
EOT	→	04_H
ETX	→	03_H

12.5.4 ASCII Parameter Set

The ASCII parameter set is used to define the functions of the ASCII driver (see Table 12.22). Depending on the mode selected, the individual parameters are preset. The defaults in modes 6 and 7 apply to the DR 211 printer.

The meaning of word 7 in the ASCII parameter set depends on the mode number (see Table 12.21).

The ASCII driver can also be parameterized in DB1 (see Chapter 11); in this case, the ASCII parameter set is generated accordingly. (The area to which the parameter set is to be transferred must already exist.)

The parameter set is read

- when activating the ASCII driver
  or
- after a mode change. Data traffic via the interface must first be terminated (i.e. bit 7 of CBR=0 and bit 7 or CBS =0);
- the parameter set is transferred after POWER ON of the PLC if the ASCII driver was previously activated.

Note

The default values are used only if the parameter set does not exist or cannot be interpreted.

Table 12.22 ASCII Parameter Set

Word	Description	Value Range	Default According to Mode							
			1	2	3	4	5	6	7	8
0	Baud rate	1 150 baud 2 200 baud 3 300 baud 4 600 baud 5 1200 baud 6 2400 baud 7 4800 baud 8 9600 baud 9 19200 baud	8	8	8	8	8	8	8	8
1	Parity	0 even 1 odd 2 mark ("1") 3 space ("0") 4 no check	0	0	0	0	0	0	0	0
2	Data format*	0 to 8	0	0	0	0	0	1	1	0
3	Waiting time CR **	0 to $00FF_H \times 10$ms	x	x	x	x	x	0	0	x
4	Waiting time LF **	0 to $00FF_H \times 10$ms	x	x	x	x	x	0	0	x
5	Waiting time FF **	0 to $00FF_H \times 10$ms	x	x	x	x	x	0	0	x
6	Character delay time (receive only)	1 to $FFFF_H \times 10$ms	10	10	10	10	10	x	x	10
7	End-of-text characters/ Number of characters received	According to mode number (see Table 12.21)								
8	Suppress LF	0/1 yes/no	x	x	x	x	x	0	0	x
9	Lines per page	1 to 255	x	x	x	x	x	72	72	x
10	Left margin	0 to 255 blanks	x	x	x	x	x	10	10	x
11	Page number	o/u top/bottom	x	x	x	x	x	u	u	x
12 . . .	Header/ Footer***	Header 1 Header 2 Footer 1 Footer 2	x	x	x	x	x	CR CR CR CR	CR CR CR CR	x

x = irrelevant

* See Table 12.23 for the meaning of data formats 0 to 8

** When sending

*** The contents of each header and footer (max. 120 characters each) must be separated by CR.

The character delay time (word 6 of the ASCII parameter set) must conform to the following formula:

$$\text{Character delay time} \geq \frac{100}{\text{Baud rate}}$$

Example:

$$\text{Baud rate} = \quad 4800 \frac{1}{s}$$

$$\Rightarrow \quad \text{Character delay time} \geq \frac{100}{4800}\ s$$

$$\approx \quad 20\ \text{ms}$$

$$\Rightarrow \quad \text{Word 6 in the ASCII parameter set} = \quad \underline{\underline{2}}$$

Data Format and Character Frame

Table 12.23 Character Frame and Order of Bits on the Line in the Case of ASCII Transmission (Depending on Word 2 of the ASCII Parameter Set)

Word 2 of the ASCII Parameter Set (Data Format)	Character Frame	Parity	Number of Bits per Character	Order of Bits on the Line
0	11 bits	0 to 4*	7	1 start bit, 7 data bits, 1 parity bit, 2 stop bits
1	11 bits	0 to 4*	8	1 start bit, 8 data bits, 1 parity bit, 1 stop bit
2	11 bits	Setting irrelevant	8	1 start bit, 8 data bits, 2 stop bits
3	10 bits	Setting irrelevant	7	1 start bit, 7 data bits, 2 stop bits
4	10 bits	0 to 4*	7	1 start bit, 7 data bits, 1 parity bit, 1 stop bit
5	10 bits	Setting irrelevant	8	1 start bit, 8 data bits, 1 stop bit
6	-	-	-	-
7 As data format 0	11 bits	0 to 4*	7	1 start bit, 7 data bits, 1 parity bit, 2 stop bits
8 As data format 1	11 bits	0 to 4*	8	1 start bit, 8 data bits, 1 parity bit, 1 stop bit

* see Table 12.22

12.5.5 Assigning Parameters to the ASCII Driver

With the help of the control program, you must define the position of the ASCII parameter set, the send and receive mailboxes and the coordination byte in a parameter block (see Table 12.24) located in the system data area of the CPU 945; you also specify the mode number there.

The ASCII driver can also be parameterized in DB1 (see Chapter 11).

Table 12.24 Parameter Block for the ASCII Driver

System Data Word	Absolute Address	High-Order Byte	Low-Order Byte
RS 48	0E 1060 0E 1061	ASCII parameter set Data ID	ASCII parameter set DB/DX or flag byte number or high-order part of S flag address
RS 49	0E 1062 0E 1063	ASCII parameter set Data word number or low-order part of S flag address	Send mailbox Data ID
RS 50	0E 1064 0E 1065	Send mailbox DB/DX or flag byte number or high-order part of S flag address	Send mailbox Data word number or low-order part of S flag address
RS 51	0E 1066 0E 1067	Receive mailbox Data ID	Receive mailbox DB/DX or flag byte number or high-order part of S flag address
RS 52	0E 1068 0E 1069	Receive mailbox Data word number or low-order part of S flag address	CBS Data ID
RS 53	0E 106A 0E 106B	CBS DB/DX or flag byte number or high-order part of S flag address	CBS Data word number or low-order part of the S flag address
RS 54	0E 106C 0E 106D	CBR Data ID	CBR DB/DX or flag byte number or high-order part of the S flag address
RS 55	0E 106E 0E 106F	CBR Data word number or low-order part of the S flag address	Mode number *

* If no valid mode number is entered in RS 55, the default value "1" is used.

The data IDs of the parameter block are described in the following table.

Table 12.25 Data IDs of the Parameter Block (ASCII Driver)

Stored In	Data ID		Specification of Memory Area *	
	Hex Code	ASCII Code		
Flag	$4D_H$	M	Flag byte number: 0 to 255	- - -
S flag	53_H	S	S flag byte number: 0 to 4095**	- - -
Data (DB)	44_H	D	DB No.: 0 to 255	Data word number: 0 to 255
Data (DX)	58_H	X	DX No.: 0 to 255	Data word number: 0 to 255

* The start addresses of the memory areas are specified for the ASCII parameter set and the send and receive addresses.

** High-order part of S flag numbers 00_H to $0F_H$, low-order part of S flag numbers 00_H to FF_H

12.5.6 Program Example for ASCII Driver

This section describes the structure of a control program for the ASCII driver.

Example: The program on hand generates a log which is output to the DR 211 printer. It initiates an automatic printout in a two-second interval.

Proceed as follows:

- Connect the DR 211 printer to SI2 of the CPU 945 (programmer module) via the respective cable.

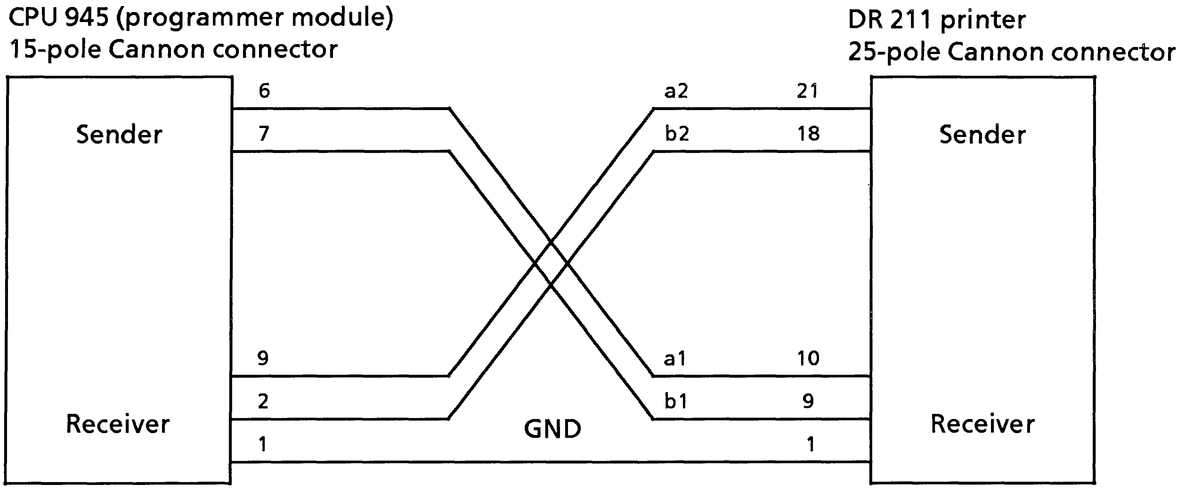


Figure 12.13** **Connector Pin Assignments of the Cable Connecting the CPU 945/SI2 to the Printer (DR 210 or DR 211; TTY Interface)

- Switch on the printer and assign the relevant parameters via the menu on the printer itself. The following list contains only those printer parameter blocks (**bold type**) in which the defaults have to be changed; the parameters to be set are printed *in italics*:

INTERNAL SETTINGS:

LINE FEED CONTROL	
PAPER FEED	*PAPER FEED + CR*
LINE OVERFLOW	*CR + LF*
CR	*CR +LF*
LF	*LF*
INTERFACE SETTING	
SERIAL INTERFACE	
RECEIVE BUFFER	*17 KB*
DATA BITS	*7 BITS*
PARITY	*EVEN*
STOP BITS	*2 STOP BITS*
BAUD RATE	*9600*
POWER-ON STATE	*ON LINE*
ESCAPE CHARACTER	*ESC*
VERTICAL SPACING	
VERTICAL SPACING	*1/72 INCH*

USER-SPECIFIC SETTINGS

LANGUAGE	
MENU LANGUAGE	*GERMAN*
PRINT DENSITY	
CHARACTER PITCH	*10 CPI*
SPACED PRINT	*NO*
CHARACTER SET	
COUNTRY	*0* (for ASCII)
VERTICAL LINE SPACING	*6 LPI*
PAPER FORMAT	
FORM LENGTH (INCH)	*12*
LINE LENGTH (1/10 INCH)	*136*

- Switch the printer to on-line mode (you are guided by the printer menu).
- Switch on the CPU 945 and initiate an overall reset of the PLC (CPU mode: STOP).
- Program the individual blocks as described below.
- Transfer the control program to the CPU 945.
- Switch the mode selector switch of the CPU to RUN.

The structure of the sample program is shown in Figures 12.14 and 12.15.

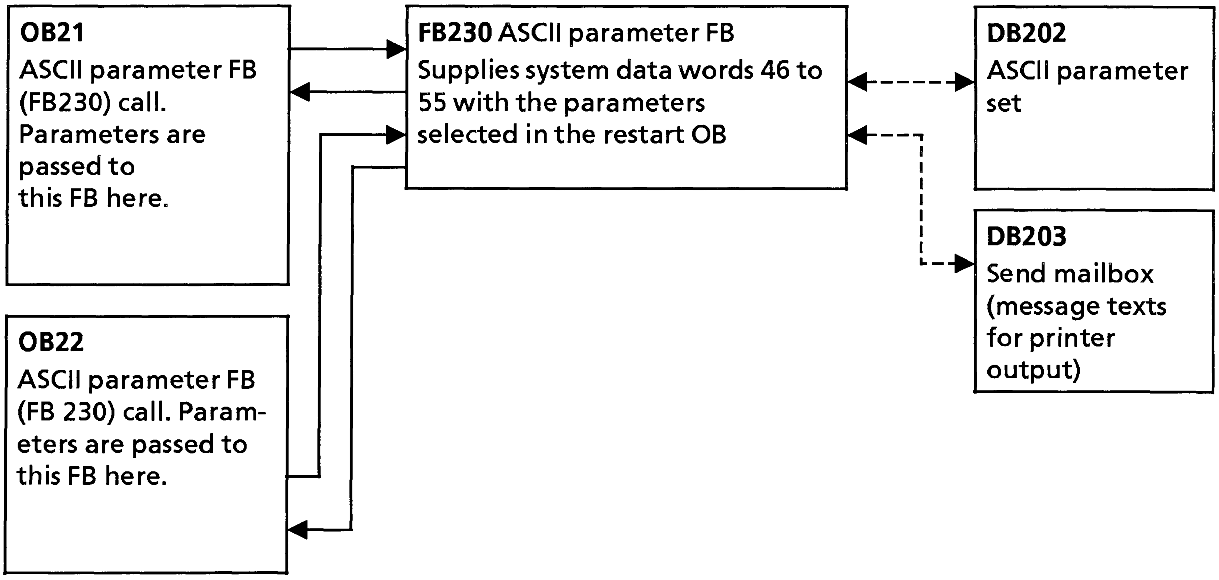

OB21
ASCII parameter FB (FB230) call. Parameters are passed to this FB here.
OB22
ASCII parameter FB (FB 230) call. Parameters are passed to this FB here.
FB230 ASCII parameter FB
Supplies system data words 46 to 55 with the parameters selected in the restart OB
DB202
ASCII parameter set
DB203
Send mailbox (message texts for printer output)

Figure 12.14 ASCII Driver Program Structure for RESTART

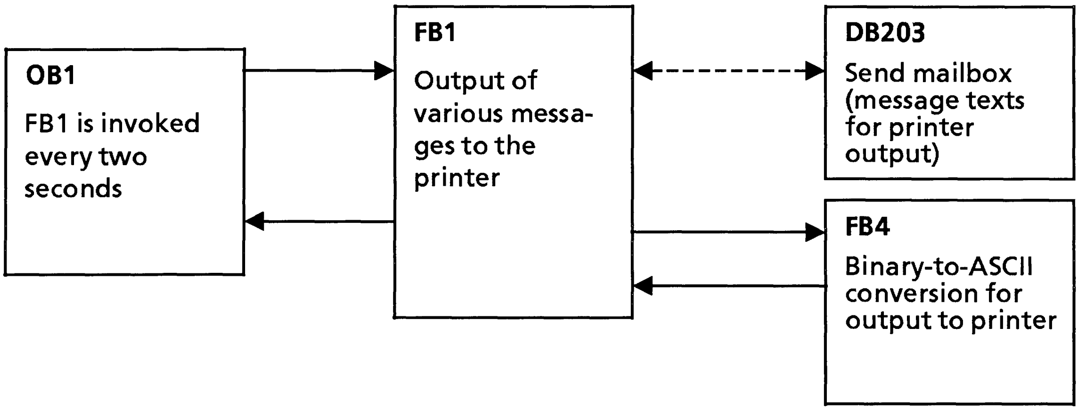

OB1
FB1 is invoked every two seconds
FB1
Output of various messages to the printer
DB203
Send mailbox (message texts for printer output)
FB4
Binary-to-ASCII conversion for output to printer

Figure 12.15 Structure of the Cyclic ASCII Driver Program

OB21 STL	Description
:	
:	ASCII parameter FB call
:JU FB 230	
NAME :ASCII-PA	
TPAR : KS DB	Data type of ASCII parameter set
NPAR : KY 202,0	Is in DB202 and begins with DW0.
TSMB : KS DB	The Send mailbox is located in
NSMB : KY 203,0	DB203 beginning at DW0.
TRMB : KS XX	Not required
NRMB : KY 0,0	Not required
TCBS : KS FY	The Send coordination byte
NCBS : KY 200,0	is FY200.
TCBR : KS FY	The receive coordination byte
NCBR : KY 201,0	is FY201.
MODE : KF +6	ASCII driver mode number 6
:	
:	
:	
:BE	

FB230 STL	Description
NAME :ASCII-PA	Initialize ASCII parameter list
DECL :TPAR I/Q/D/B/T/C: D KM/KH/KY/KS/KF/KT/KC/KG: KS	
DECL :NPAR I/Q/D/B/T/C: D KM/KH/KY/KS/KF/KT/KC/KG: KY	
DECL :TSMB I/Q/D/B/T/C: D KM/KH/KY/KS/KF/KT/KC/KG: KS	
DECL :NSMB I/Q/D/B/T/C: D KM/KH/KY/KS/KF/KT/KC/KG: KY	
DECL :TRMB I/Q/D/B/T/C: D KM/KH/KY/KS/KF/KT/KC/KG: KS	
DECL :NRMB I/Q/D/B/T/C: D KM/KH/KY/KS/KF/KT/KC/KG: KY	
DECL :TCBS I/Q/D/B/T/C: D KM/KH/KY/KS/KF/KT/KC/KG: KS	
DECL :NCBS I/Q/D/B/T/C: D KM/KH/KY/KS/KF/KT/KC/KG: KY	
DECL :TCBR I/Q/D/B/T/C: D KM/KH/KY/KS/KF/KT/KC/KG: KS	
DECL :NCBR I/Q/D/B/T/C: D KM/KH/KY/KS/KF/KT/KC/KG: KY	
DECL :MODE I/Q/D/B/T/C: D KM/KH/KY/KS/KF/KT/KC/KG: KF	

FB230 STL (Continued)	Description
:	FB230 (continued)
:L KH 0100	ID for activation
:T FW 200	ASCII driver
:L RS 47	Do not change RS 47, since it is
:T FW 202	used by the operating system
:LW =TPAR	Type of parameter list
:T FW 204	
:LW =NPAR	Address of parameter list
:T FW 205	
:LW =TSMP	Type of send mailbox
:T FW 207	
:LW =NSMP	Address of send mailbox
:T FW 208	
:LW =TRMB	Type of receive mailbox
:T FW 210	
:LW =NRMB	Address of receive mailbox
:T FW 211	
:LW =TCBS	Type of "Send" coordination byte
:T FW 213	
:LW =NKRS	Address of "Send" coordination byte
:T FW 214	
:LW =TCBS	Type of "Receive" coordination byte
:T FW 216	
:LW =NCBR	Address of "Receive" coordination byte
:T FW 217	
:LW =MODE	Specification of driver mode number
:T FY 219	
:L DH 000204DB	Absolute address of FY 219
:L DH 000E106F	Address of system data 55 (low-order byte)
:TNB 20	Initialize ASCII driver and enter
:	in RS 46 to 55
:L KB 0	Delete flag area used
:T FD 200	
:T FD 204	
:T FD 208	
:T FD 212	
:T FD 216	
:BE	

Note with TPAR, TSMB, TRMB, TCBS, TCBR: Specify XB parameter for DX blocks
Specify SY parameter for S flags

Note with NPAR, NSMB, NRMB, NCBS, NCBR: Specify flag number in KY in the case of S flags
Example: SY 258
Parameter = 1,2 ($\triangleq 1 \times 256 + 2 \times 1$)

OB1 STL	Description
:	
:AN F 0.0	
:L KT 200.0	
:SD T 0	
:A T 0	
:= F 0.0	
:	Call FB1 every 2 seconds
:JC FB 1	
NAME :DRUCKEN	
:	
:BE	

Sample function block FB1 is used to print out message texts stored in send data block DB203.

Output to printer is initiated each time the function block is invoked and the send trigger bit (CBS bit 7) reset.

Each time FB1 is invoked, the number output in the message text is incremented by 1.

Function block FB4 converts the message number from binary to ASCII.

FB1 STL	Description
NAME :DRUCKEN	Message output FB
:C DB 203	Call send mailbox DB
:	
:A F 200.7	CBS bit: "SEND" (PRINT)
:JC =ENDE	
:	
:L FW 202	Increment message output number
:ADD KF +1	For example by 1
:T FW 202	
:	
:JU FB 4	Call conversion FB
NAME :DU>ASCII	
DUAL : FW 202	Source number (BINARY)
A-TH : DW 21	ASCII Repres. TH/H (Data words to be
A-ZE : DW 22	ASCII Repres. Tens/ones updated in send DB)
:	
:L FW 204	Increment error text number
:ADD KF +2	for example by 2
:T FW 204	
:	

FB1 STL (Continued)	Description
:JU FB 4	Call conversion FB
NAME :DU>ASCII	
DUAL : FW 204	
A-TH : DW 45	Data words to be updated
A-ZE : DW 46	in send DB
:	
:AN F 200.7	CBS bit 7
:S F 200.7	Initiate printing
:	
END :BE	

FB4 STL	Description
NAME :DU>ASCII	Convert binary number to ASCII number
DECL :DUAL I/Q/D/B/T/C: I BI/BY/W/D: W	
DECL :A-TH I/Q/D/B/T/C: Q BI/BY/W/D: W	
DECL :A-ZE I/Q/D/B/T/C: Q BI/BY/W/D: W	
:L KB 0	Clear auxiliary reg.
:T FW 240	
:T FW 242	
:T FW 244	Remainder reg.
:	
:L =DUAL	Load binary number (val. range 0-9999)
:L KF +9999	
:>F	
:BEC	
:TAK	
SUBT :L KF +1000	Evaluate thousands place
:>=F	
:JC =TAUS	Jump to process thousands place
:TAK	
SUBH :L KF +100	Evaluate hundreds place
:>=F	
:JC =HUND	Jump to process hundreds place
:TAK	
SUBZ :L KF +10	Evaluate tens place
:>=f	
:JC =ZEHN	Jump to process tens place
:JU =EINE	Jump to process ones place
:	
TAUS :-F	
:T FW 244	
:L FY 240	
:ADD KF +1	
:T FY 240	Increment counting reg. for thousands
:TAK	

FB4 STL (Continued)	Description
:JU =SUBT	Jump to process thousands place
:	
HUND :-F	
:T FW 244	
:L FY 241	
:ADD KF +1	
:T FY 241	Increment counting reg. for hundreds
:TAK	
:JU =SUBH	Jump to process hundreds place
:	
ZEHN :-F	
:T FW 244	
:L FY 242	
:ADD KF +1	
:T FY 242	Increment counting reg. for tens
:TAK	
:JU =SUBZ	Jump to process tens place
:	
EINE :TAK	
:T FY 243	Describe counting reg. for ones
:	
:L KH 3030	
:L FW 240	
:OW	
:T =A-TH	
:TAK	
:L FW 242	
:OW	
:T =A-ZE	
:	
:BE	

ASCII Driver Parameter Data Block DB202 for Sample Program

DB202 STL	Description
0: KF = +00008;	Baud rate: 8=9600 baud
1: KF = +00000;	Parity: 0=even parity
2: KF = +00007;	Bits/char.: 7=7 bits
3: KH = 0000;	Waiting time after CR: (none)
4: KH = 0000;	Waiting time after LF: (none)
5: KH = 0000;	Waiting time after FF: (none)
6: KH = 000A;	Delay time between 2 char.: A = 100 ms
7: KH = 0004;	End-of-text char.: ″EOT″
8: KH = 0001;	Suppress LF: NO
9: KF = +00066;	Lines/page: 66
10: KF = +00000;	Left margin: 0 characters
11: KS =' u';	Page number at bottom of page
12: KH = 1B38;	Spaced print ON
13: KS =' MESSAGE LOG: CPU945	Header line 1

DB202 STL (Continued)	Description
25: KS ='ASCII DRIVER ';	
33: KH = 1B3C;	Spaced print OFF
34: KH = 0D0A;	CR / LF
35: KS ='========================';	Header 2
47: KS ='========================';	
59: KS ='========================';	
71: KS ='========';	
75: KH = 0D0A;	CR / LF
76: KS ='************************';	Footer 1
88: KS ='**********<Page>********';	
100: KS ='************************';	
112: KS ='********';	
116: KH = 0D0A;	CR / LF
117: KS =' Exam';	Footer 2
129: KS ='ple for CPU945 ASCII dri';	
141: KS ='ver interface ';	
151: KH = 0D0A;	CR / LF
152:	

Send Data Block DB203 for Sample Printer Output Program

DB203 STL	Description
0: KH = 0A0D;	Control char.: LF / CR
1: KH = 1B5B;	Activate control char. for
2: KH = 3477;	pitch 1/17
3: KS =' Process mess'	Message text
15: KS ='age NO.: ':	
20: KH = 1B30;	Control char.:Underline ON
21: KS ='0000';	Text message number (is used by FB4)
23: KH = 1B39;	Control char.:Underline OFF
24: KS =' *** >';	Message text
28: KH = 1B30;	Control char.:Underline ON
29: KS =' C A U T I O N B U R N';	Message text
41: KS =' E R 0000 H A S F A ';	Message text and message number
53: KS ='I L E D ! ';	Message text
60: KH = 1B39;	Control char.:Underline OFF
61: KS ='< ';	Message text
62: KH = 200D;	SPACE and CR
63: KH = 1B5B;	Activate control char.
64: KH = 3177;	for pitch 1/10
65: KH = 0A04;	End-of-text char. is EOT (see PAR-DB202)
66: KH = 0000;	
67:	

12.5.7 ASCII Code

Table 12.26 ASCII Code

Hex	ASCII	Hex	ASCII	Hex	ASCII	Hex	ASCII
00	NUL	20	SP	40	@	60	,
01	SOH	21	!	41	A	61	a
02	STX	22	″	42	B	62	b
03	ETX	23	#	43	C	63	c
04	EOT	24	$	44	D	64	d
05	ENQ	25	%	45	E	65	e
06	ACK	26	&	46	F	66	f
07	BEL	27	´	47	G	67	g
08	BS	28	(	48	H	68	h
09	HT (TAB)	29	)	49	I	69	i
0A	LF	2A	*	4A	J	6A	j
0B	VT	2B	+	4B	K	6B	k
0C	FF	2C	,	4C	L	6C	l
0D	CR	2D	-	4D	M	6D	m
0E	SO	2E	●	4E	N	6E	n
0F	SI	2F	/	4F	O	6F	o
10	DLE	30	0	50	P	70	p
11	DC1 (XON)	31	1	51	Q	71	q
12	DC2 (TAPE)	32	2	52	R	72	r
13	DC3 (XOFF)	33	3	53	S	73	s
14	DC4 (TAPE)	34	4	54	T	74	t
15	NAK	35	5	55	U	75	u
16	SYN	36	6	56	V	76	v
17	ETB	37	7	57	W	77	w
18	CAN	38	8	58	X	78	x
19	EM	39	9	59	Y	79	y
1A	SUB	3A	:	5A	Z	7A	z
1B	ESC	3B	;	5B	[	7B	{
1C	FS	3C	<	5C	\	7C	|
1D	GS	3D	=	5D	]	7D	~
1E	RS	3E	>	5E	∧	7E	}(ALT MODE)
1F	US	3F	?	5F	-	7F	DEL (RUB OUT)

12.6 Computer Link with 3964(R) Transmission Protocol

The CPU 945 permits data interchange with the 3964(R) transmission protocol via its second interface. Data can be exchanged between

- two programmable controllers (two CPUs)
  or
- a programmable controller and another communications partner.

Possible communications partners for the CPU 945 (SI2) using the 3964(R) transmission protocol:

- CPU 945 (SI2)
- CPU 944 (SI2)
- CP 523
- CP 524/525 (e.g. with S5 R006 special driver "Programmable computer link with 3964(R) protocol without reaction message frame". (Order No.: 6ES5 897-2AB11-03))
- Other communications partners (with 3964(R) transmission protocol)
  e.g.: - SICOMP PC
  - Teleperm M
  - Moby I

The control program on the CPU initiates the data interchange.
The transmission procedure controls the exchange.

The transmission procedure permits two types of data communications:

● Data interchange with block check character (BCC):	3964R procedure
● Data interchange without block check character:	3964 procedure

The block check character generates the vertical parity (exclusive OR operation) of all the bytes of a block that have been transmitted.

Data can be interchanged by means of the 3964(R) transmission procedure only if

- the parameters have been assigned according to Sections 12.6.4 and 12.6.5 and
- the computer link has been activated by the corresponding entry in the high-order byte of RS 46 (E 105C$_H$).

The ASCII driver can also be parameterized and activated by parameterizing DB1 accordingly (see Chapter 11).

After activation of the computer link, any error flags concerning the CBR or CBS are set in the low-order byte of RS 46.

Note

When the computer link is activated, SI2 is no longer available for any other functions (e.g. programmer/OP, SINEC L1, point-to-point connection).

Table 12.27 Meaning of System Data Word 46 (Computer Link)

Byte	Assignment	Meaning
High-order byte	00$_H$*	Programmer/OP and SINEC L1 operation
	02$_H$	Driver for 3964(R) computer link active
Low-order byte		Error checkback signals
	00$_H$	No error
	01$_H$	Invalid driver number
	10$_H$	CBS not available
	20$_H$	CBR not available
	40$_H$	CBS and CBR not available

* Default value (preset value) after overall reset

The communications partners are linked via a direct line.

Cable requirements:

- 4-core
- Shielded
- Cross-section ≥ 0.14 mm^2 (26 AWG)

The SIMATIC cable 6ES5 707-1AA00 is recommended.

For connector pin assignments, refer to Section 12.7 "Interface Modules".

12.6.1 3964(R) Transmission Protocol

This section describes the 3964(R) transmission protocol. The parameter assignments and activation of the computer link are explained in Sections 12.6.4 and 12.6.5.

The 3964(R) driver ensures relatively safe data transmission since the receiver must first send a ready signal to the transmitter (connection buildup) and acknowledge the reception of data after the data interchange. The block check character transmitted in addition to the data also enhances data security.

The 3964(R) driver interprets the following control characters:

●	DLE	(10_H)	Data Link Escape
●	STX	(02_H)	Start of Text
●	NAK	(15_H)	Negative Acknowledgement
●	ETX	(03_H)	End of Text

Transmission Procedure

Control characters and useful data are transmitted bit by bit.

When mode 2 is selected, the data unit transmitted, also called message frame, is followed by a block check character. Like all other data bytes, the BCC is protected by the relevant parity check and transmitted at the end of the message frame. Mode 2 must be set in system data word 55 (see Section 12.6.4).

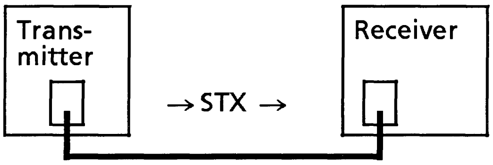
Prior to a transmission, the data is buffered in a 1024-byte output buffer. If the amount of data to be transmitted is too large for the output buffer, an error bit is set (see Section 12.6.3).

The data received is first entered in the 1024-byte input buffer of the receiver before the user program initiates the transfer to the receive mailbox of the CPU.

Sending/Receiving with the 3964/3964R Line Procedure in Detail

Connection Buildup

The 3964(R) line procedure executes the following steps automatically.



When there is no Send order to process, the 3964(R) driver waits for the peer in the link to establish a connection.

STX is a control character (02_H) which initiates connection buildup.

Possible Responses from the Receiver	Response (from the Transmitter's Point of View)
Receiver acknowledges with DLE (10_H) prior to time-out (QVZ).	Connection buildup was successful; the transmitter sends the first character from the send buffer. (QVZ: Word 5 in the parameter list)
Receiver acknowledges with a character other than DLE or STX prior to time-out (QVZ) or receiver does not acknowledge prior to time-out	Connection buildup was initially unsuccessful; the transmitter makes another attempt to establish a connection. (Number of attempts to establish a connection: Word 7 in the parameter list). If the last attempt also proves unsuccessful, the transmitter enters a code in the Send coordination byte (CBS).
Receiver transmits an STX control character prior to time-out (QVZ)	Initiation conflict, i.e. both peers in the link want to transmit. The peer with the lower priority sends DLE, thus enabling the higher-priority node to transmit first; the lower-priority node then transmits its data. The two peers must never have the same priority! (Priority: Word 3 in the parameter set)

Sending and Receiving Frames

- Each character whose value is 10_H (DLE) is transmitted twice in succession so that the receiver does not interpret it as the control character for connection buildup. The receiver enters only one of the two characters in its Receive buffer.

- The receiver monitors the time that elapses between transmission of two consecutive characters. If it exceeds the specified character delay time (ZVZ), the receiver sends a NAK and waits the amount of time defined in word 6 of the parameter list for the frame to be retransmitted. (Character delay time: Word 3 in the parameter list).

- The following occurs when the receiver's Receive buffer is full before the transmitter has initiated a connection cleardown:
  - receiving continues until the connection has been cleared down
  - the receiver subsequently transmits the NAK control character
  - the error is flagged in the CBR.

- If the receiver sends a NAK character to the transmitter while transmission is in progress, the transmitter aborts the data transfer and retransmits the entire frame, beginning with the first character.

- If the receiver sends a character other than NAK while transmission is in progress, the transmitter ignores it and continues its transmission.

- The receiver reacts to a transmission error (character lost, bad frame, parity error, BCC error) as follows:
  - Reception continues until the connection is cleared down
  - NAK is then transmitted
  - If an attempt to transmit is still possible (word 8 of the parameter list), the receiver waits for the frame to be retried. How long the receiver waits depends on the frame delay time (word 6 in the parameter list).

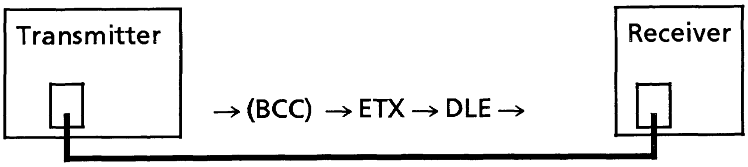
  The receiver aborts transmission and reports an error in CBR
  - if the data block could not be received at the last send attempt
    or
  - if the sender does not start sending within the block waiting time.

- The "BREAK" signal causes the transmitter to
  - abort the current transmission
  - send NAK
  - flag an error in the CBS.

- If a message frame is not accepted (no positive acknowledgement) after the set number of attempts to establish a connection or transmit, the sender responds by sending a NAK.

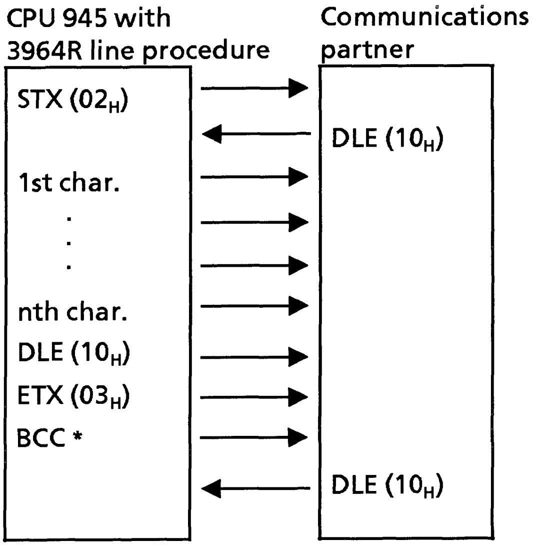
Connection Cleardown

When all characters in the Send buffer have been transmitted, the transmitter initiates connection cleardown by transmitting in succession the control characters DLE (10_H), ETX (03_H) and, if specified, BCC (block check character for 3964R).



Possible Reactions from the Receiver	Explanation
Receiver sends a DLE control character prior to time-out (QVZ)	The frame was received without error and the connection cleared down.
Receiver sends a NAK control character or any other character (except DLE !) prior to time-out or the receiver does not send any character prior to time-out	If the specified number of transmission attempts is greater than 1, the frame is retransmitted (number of attempts: word 8 in the parameter list). If the last attempt is unsuccessful, the transmitter aborts the transmission and flags an error in the CBS.

Example of an Error-Free Send Procedure



* with 3964R transmission protocol only

Figure 12.16 Error-Free Send Procedure (Computer Link)

Example of an Error-Free Send Procedure:

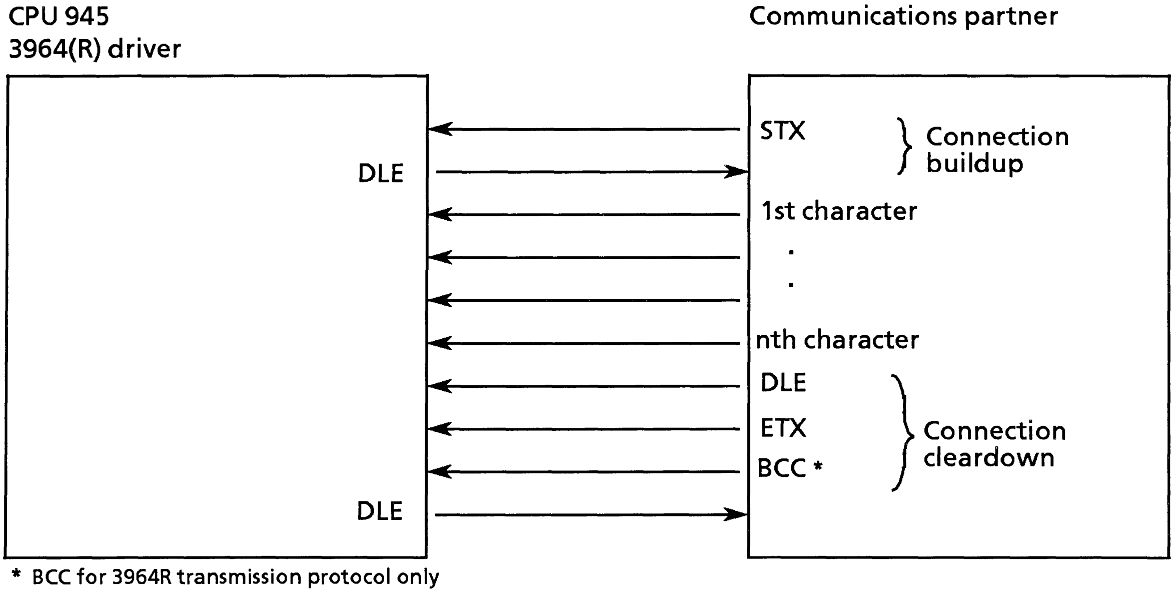
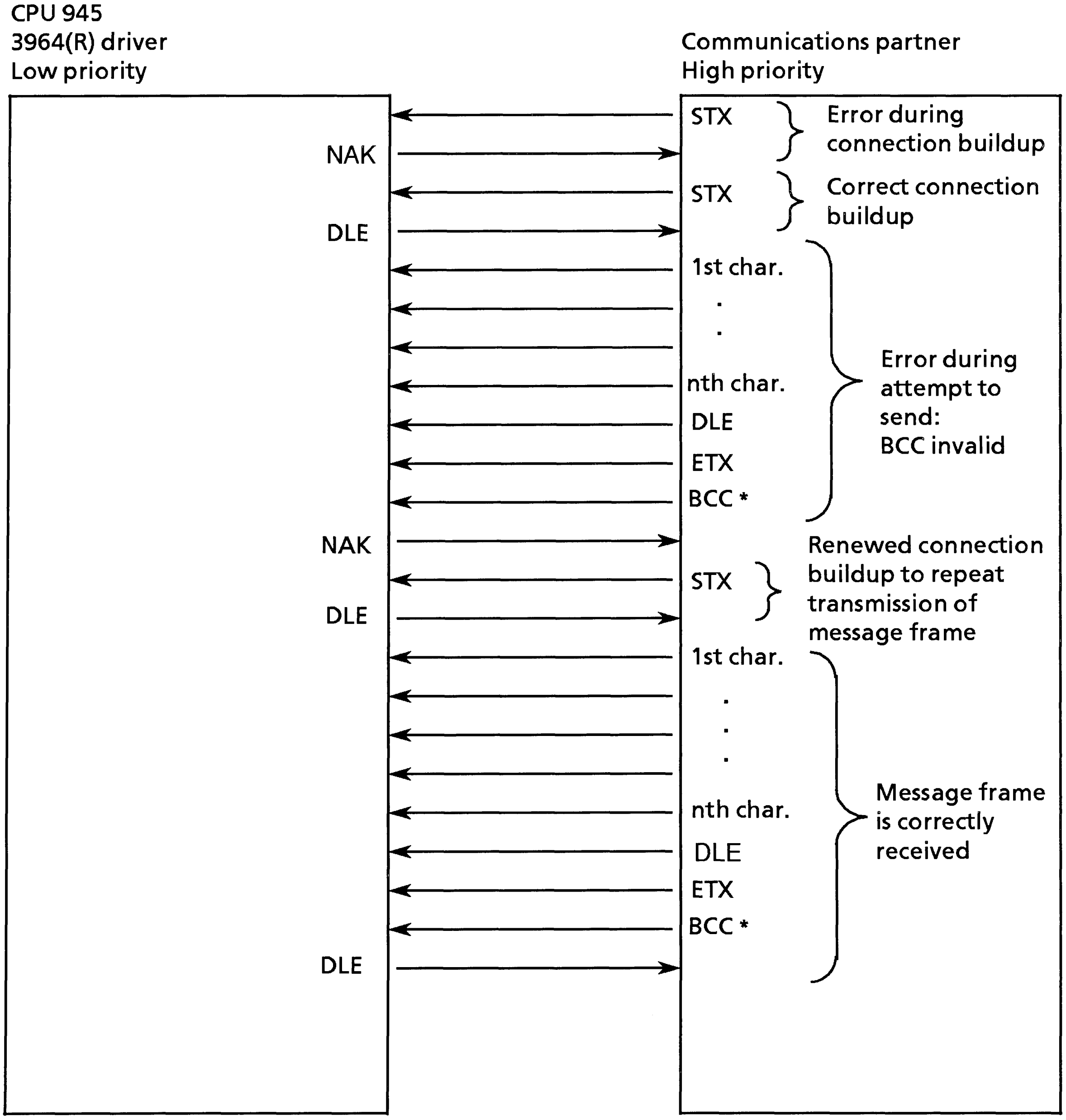


Figure 12.17 Error-Free Receive Procedure (Computer Link)

Examples of Errors During Data Transmission:


CPU 945
3964(R) driver
Low priority
Communications partner
High priority
STX
Error during connection buildup
NAK
STX
Correct connection buildup
DLE
1st char.
nth char.
DLE
ETX
BCC *
Error during attempt to send: BCC invalid
NAK
STX
Renewed connection buildup to repeat transmission of message frame
DLE
1st char.
nth char.
DLE
ETX
BCC *
Message frame is correctly received
DLE

* BCC for 3964R transmission protocol only

Figure 12.18 Errors During Data Transmission (Computer Link)

Example of How to Solve an Initiation Conflict

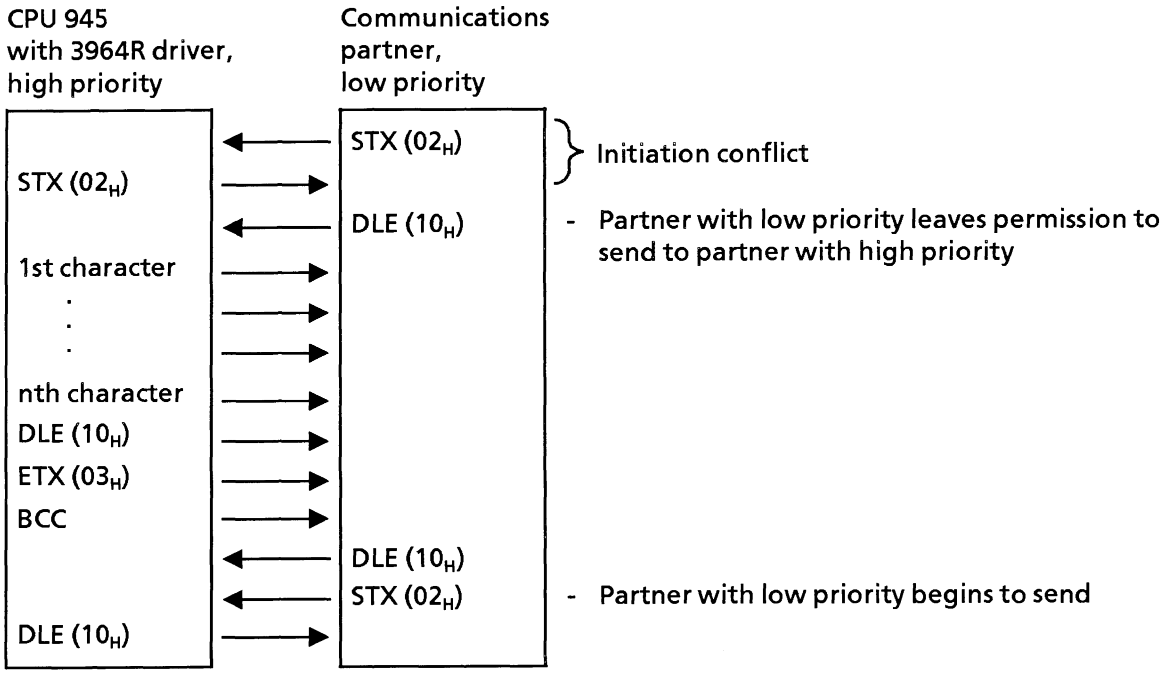


Figure 12.19 Solving an Initiation Conflict (Computer Link)

12.6.2 Data Interchange over the SI2 Interface with 3964(R) Transmission Protocol

The data to be transferred must be entered in an area of memory designated as the "Send mailbox".
Conversely, the data to be received requires a "Receive mailbox", and an area in memory must therefore also be designated for this purpose (detailed information is presented in the next section).
The data is stored temporarily in interface SI2's input or output buffer. Figure 12.20 illustrates the procedures involved in data interchange.

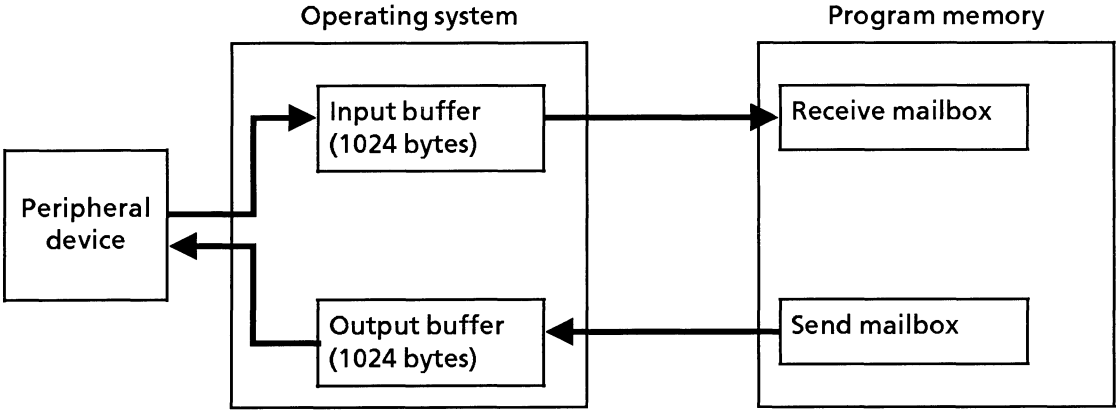


Figure 12.20 Data Interchange over the SI2 Interface (Computer link)

Transmitting Data

- The length of the frame to be transmitted (in bytes) must be entered in the first word of the Send mailbox.

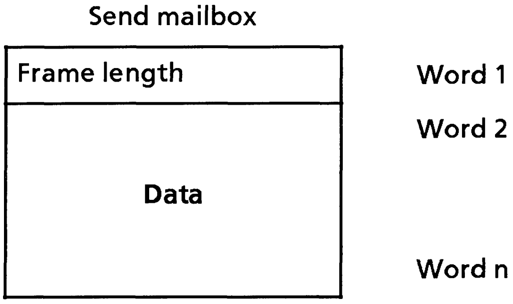


Figure 12.21 Structure of the Send Mailbox (Computer Link)

- The data to be transmitted must be entered in the remaining words of the Send mailbox.
- Set bit 7 in the CBS (a rising edge triggers the transmission). The computer link resets this bit when transmission has been completed.
  The bit is reset irrespectively of whether the send procedure was correct or errored.

If an error occurred during data transmission, an error code is entered in bits 0 to 6 of the CBS which describes the type of error (see Section 12.6.3).

Receiving data

Receive data (message frames) are automatically written in the input buffer of the computer link (buffer size: 1024 bytes) if there is sufficient space or less than 100 message frames are in the buffer. Otherwise, an error flag is written in the input buffer and can be evaluated in the CBR. The error flag is ***not*** stored in the input buffer if it has been stored there immediately before (see Section 12.6.3).

In order to transfer this data to the receive mailbox, bit 7 must be set in the CBR by the control program. The computer link automatically enters the number of bytes received in the first word of the receive mailbox. When the data of a message received is stored in the receive mailbox, the computer link resets bit 7 of the CBR. If the data was not correctly received, an error code is written in bits 0 to 6 of the CBR and bit 7 of the CBR is reset (see Section 12.6.3).

Since various errors can occur in one receive request, the computer link assigns priorities to the individual errors. The CBR always contains the error that had the highest priority during the last attempt to receive. 0 indicates the highest priority, 6 the lowest.

12.6.3 Coordination Bytes of the 3964(R) Driver

The 3964(R) driver monitors data communications via

- the coordination byte for "Send" (CBS) and
- the coordination byte for "Receive" (CBR).

The 3964(R) driver enters status and error messages in these bytes.

A coordination byte can either be

- a flag byte (FY/SY) or
- a high-order byte in a data word (DB/DX)

Tables 12.28 and 12.29 describe the meanings of the individual bits of the coordination byte.

Table 12.28 Meanings of the Individual Bits of the "Send" Coordination Byte (CBS) in a Computer Link

Bit	Meaning		
	Hex Code	**Error Flags**	**Reaction**
0 ... 6	07_H	Frame exceeds output buffer capacity	Data transmission not possible
	09_H	Negative acknowledgement from receiver during connection cleardown	Receive data invalid
	$0B_H$	Negative acknowledgement from receiver during connection buildup	Data transmission not possible
	$0D_H$	Parameter assignment error	
	$0F_H$	Receiver aborted transmission	Receive data invalid
	11_H	No Send mailbox	Data transmission not possible
	13_H	Frame exceeds send mailbox capacity	
	15_H	Time-out during connection buildup	
	17_H	Time-out during connection cleardown	Receive data invalid
	19_H	Initiation conflict, both partners have high priority	Data transmission not possible
	$1B_H$	Break	Transmission is aborted
	$1D_H$	Initiation conflict, both partners have low priority	Data transmission not possible
7	**Permission to send** The bit is set by the control program and reset by the 3964(R) driver irrespective of whether the send procedure is terminated with or without error. The transmission is initiated by a positive-going edge at bit 7. The send data and the location of the send mailbox must not be modified as long as this bit is "1".		

Table 12.29 Meanings of the Individual Bits of the Coordination Byte for "Receive" (CBR) in a Computer Link

Bit	Meaning			
	Hex Code	Error Flags	Priority	Reaction
0 ... 6	03_H	Parity error	5	Data rejected
	05_H	Frame length is 0	6	
	07_H	Input buffer full	2	
	09_H	Too many frames received (more than 100)	2	Data valid, subsequent frames rejected
	$0B_H$	Frame longer than Receive mailbox	0	
	$0D_H$	DLE not transmitted twice in succession or no ETX after DLE*	3	Data rejected
	$0F_H$	No Receive mailbox	0	
	11_H	STX error: Handshaking was not started**	3	
	13_H	Character delay time ZVZ exceeded	4	
	15_H	Frame delay time BWZ exceeded	2	
	17_H	Check sum error	5	
	19_H	Framing error	5	
	$1B_H$	Break	1	
7		**Permission to receive** The bit is set by the user and reset by the 3964(R) driver when a frame received has been transferred from the input buffer to the Receive mailbox or an error flag has been set in the CBR. ● As long as the bit is "1", the receive data and the location of the receive mailbox must not be modified. ● As long as the bit is "0", the user must not access the receive mailbox. As long as the input buffer is not full, the messages are stored there and are not entered in the Receive mailbox. Up to 100 message frames can be entered in the receive buffer.		

* DLE and ETX are control characters for connection buildup and connection cleardown (DLE = Data Link escape, ETX = End of Text).
The line procedure automatically doubles a byte of data that has the same code as a control character (DLE in this case) in order to be able to distinguish the data from the control character.
DLE-ETX is a fixed sequence that is required for error-free connection buildup/cleardown.

** STX is the control character which establishes the connection to the communications partner (STX = start of text)

Note

The bits in the coordination bytes can be set or reset by the operating system after any operation and irrespective of the PLC cycle. This means that a multiple scan of a coordination byte within one program cycle can have different results (This must be taken into consideration for an edge evaluation!)

12.6.4 Parameter Set of the 3964(R) Driver

The parameter set contains the defaults necessary for a data interchange. The location of the parameter set is defined by system data word 48 (or both 48 and 49) (see Table 12.33).

For permissible parameter settings and defaults, refer to Table 12.30.

The 3964(R) driver can also be parameterized in DB 1 (see Chapter 11).

Note

The default values are used only if the parameter set is not available or cannot be interpreted.

Table 12.30 Parameter Set (Computer Link)

Word	Description	Value Range	Default
0	Baud rate	1 150 baud 2 200 baud 3 300 baud 4 600 baud 5 1200 baud 6 2400 baud 7 4800 baud 8 9600 baud 9 19200 baud	8
1	Parity	0 even 1 odd 2 mark (filler bit high) 3 space (filler bit low) 4 no check	0
2	Data format*	0 to 8	1
3	Priority	0 low 1 high	1
4	Character delay time (The max. amount of time which may elapse between transmission of two consecutive characters)	1 to 65535×10 ms	22
5	Acknowledgement delay-time (time-out) (The time within which a request-to-send or a complete frame (DLE, ETX) must be acknowledged)	1 to 65535×10 ms	200
6	Frame delay time (If the character delay time is exceeded, the complete retransmitted frame must arrive in the receiver before the frame delay time is exceeded)	1 to 65535×10 ms	400
7	Connection buildup attempts (Maximum number of attempts that may be made to build up a connection)	1 to 255	6
8	Number of transmission attempts (Maximum number of attempts that may be made to transmit a block)	1 to 255	6

* Meaning of word 2 (data format) see Table 12.31

The parameter set is read at activation of the computer link or after a mode change; data traffic at the interface must have previously been terminated, however (bit 7 in CBR and and bit 7 in CBS=0). The parameter set is also transferred after PLC POWER ON if the computer link had been previously activated.

The parameter settings on the CPU and in the communications partner must be identical to word 3 (priority). The opposite priority must be the default in the communications partner so that a parameterization conflict can be resolved.

Please note these time relationships when setting the following:

Character delay time < timeout < block waiting time!

The send or receive process can be initiated when these defaults have been completed.

Table 12.31 Character Frame and Order of Bits on the Line in the Case of a Computer Link (Depending on Word 2 of the Parameter Set)

Word 2 of the Parameter Set (Data Format)	Character Frame	Parity	Number of Bits per Character	Order of Bits on the Line
0	11 bits	0 to 4*	7	1 start bit, 7 data bits, 1 parity bit, 2 stop bits
1	11 bits	0 to 4*	8	1 start bit, 8 data bits, 1 parity bit, 1 stop bit
2	11 bits	Setting irrelevant	8	1 start bit, 8 data bits, 2 stop bits
3	10 bits	Setting irrelevant	7	1 start bit, 7 data bits, 2 stop bits
4	10 bits	0 to 4*	7	1 start bit, 7 data bits, 1 parity bit, 1 stop bit
5	10 bits	Setting irrelevant	8	1 start bit, 8 data bits, 1 stop bit
6	-	-	-	-
7 As data format 0	11 bits	0 to 4*	7	1 start bit, 7 data bits, 1 parity bit, 2 stop bits
8 As data format 1	11 bits	0 to 4*	8	1 start bit, 8 data bits, 1 parity bit, 1 stop bit

* see Table 12.30

Assigning a Mode Number (System Data 55)

Data can be transmitted in two different modes. The mode selected, i.e. its number, is to be entered in the low-order byte of system data word 55 (0E 106E$_H$) (see Table 12.33).

Refer to Table 12.32 for the mode types.

Table 12.32 Meanings of the Mode Numbers (Computer Link)

Mode	Meaning
1	***No*** block check character (BCC) is transmitted at the end of a frame (3964)
2	A block check character (BCC) is transmitted at the end of each frame (3964R)

12.6.5 Assigning Parameters to the 3964(R) Driver

The location of the parameter set, of the send and receive mailboxes, the coordination bytes and the mode number are to be defined by means of the control program in a parameter block (see Table 12.33) in the system data area of the CPU 945.

Parameters can also be assigned in DB1 (see Chapter 11).

Table 12.33 Parameter Block for Computer Link

System Data Word	High-Order Byte	Low-Order Byte	Absolute Address
RS 48	Parameter set Data ID	Parameter set DB/DX- or flag byte, or high-order part of S flag address	0E 1060 0E 1061
RS 49	Parameter set Data word No. or low-order part of S flag address	Send mailbox Data ID	0E 1062 0E 1063
RS 50	Receive mailbox DB/DX or flag byte, or high-order part of S flag address	Send mailbox Data word No. or low-order part of S flag address	0E 1064 0E 1065
RS 51	Receive mailbox Data ID	Receive mailbox DB/DX- or flag byte, or high-order part of S flag address	0E 1066 0E 1067
RS 52	Receive mailbox Data word No. or low-order part of S flag address	CBS Data ID	0E 1068 0E 1069
RS 53	CBS DB/DX or flag byte, or high-order part of S flag address	CBS Data word No. or low-order part of S flag address	0E 106A 0E 106B
RS 54	CBR Data ID	CBR DB/DX- or flag byte, or high-order part of S flag address	0E 106C 0E 106D
RS 55	CBR Data word No. or low-order part of S flag address	Mode number*	0E 106E 0E 106F

* If no valid mode number is entered, the default number is "1".

12

The data IDs of the parameter block are described in the following table.

Table 12.34 Data IDs of the Parameter Block (Computer Link)

Stored in	Data ID		Memory Area*	
	Hex Code	ASCII Code		
Flag	$4D_H$	F	Flag byte No.: 0 to 255	- - -
S flag**	53_H	S	S flag byte No.: 0 to 4095	- - -
Data (DB)	44_H	D	DB No.: 0 to 255	Data word No.: 0 to 255
Data (DX)	58_H	X	DX No.: 0 to 255	Data word No.: 0 to 255

* The start addresses of the relevant memory areas, both for the parameter set of the computer link and for the send and receive addresses, are stated here.

** High-order part of S flag numbers 00_H ... $0F_H$; low-order part of S flag numbers 00_H ... FF_H

12.6.6 Program Example for Transmitting Data

During restart, the relevant computer link parameters are assigned to system data words 46 to 55 and the computer link is activated. For this purpose, a programmable function block (FB220) is used.

The parameters for the communications link are as follows:

- Parameter set in DB202, beginning with DW 0
- Send mailbox in DB203, beginning with DW 0
- Receive mailbox in DB204, beginning with DW 0
- The CBS is flag byte FY 100
- The CBR is flag byte FY 101
- The mode setting is: Mode 2 (with BCC)

The data to be transferred is in data words DW 1 to DW 5 of DB203. The frame length specification must therefore be 10 bytes.

The example describes the program of the communications partner. It can be used analogously for a CPU 945 which acts as a communications partner if the priority (DB202, DW 3) is changed to "low priority".

OB21/OB22 STL	Description
:	Initialize system data area
:JU FB 220	for communications link
NAME :PA-3964	
TPAR : KS DB	Parameter set for communications link
NPAR : KY 202,0	is located in DB202 beginning DW0
TSMB : KS DB	The send mailbox is located in
NSMB : KY 203,0	DB203 beginning DW0
TRMB : KS DB	The receive mailbox is located in
NRMB : KY 204,0	DB204 beginning DW0
TCBS : KS FY	The coordination byte for
NCBS : KY 100,0	send is flag byte FY100
TCBR : KS FY	The coordination byte for
NCBR : KY 101,0	receive is flag byte FY101
MODE : KF +2	Mode number: 2 (with BCC)
:	
:AN F 101.7	Receive enable
:S F 101.7	
:BE	

Notes for TPAR, TSMB, TRMB, TCBS, TCBR: For DX blocks → Specify "XB" parameter
For S flags → Specify "SY" parameter

Notes for NPAR, NSMB, NRMB, NCBS, NCBR: For S flags → Specify S flag No. in KY

Example: SY 258
Parameter=1.2 (≙1×256+2×1)

```
              FB220 STL                                      Description

             NAME :PA-3964
     DECL :TPAR        I/Q/D/B/T/C: D  KM/KH/KY/KS/KF/KT/KC/KG: KS
     DECL :NPAR        I/Q/D/B/T/C: D  KM/KH/KY/KS/KF/KT/KC/KG: KY
     DECL :TSMB        I/Q/D/B/T/C: D  KM/KH/KY/KS/KF/KT/KC/KG: KS
     DECL :NSMB        I/Q/D/B/T/C: D  KM/KH/KY/KS/KF/KT/KC/KG: KY
     DECL :TRMB        I/Q/D/B/T/C: D  KM/KH/KY/KS/KF/KT/KC/KG: KS
     DECL :NRMB        I/Q/D/B/T/C: D  KM/KH/KY/KS/KF/KT/KC/KG: KY
     DECL :TCBS        I/Q/D/B/T/C: D  KM/KH/KY/KS/KF/KT/KC/KG: KS
     DECL :NCBS        I/Q/D/B/T/C: D  KM/KH/KY/KS/KF/KT/KC/KG: KY
     DECL :TCBR        I/Q/D/B/T/C: D  KM/KH/KY/KS/KF/KT/KC/KG: KS
     DECL :NCBR        I/Q/D/B/T/C: D  KM/KH/KY/KS/KF/KT/KC/KG: KY
     DECL :MODE        I/Q/D/B/T/C: D  KM/KH/KY/KS/KF/KT/KC/KG: KF

          :                              Preset system data area
          :L    KH 0200                  Driver number for
          :T    FW 200                   computer link
          :L    RS 47                    Retain entry
          :T    FW 202                   in RS 47
          :LW   =TPAR                    Data type of parameter set
          :T    FW 204
          :LW   =NPAR                    Address of parameter set
          :T    FW 205
          :LW   =TSMB                    Type of Send mailbox
          :T    FW 207
          :LW   =NSMB                    Address of Send mailbox
          :T    FW 208
          :LW   =TRMB                    Type of Receive mailbox
          :T    FW 210
          :LW   =NRMB                    Address of Receive mailbox
          :T    FW 211
          :LW   =TCBS                    Type of CBS
          :T    FW 213
          :LW   =NCBS                    Address of CBS
          :T    FW 214
          :LW   =TCBR                    Type of CBR
          :T    FW 216
          :LW   =NCBR                    Address of CBR
          :T    FW 217
          :LW   =MODE                    Mode
          :T    FY 219
          :L    DH 000204DB              Abs. address of FY219 (source)
          :L    DH 000E106F              SD55 address (low-order byte)(dest.)
          :TNB      20                   Enter computer link parameter in RS 46 to 55
          :L    KB 0                     Reset flag area
          :T    FD 200
          :T    FD 204
          :T    FD 208
          :T    FD 212
          :T    FD 216
          :BE
```

OB1 STL	Description
:JU FB 1	SEND
NAME :SEND	
:	
:JU FB 2	RECEIVE
NAME :RECEIVE	
:BE	

FB1 STL	Description
NAME :SEND	
:C DB 203	Open Send mailbox
:O F 100.7	Go to end when send in progress
:ON I 0.0	or when no send request is pending
:BEC	(enable send with I 0.0)
:	
:A F 100.0	ERROR during last SEND?
:JC PB 1	Then evaluate error in PB1
:	
:	Load data into Send mailbox
:L KF +10	10 bytes are to be transferred
:T DW 0	(first word in Send mailbox)
:	
:L DW 1	Modify send data
:ADD KF +1	
:T DW 1	
:	
:AN F 100.7	Initiate send
:S F 100.7	
:BE	

FB2 STL	Description
NAME :RECEIVE	
:C DB 204	Open Receive mailbox
:A F 101.7	END if no data were received.
:BEC	
:	
:A F 101.0	Receive ERROR?
:JC PB 2	Then evaluate error in PB2
:A F 101.0	If errors occured, re-enable Receive mailbox and do not evaluate data
:S F 101.7	Evaluate Receive mailbox
:BEC	
:L DW 0	Evaluate number of bytes received
:T QW 0	
:L DW 1	Evaluate data received
:T QW 2	
:	
:AN F 101.7	Re-enable Receive mailbox
:S F 101.7	
:BE	

DB202	Description
0: KF = +00008;	Baud rate = 9600 baud
1: KF = +00000;	Even parity
2: KF = +00001;	Data format 1
3: KF = +00001;	High priority
4: KF = +00022;	Character delay time = 220 ms
5: KF = +00200;	Acknowledgement delay = 2 s
6: KF = +00400;	Frame delay time = 4 s
7: KF = +00006;	Max. number of conn. buildup attempts
8: KF = +00006;	Max. number of attempts to send
9:	

12.7 Interface Modules

This section describes the interface modules that can be used in conjunction with the CPU 945. Furthermore, the functions of the individual interface modules are explained.

Besides the first serial interface (SI1), the CPU 945 offers a receptacle for an interface module which serves as the second serial interface (SI2).

The following interface modules can be inserted in the CPU 945:

• Programmer module (15-pole)	6ES5 752-0LA52
• V.24 module (25-pole)	6ES5 752-0LA22
• TTY module (25-pole)	6ES5 752-0LA12
• RS 422-A/485 module (15-pole)	6ES5 752-0LA42
• SINEC L1 module (15-pole)	6ES5 752-0LA62

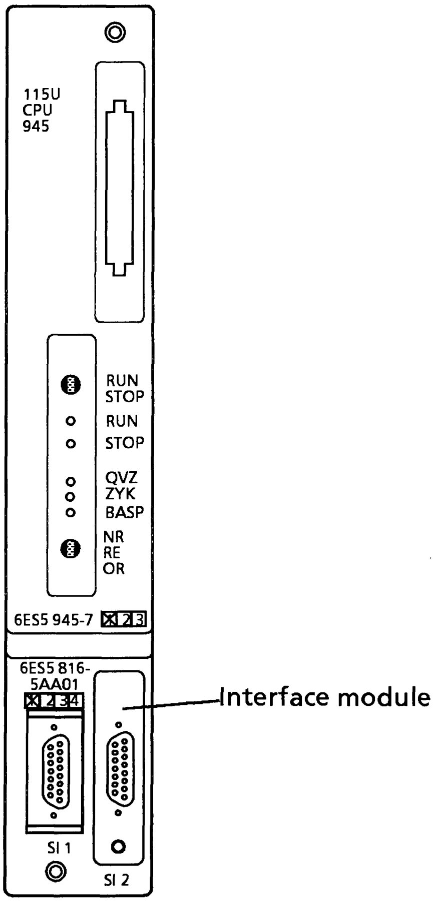


Figure 12.22 Location of the Interface Module in the CPU 945

Inserting the Interface Module in the CPU 945

Warning

The interface module can only be inserted or withdrawn at POWER-OFF.

Proceed as follows when connecting an interface module to SI2:

- Unscrew the cover plate of the interface module receptacle and remove it from the housing of the CPU 945.
- Insert the interface module in the receptacle.
- Fix the interface module by means of the screws.

The following table shows which interface modules can be used for the various types of links.

Table 12.35 Applications of the Interface Modules

	Programmer Module	V.24 Module	TTY Module	RS 422-A/ 485 Module	SINEC L1 Module
Programmer interface	X	X*	X	X	X
SINEC L1					X
Point-to-point connection (SINEC L1 protocol)					X
Computer link 3964(R)	X	X*	X	X	X
ASCII driver	X	X*	X	X	X

* Only V.24 signals RxD and TxD are supported.

12.7.1 Programmer Module

The programmer module permits the connection of programmers and operator panels to the second serial interface in addition to those connected to the first interface of the CPU 945.

Other permissible links:

- ASCII driver
- Computer link

The maximum permissible transmission rate is 9600 baud.

The programmer functions are limited if they are requested simultaneously at the first interface and at the programmer module. Depending on the functions activated at one interface, certain requests cannot be made by a programmer/OP connected to the other interface.

If a function requested at one interface clashes with the function initiated at the other interface, the operating system of the CPU interrupts this function and issues an error message to tell the operator that the interface function is disabled.

The programmer module incorporates both a transmitter and a receiver for 20 mA current-loop signals. The loop current is always fed from the programmer if a PG 7xx programmer is connected. The following diagram shows the connection of the loop current signals.

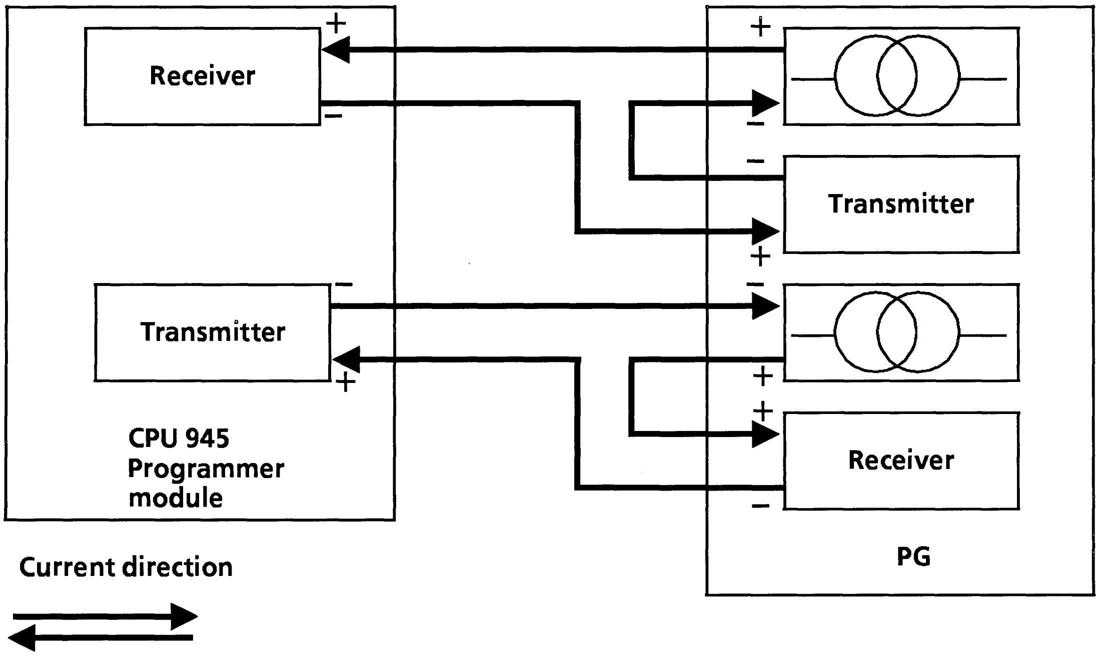
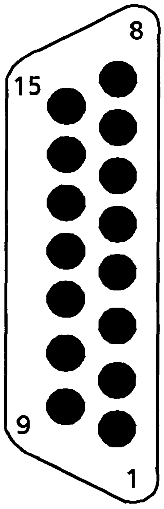


Figure 12.23 Programmer Module: Direction of Loop Current

Data is generally transmitted via the programmer interface at a rate of 9600 baud.

The following Figure shows the connector pin assignments of the 15-pole Cannon socket in the front plate of the programmer module (corresponds to assignments of SI1):



Pin	Signal	Comment
1	M_{ext}	
2	- RXD	
3	+ 5,2 V	
4	+ 24 V	
5	M	
6	+TXD	
7	- TXD	
8	M_{ext}	
9	+RXD	
10	M 24 V	
11	+20 mA	Current source
12	M	
13	+20 mA	Current source
14	+ 5.2 V	
15	M	

Figure 12.24 Programmer Module: Pin Assignments

Jumper settings on the programmer module

The programmer module is supplied with the jumper settings shown in Figure 12.25. The module is thus ready for use.

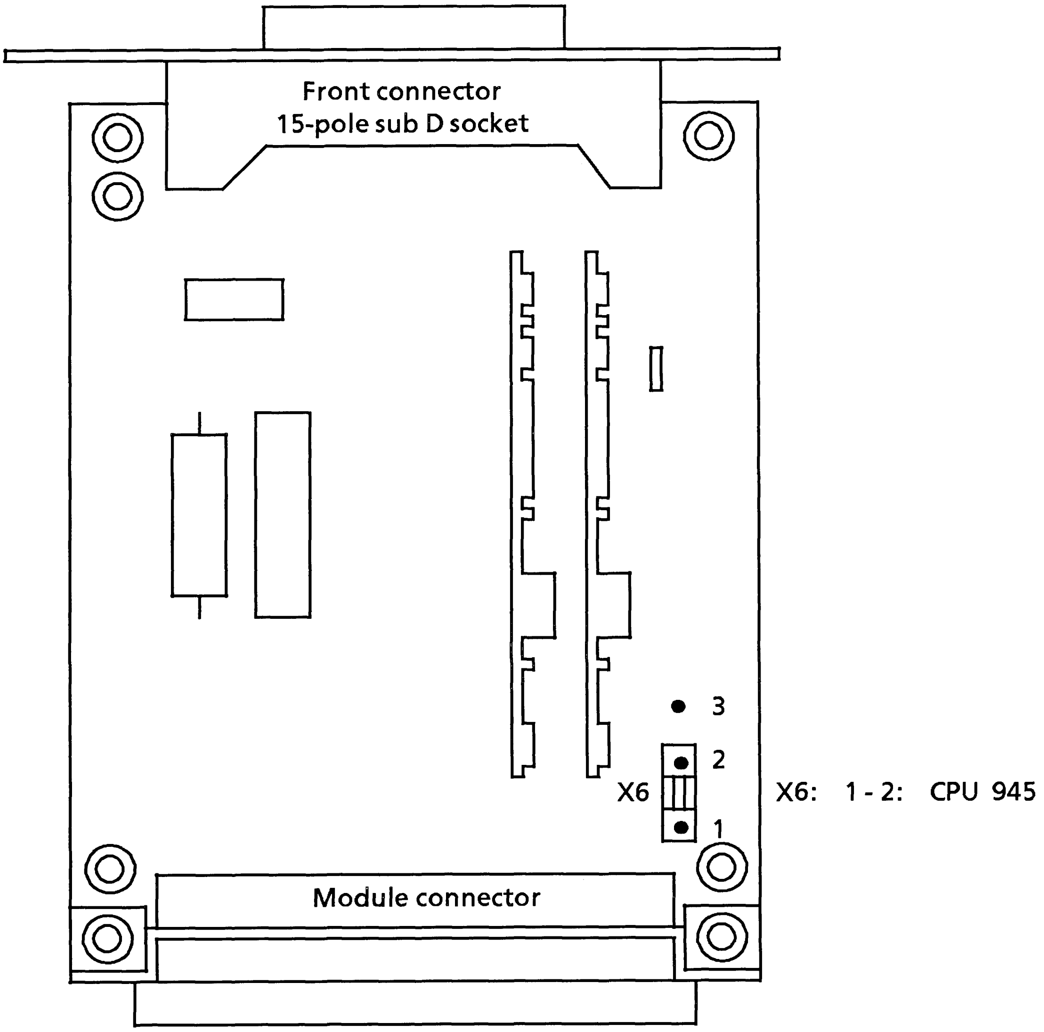


Figure 12.25 Programmer Module: Jumper Settings

Standard connecting cables for the programmer module

Standard connecting cables with lengths up to 1000 m (3300 ft) are available for linking the programmer module in the CPU 945 to the programmer.

CPU 945/programmer connecting cable

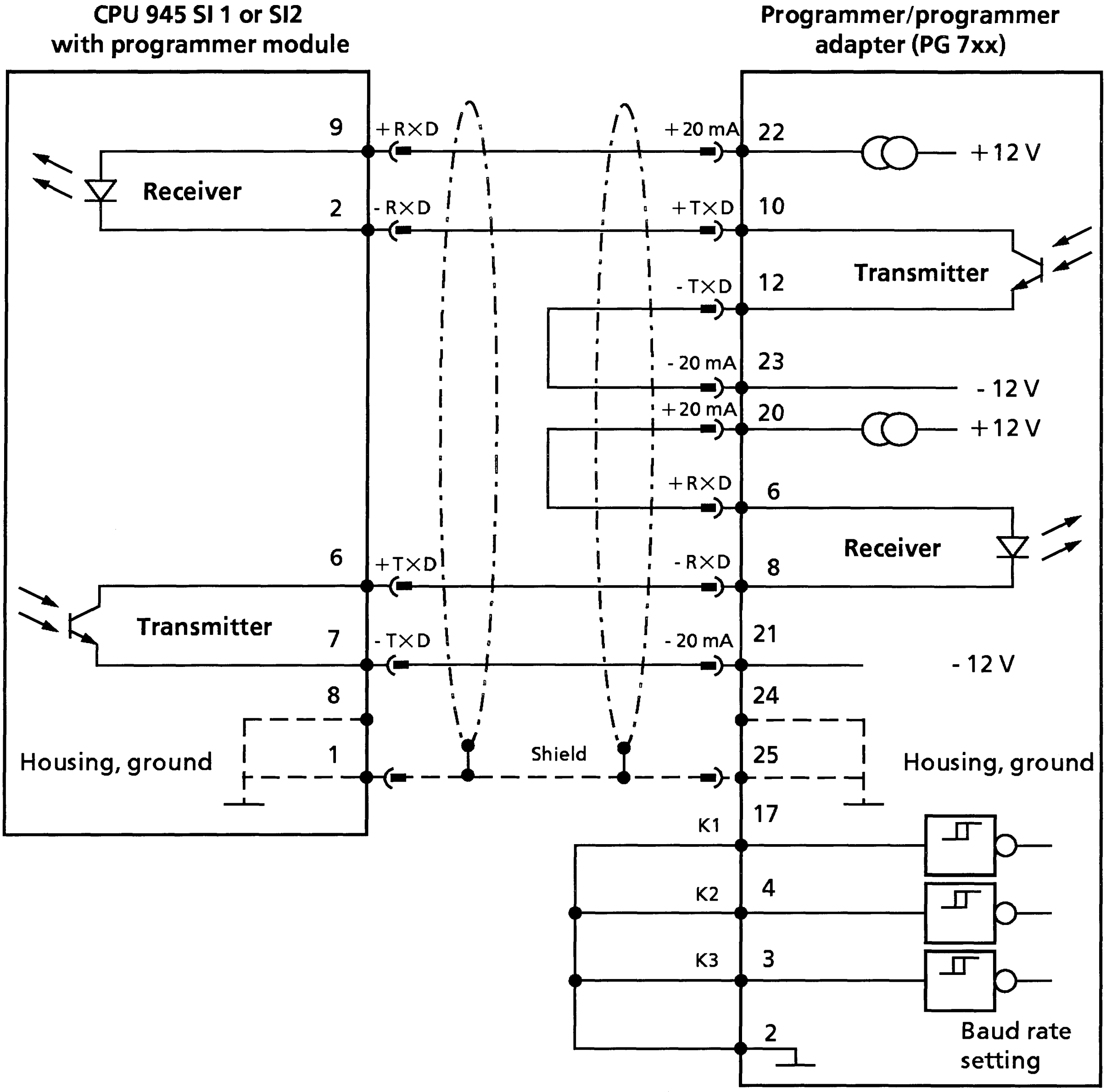


Figure 12.26 Programmer Module: Standard Connecting Cable

Note

Incorrect wiring can cause destruction of the optical couplers in the interface.

Connecting cable for point-to-point connection (programmer module)

The following Figure shows the assignments of the connecting cable for point-to-point connections.

- ASCII driver (see Section 12.5)
- Computer link (see Section 12.6)

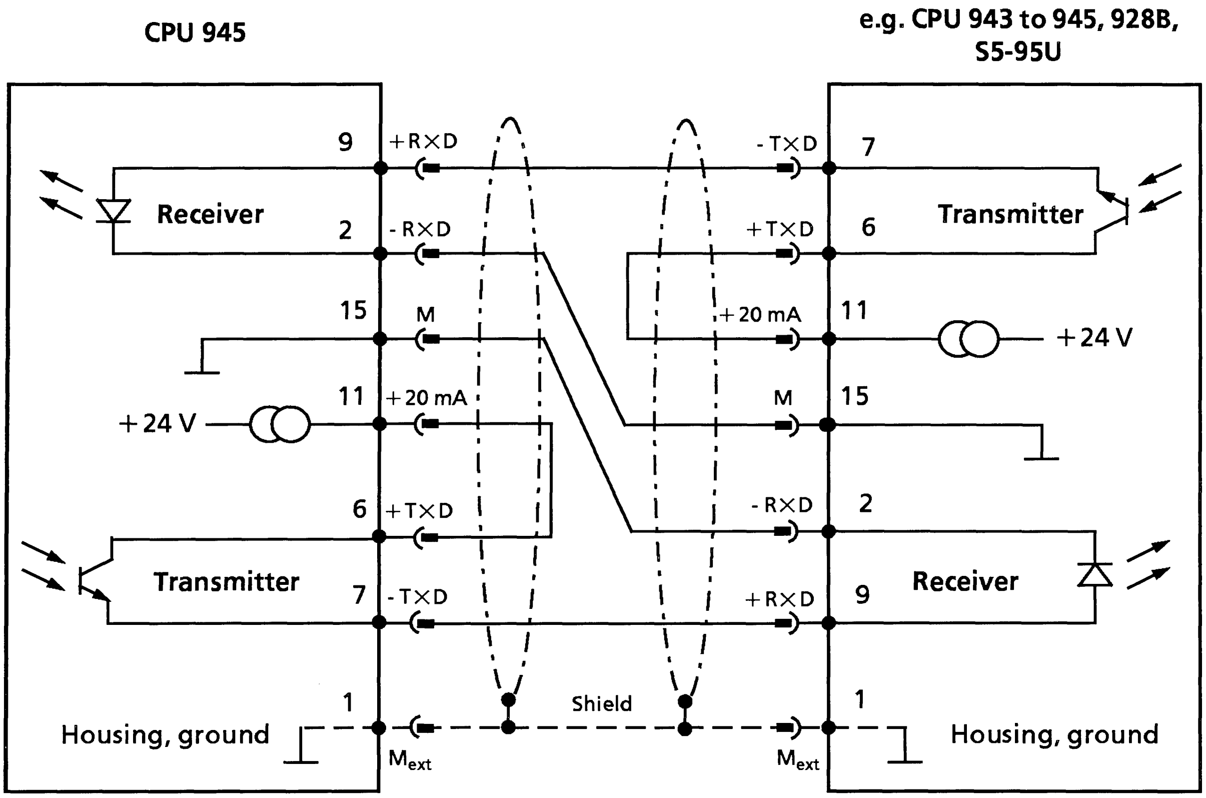


Figure 12.27 Programmer Module: Connecting Cable for Point-To-Point Connection

12.7.2 V.24 Module

The V.24 module can be used for the following types of link:

- Data link with 3964/3964R procedure
- Data link with ASCII driver
- Programmer interface

The following Figure shows the pin assignments of the V.24 interface (send and receive lines):

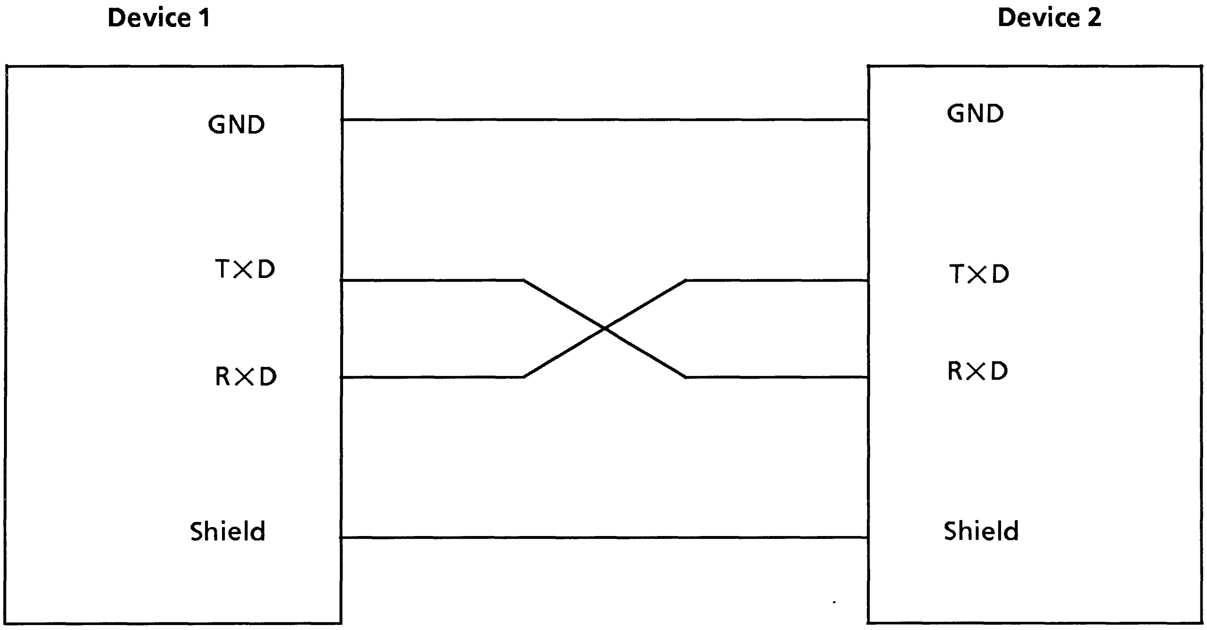


Figure 12.28 Pin Assignments of the V.24 Interface

Besides send and receive lines, the V.24 module incorporates a number of control and signal lines complying with the CCITT recommendation for V.24/V.28. Neither the 3964/3964R standard procedures nor the ASCII driver use these signals.

Voltage ranges for V.24 signals:

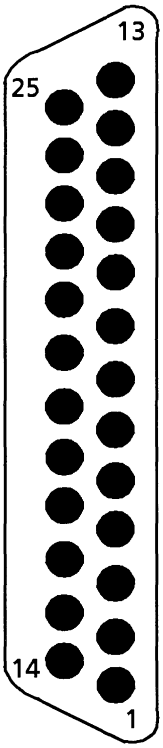
Logic "0" corresponds to a voltage of	$V \geq +3$ V
Logic "1" corresponds to a voltage of	$V \leq -3$ V

If you assemble the connecting cables to suit your own requirements, note that unassigned inputs of the partner must possibly be connected to an open-circuit potential. For detailed information, refer to the relevant manuals and the CCITT recommendations for V.24 or V.28.

Note

For data transmission via the V.24 module, baud rates up to 19,200 baud are permissible.

The following Figure shows the connector pin assignments of the 25-pole Cannon socket in the front plate of the V.24 module:



Pin	Designation to DIN 66020	Designation to CCITT V.24	Internal designation	Input/ Output
1			Shield	
2	D1	103	TXD	Output
3	D2	104	RXD	Input
4*	S2	105	RTS	Output
5*	M2	106	CTS	Input
6*	M1	107	DSR	Input
7	E2	102	GND	
8*	M5	109	DCD	Input
15*	T2	114	TXC	Input
17*	T4	115	RXC	Input
18*	PS3	141		Output
20*	S1.2	108.2	DTR	Output
22*	M3	125	RI	Input
23*	S4	111		Output
24*	T1	113		Output
25*	PM1	142		Input

* Not supported by drivers.

Figure 12.29 V.24 Module: Pin Assignments

The signal numbers are specified in accordance with DIN 66020 (V.24/RS 232C), the signal designations comply with the international standards (RS 232C).

Jumper settings on the V.24 module

The V.24 module is supplied with the jumper settings shown in Figure 12.30. The V.24 module is thus ready for use.

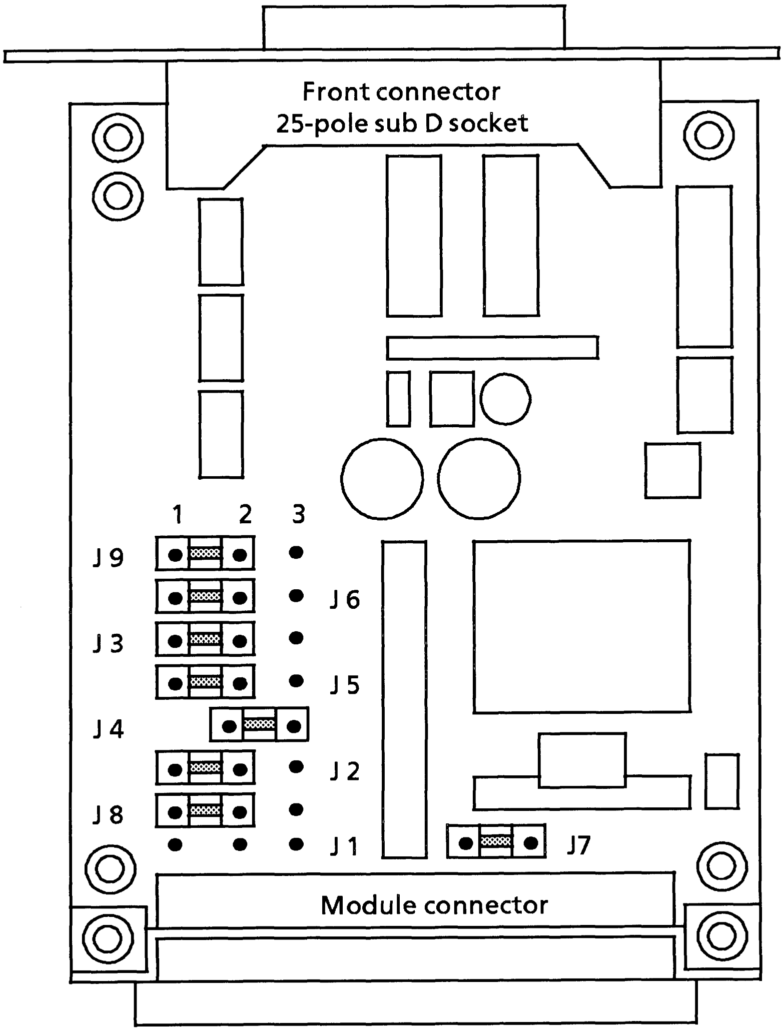
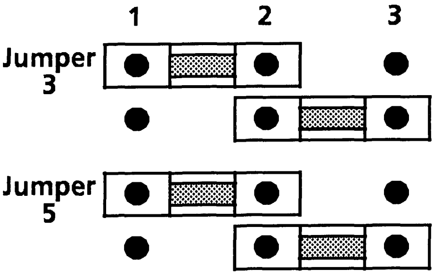


Figure 12.30 V.24 Module: Jumper Settings on Delivery

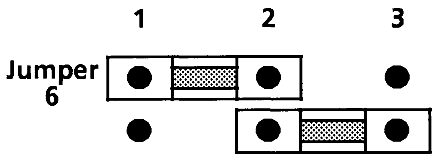
Jumpers 3 and 5 are used for reversing the polarity of the send and receive data.



Send data with normal polarity
Negated send data

Receive data with normal polarity
Negated receive data

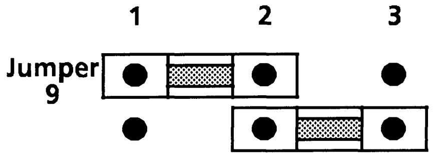
Jumper 6 can be used to switch all V.24 receivers in such a way that they need only the positive voltage range.



All signals received must have a V.24-specific signal level

All signals received may be within the positive signal range

Jumper 9 can be used to connect CTS permanently to an open-circuit potential or to be connected directly to the front connector.



CTS connected to open-circuit potential

CTS connected to pin 5

Standard connecting cables for V.24 module

Standard connecting cables are available at different lengths up to 16 m (52 ft) for linking the V.24 module of the CPU 945 to the partner.

Cable for connecting the CPU 945 and CP 525, CP 524, CPU 945, CPU 928B

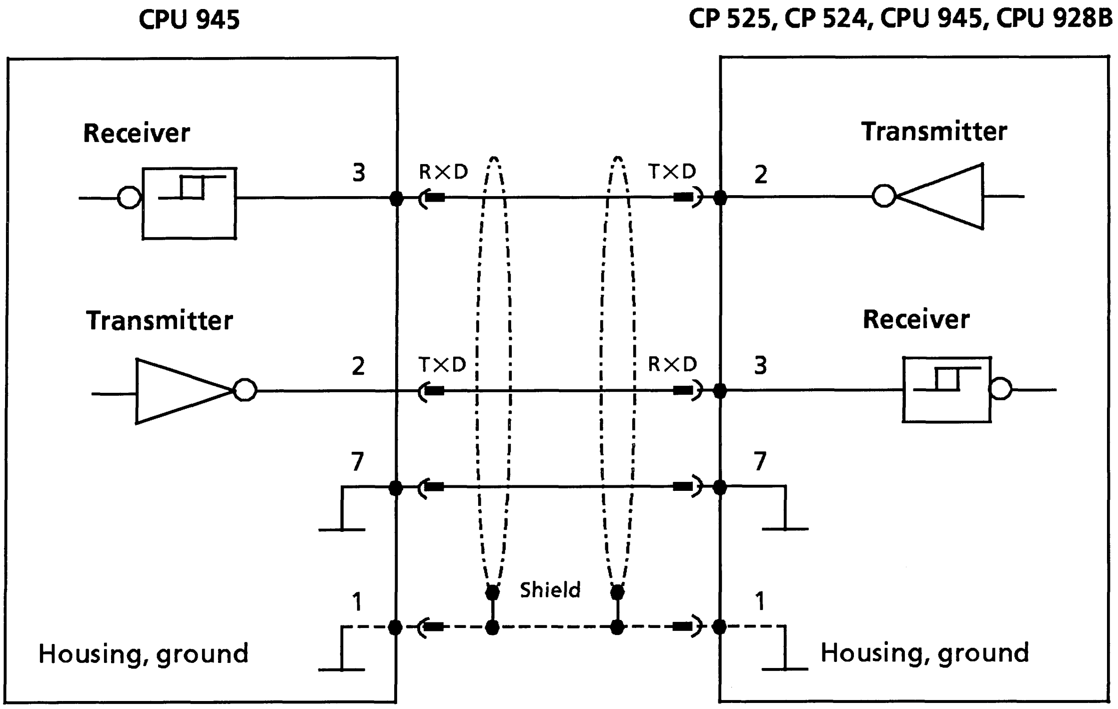


Figure 12.31 V.24 Module: Cable for Connecting the CPU 945 and CP 525, CP 524, CPU 945, CPU 928B

Cable connecting the CPU 945 and DR 210/DR 211

This connecting cable can be used for both the V.24 and the TTY module. Make sure you have selected the same interface type in the CPU 945 and the printer.

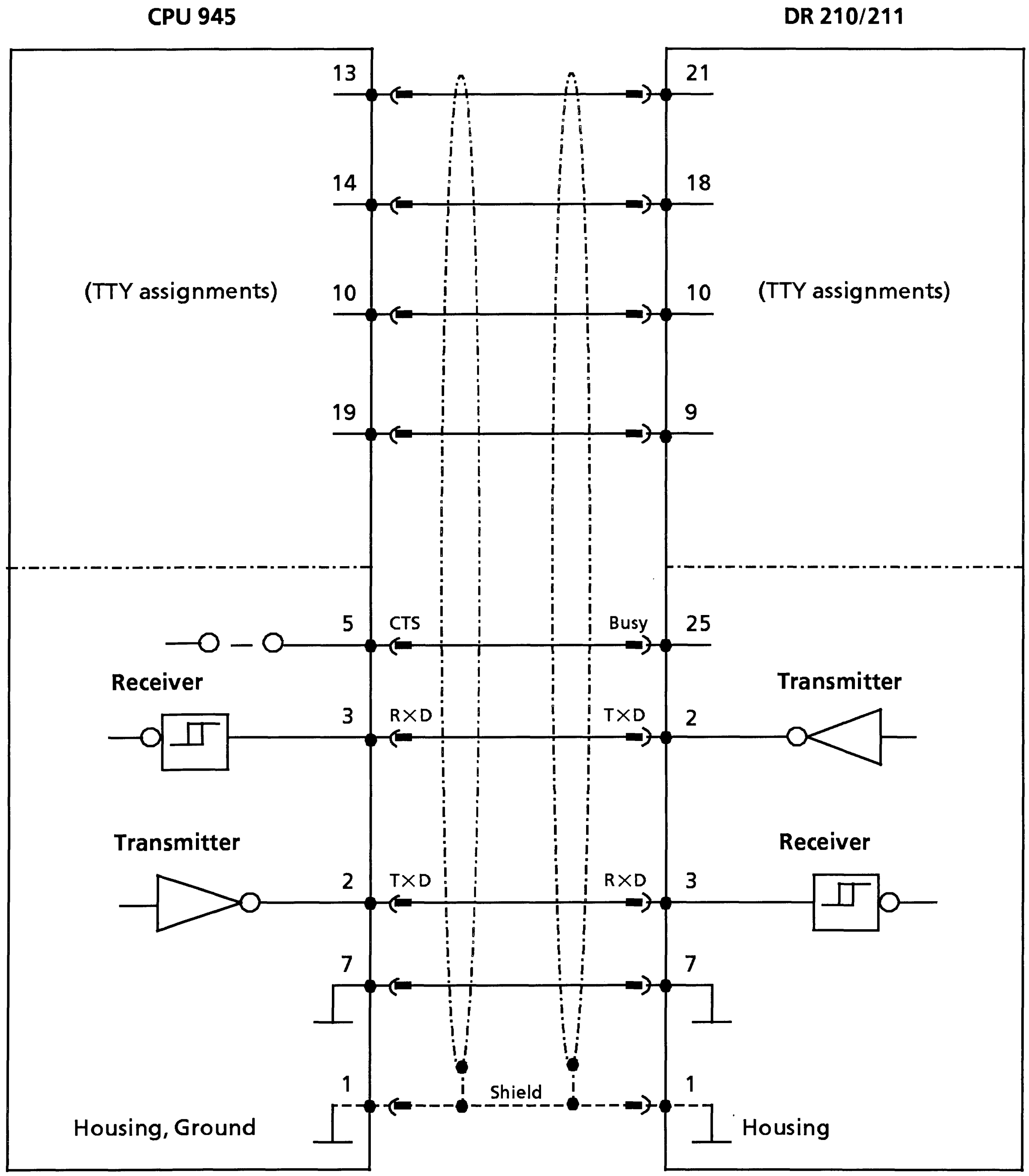


Figure 12.32 V.24 module: Cable Connecting the CPU 945 and the DR 210/211

12.7.3 TTY Module

The TTY module can be used for the following types of link:

- Data link with 3964/3964R procedure
- Data link with ASCII driver
- As programmer interface

The TTY module incorporates a transmitter and a receiver for 20 mA current-loop signals. The Figure below illustrates a typical connection of the current-loop signals.

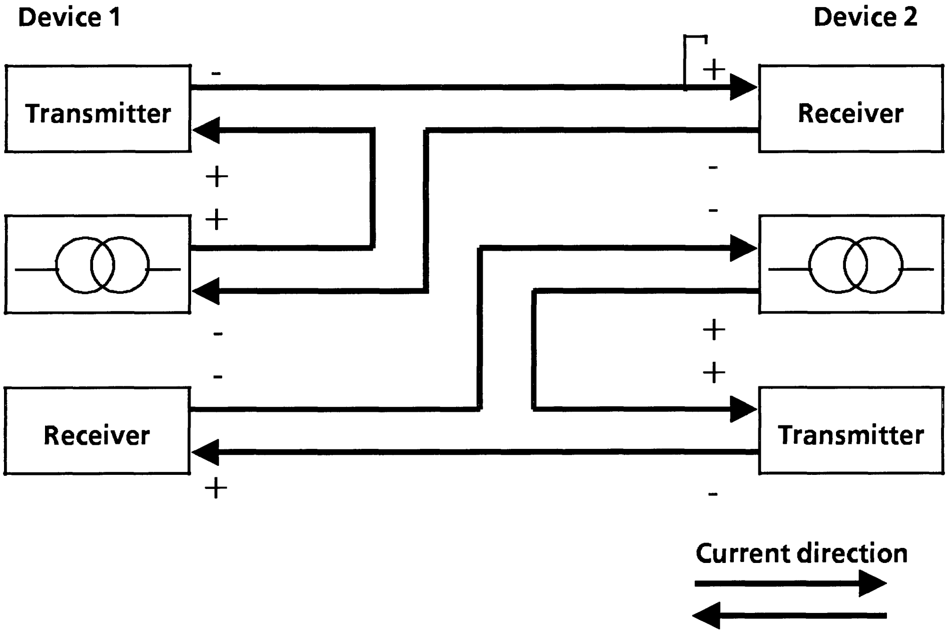


Figure 12.33 TTY Module: Current Direction

The loop current can be fed both from the TTY module and the partner. Only the device that supplies the power must be non-floating.

Caution

For greater cable lengths, the lines should be connected in such a way that the transmitter always supplies the power.

The TTY module supplies the current (20 mA) via jumpers in the connector of the standard connecting cable. The 24 V required for generating the loop current are provided by the power supply unit of the PLC. A current of 20 mA (logic "1") flows in the inactive state of a correct current-loop circuit. An open circuit is indicated by a logic "0".

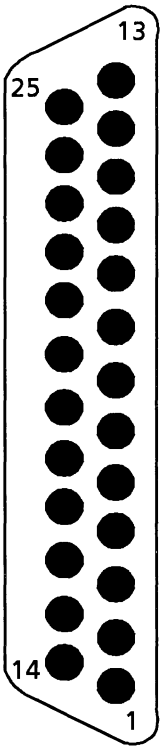
Meanings of the TTY signals:	Logic "0": No current
	Logic "1": Current (20 mA)

Note

The maximum permissible transmission rate for a data link with the TTY module is 9600 baud.

The TTY module complies with DIN 66 258, Part 1.

Figure 12.34 shows the pin assignments of the 25-pole Cannon socket in the front plate of the TTY module:



Pin	Designation	Comment
1	M_{ext}	
9	24 V external	Jumper 3 is used to switch this connection from internal 24 V to external 24 V (see next page)
10	+TXD	
12	+20 mA	Current source
13	+RXD	
14	- RXD	
16	+20 mA	Current source
19	- TXD	
21	M 24	
24	M 24	

Figure 12.34 TTY Module: Pin Assignments

Jumper settings on the TTY module

The TTY module is supplied with the jumper settings shown in Figure 12.35. In this way, the TTY module is ready for use.

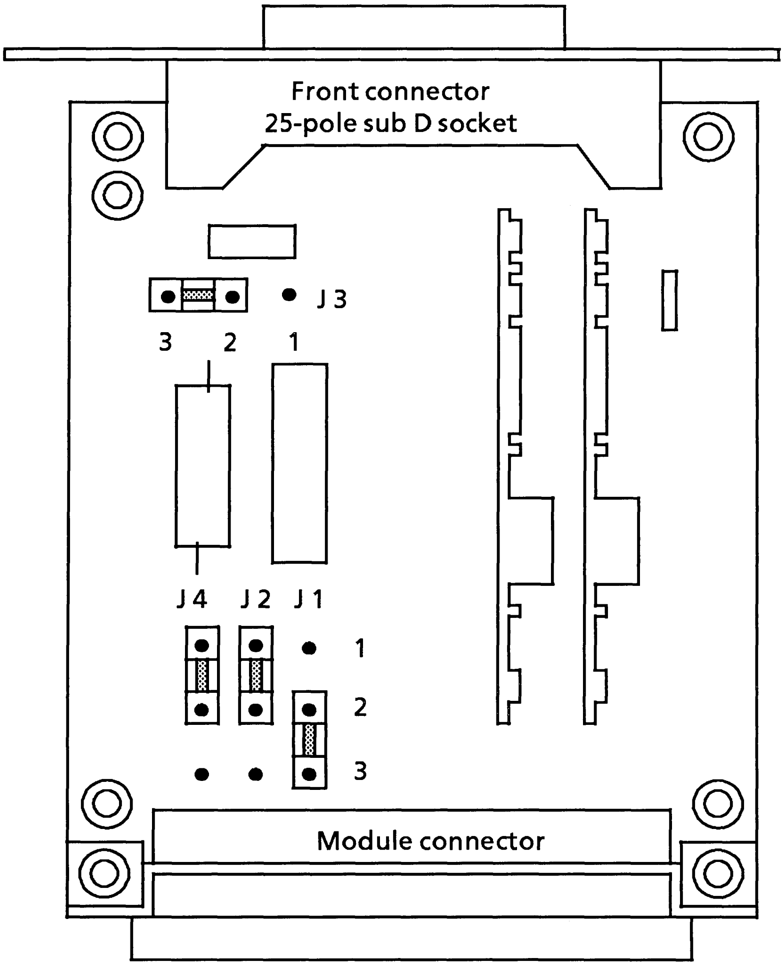
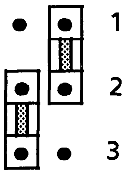


Figure 12.35 TTY Module: Jumper Settings on Delivery

Jumpers 1 and 2 are used for reversing the polarity of the send and receive data:

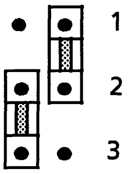
Jumper 1



Negated send data

Send data with normal polarity

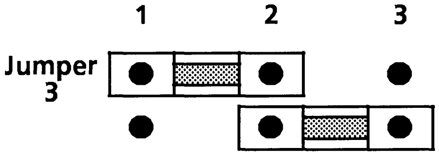
Jumper 2



Receive data with normal polarity

Negated receive data

Jumper 3 can be used to route the 24 V source voltage for generating the loop current:



24 V are connected to pin 9 of the sub D socket in the front plate

24 V are supplied via the module connector (internal)

Standard connecting cables for the TTY module

Standard connecting cables are available in various lengths up to 1000 m (3300 ft) for linking the TTY module in the CPU 945 with the partner.

Cable connecting the CPU 945 and CP 524, CP 525, CPU 945, CPU 928B

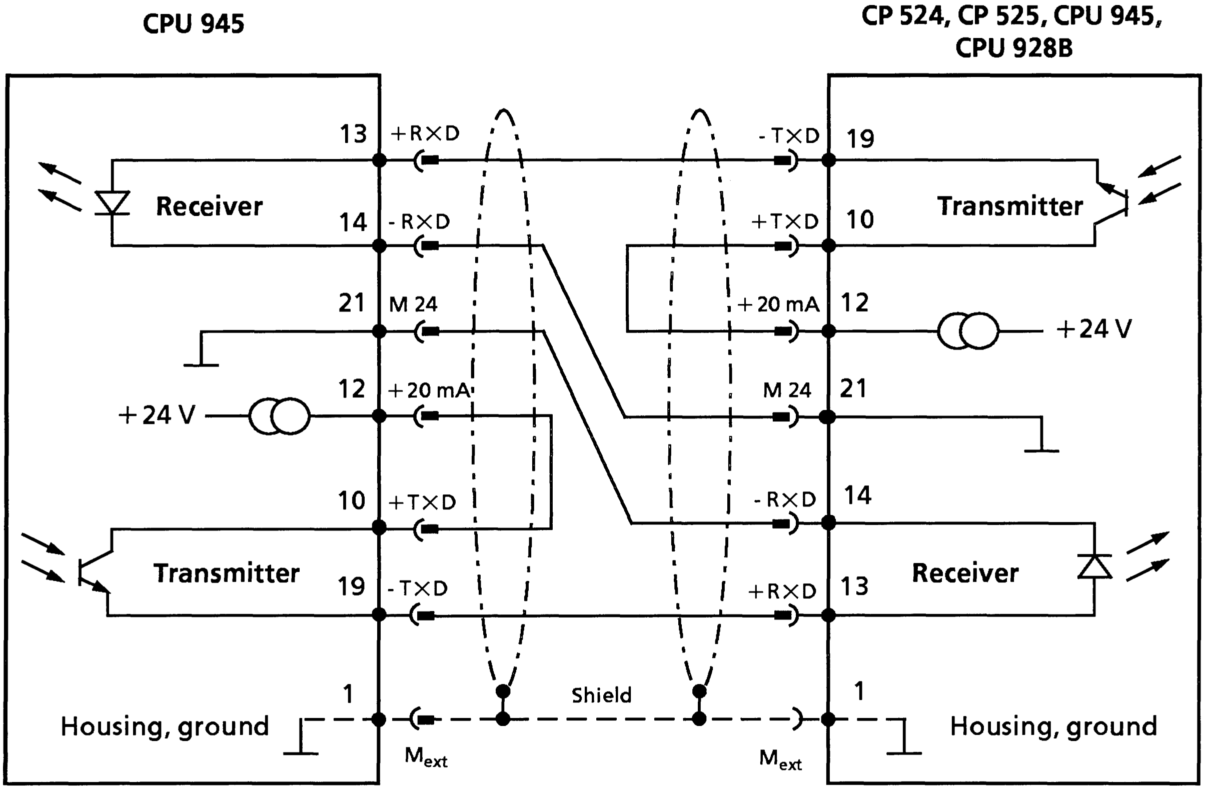


Figure 12.36 TTY Module: Cable Connecting the CPU 945 and CP 524, CP 525, CPU 945, CPU 928B

Note

Incorrect wiring can cause destruction of the optical couplers in the interface.

Cable connecting the CPU 945 and the DR 210/DR 211

This connecting cable can be used for both the TTY and the V.24 module. Make sure you have selected the same interface types in the CPU 945 and the printer.

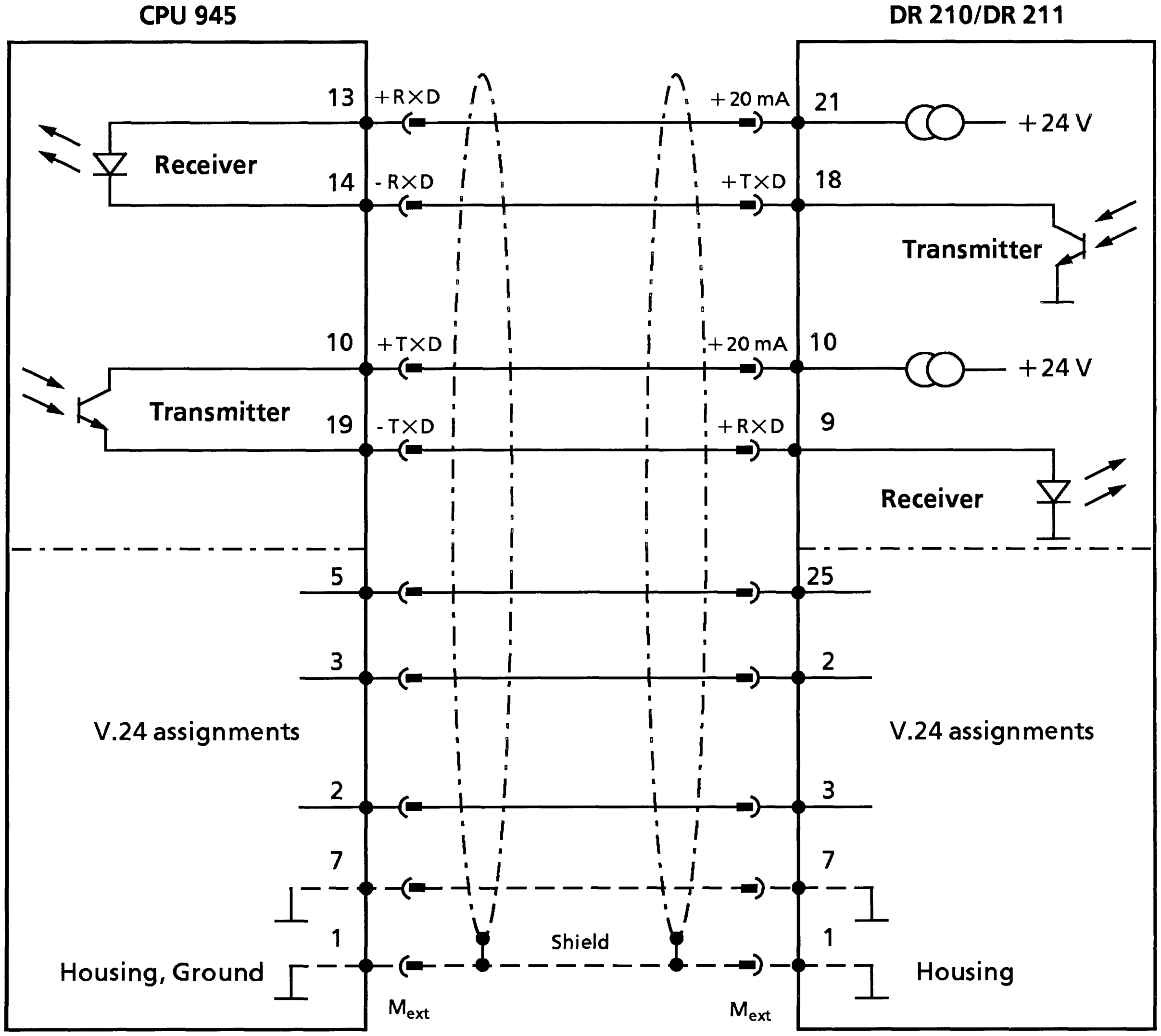


Figure 12.37 TTY Module: Cable Connecting the CPU 945 and the DR 210/211

Note

Incorrect wiring can cause destruction of the optical couplers in the interface.

12.7.4 RS 422-A/485 Module

The RS 422-A/485 module can only be used in the RS 422-A mode

- in a 3964(R) link,
- with the ASCII driver,
- as programmer interface.

RS422-A mode means that the module incorporates the necessary hardware components for full-duplex mode according to the EIA standard RS422-A (CCITT recommendation V.11).

The following Figure shows the pin assignments of the RS422-A/485 interface (send and receive lines):

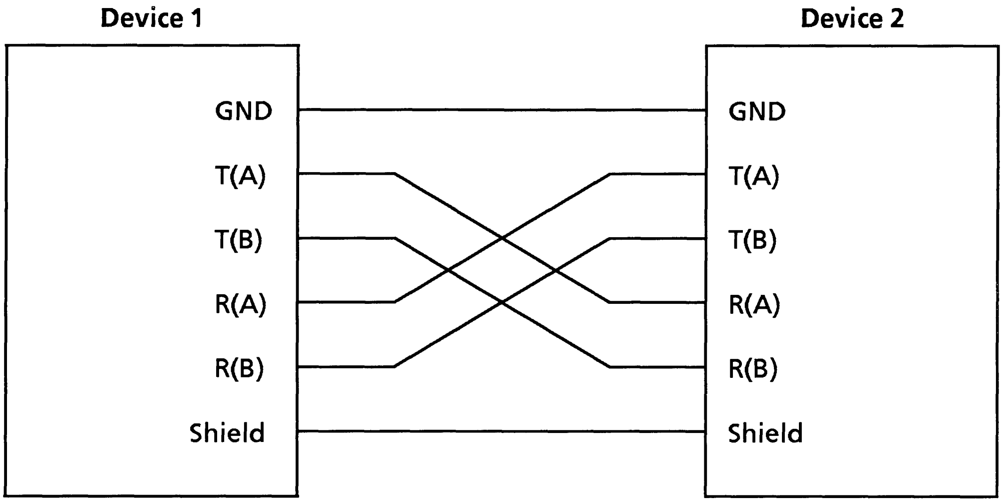


Figure 12.38 Pin Assignments of the RS422-A/485 Interface

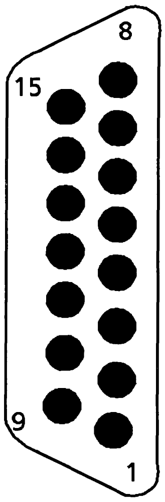
Besides the send and receive lines, the RS422-A/485 module has a number of control and signal lines conforming to the CCITT recommendation X.24 and ISO 8481. However, the above-mentioned links neither require nor use these signals. Therefore the relevant pins need not be assigned. The RS422-A/485 interface operates on the differential voltage principle and therefore has a higher degree of interference immunity than a TTY or V.24 interface.

The following applies to the signals complying with the EIA standard RS 422-A (V.11):

Logic "0" (ON) corresponds to: $V_A > V_B$
Logic "1" (OFF) corresponds to: $V_A < V_B$

In the RS422-A/485 module, the interface signals are electrically isolated from the supply voltage of the PLC.

The following Figure shows the pin assignments of the 15-pole Cannon socket in the front plate of the RS422-A/485 module:



Pin	Designa-tion	Input/ Output	Comment
1	Shield		
2	T(A)	Output	
3	C(A)	Output	
4	R(A)	Input/ output	In full-duplex mode, this two-wire line is reserved for receiving data.
5	I(A)	Input	
6	S(A)	Input	
7	B(A)	Output	
8	GND		
9	T(B)	Output	
10	C(B)	Output	
11	R(B)	Input/ output	In full-duplex mode, this two-wire line is reserved for receiving data.
12	I(B)	Input	
13	S(B)	Input	
14	B(B)	Output	
15	X(B)	Input	

Figure 12.39 RS422-A/485 Module: Pin Assignments

Jumper settings on the RS422-A/485 module

These jumpers are inserted as shown in Figure 12.40 when the module is supplied. In this way the RS422-A/485 module is ready for immediate use.

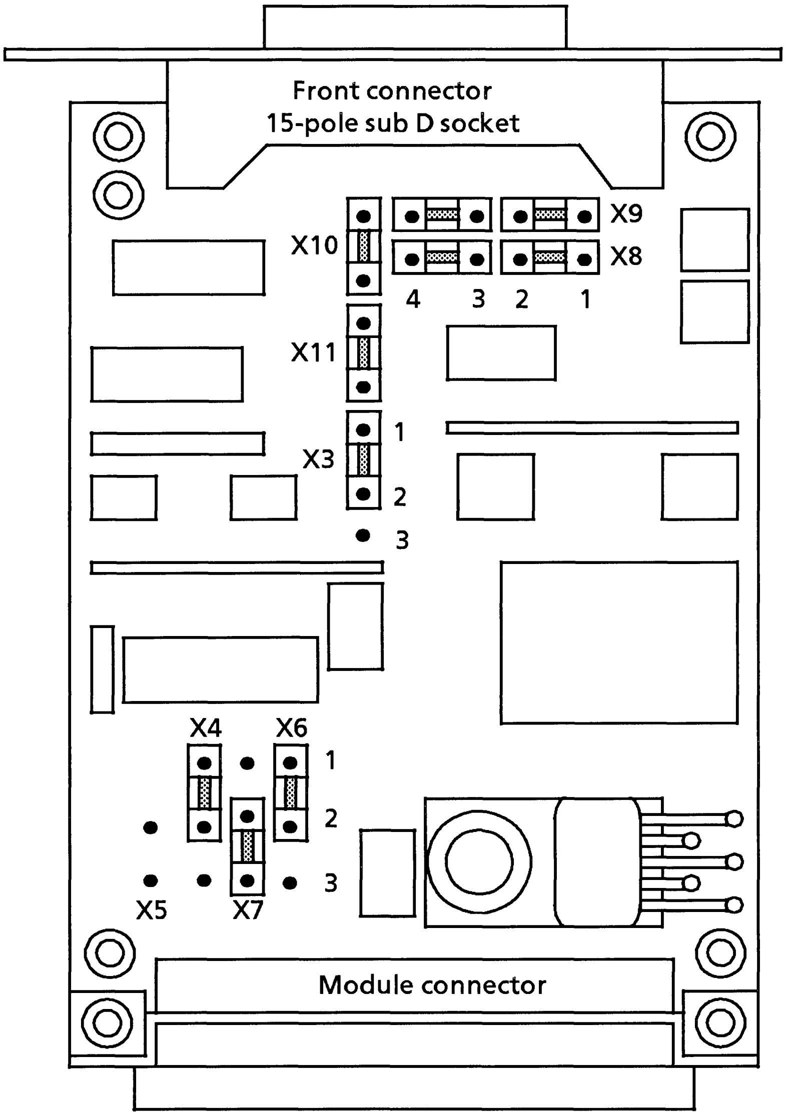
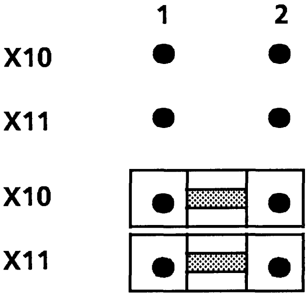


Figure 12.40 RS422-A/485 Module: Jumper Settings on Delivery

Jumpers X10 and X11 can be used to change the default for the two-wire line R, which enables the detection of a wire break.



No presetting for two-wire line R;
the break cannot be detected definitely

Presetting for detecting a wire break
on the two-wire line R.

All other jumper settings must not be changed.

Standard connecting cables for RS422-A/485 module

Standard connecting cables with lengths of up to 1200 m (3900 ft) are available for linking the RS422-A/485 module in the **CPU 945** and the partner.

Standard cables connecting the CPU 945 and CP 524, CPU 945 and CPU 928B

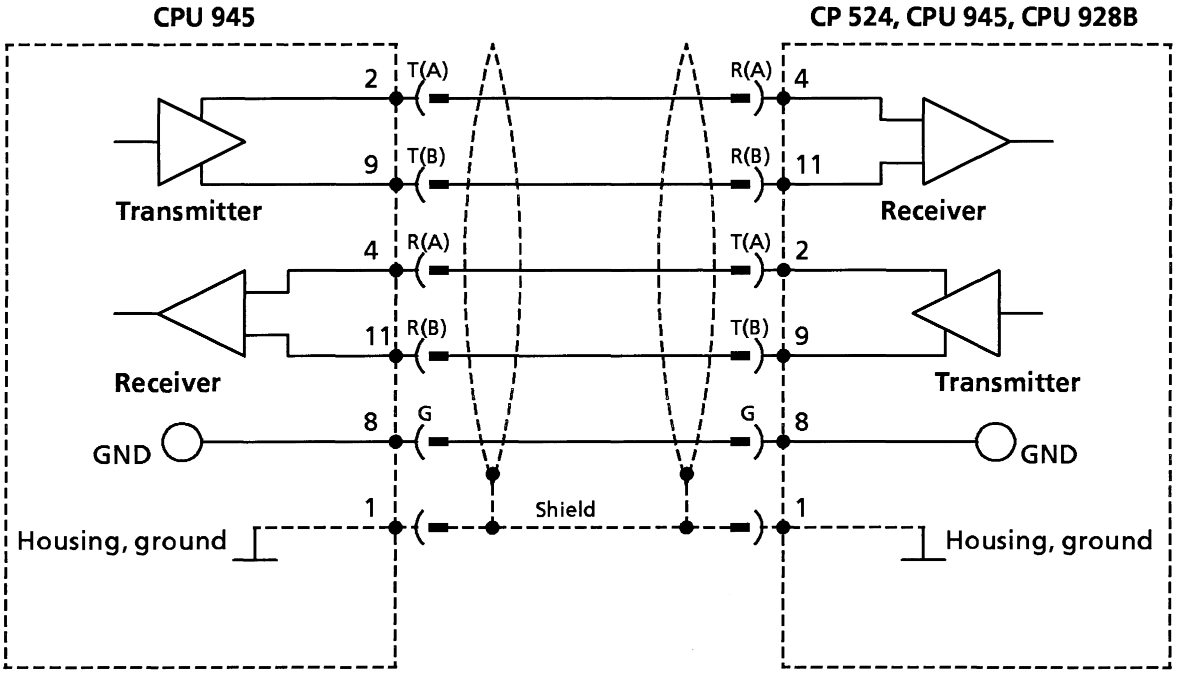


Figure 12.41 RS422-A/485 Module: Cable Connecting the CPU 945 and CP 524, CPU 945, CPU 928B

12.7.5 SINEC L1 Module

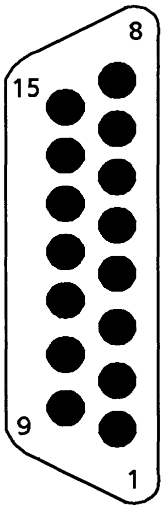
The SINEC L1 module enables the connection of the CPU 945 to the SINEC L1 LAN via its second interface.

Other links permitted by the SINEC L1 module:

- Connection of a programmer
- Point-to-point connection (with SINEC L1 protocol)
- ASCII driver
- Computer link

The maximum permissible transmission rate is 9600 baud.

The Figure below shows the connector pin assignments of the 15-pole Cannon connector in the front plate of the SINEC L1 module (it corresponds to the assignments of SI1):



Pin	Designation	Comment
1	M_{ext}	
2	- RXD	
3	+ 5.2 V	
4	+ 24 V	
5	M	
6	+TXD	
7	- TXD	
8	M_{ext}	
9	+RXD	
10	M 24 V	
11	+20 mA	Current source
12	M	
13	+20 mA	Current source
14	+ 5.2 V	
15	M	

Figure 12.42 SINEC L1 Module: Pin Assignments

Jumper settings on the SINEC L1 module

The SINEC L1 module is supplied with the jumper settings shown in Figure 12.43. In this way, the SINEC L1 module is ready for use.

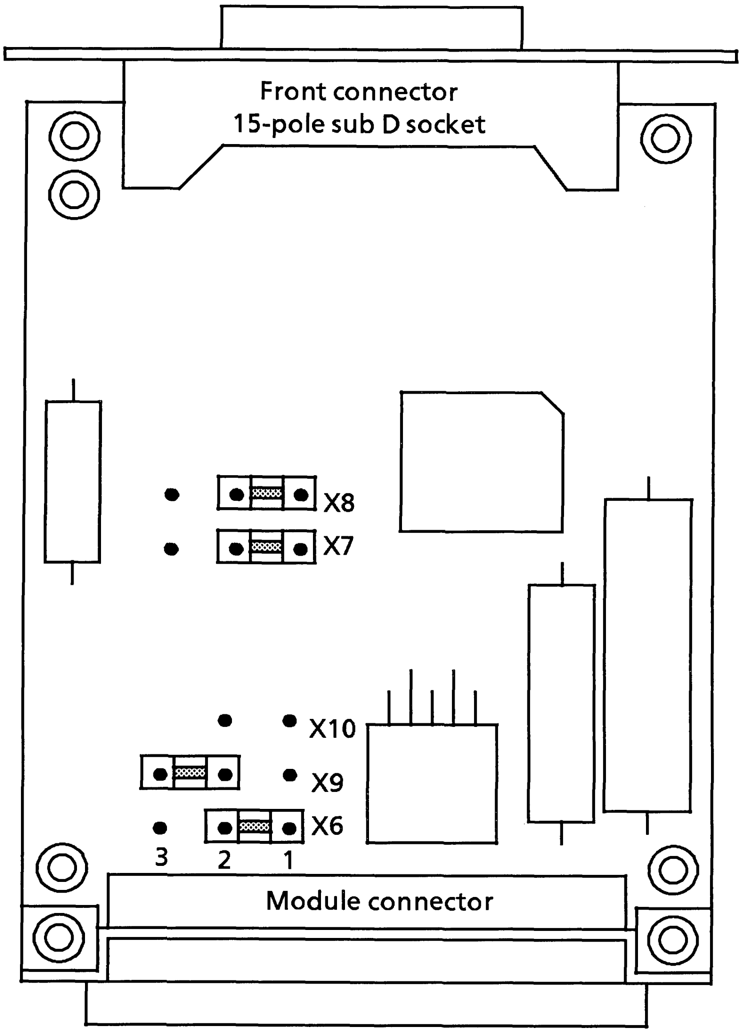


Figure 12.43 SINEC L1 Module: Jumper Settings

Connecting cable for point-to-point connection (SINEC L1 module)

The following Figure shows the connecting cable for point-to-point links.

- Point-to-point connection (see Section 12.4)
- ASCII driver (see Section 12.5)
- Computer link (see Section 12.6)

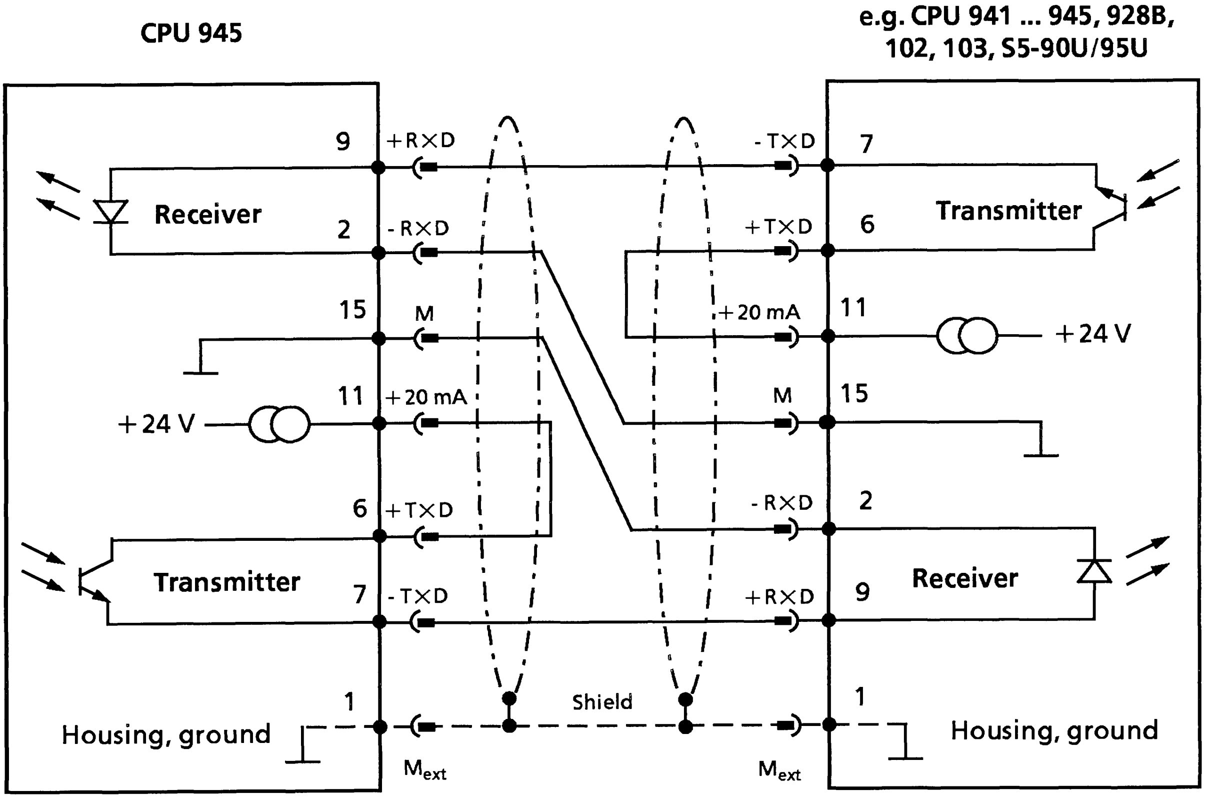


Figure 12.44 SINEC L1 Module: Connecting Cable for Point-To-Point Connection

Use of programmer interface

If you connect a programmer to the SINEC L1 module, the same requirements and restrictions as for the programmer module are valid (see Section 12.7.1).

12.7.6 Technical Specifications of the Interface Modules

Table 12.36 Technical Specifications of the Interface Modules

Degree of protection **Ambient temperature** **Relative humidity** **Altitude** **Supply voltage**	as for CPU 945 (see Section 15)	
Transmission rate		
Programmer module	Fixed 9 600 bit/s	
V.24 module	max. 19 200 bit/s	
TTY module	max. 9 600 bit/s	
RS422-A/485 module	max. 19 200 bit/s	
SINEC L1 module	max. 9 600 bit/s	
Front socket connector		
Programmer module	15-pin Cannon	
RS422-A/485 module	15-pin Cannon	
V.24 module	25-pin Cannon	
TTY module	25-pin Cannon	
SINEC L1 module	15-pin Cannon	
Connecting cable	Shielded four-wire cable (five-wire cable for RS422-A) with braided screen and metal connector housing; must be earthed at both ends)	
Cable lengths		
Programmer module	max. 1 000 m (3300 ft)	
V.24 module	max. 16 m (52 ft)	
TTY module	max. 1 000 m (3300 ft)	
RS422-A/485 module	max. 1 200 m (3900 ft)	
SINEC L1 module	max. 1 000 m (3300 ft)	
Current consumption at 5 V/ 24 V	**5 V**	**24 V**
Programmer module	max. 40 mA	380 µA
V.24 module	max. 200 mA	-
TTY module	max. 100 mA	60 mA
RS422-A/485 module	max. 500 mA	-
SINEC L1 module	max. 155 mA	70 mA
Dimensions (W×H×D)	16.1 mm × 68.4 mm × 102.7 mm 0.633 in × 2.692 in × 4.043 in	
Weight of one interface module	approx. 0.1 kg	

13 Real-Time Clock

Figures

Tables

13 Real-Time Clock

The real-time clock offers the following methods of controlling the process:

- Alarm clock function
  e.g. for monitoring the duration of a process

- Operating hours counter
  e.g. for monitoring inspection intervals

- Real-time clock function
  e.g. for establishing the time at which the CPU stopped in the event of a fault

The clock has an accuracy of ±2 seconds per day at a temperature of 15 °C. This accuracy changes with temperature according to the following formula:
Temperature dependency (T_{amb} in °C): Δt in ms/day $= \pm 2$s/day - $3.5 \cdot (T_{amb} - 15)^2$ ms/day

Example: Tolerance at 40 ° C: ± 2 s/day - $3.5 \cdot (40 - 15)^2$ ms/day → ca. 0 to 4 s/day.

13.1 Parameterizing the Real-Time Clock

The real-time clock cannot be used unless you initialize a clock data area and a status word.

The parameters can be assigned either

- in DB1 (→ see Capter 11)
  or
- in system data 8 to 10.

The following information must be parameterized in system data 8 to 10:

- Location of the clock data area
- Location of the status word

Proceed as follows for parameterizing the clock:

Initialization of the real-time clock in the system data area

The clock is initialized in a function block that you first have to program. In this function block, you can store parameters in the relevant system data by means of transfer operations (e.g. "T RS, TNB").
It is advisable to call the function block in the restart organization blocks OB 21 and OB 22.
The location of the clock data area and the status word is stored in system data words 8 to 10.
In these system data words, you can determine

- whether the area concerned is a flag/S flag area or a data block (DB/DX)
  and
- the exact location within the defined area.

The operating system makes no standard assignments in these system data cells so that clock can only be accessed if the necessary parameters have been entered.

Table 13.1 describes the meanings of the individual bytes of system data words 8 to 10. System data 12 will be explained subsequent to Table 13.1 and the following example.

Table 13.1 System Data Area of the Clock

Absolute RAM Address	System Data Word	Meaning	Permissible Parameters
0E 1010	8	Operand area of the clock data	ASCII characters: D for DB area X for DX area F for flag area S for S flag area
0E 1011		Initial clock data address Operand area D Operand area X Operand area M Operand area S	 DB number (DB0 to DB255) DX number (DX0 to DX 255) Flag byte number High-order part of S flag number
0E 1012	9	Initial clock data address (only relevant for operand areas D, X and S)	Data word number DW 0 to DW 255 For S: Low-order part of S flag number
0E 1013		Operand area of the status word	ASCII characters: D for DB area X for DX area F for flag area S for S flag area
0E 1014	10	Status word address Operand area D Operand area X Operand area M Operand area S	 DB number (DB0 to DB255) DX number (DX0 to DX255) Flag word number High-order part of S flag number
0E 1015		Status word address (only relevant for operand areas D, X and S)	Data word number DW 0 to DW 255 For S: Low-order part of S flag number
0E 1016	11	Startup check of the clock block	Bits 0 and 1 are set when the clock is running
0E 1017			
0E 1018	12	Correction value*	- 400 to 0 to +400
0E 1019			

* Checked and processed only once per hour

Parameterizing the clock

Example: Parameterizing the clock during restart of the PLC (OB21 and OB22)
The clock data are to be stored in DB2 from DW0. The status word is entered in flag word 10.

OB21 STL	Description
:JU FB 101	INITIALIZE CLOCK
NAME :UHR-INIT	
TUDA : KS DB	Clock data area is in DB.
NUDA : KY 2,0	Here: DB2 from DW0
TUSW : KS FW	Status word of clock is an FW
NUSW : KY 10,0	Here: FW10
CORR : KF -12	Correction value
:	
:L KM 00000010 00110000	Preset status word
:T FW 10	(Here ,e.g. operating hours counter
:	enabled, last RUN-STOP change stored,
:	clock time updated during CPU STOP)
:BE	

Programming note:

- TUDA/TUSW: For data area DX → KS XB
  For S flag area → DC SY
- NUDA/NUSW: State the S flag number in the KY format
  (Example: 1,2 for 1 x 256 + 2 x 1 = 258) when using the S flag area

OB21 STL	Description
NAME :UHR-INIT	Initialize clock
DECL :TUDA I/Q/D/B/T/C: D	KM/KH/KY/KS/KF/KT/KC/KG: KS
DECL :NUDA I/Q/D/B/T/C: D	KM/KH/KY/KS/KF/KT/KC/KG: KY
DECL :TUSW I/Q/D/B/T/C: D	KM/KH/KY/KS/KF/KT/KC/KG: KS
DECL :NUSW I/Q/D/B/T/C: D	KM/KH/KY/KS/KF/KT/KC/KG: KY
DECL :CORR I/Q/D/B/T/C: D	KM/KH/KY/KS/KF/KT/KC/KG: KF
:LW =TUDA	Operand area type for
:T FW 250	clock data
:LW =NUDA	Start address of clock data area
:T FW 251	
:LW =TUSW	Operand area type for
:T FW 253	status word
:LW =NUSW	Address of status word
:T FW 254	
:	
:L DH 0002 04FE	Final address of source area (FW254)
:L DH 000E 1014	End address target area (RS10)
:TNW 3	FY 250-255 in RS 8 to 10 transfer.
:LW =KORR	Enter correction value
:T RS 12	
:L KB +0	Delete scratch flag
:T MD 250	
:T FW 254	
:BE	

DB2 STL	Description
0: KH = 0003;	--,WEEKDAY //Current time
1: KH = 1402;	DAY, MONTH
2: KH = 8908;	YEAR, HOUR + AM/PM bit
3: KH = 0000;	MINUTE, SECOND
4: KH = 0102;	LEAP YEAR, WEEKDAY//Setting clock
5: KH = 0504;	DAY, MONTH
6: KH = 9308;	YEAR, HOUR + AM/PM bit
7: KH = 0000;	MINUTE, SECOND
8: KH = 0002;	--,WEEKDAY //Prompt time (setting)
9: KH = 0504;	DAY, MONTH
10: KH = 0009;	--,HOUR + AM/PM bit
11: KH = 0000;	MINUTE,SECOND
12: KH = 0000;	--,SECONDS //Current operating hours
13: KH = 0001;	MINUTES, HOURS
14: KH = 0000;	HOURS X 100, HOURS X 10000
15: KH = 0000;	--,SECONDS //Operating hours setting
16: KH = 0012;	MINUTES, HOURS
17: KH = 0000;	HOURS X 100, HOURS X 10000
18: KH = 0000;	--,WEEKDAY // Clock after STP/RUN
19: KH = 0000;	DAY,MONTH
20: KH = 0000;	YEAR,HOUR
21: KH = 0000;	MINUTE,SECOND
22:	

Correction Value

To compensate for clock inaccuracy due to the effect of temperature, you can enter a correction value in system data word (RS) 12 (E 1018_H).
The correction value (in seconds) is based on an operating time of 30 days, i.e. if you see that the clock of the CPU has lost, say, 20 seconds in 30 days, the correction value is +20.
Internally, the operating system corrects the clock every hour by a value smaller than one second. This ensures that the clock does not "jump" a second (the correction value is read and checked only once per hour). This compensation is unaffected by the mode selected, i.e. it functions in both STOP and RUN mode.
Correction value range:
- 400 to 0 to +400 (no correction at "0").
You must specify the correction value in "KF" format.
Following an OVERALL RESET, the default value "0" is in RS 12.

If a nonvalid correction value is used, the operating system sets bit No.15 in RS 11; in this case, the correction value is "0".

The time is not corrected on POWER OFF. On POWER-UP, the time correction is updated if the CPU had battery backup during this time.

13.2 Structure of the Clock Data Area

The location of the clock data area must be stored in system data words 8 and 9. Data is always exchanged between the control program and the integrated clock via the clock data area. The integral clock stores current values of clock time, date and operating hours counter in the clock data area (flag/S flag area or data block (DB/DX)) and, in this same clock data area, the control program stores settings for prompting times and operating hours counters. The control program can only read or write to the clock data area, but can never access the clock direct. Figure 13.1 illustrates this relationship.

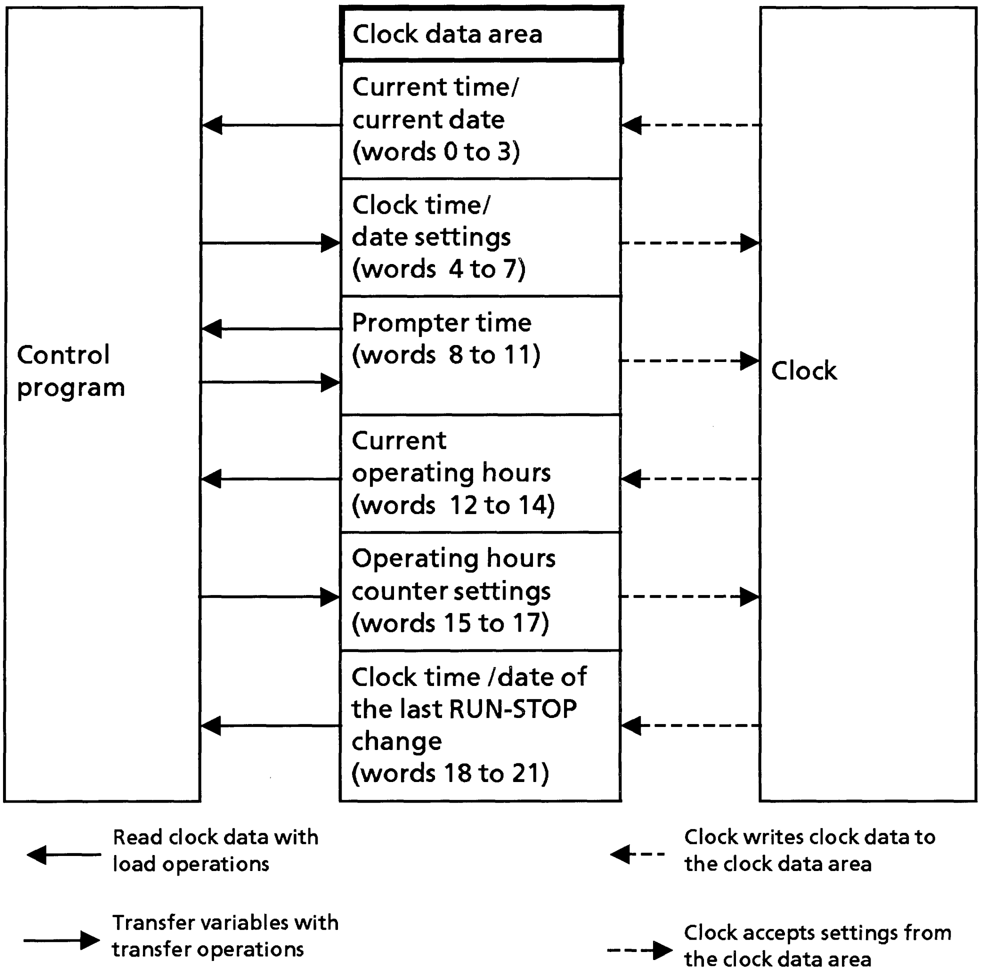

Control program
Clock data area
Current time/ current date (words 0 to 3)
Clock time/ date settings (words 4 to 7)
Prompter time (words 8 to 11)
Current operating hours (words 12 to 14)
Operating hours counter settings (words 15 to 17)
Clock time /date of the last RUN-STOP change (words 18 to 21)
Clock
Read clock data with load operations
Transfer variables with transfer operations
Clock writes clock data to the clock data area
Clock accepts settings from the clock data area

Figure 13.1 Control Program and Clock Access to the Clock Data Area

When setting the clock, you need only transfer the data required for implementing the function in question. For example, if you only want to change the data for the clock function, you need not specify the data for the prompter function or for the operating hours counter.

Table 13.2 gives information on the location of certain clock data within the clock data area, regardless of the memory area selected (DB/DX area or flag/S flag area). You will find explanations of the entries in the clock data area following Table 13.2.

Table 13.2 Clock Data in the Clock Data Area

Clock Data Area Word Number	Meaning	Left Word	Right Word
0	Current clock time/ current date	---	Weekday
1		Date	Month
2		Year	AM/PM (Bit No.7), Hour
3		Minute	Second
4	Time/ Date settings	Leap year	Weekday
5		Date	Month
6		Year	AM/PM (Bit No.7) , Hour
7		Minute	Second
8	Prompt time	---	Weekday
9		Date	Month
10		---	AM/PM (Bit No.7) , Hour
11		Minute	Second
12	Current operating hours	-	Seconds
13		Minutes	Hours
14		Hours · 100	Hours · 10,000
15	Operating hours counter settings	-	Seconds
16		Minutes	Hours
17		Hours · 100	Hours · 10,000
18	Clock time/ date after a RUN/STOP change or POWER OFF (only if bit No. 5 in the status word = 1)	---	Weekday
19		Date	Month
20		Year	AM/PM (Bit No.7) , Hour
21		Minute	Second

Please note the following:

- Entries in the clock data area must be in BCD.

- By changing bit No. 1 in the status word, you can select the 12-hour or 24-hour mode for the clock (see Section 13.3)
  The AM/PM flag (0 = AM; 1 = PM) is only of significance if the hardware clock is operating in 12-hour mode. It corresponds to bit 7 of the following words:
  - Word 2
  - Word 6
  - Word 10
  - Word 20.

  In this mode, the hours and the AM/PM flag cannot be set independently of each other when specifying the settings of the clock and the prompting time.

  If an AM/PM flag is set in 24-hour mode, this is recognized when the settings for the clock and prompting time are entered and the relevant error bit is set.

- Settings must lie within the definition ranges given in Table 13.4:

Table 13.3 Clock Data Definition Areas

Variable	Permissible Parameters	Variable	Permissible Parameters
Seconds	0 to 59	Day	1 to 31
Minutes	0 to 59	Month	1 to 12
Hours	in 24-hour mode: 0 to 23 in 12-hour mode: for AM 1 to 12 (12≙00.00 hours) for PM 81 to 92 (92≙noon if AM/PM bit set) 0 to 99 if operating hour counter specified	Year Leap year	0 to 99 0 to 3 0=Leap year is current year 1=Leap year was last year 2=Leap year was two years ago 3=Leap year was three years ago
Weekday	1 to 7 1=Sunday 2=Monday 3=Tuesday 4=Wednesday 5=Thursday 6=Friday 7=Saturday		

Any other entries will lead to operating system error messages which are flagged in the status word. If the settings are within the definition range, error bits in the status word are reset by the operating system the next time the clock, the prompting time or the operating hours counter is set.

If a setting (prompting time or operating hours counter) is not to be transferred to the clock or if the current value is not to be changed on entry of the setting, enter "FF" (hexadecimal) for this variable.

If the clock data area is located at the end of the individual areas (flags, data block) and if there is insufficient space for the clock data area, only the actual clock data transferred will be accomodated in this area.
Settings outside the range are ignored.

- If the clock data is in the nonretentive flag area or nonretentive S flag area, all settings and the time of the last RUN/STOP change will be lost after POWER OFF or COLD RESTART!

- Please remember that you can define the location of the clock data area and that the word numbers in Table 13.2 are relative.

  - If your data word area is in a data block (DB/DX) and if it does not begin with DW 0 but DW X, you must add the value X to the word number in Table 13.2.

    Example: Your clock data area begins at DW 124. The data for clock time/date are stored from DW 124 to DW 127.

  - If you store the clock data area in the flag area from flag word 0, you must multiply the word number given in Table 13.2 by a factor of 2 in order to obtain the corresponding word address.

    Example: Store the clock data area in the flags operand area from FW 0 onward. The operating hours counter data is stored from address FW 244 onward.

  If your clock data area does not begin at flag word 0, you must add this value.

13.3 Structure of the Status Word

The status word can be scanned to detect, for example, errors in the entry of clock settings, or alternatively, specific bits can be changed in the status word to disable or enable transfer or read operations.
The response of the CPU when changing from RUN to STOP or during POWER OFF can be determined with the bits (flags) reserved for this purpose.

- The status word can be located in the flag area/S flag area or in a data block (DB/DX). The location must be defined in system data words 9 and 10.

- The clock runs independently of the mode set.
  Updating of the clock data area is dependent on the mode set and the states of bits 4 and 5 of the status word. You can set or reset these bits with "S" or "R" operations in the control program.
  When monitoring the program with an operator panel (e.g. the OP 396), it is advantageous to have the CPU update the clock (current date) also in STOP mode.

- The "Transfer settings" bits (bits 2, 10 and 14 in the status word) are reset by the operating system if
  - the settings have been transferred
  - the settings have not been transferred because they were outside the permissible range. In this case, the relevant error is set (bits 0, 8 and 12 of the status word).

- The "Transfer settings" bits (bits 2, 10 and 14 of the status word) are not reset by the operating system if
  - the system data for the clock are incorrect or not available
  - the clock data area is too small

- The bits of the status word are divided into
  - clock flags
  - operating system flags
  - operating hours counter flags
  - prompting time flags.

Tables 13.4 to 13.7 contain information on the meaning of the signal states of the flags.

Clock Flags

Table 13.4 Meaning of the Clock Flags (Bits 0, 1, 2 and 3 of the Status Word)

Bit Number	Signal State	Meaning
0	1	Error when entering settings
	0	No error when entering settings
1	1	12-hour representation (clock mode)
	0	24-hour representation (clock mode)
2	1	Transfer settings
	0	Do not transfer settings
3	1	Clock time can be read
	0	Clock time cannot be read

Operating System Flags

Table 13.5 Meaning of Bits 4 and 5 of the Status Word

Operating Mode	Bit in the Status Word	Signal State	Meaning
STOP	4	1	The clock updates only words 0 to 3 in the clock area (current clock time/ current date). The clock can be set with the "FORCE VAR" programmer function.
		0	The clock does not update the clock data area. Word 0 to 3 contain the time of the last RUN/STOP change.
	5	1	Words 18 to 21 contain the time of the last RUN/STOP change or the time of the last POWER OFF if bit 4 is also set.
		0	Words 18 to 21 are not used.
RUN	4	1/0	The clock updates the clock data area continuously (words 0 to 17).
	5	1	Words 18 to 21 contain the time of the last RUN/STOP change or the time of the last POWER OFF.
		0	Words 18 to 21 are not used.

Operating Hours Counter Flags

Table 13.6 Meaning of the Operating Hours Counter Flags (Bits 8, 9 and 10 of the Status Word)

Bit Number	Signal State	Meaning
8	1	Error when entering settings
	0	No error when entering settings
9	1	Enable operating hours counter
	0	Disable operating hours counter
10	1	Transfer settings
	0	Do not transfer settings

Alarm Clock Flags

Table 13.7 Meaning of the Alarm Clock Flags (Bits 12, 13 and 14 of the Status Word)

Bit Number	Signal State	Meaning
12	1	Error when entering settings
	0	No error when entering settings
13	1	Set prompting time reached
	0	Set prompting time not reached
14	1	Transfer settings
	0	Do not transfer settings

Bits 6, 7, 11 and 15 are required by the operating system, and cannot be used by the user.

13.4 Backup of the Clock

With battery backup, the clock will continue to operate even after "POWER OFF". If the PLC does not have battery backup, the clock will show the settings 01.01.93 12.00.00, Weekday: 6 when the clock is initialized following "POWER ON". The 24-hour mode is set as default. Batteries should therefore only be changed while the power is on, otherwise the clock data will be lost.

13.5 Programming the Clock

Transferring Settings to the Clock

- Settings are stored in the clock data area with Transfer operations (see Table 13.2).
- The AM/PM flag (bit No. 7) is only significant in 12-hour mode.
  Bit 7 = 1 → PM
  Bit 7 = 0 → AM
- Clock data must be transferred in BCD.
- If a setting is not to be transferred, identify the corresponding byte with the number "255_D" or "FF_H". The value of this variable in the clock is then retained when the clock is set.
- Once you have transferred the settings to the clock data area, you must set bit 2 of the status word before the clock can accept the clock data.
- After completing the transfer of clock data to the clock, bit 2 of the status word is reset.
- Incorrect settings are flagged in the status word by setting bit 0. The clock continues to operate with old values.

Example: Transferring new settings (clock time/date) to the clock, using the programmer.

The clock is to be set with the following data: Mo 05.04.93; 12:00:00. The status word is assigned to flag word 10 and the clock data is stored in DB2 from data word 0. The settings for the clock data are transferred:

- With the "FORCE VAR" programmer function if the PLC is in RUN Mode
- With the "FORCE VAR" programmer function if the PLC is in STOP mode and status word bit 4 = 1.

Note

When using the "FORCE VAR" function, you must enter the clock data first and then the status word.

Operand	Signal States	Meaning
DB2		
DW 4	KH = 0102	Leap year was last year and weekday (MO)
DW 5	KH = 0504	Date (05) and month (04)
DW 6	KH = 9312	Year (93) and hour (12)
DW 7	KH = 0000	Minute (00) and second (00)
FW 10	KM = 0000 0000 0001 0100	In "STOP" and "RUN": Bit 4=1: Clock data area is updated during "STOP". Bit 2=1: Transfer settings
	or	
FW 10	KM = 0000 0000 0000 0100	In "RUN" mode only: Bit 4=0: Clock data are not updated during "STOP". Bit 2=1: Transfer settings

Example: Program for setting clock time and date.

Settings for clock time and date are transferred depending on the signal state at input 12.1. These settings must be transferred to flag bytes 120 and 127 before setting input 12.1 (cf. OB1). Values which are not to be changed must be preset with FF$_H$. Clock mode can be defined with input 14.0 (1 = 12-hour mode). Input 13.0 is the AM/PM bit for 12-hour mode.
The clock data area is in DB2 from DW 0, and the status word is FW 10.

OB1 STL	Description
:	================================
:	SETTING CLOCK TIME AND DATE
:	================================
:	LOAD time and date values
:	into FY 120 to FY 127 first!
:A I 12.1	clock setting triggered
:S F 20.0	by setting F 20.0 (reset in FB 10)
:	
:JU FB 10	
NAME :SETCLK	
LPYR : FY 120	LEAP YEAR
WDAY : FY 121	WEEKDAY
DAY : FY 122	DAY
MON : FY 123	MONTH
YEAR : FY 124	YEAR
HOUR : FY 125	HOUR
AMPM : I 13.0	AMPM BIT (ONLY IMPORTANT IN 12-HR MODE)
MIN : FY 126	MINUTES
SEC : FY 127	SECONDS
ERR : F 12.1	ERROR BIT
MODE : I 14.0	12-HR MODE: I 14.0 = 1
:BE	

FB10 STL	Description
NAME :SETCLK	SETTING THE CLOCK
DECL :LPYR I/Q/D/B/T/C: I BI/BY/W/D: BY	
DECL :WDAY I/Q/D/B/T/C: I BI/BY/W/D: BY	
DECL :DAY I/Q/D/B/T/C: I BI/BY/W/D: BY	
DECL :MON I/Q/D/B/T/C: I BI/BY/W/D: BY	
DECL :YEAR I/Q/D/B/T/C: I BI/BY/W/D: BY	
DECL :HOUR I/Q/D/B/T/C: I BI/BY/W/D: BY	
DECL :AMPM I/Q/D/B/T/C: I BI/BY/W/D: BI	
DECL :MIN I/Q/D/B/T/C: I BI/BY/W/D: BY	
DECL :SEC I/Q/D/B/T/C: I BI/BY/W/D: BY	
DECL :ERR I/Q/D/B/T/C: Q BI/BY/W/D: BI	
DECL :MODE I/Q/D/B/T/C: I BI/BY/W/D: BI	
:A =MODE	24-HR mode = 0, 12-HR mode = 1
:= F 11.1	(clock mode status word bit 1)
:AN F 20.0	Flag is reset if settings already
:JC =M001	read into clock data area
:R F 20.0	
:	
:C DB 2	Clock data area
:L =LPYR	Store value for LEAP YEAR
:T DL 4	
:L =WDAY	Store value for WEEKDAY
:T DR 4	
:L =DAY	Store value for DAY
:T DL 5	
:L =MON	Store value for MONTH
:T DR 5	
:L =YEAR	Store value for YEAR
:T DL 6	
:L =HOUR	Store value for hour
:ON =AMPM	IF 12-HR mode is set, and
:ON =MODE	AMPM bit = 1 (afternoon), the
:JC =MORN	relevant bit in the clock area
:L KH 0080	is set
:OW	
MORN :T DR 6	
:L =MIN	Store value for MINUTES
:T DL 7	
:L =SEC	Store value for SECONDS
:T DR 7	
:AN F 11.2	Transfer settings
:S F 11.2	(Status word is FW10)
:L KT 020.1	Start monitoring time
:SV T 10	
M001 :A T 10	BEC, if monitoring time
:BEC	not yet elapsed

FB10 STL (Continued)	Description
:AN F 11.2	Have settings been Transferred?
:JC =M002	if YES, jump to M002
:AN F 11.0	Error when entering settings?
:RB =ERR	Reset error bit if NO
:BEC	BEC if no error
M002 :S =ERR	Set error bit if error
:BE	

Reading the Current Time/Date

Current data is stored in the first four data words of the clock data area. This data can be read out from there with Load operations.

To be able to read a correct time, bit 3 of the status word must be set in the control program before the read access. The clock data area is no longer updated when bit 3 is set. You must reset this bit after reading the clock.

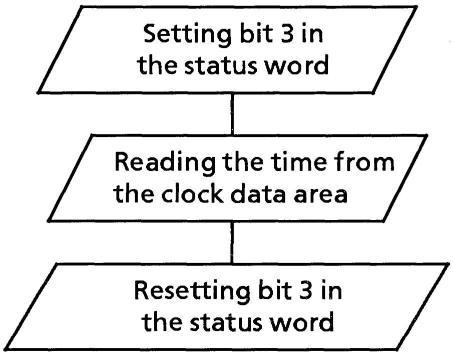

Setting bit 3 in the status word
Reading the time from the clock data area
Resetting bit 3 in the status word

Figure 13.2 Procedure for Reading the Current Date/Time

Example: Reading the time and the date.

The time is stored in flag bytes 30 to 36 depending on an external event, simulated here by a positive edge at input 12.0. Flag 13.1 indicates which mode the clock is operating in. Flag 13.0 is the AM/PM bit in 12-hour mode.
The clock data area is in DB2 from DW 0 onwards, and the status word is FW 10.

OB1 STL	Description
:	================================
:	READING TIME AND DATE
:	================================
:A I 12.0	Time and date are to be stored
:AN F 0.1	in FY 30 to FY 36 in the case of
:	positive edge at I 12.0
:	(external event).
:JC FB 13	
NAME :READCL	
WDAY : FY 30	WEEKDAY
DAY : FY 31	DAY
MON : FY 32	MONTH
YEAR : FY 33	YEAR
HOUR : FY 34	HOUR
AMPM : F 13.0	F13.0=1, afternoon in 12-HR mode
MIN : FY 35	MINUTES
SEC : FY 36	SECONDS
MODE : F 13.1	F13.1=1, in 12-HR mode
:A I 12.0	
:= F 0.1	
:BE	

FB13 STL	Description
NAME :READCL	READING THE CLOCK
DECL :WDAY I/Q/D/B/T/C: Q BI/BY/W/D: BY	
DECL :DAY I/Q/D/B/T/C: Q BI/BY/W/D: BY	
DECL :MON I/Q/D/B/T/C: Q BI/BY/W/D: BY	
DECL :YEAR I/Q/D/B/T/C: Q BI/BY/W/D: BY	
DECL :HOUR I/Q/D/B/T/C: Q BI/BY/W/D: BY	
DECL :AMPM I/Q/D/B/T/C: Q BI/BY/W/D: BI	
DECL :MIN I/Q/D/B/T/C: Q BI/BY/W/D: BY	
DECL :SEC I/Q/D/B/T/C: Q BI/BY/W/D: BY	
DECL :MODE I/Q/D/B/T/C: Q BI/BY/W/D: BI	
:SU F 11.3	Time can be read
:	(set bit 3 in the status word)
:C DB 2	
:L DR 0	WEEKDAY
:T =WDAY	
:L DL 1	DAY
:T =DAY	
:L DR 1	MONTH
:T =MON	
:L DL 2	YEAR
:T =YEAR	
:L DR 2	HOUR
:L KH 007F	Erase AMPM bit
:AW	(only relevant in 12-HR mode)
:T =HOUR	
:A D 2.7	Display AMPM bit
:= =AMPM	(only relevant in 12-HR mode)
:L DL 3	MINUTE
:T =MIN	
:L DR 3	SECOND
:T =SEC	
:RU F 11.3	time is updated again
:A F 11.1	Display clock mode
:= =MODE	MODE = 1, IN 12-HR mode
:BE	

Storing the Current Time/Date After a RUN/STOP Change

Note

This clock data area is only written to if

- bit 5 in the status word is set to "1"
- a RUN/STOP change or a POWER OFF has taken place
- the necessary memory space is available in the operand area

This enables you to detect a RUN/STOP change or a POWER OFF even if the PC has since gone back to RUN mode. The time and date of the last RUN/STOP change or POWER OFF are in words 18 to 21 (see Table 13.2).

If several RUN/STOP changes have occurred before you read out this clock data area, you will only be able to determine the time of the last change.

If you do not have sufficient memory for this clock data area, you cannot use this area or only part of it. This has no other effects.

Programming the Prompt Function

Transferring Settings to the Clock

- The settings are stored in the clock data area using Transfer operations (→ see Table 13.2).
- The AM/PM flag (bit No. 7) is only significant in 12-hour mode
  Bit 7 = 1→ PM
  Bit 7 = 0→ AM
- The clock data must be transferred in BCD.
- If you enter the number "255_D" or "FF_H" in a byte in the prompting time, this byte will be ignored when evaluating "Prompt time reached". This makes it easy to program, for example, a prompt which is repeated daily by entering the value "255_D or FF_H" in the "Weekday", "Date" and "Month" variables.
- Transfer of the prompt function settings to the clock is initiated by bit 14 in the status word.
- Incorrect settings are flagged by bit 12 in the status word.

Prompter Time Sequence

- Bit 13 in the status word is set after the prompter time has elapsed.
- Bit 13 remains set until you reset it in the control program.
- The prompting time can be read at any time.

> **Caution**
>
> If the prompting time is reached in STOP mode or in POWER OFF, the prompting time bit is not set.

Example: Setting and evaluating the prompting time.

In the example program, the settings for the prompting time are transferred as a function of the state of input 12.2. You must transfer the settings to flag bytes 130 and 135 before setting input 12.2. Values that are not to be changed must be preset with FF_H.
The clock mode is set with input 14.0. Use input 13.0 to set the AM/PM bit for 12-hour mode.
Flag 13.2 is set when the preset prompting time has been reached. Any errors made when entering the prompter time are flagged in F 12.2.
The clock data is stored in DB2 from DW 0 onwards, and the status word is FW 10.

OB1 STL	Description
:	==
:	SETTING AND EVALUATING THE PROMPTING TIME
:	==
:	Load values into FY130 to FY135
:	first!
:A I 12.2	Initiate setting of prompting time
:S F 20.1	by setting F 20.1 (reset in FB11).
:	
:JU FB 11	
NAME :WECKZ-ST	
WDAY : FY 130	WEEKDAY
DAY : FY 131	DAY
MON : FY 132	MONTH
HOUR : FY 133	HOUR
AMPM : I 13.0	AMPM-BIT (only important in 12-HR mode)
MIN : FY 134	MINUTES
SEC : FY 135	SECONDS
ERR : F 12.2	Error bit
ALRM : F 13.2	Flag for prompting time reached
MODE : I 14.0	12-HR-mod: I 14.0 = 1
:BE	

FB11 STL	Description
NAME :WECKZ-ST	SETTING THE PROMPTING TIME
DECL :WKDAY I/Q/D/B/T/C: I BI/BY/W/D: BY	
DECL :DAY I/Q/D/B/T/C: I BI/BY/W/D: BY	
DECL :MON I/Q/D/B/T/C: I BI/BY/W/D: BY	
DECL :HOUR I/Q/D/B/T/C: I BI/BY/W/D: BY	
DECL :AMPM I/Q/D/B/T/C: I BI/BY/W/D: BI	
DECL :MIN I/Q/D/B/T/C: I BI/BY/W/D: BY	
DECL :SEC I/Q/D/B/T/C: I BI/BY/W/D: BY	
DECL :ERR I/Q/D/B/T/C: Q BI/BY/W/D: BI	
DECL :ALRM I/Q/D/B/T/C: Q BI/BY/W/D: BI	
DECL :MODE I/Q/D/B/T/C: I BI/BY/W/D: BI	
:A =MODE	24-HR mode = 0, 12-HR mode = 1
:= F 11.1	(Set clock mode)
:A F 10.5	Display prompting time reached
:S =ALRM	(Bit 13 in the status word)
:R F 10.5	Reset bit after evaluation
:	
:AN F 20.1	Flag is reset if settings have
:JC =M001	been read into the clock data
:R F 20.1	area
:	
:C DB 2	Clock data area
:L =WKDAY	Store value for WEEKDAY
:T DR 8	
:L =DAY	Store value for DAY
:T DL 9	
:L =MON	Store value for MONTH
:T DR 9	

FB11 STL (Continued)	Description
:L =HOUR	Store value for HOUR
:ON =AMPM	if AMPM = 1 (afternoon) and
:ON =MODE	12-HR mode are set, the
:JC =MORN	corresponding bit on the clock
:L KH 0080	data area will be set
:OW	
MORN :T DR 10	
:L =MIN	Store value for MINUTES
:T DL 11	
:L =SEC	Store value for SECONDS
:T DR 11	
:SU F 10.6	Transfer settings
:	(Bit 14 in status word FW 10)
:L KT 020.1	Start monitoring time
:SE T 11	
M001 :A T 11	BEC if monitoring time not yet
:BEC	elapsed
:A F 10.6	Have settings been transferred?
:JC =M002	If NO, jump to M002
:AN F 10.4	Error when entering settings?
:RB =ERR	if NO, reset error bit
:BEC	BEC if no errors
M002 :S =ERR	If error, set error bit
:BE	

Programming the Operating Hours Counter

The operating hours counter is enabled with bit 9 of the status word. This allows you to establish, for example, the number of hours a motor has been in operation. The operating hours counter is only active in RUN mode.

Transferring Settings to the Operating Hours Counter

You can preset the operating hours counter to a specific initial value (e.g. after change of CPU).

- The clock data must be transferred in BCD.

- If a variable is not to be transferred when you are entering settings for the operating hours counter, identify the relevant byte with the number "255_D" or "FF_H". The value of this variable in the operating hours counter will then be retained when setting the counter.
- After you have transferred the settings to the clock data area, you must set bit 10 of the status word to have the clock data area accepted by the clock.
- Incorrect settings are flagged by bit 8 in the status word.

Example: Setting the operating hours counter

Transferring the settings for the operating hours counter is a function of the state of input 12.3. You must transfer these values to flag bytes 136 to 140 before setting input 12.3 (not implemented in the example program). Values that are not to be changed should be preset with FF_H.

Incorrect settings are flagged in F 12.3.

The clock data area is in DB2 from DW 0 onwards, and the status word in FW 10.

OB1 STL	Description
:	===================================
:	SETTING THE OPERATING HOURS COUNTER
:	===================================
:	Load values into FY 136 to FY 140
:	first!
:A I 12.3	Initiate transfer of settings for
:S F 20.2	operating hours counter by setting
:	F 20.2
:JU FB 12	
NAME :BETRST-S	
SEC : FY 136	SECONDS
MIN : FY 137	MINUTES
HOUR0: FY 138	HOURS
HOUR2: FY 139	HOURS X 100
HOUR4: FY 140	HOURS X 10000
ERR : F 12.3	ERROR BIT
:BE	

FB12 STL	Description
NAME :BETRST-S	SETTING THE OPERATING HOURS COUNTER
DECL :SEC I/Q/D/B/T/C: I BI/BY/W/D: BY	
DECL :MIN I/Q/D/B/T/C: I BI/BY/W/D: BY	
DECL :HOUR0 I/Q/D/B/T/C: I BI/BY/W/D: BY	
DECL :HOUR2 I/Q/D/B/T/C: I BI/BY/W/D: BY	
DECL :HOUR4 I/Q/D/B/T/C: I BI/BY/W/D: BY	
DECL :ERR I/Q/D/B/T/C: Q BI/BY/W/D: BI	
:AN F 20.2	Flag is reset if settings have
:JC =M001	already been read into the
:R F 20.2	clock data area
:	
:C DB 2	Clock data area
:L =SEC	Store value for SECONDS
:T DR 15	
:L =MIN	Store value for MINUTES
:T DL 16	
:L =HOUR0	Store value for HOURS
:T DR 16	
:L =HOUR2	Store value for HOURS X 100
:T DL 17	
:L =HOUR4	Store value for HOURS X 10000
:T DR 17	
:AN F 10.2	Transfer settings
:S F 10.2	(Bit 10 in status word FW 10)
:S F 10.1	Enable operating hours counter
:	if not already enabled
:L KT 020.1	start monitoring time
:SE T 12	
M001 :A T 12	BEC if monitoring time not yet
:BEC	elapsed
:AN F 10.2	Have settings been transferred?
:JC =M002	if NO, jump to M002
:AN F 10.0	Error when entering settings?
:RB =ERR	if NO, reset error bit
:BEC	BEC if no error
M002 :S =ERR	if error, set error bit
:BE	

Reading the Current Operating Hours

The current data is stored in words 12 to 14 of the clock data area. The data can be read from there with Load operations.

Bit 9 in the status word of the control program must be reset before the read access in order to be able to read the operating hours counter correctly. The clock data area is no longer updated when bit 9 is reset. You must set this bit again after reading the clock.

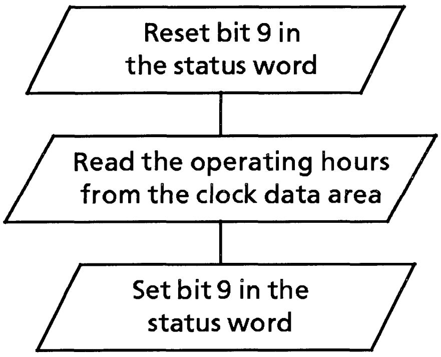

Reset bit 9 in the status word
Read the operating hours from the clock data area
Set bit 9 in the status word

Figure 13.3 Procedure for Reading the Operating Hours Counter

Example: Reading the operating hours counter

A machine is to be switched off after 300 hours of operation for inspection purposes. Flag 12.4 is set when the machine is switched off. After 300 hours of operation, a jump is made to PB5 to switch the machine off (not programmed in the example).

The clock data area is in DB2 from FW 0 onwards, and the status word is FW 10.

OB1 STL	Description
:JU FB 14 NAME :BETR-LES : :BE	EVALUATE OPERATING HOURS COUNTER

FB14 STL	Description
NAME :BETR-LES	READING THE OPERATING HOURS COUNTER
:A F 12.4	If auxiliary flag 12.4 is set,
:BEC	the machine is already
:	off. --> block END
:RU F 10.1	Disable operating hours counter
:	(Bit 9 in the status word)
:C DB 2	DB containing clock data.
:L DL 14	Load HOUR VALUE x 100 into
:	ACCU 1
:SU F 10.1	Enable operating hours counter
:L KB 003	Compare with 3 (= 300 hours)
:><F	
:BEC	END if 300 hours not yet
:	reached
:S F 12.4	Set auxiliary flag
:JU PB 5	Jump to PB5 when 300 operating
:	hours reached
:BE	

14 Reliability, Availability and Safety of Electronic Control Equipment

Figures

14 Reliability, Availability and Safety of Electronic Control Equipment

The terms reliability, availability and safety of electronic control equipment are not always clear and sometimes even misinterpreted. This can be explained on the one hand by the different failure characteristics of electronic control systems compared with conventional systems. On the other hand, some of the safety regulations have been made considerably more stringent in a number of application areas in the course of the last few years. The following chapter is intended to familiarize the large number of users of SIMATIC electronic control systems with the basics of this problem complex.

The information given is of a predominantly fundamental nature and applies regardless of the type of electronic control system and its manufacturer.

14.1 Reliability

Reliability is the capability of an electronic control system to satisfy, over a specified period and within the specified limits (i.e. technical data), the requirements placed upon it by its application.

Despite all the measures taken to prevent failures, there is no such thing as 100 % reliability.

The failure rate λ is a measure of the reliability:

$$\lambda = \frac{n}{N_0 \times t}$$

and n = Number of failures during time t

N_0 = Remaining components

14.1.1 Failure Characteristics of Electronic Devices

The failure-rate-versus-time curve can be broken down roughly into three periods of time.

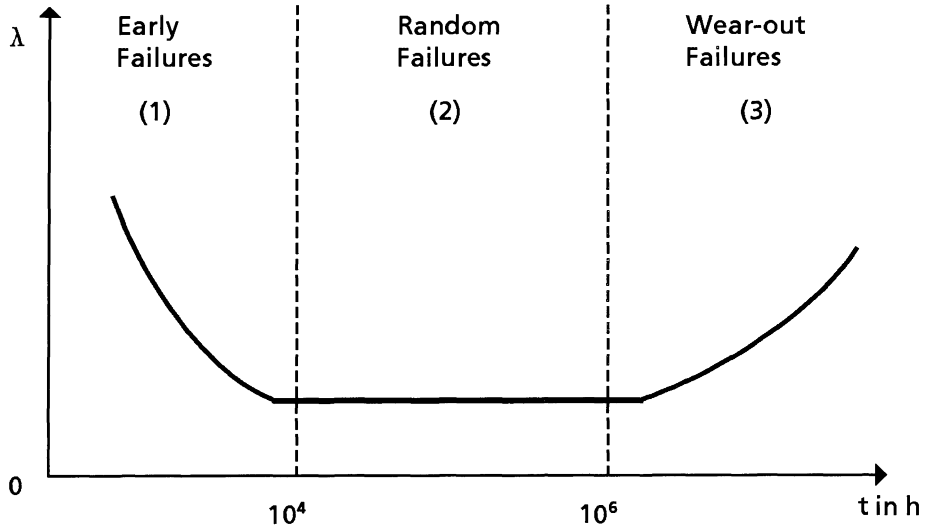


Figure 14.1 Failure Characteristics of Electronic Devices ("Bathtub" Curve)

(1) Early failures are caused by material and manufacturing defects and the failure rate falls steeply during the initial period of operation.

(2) The random failure phase is characterized by a constant failure rate. Provided the systems are used in accordance with the specifications, only random failures occur during this period. This period covers the normal behaviour of system components and is the basis for the calculation of all reliability parameters.

(3) The failure rate increases with time. Wear-out failures become more frequent, indicating that the end of the useful life is approaching. The transition to this phase is gradual. There is no sudden increase in the failure rate.

14.1.2 Reliability of SIMATIC S5 Programmable Controllers and Components

A very high degree of reliability can be achieved by taking the following extensive and cost-intensive measures during the development and manufacture of SIMATIC S5 systems:

- The use of high-quality components;
- Worst-case design of all circuits;
- Systematic and computer-controlled testing of all components supplied by subcontractors;
- Burn-in of all LSI circuits (e.g. processors, memories etc.);
- Measures to prevent static charge building up when handling MOS ICs;
- Visual checks at different stages of manufacture;
- In-circuit testing of all components, i.e. computer-aided testing of all components and their interaction with other components in the circuit;
- Continuous heat-run test at elevated ambient temperature over a period of several days;
- Careful computer-controlled final testing;
- Statistical evaluation of all failures during testing to enable the immediate initation of suitable corrective measures.

14.1.3 Failure Distribution

Despite the extensive measures described above, one must still reckon with the occurence of failures. Experience has shown that, in installations with programmable controllers, failures can be distributed approximately as follows:

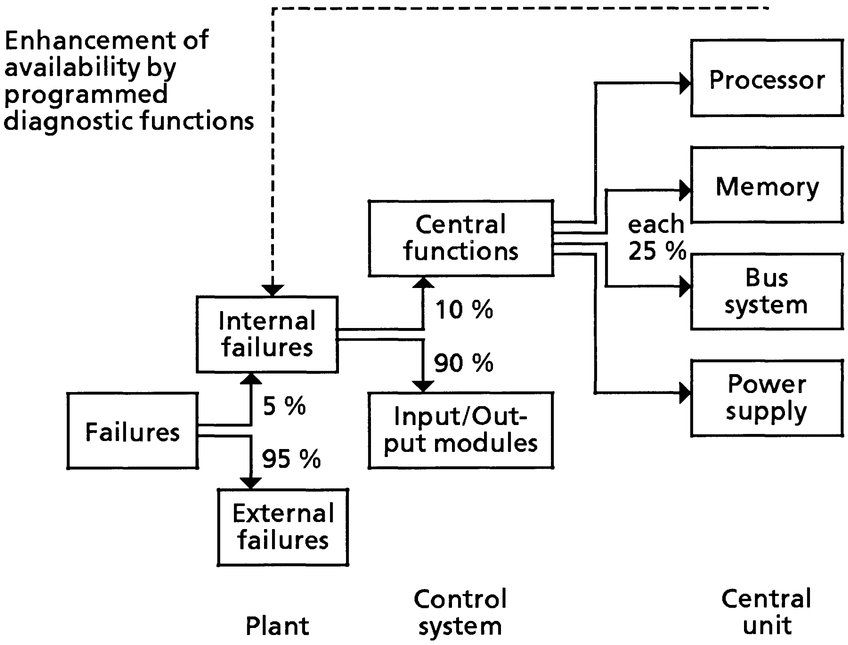


Figure 14.2 Distribution of Failure Occurrences in Installations Incorporating Programmable Controllers

Meaning of error distribution:

- Only a small number (approx. 5 %) of failures occur inside the electronic control system. These can be broken down as follows:
  - CPU failures (about 10 %, i.e. only 0.5 % of all failures); these failures are evenly divided among the processor, memory, bus system and power supply.
  - I/O module failures (about 90 %, i.e. only 4.5 % of all failures)
- The highest number of all failures (about 95 %) occur in the sensors, actuators, drives, cabling etc.

14.2 Availability

Availability "V" is the probability of finding a system in a functional state at a specified point in time.

$$V = \frac{MTBF}{MTBF + MTTR}$$

MTBF= Mean Time Between Failures;
MTTR= Mean Time To Repair;

Ideal availability, i.e. V=1, can never be attained owing to the residual failure probability that always exists.

However, it is possible to get near this ideal state by using, for example, voter systems. Such systems include the following:

- Standby sytems
- 2-out-of-3 voter systems
- Multi-channel voter systems with mutual check functions (for maximum safety requirements).

Availability can also be enhanced by reducing the mean time to repair. Such measures include, for instance:

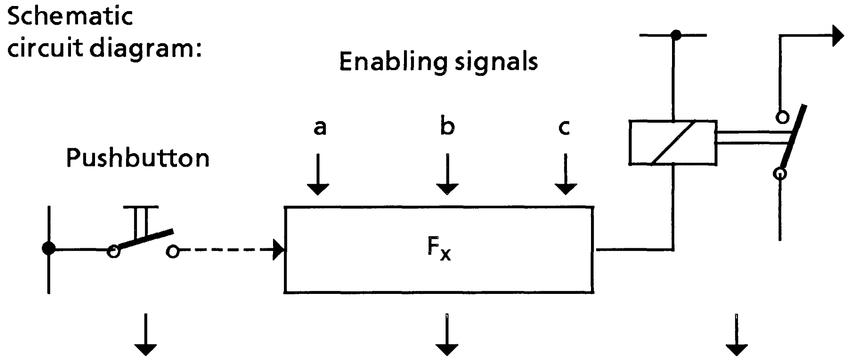
- the stocking of spare parts
- the training of operating personnel
- fault indicators on the devices
- higher memory and software overhead for implementing programmed diagnostic functions.

14.3 Safety

14.3.1 Types of Failures

The nature of a failure is decided by the effect it has. A distinction is made between active and passive failures, as well as fatal and non-fatal failures.

Example: Control of function "F_x"



Input	Control System	Output
0 1	No fault	No output command Output command
0 1	Active failure Passive failure	Output command No output command

Figure 14.3 Control of Function "F_x"

Depending on the job a control system has to do, active or passive failures can also be fatal faults.

Examples:

- In a drive control system, an active failure results in the unauthorized starting of the drive.
- In an indicating system, a passive fault can be fatal since it blocks the indication of a dangerous operating state.

In all cases where the occurence of failures can result in severe material damage or even injury to persons, i.e. where the failure may be dangerous or fatal, measures must be taken to enhance the safety of the control system. In this connection, the relevant regulations and specifications must be observed.

14.3.2 Safety Measures

Single-Channel Configurations

In the case of single-channel programmable controllers, the means available for enhancing safety are limited:

- Programs or parts can be stored and executed more than once.
- Outputs can be monitored per software by parallel feedback to inputs of the same device.
- Diagnostic functions within the programmable control system, which bring the output of the controller into a defined state (generally the FF state) when a failure occurs.

Failure characteristics of electromechanical and electronic control systems:

- Relays and contactors pick up only if a voltage is applied to the coil. With such a control element, therefore, active failures are less probable than passive failures.
- In electronic control systems, however, the probability of both types of failure occurring (active and passive) is approximately equal. The failing of an output transistor, for instance, may cause this transistor to become either continuously non-conducting or continuously conducting.

The safety of electronic control systems can therefore be enhanced as follows.

- All functions not relevant to the safety of the plant are controlled electronically.
- Functions that are relevant to the safety of the plant are implemented with conventional control elements.

Multi-Channel Configurations

If the measures taken to improve safety in single-channel control systems are not sufficient to satisfy safety requirements, electronic control systems should be designed as redundant, i.e. multi-channel, systems.

- Two-channel control systems
  Both "channels" monitor each other mutually and the output commands are evaluated on a "1-out-of-2" or "2-out-of-2" basis.
  Typical PLC: S5-115F
  This programmable controller consists of two submits that are identically programmed and operate in clock synchronism; monitoring is implemented via two comparator modules. Failures are displayed and the corresponding safety functions initiated.
- Multi-channel control systems
  Further voter systems (e.g. on the 2-out-of-3 principle) can be implemented by adding further "channels".

14.4 Summary

- In electronic control systems, failures of any kind can occur at any point in the system.
- Even when the greatest efforts are made to obtain maximum reliability, the probability of such a failure occurring can never be zero.
- The following is decisive for the effects of such failures: depending on the job a control system has to do, active or passive failures may be fatal or non-fatal.
- When safety requirements are very high, fatal failures must be recognized by taking additional measures and prevented from affecting other parts of the system.
- In the case of single-channel systems, the means available to do this are relatively limited.
  For this reason, safety-oriented functions should generally be implemented outside the electronics by interposing conventional components.
- In order to satisfy safety functions, electronic control systems should be of the multi-channel (redundant) type.
- These fundamental considerations are independent of
  - the type of control systems (hard-wired or programmable)
  - the vendor
  - the country of origin (Europe, US, etc.).

15 Technical Specifications

Tables

15 Technical Specifications

15.1 General Technical Specifications

The general technical specifications include standards and test specifications which the S5-115U meets and fulfills and which were used during testing of the S5-115U.

UL/CSA Approbations

The following approbations have been granted for the S5-115U:
UL-Recognition Mark
Underwriters Laboratories (UL) to UL standard 508, Report 116536
CSA Certification Mark
Canadian standard Association (CSA) to C22.2 standard No. 142, Report LR 48323

CE-Marking

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

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

Siemens Aktiengesellschaft
Bereich Automatisierungstechnik
AUT E 14
Postfach 1963

D-92209 Amberg
Federal Republic of Germany

Area of Application

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

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

Observing the Installation Guidelines

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

Notes for the Machine Manufacturer

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

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

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

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

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

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

Climatic
Environmental Conditions

Temperature

Operation	
- Open Design air intake temperature (measured at the bottom of the modules)	0 to + 55 ° C
- Cabinet design (Where cabinet design is concerned, the dissipatible heat loss depends on the type of construction, the ambient temperature, and the arrangement of the devices.) Air intake temperature (measured at the bottom of the modules)	0 to + 55 ° C
Storage/Shipping	- 40 to + 70 ° C
Temperature change	
- operation	maximum 10 K/h
- storage/shipping	maximum 20 K/h
Relative Humidity	
- operation	≤95 % (according to DIN 40040)
- storage/shipping	≤95 % (noncondensing)
Atmospheric Pressure	
- operation	860 to 1060 hPa [1]
- storage/shipping	660 to 1060 hPa [1]
Pollutants	
- SO_2	≤0.5 ppm, (rel. humidity ≤60 %, noncondensing)
- H_2S	≤0.1 ppm, (rel. humidity ≤60 %, noncondensing)

[1] For use under 900 hPa (= 1000 m above sea level), check with the manufacturer on the cooling requirements.

Mechanical
Environmental Conditions

Vibration*

- tested to IEC 68-2-6	
10 Hz ≤ f < 57 Hz,	Constant amplitude 0,075 mm
57 Hz ≤ f < 150 Hz,	Constant acceleration 1 g
Mode of vibration	Frequency sweeps with sweep rate of 1 oktave/min.
Vibration period	10 frequency sweeps per axis in each of the 3 axes vertical to each other

- Operation conditions in accordance with IEC 1131-2

Shock*

- tested to IEC 68-2-27	
Type of shock	Half-sine
Shock intensity	15 g peak value, 11 ms duration
Direction of shocks	2 shocks in each of the 3 axes vertical to each other

Free fall
- tested to IEC 68-2-3
  Height of fall 1 m

* Vibrations and shocks persisting for a prolonged period at the maximum values stated above and continuous shocks should be avoided by taking the appropriate measures.

Electromagnetic Compatibility (EMC) **Noise Immunity**	
Immunity to static electrical discharge	
- tested in accordance with EN 61000-4-2	
- Discharge to air	8 kV
- Discharge on contact	4 kV
Immunity to electromagnetic fields	
- tested in accordance with EN V 50140	
(amplitued-modulated HF)	80 to 1000 MHz 10 V/m 80 % AM (1 kHz)
- tested in accordance with EN V 50204	
(pulse-modulated HF)	900 MHz 10 V/m 50 % ED, 200 Hz repetition frequency
Immunity to fast transient bursts	
- tested in accordance with EN 61000-4-4	
- Supply lines for 120/230 V AC	2 kV
- Supply lines for 24 V DC	2 kV
- Signal lines (I/O and bus lines)	2 kV*
Immunity to high frequency	
- tested in accordance with EN V 50141	0.15 to 80 MHz 10 V 80 % AM (1 kHz) Source impedance 150 Ω
Emitted interference	
- tested in accordance with EN 55011	
- Emmission of electromagnetic fields	Limit value class A, Group 1
- Emitted interference over supply cable	Limit value class A, Group 1
Damped Oscillatory Wave Test	according to IEC 255-4
- AC power supply modules	2.5 kV
- DC power supply modules	1 kV
- output 24 V DC	1 kV
- input 115/230 V AC	2.5 kV
- digital input/output modules	2.5 kV
- analog input/output modules	1 kV
- communications interfaces	1 kV

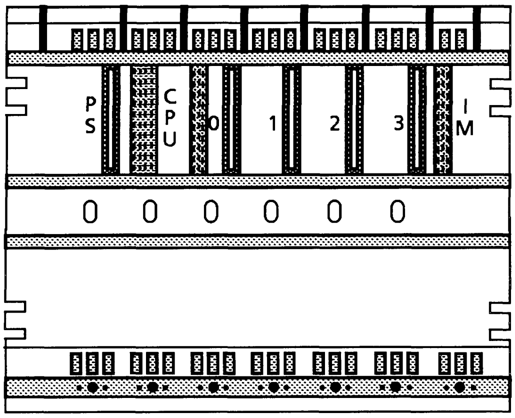
* Signal lines that are not used for process control, for example, connections to external printers: 1 kV

IEC / VDE **Safety Information**	
Degree of Protection	according to IEC 529
- Type	IP 20
- Class	I according to IEC 536
Isolation Rating	
- between electrically independent circuits **and** with circuits connected by a central grounding point	according to VDE 0160
- between all circuits **and** central grounding point (standard mounting rail)	according to VDE 0160
Test Voltage for a rated voltage V_e of the circuits (AC/DC)	Sine, 50 Hz
V_e=0 to 50 V	500 V DC
V_e=50 to 125 V	1250 V AC
V_e=125 to 250 V	1500 V AC
Impulse voltage for a rated voltage V_e of the circuits (AC/DC)	according to IEC 255-4
V_e=0 to 50 V	1 kV, 1.2/50 µsec.
V_e=50 to 125 V	1 kV, 1.2/50 µsec.
V_e=125 to 250 V	3 kV, 1.2/50 µsec.

15.2 Description of Modules

15.2.1 Mounting Racks (CRs, ERs)

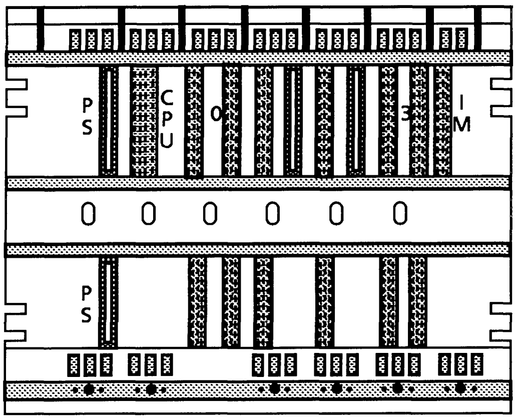
Mounting Rack CR 700-0 for Central Controller 0 (6ES5 700-0LA12)



Technical Specifications

Number of input/output modules that can be plugged in	maximum	4
Number of expansion units that can be connected - central	maximum	3
Dimensions w×h×d (mm)		353 x 303 x 47
Weight		4 kg (8.82 lb.)

Mounting Rack CR 700-0 for Central Controller 0 (6ES5 700-0LB11)

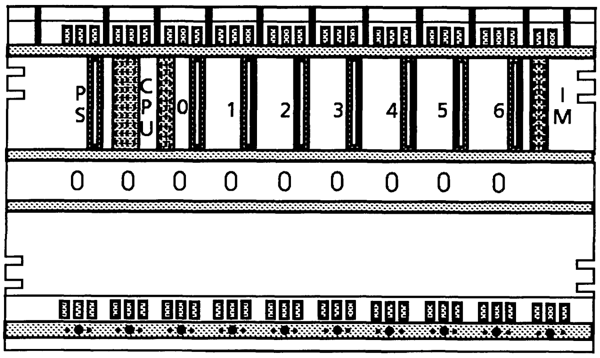


Technical Specifications

Number of input/output modules that can be plugged in	maximum	6
Number of expansion units that can be connected - central	maximum	3
- distributed *	maximum	63
Dimensions w x h x d (mm)		353 x 303 x 47
Weight		4 kg (8.82 lb.)

* see Section 3.3.2

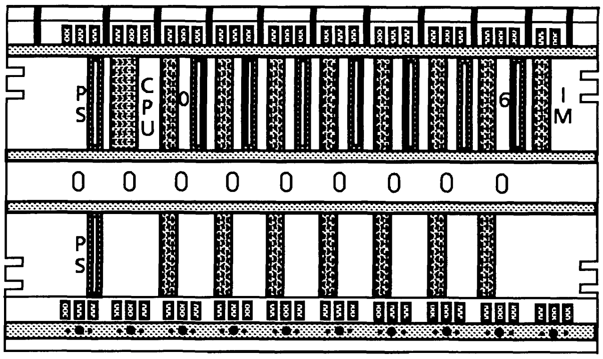
Mounting Rack CR 700-1 for Central Controller 1 (6ES5 700-1LA12)



Technical Specifications

Number of input/output modules that can be plugged in	maximum 7
Number of expansion units that can be connected - central	maximum 3
Dimensions w x h x d (mm)	483 x 303 x 47 (18.84 in. x 11.82 in. x 1.83 in.)
Weight	5 kg (11.02 lb.)

Mounting Rack CR 700-2 for Central Controller 2 (6ES5 700-2LA12)

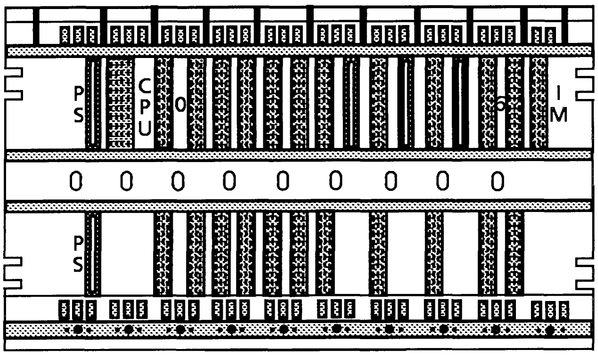


Technical Specifications

Number of input/output modules that can be plugged in	maximum 7
Number of expansion units that can be connected - central - distributed *	 maximum 3 maximum 63
Dimensions w x h x d (mm)	483 x 303 x 47 (18.84 in. x 11.82 in. x 1.83 in.)
Weight	5 kg (11.02 lb.)

* see Section 3.3.2

Mounting Rack CR 700-3 for Central Controller 3 (6ES5 700-3LA12)

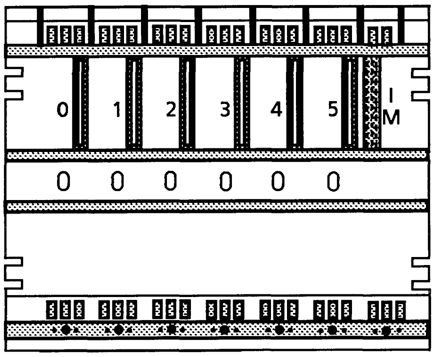


Technical Specifications

Number of input/output modules that can be plugged in	maximum	11
Number of expansion units that can be connected		
- central connection	maximum	3
- distributed *	maximum	63
Dimensions w x h x d (mm)		483 x 303 x 47 (18.84 in. x 11.82 in. x 1.83 in.)
Weight		5 kg (11.02 lb.)

* see Section 3.3.2

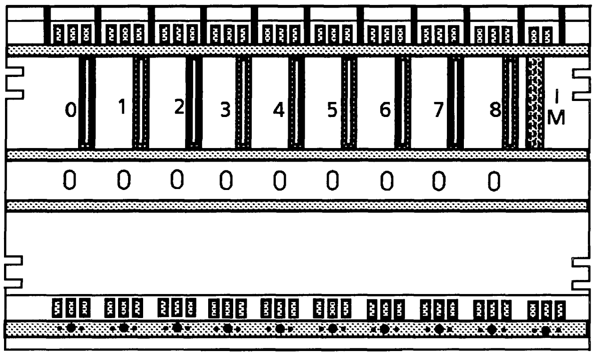
Mounting Rack ER 701-0 for Expansion Unit 0 (6ES5 701-0LA11)



Technical Specifications

Number of input/output modules that can be plugged in	maximum	6
Interface module		
- central connection		IM 305/IM 306
Interrupt evaluation		not possible
Dimensions w x h x d (mm)		353 x 303 x 47
Weight		4 kg (11.02 lb.)

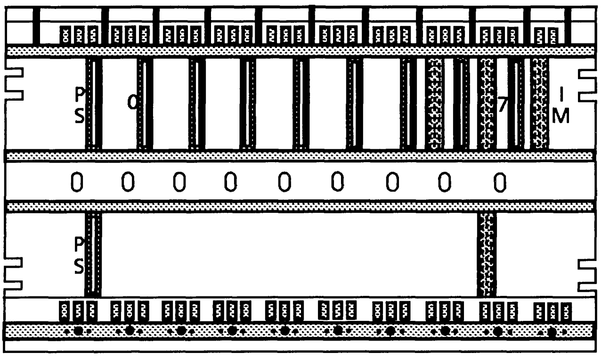
Mounting Rack ER 701-1 for Expansion Unit 1 (6ES5 701-1LA12)



Technical Specifications

Number of input/output modules that can be plugged in	**maximum**	**9**
Interface module		
- central connection		**IM 305/IM 306**
Interrupt evaluation		**not possible**
Dimensions w x h x d (mm)		**483 x 303 x 47 (18.84 in. x 11.82 in. x 1.83 in.)**
Weight		**5 kg (11.02 lb.)**

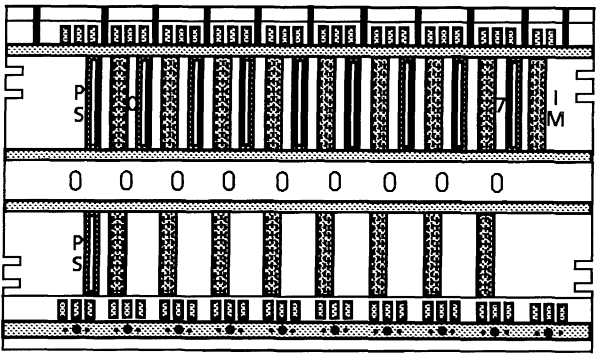
Mounting Rack ER 701-2 for Expansion Unit 2 (6ES5 701-2LA12)



Technical Specifications

Number of input/output modules that can be plugged in	**maximum**	**7**
Interface module		
- central connection		**IM 306**
- distributed connection		**AS 310/AS 311 IM 314/IM 317/ IM 318**
Interrupt evaluation		**not possible**
Dimensions w x h x d (mm)		**483 x 303 x 47 (18.84 in. x 11.82 in. x 1.83 in.)**
Weight		**5 kg (11.02 lb.)**

Mounting Rack ER 701-3 for Expansion Unit 3 (6ES5 701-3LA12)

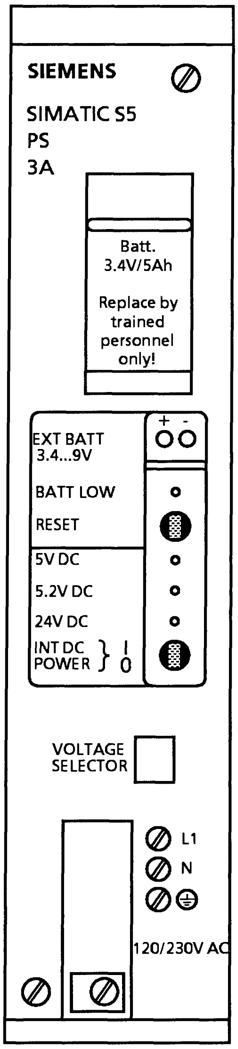


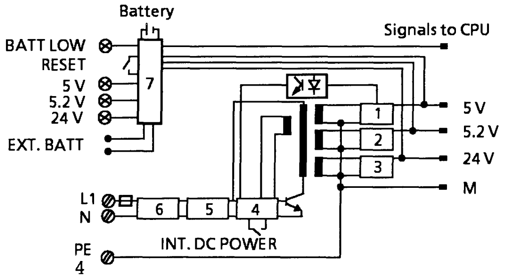
Technical Specifications

Number of input/output modules that can be that can be plugged in	maximum	7
Interface module		
- central connection		IM 306
- distributed connection		AS 310/AS 311 IM 314/IM 317/ IM 318
Interrupt evaluation		possible with IM 307/317
Dimensions w x h x d (mm)		483 x 303 x 47 (18.84 in. x 11.82 in. x 1.83 in.)
Weight		5 kg (11.02 lb.)

15.2.2 Power Supply Modules

Power Supply Module PS 951 120/230 V AC; 5 V, 3 A (6ES5 951-7LB21)



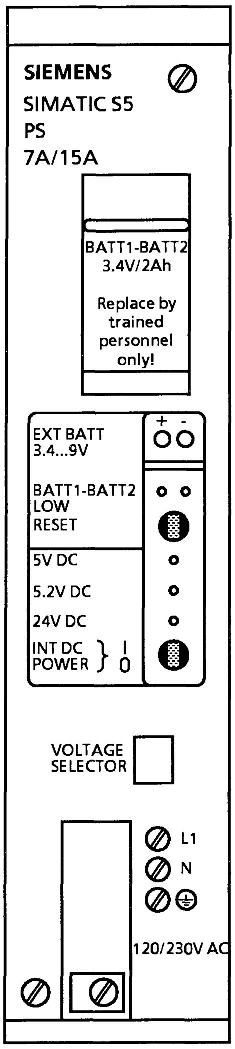


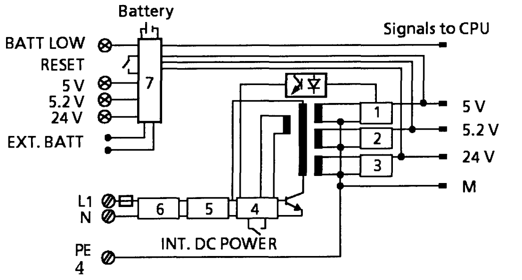
1 Chopper controller
2 Linear controller
3 Linear controller
4 Control electronics
5 Rectifier
6 Radio interference suppression filter
7 Monitoring electronics

Simplified schematic

Technical specifications		
Input voltage L1		
- Rated value		120/230 V AC
- Permissible range		94 to 132 V/ 187 to 264 V
Line frequency		
- Rated value		50 Hz
- Permissible range		47 to 63 Hz
Input current at 120/230 V		
- Rated value		0.55/0.33 A
- Inrush current	max.	$15 \times I_N$
- I^2t		0.135 A^2s
Power consumption (active power)		44.8 W
Output voltage		
- Rated value		5 V
- Tolerance		±1.5%
Output current		
- Rated value without fan		3 A
- Rated value with fan		3 A
- Permissible range		0.3 A to 3 A
Output voltage (PG/OP)		
- Rated value		5.2 V
- Tolerance		±1.5%
Output current		max. 1 A
Output voltage (auxiliary voltage)		
- Rated value		24 V
- Tolerance		±5%
Output current		max. 0.2 A
Backup battery		Lithium battery, size C (3.6 V/5 Ah)
- Backup time	min.	1 year (at 0.3 mA, 25°C and uninterrupted backup)
Mains buffering (at $L1_{min}$)	min.	20 ms
Short-circuit protection		Electronic
Fault indicator		No
Fuse (primary circuit)		Integral
Galvanic isolation		Yes
Isolation rating		Safe electrical isolation according to VDE 0160
- Tested with		2700 V DC
Initial discharge current to VDE 0160 at 230 V AC		2.6 mA
RI specifications		A according to VDE 0871
Power losses of the module	typ.	19.8 W
Weight	approx.	1.6 kg (3.53 lb.)

Power Supply Module PS 951 120/230 V AC; 5 V, 7/15 A (6ES5 951-7LD21)





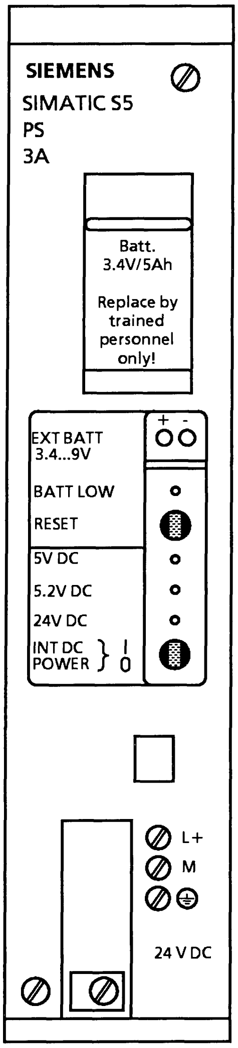
1 Chopper controller
2 Linear controller
3 Linear controller
4 Control electronics
5 Rectifier
6 Radio interference suppression filter
7 Monitoring electronics

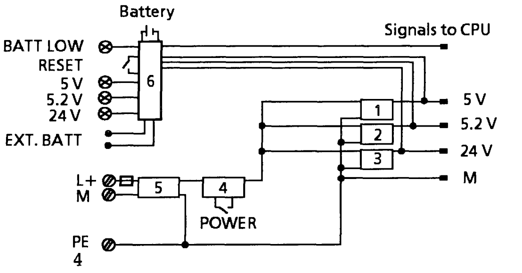
Simplified schematic

Technical specifications

Input voltage L1		
- Rated value		120/230 V AC
- Permissible range		94 to 132 V 187 to 264 V
Line frequency		
- Rated value		50 Hz
- Permissible range		47 to 63 Hz
Input current at 120/230 V		
- Rated value		1.4/0.8 A
- Inrush current	max.	15 × I_N
- I^2t		1.8 A^2s
Power consumption (active power)		133 W
Output voltage		
- Rated value		5 V
- Tolerance		± 1.5%
Output current		
- Rated value without fan		7 A
- Rated value with fan		15 A
- Permissible range		0.3 to 15 A
Output voltage (PG/OP)		
- Rated value		5.2 V
- Tolerance		± 1.5%
Output current		max. 2.5 A
Backup battery		2 lithium batteries, size AA (3.6 V/2x1.75 Ah)
- Backup time	min.	1 year (at 0.3 mA, 25°C and uninterrupted backup)
Mains buffering (at $L1_{min}$)	min.	20 ms
Output voltage (auxiliary voltage)		
- Rated value		24 V
- Tolerance		± 5%
Output current		max. 0.35 A
Short-circuit protection		Electronic
Fault indicator		No
Fuse (primary circuit)		Integral
Class of protection		Class 1
Galvanic isolation		Yes
Isolation rating		Safe electrical isolation according to VDE 0160
- Tested with		2700 V DC
Initial discharge current to VDE 0160 at 230 V AC		2.6 mA
RI specifications		A according to VDE 0871
Power losses of the module	typ.	36 W
Weight	approx.	1.9 kg (4.19 lb.)

Power Supply Module PS 951 24 V DC; 5 V, 3 A (6ES5 951-7NB21)





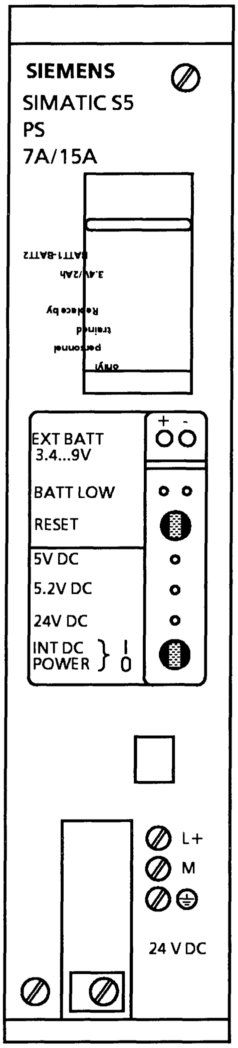
1 Chopper controller
2 Chopper controller
3 Linear controller
4 Control electronics
5 Radio interference suppression filter
6 Monitoring electronics

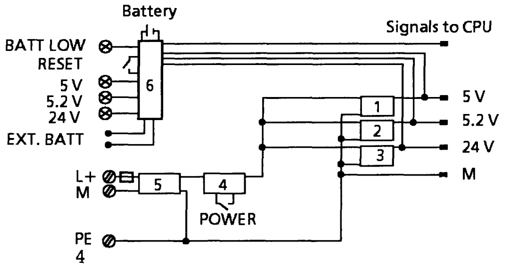
Simplified schematic

Technical specifications

Input voltage L+		
- Rated value		24 V DC
- Permissible range		19.2 to 30 V
Input current at 24 V		
- Rated value		1.51 A
- Inrush current	max.	15 × I_N
- I^2t		0.4 A^2s
Power consumption		36.2 W
Output voltage		
- Rated value		5 V
- Tolerance		± 1.5%
Output current		
- Rated value without fan		3 A
- Rated value with fan		3 A
- Permissible range		0.3 to 3 A
Output voltage (PG/OP)		
- Rated value		5.2 V
- Tolerance		± 1.5%
Output current		max. 1 A
Backup battery		Lithium battery, size C (3.6 V/5 Ah)
- Backup time	min.	1 year (at 0.3 mA, 25°C and uninterrupted backup)
Mains buffering (at L+ $_{min}$)	min.	20 ms
Output voltage (auxiliary voltage)		
- Rated value		24 V
- Tolerance		± 5%
Output current		max. 0.2 A
Short-circuit protection		Electronic
Fault indicator		No
Fuse (primary circuit)		Integral
Class of protection		Class 1
Galvanic isolation		No
RI specifications		A according to VDE 0871
Power losses of the module	typ.	11.2 W
Weight	approx.	1.6 kg (3.53 lb.)

Power Supply Module PS 951 24 V DC; 5 V, 7/15 A (6ES5 951-7ND51)





1 Chopper controller
2 Chopper controller
3 Linear controller
4 Control electronics
5 Radio interference suppression filter
6 Monitoring electronics

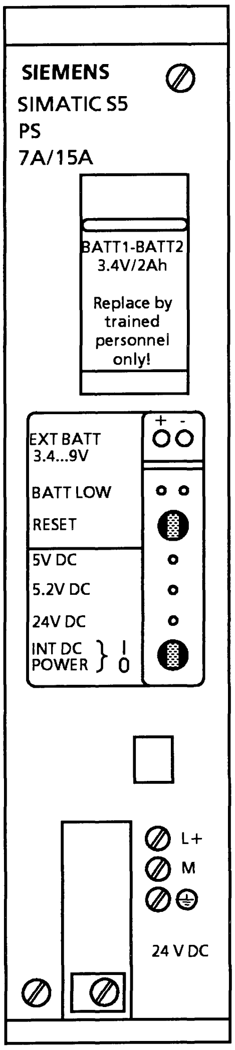
Simplified schematic

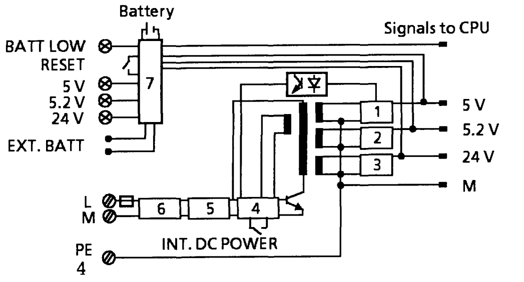
Technical specifications

Input voltage L+		
- Rated value		24 V DC
- Permissible range		19.2 to 30 V
Input current at 24 V		
- Rated value		5.04 A
- Inrush current	max.	15 × I_N
- I^2t		16 A^2s
Power consumption		120.5 W
Output voltage		
- Rated value		5 V
- Tolerance		±1.5%
Output current		
- Rated value without fan		7 A
- Rated value with fan		15 A
- Permissible range		0.3 to 15 A
Output voltage (PG/OP)		
- Rated value		5.2 V
- Tolerance		±1.5%
Output current		max. 2.5 A
Backup battery		2 lithium batteries, size AA (3.6 V/ 2 × 1.75Ah)
- Backup time	min.	1 year (at 0.3 mA, 25°C and uninterrupted backup)
Mains buffering (at L+ $_{min}$)	min.	20 ms
Output voltage (auxiliary voltage)		
- Rated value		24 V
- Tolerance		±5%
Output current		max. 0.35 A
Short-circuit protection		Electronic
Fault indicator		No
Fuse (primary circuit)		Integral
Class of protection		Class 1
Galvanic isolation		No
RI specifications according to VDE		A according to VDE 0871
Power losses of the module	typ.	24.1 W
Weight	approx.	1.7 kg (3.75 lb.)

15

Power Supply Module PS 951 24 V DC; 5 V, 7/15 A (6ES5 951-7ND41)





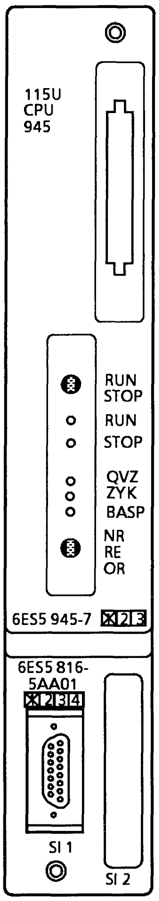
1 Chopper controller
2 Linear controller
3 Linear controller
4 Control electronics
5 Rectifier
6 Radio interference suppression filter
7 Monitoring electronics

Simplified schematic

Technical specifications		
Input voltage L+		
- Rated value		**24 V DC**
- Permissible range		**19.2 to 30 V**
Input current at 24 V		
- Rated value		**5.6 A**
- Inrush current	**max.**	**15 × I_N**
- I^2t		**4.5 A^2 s**
Power consumption		**134.4 W**
Output voltage		
- Rated value		**5 V**
- Tolerance		**±1.5%**
Output current		
- Rated value without fan		**7 A**
- Rated value with fan		**15 A**
- Permissible range		**0.3 to 15 A**
Output voltage (PG/OP)		
- Rated value		**5.2 V**
- Tolerance		**±1.5%**
Output current		**max. 2.5 A**
Backup battery		**2 lithium batteries, size AA (3.6 V/2 x 1.75 Ah)**
- Backup time	**min.**	**1 year (at 0.3 mA, 25°C and uninterrupted backup)**
Mains buffering (at L+ $_{min}$)	**min.**	**20 ms**
Output voltage (auxiliary voltage)		
- Rated value		**24 V**
- Tolerance		**±5%**
Output current		**max. 0.35 A**
Short-circuit protection		**Electronic**
Fault indicator		**No**
Fuse (primary circuit)		**Integral**
Class of protection		**Class 1**
Galvanic		**Yes**
Isolation rating		**Safe electrical isolation according to VDE 0160**
- Tested with		**2700 V DC**
RI specifications		**A according to VDE 0871**
Power losses of the module	**typ.**	**38 W**
Weight	**approx.**	**1.7 kg (3.75 lb.)**

15.2.3 Central Processing Units

Central Processing Unit CPU 941 (6ES5 945-7UA1./2.)



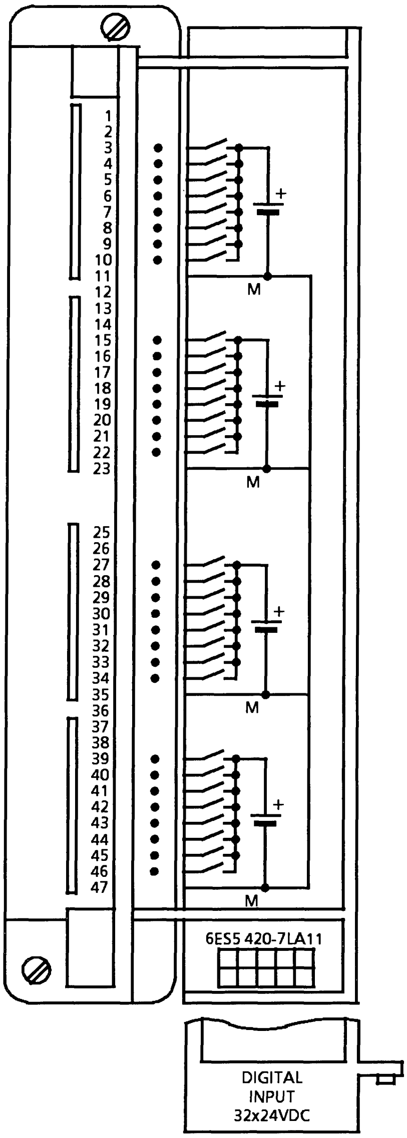
Technical Specifications

Memory capacity (total)	maxim.	256/384 Kbyte
- internal memory	maxim.	256/384 Kbyte
- memory submodule (Flash-EPROM)	maxim.	128/256/384 Kbyte
Execution time per operation	approx.	0.1 µs - 1.35 µs
Clock		
- Accuracy t_a		± 2s /day
- Temperature dependency t_D (ambient temperature T_{amb} in °C)		-3.5 x (T_{amb}-15)^2 ms /day
- e. g. Tolerance at 40°C		± 2 s - 3.5 x (40 -15)^2 ms/day approx. 0 to -4s /day
Scan time monitoring	approx.	500 msec. (can be modified)
Flags		2048; optionally half or all retentive [1]
S flags		32768; optionally half or all retentive [1]
Timers		
- number		256; optionally half or T 0 to 63 retentive[1]
- range		0.01 to 9990 sec.
Counter		
- number		256; optionally half or C 0 to 63 retentive [1]
- range		0 to 999 (up, down)
Digital inputs Digital outputs - total	maxim.	8192
Analog inputs Analog outputs - total	maxim.	256
Organization blocks	maxim.	256
Program blocks	maxim.	256
Funktion blocks (FB/FX)	maxim.	512 (can be assigned parameters)
Sequence blocks	maxim.	256
Data bocks (DB/DX)	maxim.	512
Operation set	approx.	270 operations
Required backup current from the backup battery at power off		
- internal RAM	approx.	100 µA
Current consuption (without interface module)		
- from 5 V (internal)	typically	0.5 A
- from 24 V		0.06 A
Power losses of the module (without interface module)	typically	4.2 W
Weight	approx.	0.8 kg (1.7 lb)

[1] If backup battery used

15.2.4 Digital Input Modules

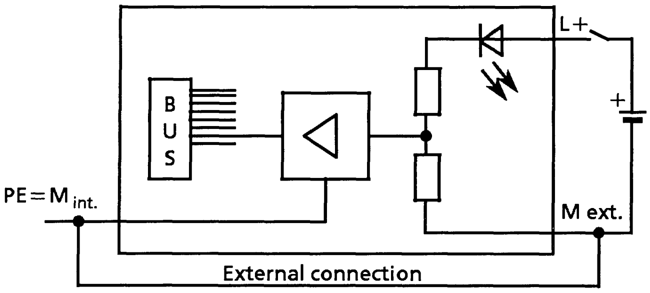
Digital Input Module 32 x 24 V DC, Nonfloating (6ES5 420-7LA11)



Terminal Assignment

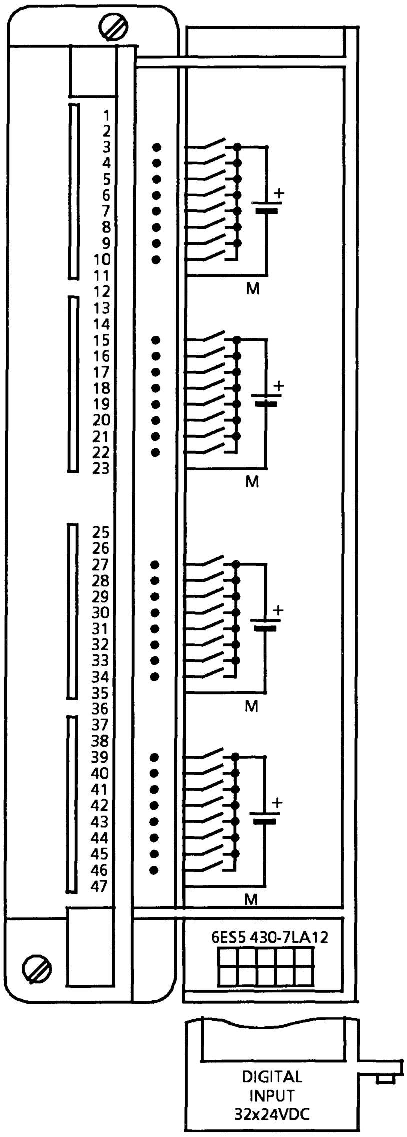
Technical Specifications

Number of inputs		32
Floating		no
Input voltage L +		
- rated value		24 V DC
- for "0" signal		- 30 to + 5 V
- for "1" signal		13 to 30 V
Input current		
- for "1" signal	typically	8.5 mA
Response time		
- from "0" to "1"		1.4 to 5 msec.
- from "1" to "0"		1.4 to 5 msec.
Cable length		
- shielded	maximum	1000 m
- unshielded	maximum	600 m
Isolation rating		according to VDE 0160
Connection of 2-wire BERO proximity switches		possible
- leakage (quiescent) current		≤1.5 mA
Current consumption		
- from 5 V (internal)		≤5 mA
Power losses of the module	typically	6.5 W
Weight	approx.	0.7 kg (1.54 lb.)



Simplified Schematic

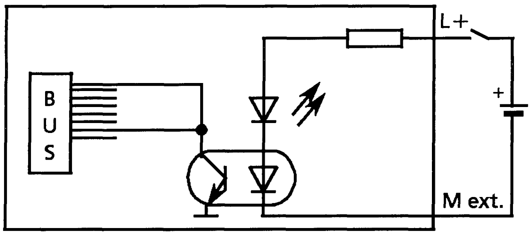
Digital Input Module 32 x 24 V DC, Floating (6ES5 430-7LA12)



Terminal Assignment

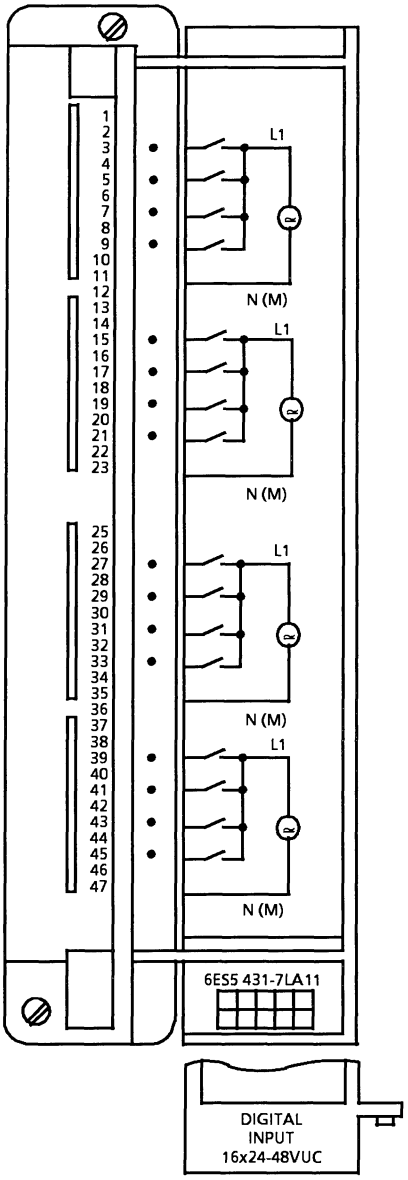
Technical Specifications

Number of inputs		32
Floating		yes (optocoupler)
- in groups of		8
Input voltage L +		
- rated value		24 V DC
- for "0" signal		- 30 to + 5V
- for "1" signal		13 to 30V
Response time		
- for "1" signal	typically	8.5 mA
Response time		
- from "0" to "1"	typically	2.6 msec.;
	maximum	5.5 msec.
- from "1" to "0"	typically	2.3 msec.;
	maximum	5 msec.
Cable length		
- shielded	maximum	1000 m (3281 ft.)
- unshielded	maximum	600 m (1969 ft.)
Isolation rating		according to VDE 0160
Rated isolation voltage		
(between groups)		30 V
- isolation group		C
- tested with		500 V
Rated isolation voltage		
(L + to ⏚)		30 V
- isolation group		C
- tested with		500 V
Connection of 2-wire		
BERO proximity switches		possible
- quiescent current		≤1.5 mA
Current consumption		
- from 5 V (internal)		≤5 mA
Power losses of		
the module	typically	6.5 W
Weight	approx.	0.7 kg (1.54 lb.)

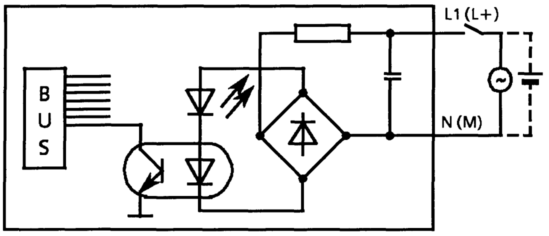


Simplified Schematic

Digital Input Module 16 x 24 to 48 V UC (6ES5 431-7LA11)



Terminal Assignment



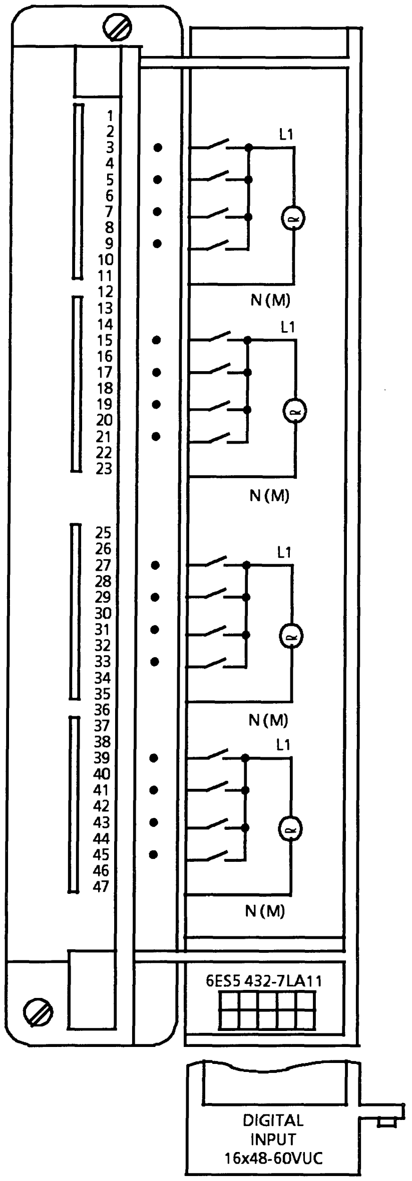
Simplified Schematic

Technical Specifications

Number of inputs		16
Floating		yes (optocoupler)
- in groups of		4
Input voltage L +		
- rated value		24 to 48 V UC
- frequency		0 to 63 Hz
- for "0" signal		0 to 5 V
- for "1" signal		13 to 60 V
Input current at "1" signal		
- for 24 V AC	typically	8.5 mA
- for 24 V DC	typically	9.0 mA
- for 48 V AC	typically	10.5 mA
- for 48 V DC	typically	10.5 mA
Response time		
- from "0" to "1"		2 to 13 msec.
- from "1" to "0"		10 to 25 msec.
Cable length		
- shielded	maximum	1000 m
- unshielded	maximum	600 m
Isolation rating		according to VDE 0160
Rated isolation voltage [1] (between groups)		60V
- isolation group		C
- tested with		500 V
Rated isolation voltage (L1 to ⏚)		250V
- isolation group		C
- tested with		1500 V
Connection of 2-wire BERO proximity switches		possible
- quiescent current		≤2 mA
Current consumption		
- from 5 V (internal)		≤5 mA
Power losses of the module	typically	9 W
Weight	approx.	0.7 kg (1.54 lb.)

[1] Connection of different phases is not permissible.

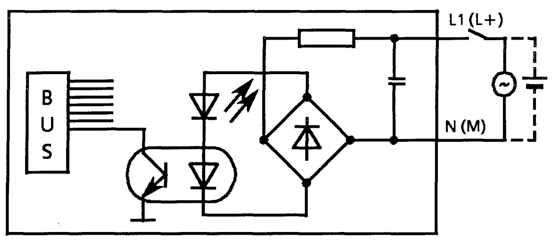
Digital Input Module 16 x 48 to 60 V UC, Floating (6ES5 432-7LA11)



Terminal Assignment

Technical Specifications

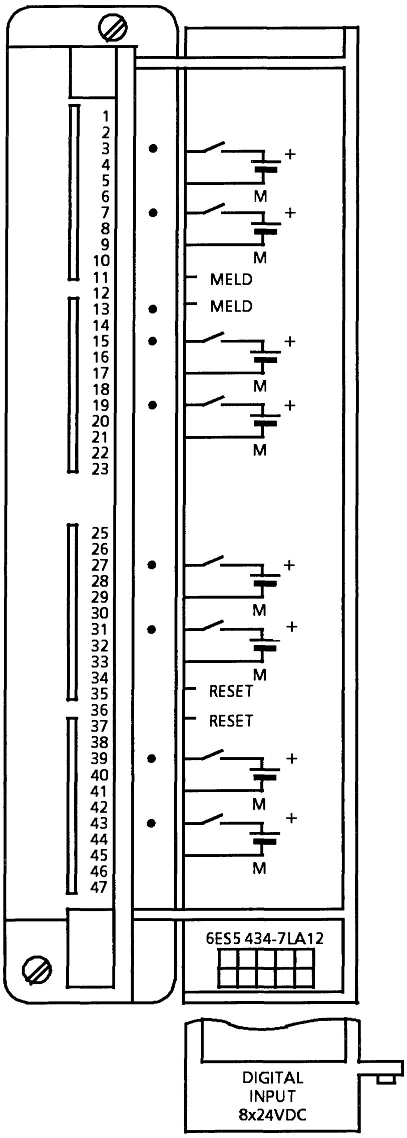
Number of inputs		16
Floating		yes (optocoupler)
- in groups of		4
Input voltage L1		
- rated value		48 to 60 V UC
- frequency		0 to 63 Hz
- for "0" signal		0 to 10 V
- for "1" signal		30 to 72 V
Input current at "1" signal		
- for 48 V AC /50 Hz	typically	8.5 mA
- for 48 V DC	typically	9.5 mA
- for 60 V AC /50 Hz	typically	9.5 mA
- for 60 V DC	typically	10.0 mA
Response time		
- from "0" to "1"		2 to 13 msec.
- from "1" to "0"		10 to 25 msec.
Cable length		
- shielded	maximum	1000 m (3281 ft.)
- unshielded	maximum	600 m (1969 ft.)
Isolation rating		according to VDE 0160
Rated isolation voltage[1] (between groups)		250 V
- isolation group		C
- tested with		1500 V
Rated isolation voltage (L1 to ⏚)		250 V
- isolation group		C
- tested with		1500 V
Connection of 2-wire BERO proximity switches		possible
- quiescent current		≤5 mA
Current consumption		
- from 5 V (internal)		≤5 mA
Power losses of the module	typically	10 W
Weight	approx.	0.7 kg (1.54 lb.)



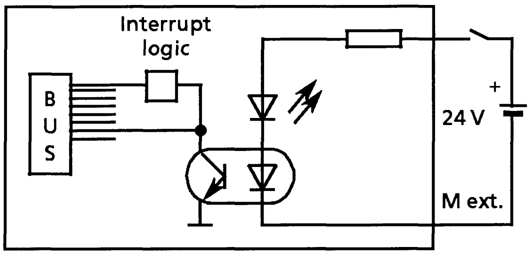
Simplified Schematic

[1] Connection of different phases is not permissible.

Digital Input Module 8 x 24 V DC (with P Interrupt), Floating (6ES5 434-7LA12)



Terminal Assignment

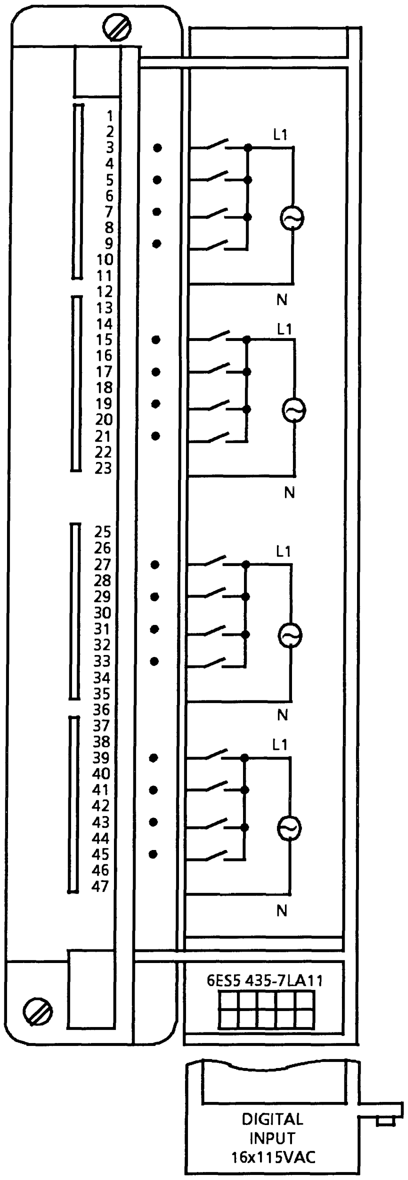


Simplified Schematic

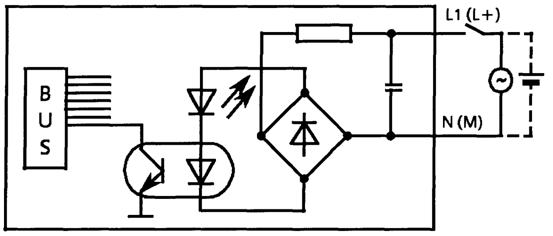
Technical Specifications

Number of inputs		8
Floating		yes (optocoupler)
- in groups of		1
Input voltage L +		
- rated value		24 V DC
- for "0" signal		- 30 to + 5 V
- for "1" signal		13 to 30 V
Input current at "1" signal		
- for 24 V DC	typically	8.5 mA
Response time		
- from "0" to "1"		0.5 to 1.5 msec.
- from "1" to "0"		0.5 to 1.5 msec.
Interrupt signal (external)		latching relay contact (permissible load: max. 0.2 A 50 V DC - switching capacity max. 20 W or 35 VA)
Interrupt signal (internal)		via bus line PRAL-N
Acknowledgement		external via input reset 24 V DC
Cable length		
- shielded	maximum	1000 m (3281 ft.)
- unshielded	maximum	600 m (1969 ft.)
Isolation rating		according to VDE 0160
Rated isolation voltage (between groups)		30V
- isolation group		C
- tested with		500 V
Rated isolation voltage (L + to ⏚)		30 V
- isolation group		C
- tested with		500 V
Connection of 2-wire BERO proximity switches		possible
- quiescent current	maximum	≤1.5 mA
Current Consumption		
- from 5 V (internal)		<70 mA
Power losses of the module	typically	2 W
Weight	approx.	0.7 kg (1.54 lb.)

Digital Input Module 16 x 115 V UC, Floating (6ES5 435-7LA11)



Terminal Assignment



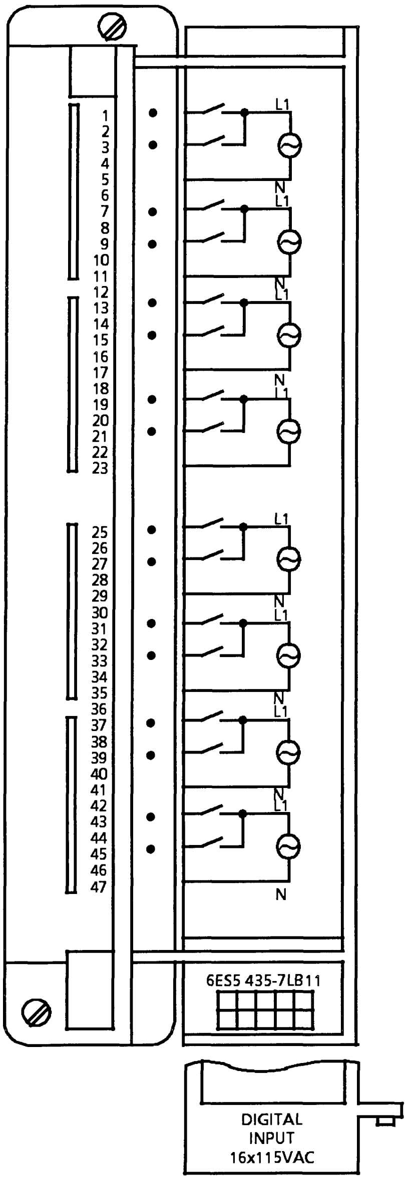
Simplified Schematic

Technical Specifications

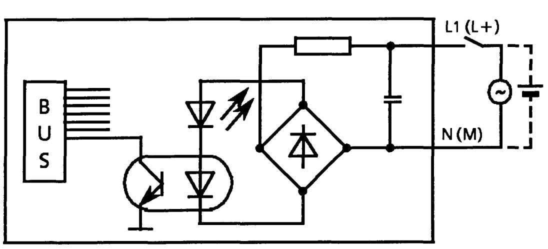
Number of inputs		16
Floating		yes (optocoupler)
- in groups of		4
Input voltage L1		
- rated value		115 V UC
- frequency		47 to 63 Hz
- for "0" signal		0 to 40 V
- for "1" signal		85 to 135 V
Input current at "1" signal		
- for AC, 50 Hz	typically	15 mA
- for DC	typically	6 mA
Response time		
- from "0" to "1"		2 to 13 msec.
- from "1" to "0"		10 to 25 msec.
Cable length		
- shielded		1000 m (3281 ft.)
- unshielded		600 m (1969 ft.)
Isolation rating		according to VDE 0160
Rated isolation voltage [1] (between groups)		250 V
- isolation group		C
- tested with		1500 V
Rated isolation voltage (L1 to ⏚)		250 V
- isolation group		C
- tested with		1500 V
Connection of 2-wire BERO proximity switches		possible
- quiescent current		≤5 mA
Current consumption		
- from 5 V (internal)		≤5 mA
Simultaneity factor (per group, L1 = 135 V)		
- at 25 °C		100 %
- at 55 °C		75 %
Power losses of the module	typically	11 W
Weight	approx.	0.7 kg (1.54 lb.)

[1] Connection of different phases is permissible.

Digital Input Module 16 x 115 V UC (6ES5 435-7LB11)



Terminal Assignment



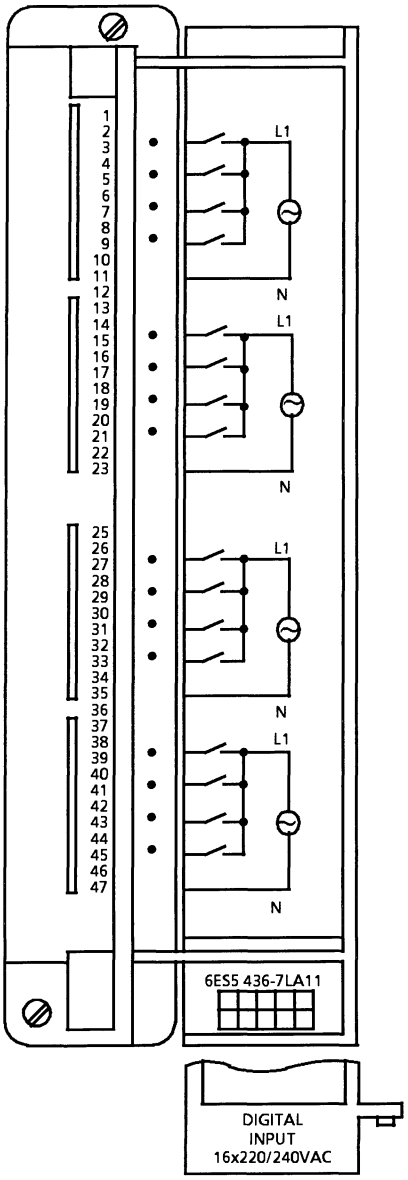
Simplified Schematic

Technical Specifications

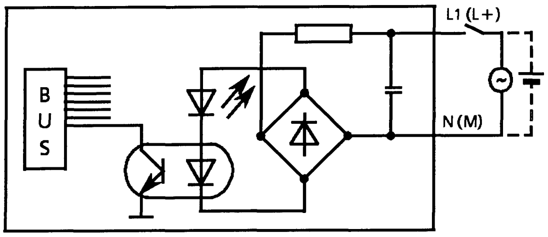
Number of inputs		16
Floating		yes (optocoupler)
- in goups of		2
Input voltage L1		
- rated value		115 V UC
- frequency		47 to 63 Hz
- for "0" signal		0 to 40 V
- for "1" signal		85 to 135 V
Input current at "1" signal		
- for AC, 50 Hz	typically	10 mA
- for DC	typically	6 mA
Response time		
- from "0" to "1"		2 to 13 msec.
- from "1" to "0"		10 to 25 msec.
Cable length		
- shielded		1000 m (3281 ft.)
- unshielded		600 m (1969 ft.)
Isolation rating		according to VDE 0160
Rated isolation voltage [1] (between groups)		250 V
- isolation group		C
- tested with		1500 V
Rated isolation voltage (L1 to ⏚)		250 V
- isolation group		C
- tested with		1500 V
Connection of 2-wire BERO proximity switches		possible
- quiescent current		≤5mA
Current consumption		
- from 5 V (internal)		≤5mA
Simultaneity factor (per group, L1 = 135 V)		
- at 25 °C		100 %
- at 55 °C		75 %
Power losses of the module	typically	11 W
Weight	approx.	0.7 kg (1.54 lb.)

[1] Connection of different phases is permissible.

Digital Input Module 16 x 230 V UC, Floating (6ES5 436-7LA11)



Terminal Assignment



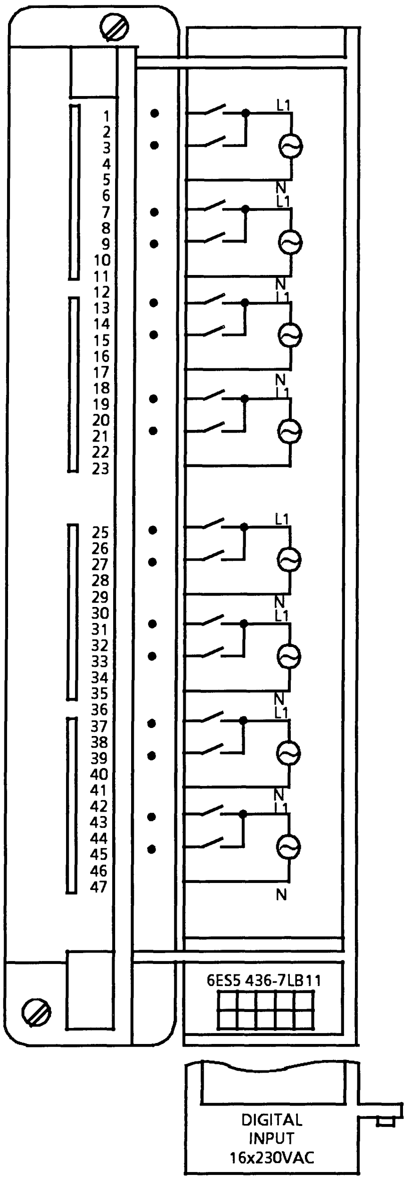
Simplified Schematic

Technical Specifications

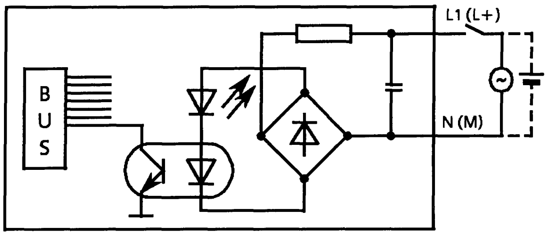
Number of inputs		**16**
Floating		**yes (optocoupler)**
- in groups of		**4**
Input voltage L1		
- rated value		**230 V UC**
- frequency		**47 to 63 Hz**
- for "0" signal		**0 to 70 V**
- for "1" signal		**170 to 264 V**
Input current		
at "1" signal		
- for AC, 50 Hz	**typically**	**15 mA**
- for DC	**typically**	**2.2 mA**
Response time		
- from "0" to "1"		**2 to 13 msec.**
- from "1" to "0"		**10 to 35 msec.**
Cable length		
- shielded		**1000 m (3281 ft.)**
- unshielded		**600 m (1969 ft.)**
Isolation rating		**according to VDE 0160**
Rated isolation voltage [1]		
(between groups)		**250 V**
- isolation group		**C**
- tested with		**1500 V AC**
Rated isolation voltage		
(L1 to ⏚)		**250 V**
- isolation group		**C**
- tested with		**1500 V AC**
Connection of 2-wire		
BERO proximity switches		**possible**
- quiescent current		**≤3 mA**
Current consumption		
- from 5 V (internal)		**≤5 mA**
Simultaneity factor		
(per group, for L1 = 264 V)		
- at 25 °C		**100 %**
- at 55 °C		**75 %**
Power losses of		
the module	**typically**	**11 W**
Weight	**approx.**	**0.7 kg (1.54 lb.)**

[1] Connection of different phases is not permissible.

Digital Input Module 16 x 230 V UC (6ES5 436-7LB11)



Terminal Assignment



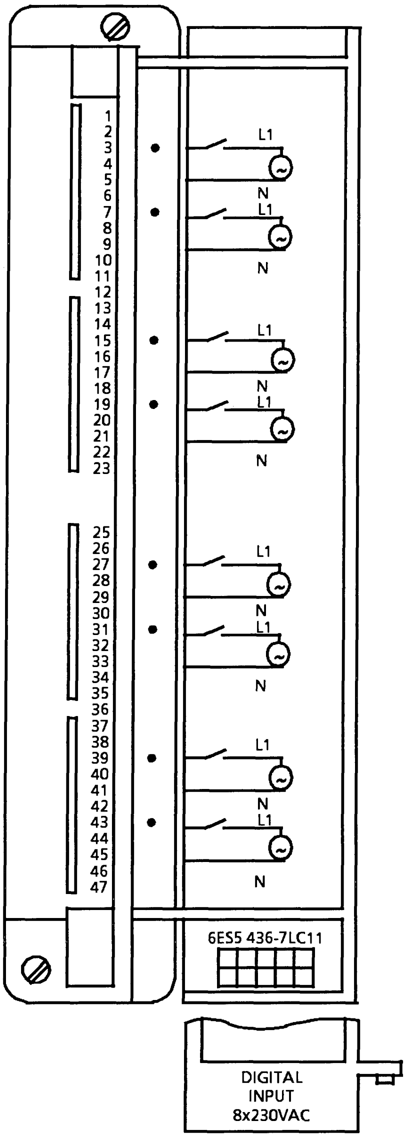
Simplified Schematic

Technical Specifications

Number of inputs		16
Floating		yes (optocoupler)
- in groups of		2
Input voltage L1		
- rated value		230 V UC
- frequency		47 to 63 Hz
- for "0" signal		0 to 70 V
- for "1" signal		170 to 264 V
Input current at "1" signal		
- for AC, 50 Hz	typically	15mA
- for DC	typically	2.2 mA
Response time		
- from "0" to "1"		2 to 13 msec.
- from "1" to "0"		10 to 35 msec.
Cable length		
- shielded		1000 m (3281 ft.)
- unshielded		600 m (1969 ft.)
Isolation rating		according to VDE 0160
Rated isolation voltage [1] (between groups)		250 V
- isolation group		C
- tested with		1500 V
Rated isolation voltage (L1 to ⏚)		250 V
- isolation group		C
- tested with		1500 V
Connection of 2-wire BERO proximity switches		possible
- quiescent current		≤3 mA
Current consumption		
- from 5 V (internal)		≤5mA
Simultaneity factor (per group, for L1 = 264 V)		
- at 25 °C		100 %
- at 55 °C		75 %
Power losses of the module	typically	11 W
Weight	approx.	0.7 kg (1.54 lb.)

[1] Connection of different phases is not permissible.

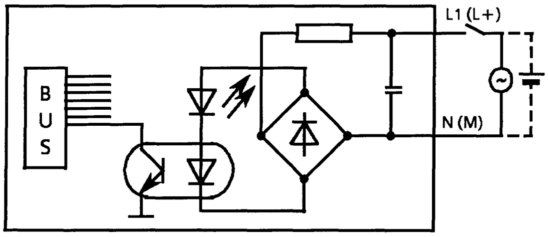
Digital Input Module 8 x 230 V UC (6ES5 436-7LC11)



Terminal Assignment

Technical Specifications

Number of inputs		8
Floating		yes (optocoupler)
- in groups of		1
Input voltage L1		
- rated value		230 V UC
- frequency		47 to 63 Hz
- for "0" signal		0 to 100 V
- for "1" signal		170 to 264 V
Input current at		
"1" signal AC	typically	16 mA
DC	typically	2.2 mA
Response time		
- from "0" to "1"		2 to 13 msec.
- from "1" to "0"		10 to 25 msec.
Cable length		
- shielded		1000 m (3281 ft.)
- unshielded		600 m (1969 ft.)
Isolation rating		according to VDE 0160
Rated isolation voltage[1]		
(between groups)		250 V
- isolation group		C
- tested with		2700 V
Rated isolation voltage		
(L1 to ⏚)		250 V
- isolation group		C
- tested with		2700 V
Connection of 2-wire		
BERO proximity switches		possible
- quiescent current		≤5 mA
Current consumption		
- from 5 V (internal)		≤5 mA
Power losses of		
the module	typically	5 W
Weight	approx.	0.7 kg (1.54 lb.)

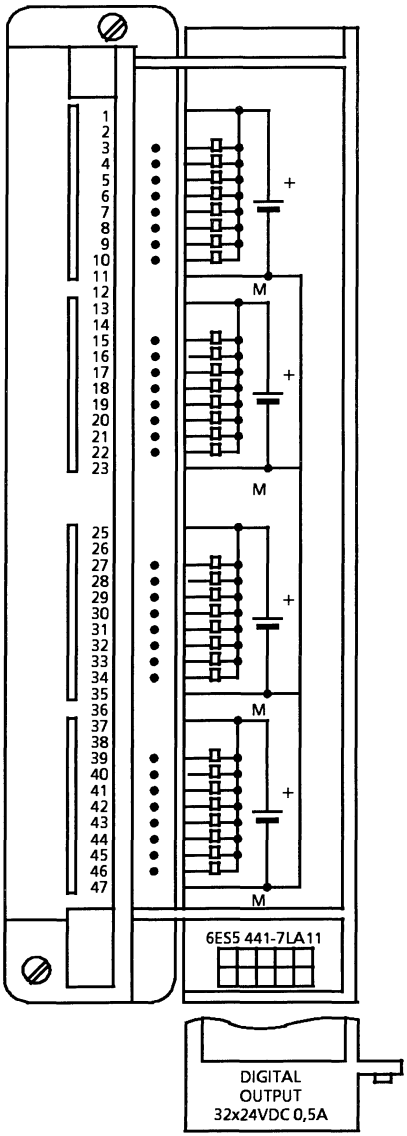


Simplified Schematic

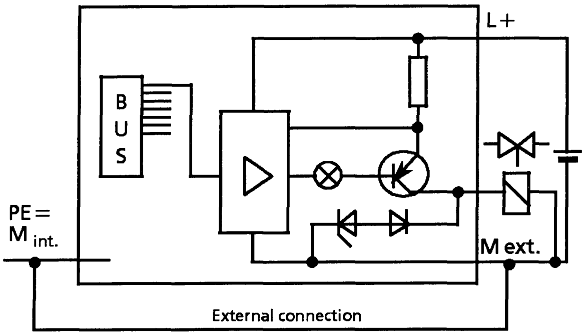
[1] Connection of different phases is not permissible.

15.2.5 Digital Output Modules

Digital Output Module 32 x 24 V DC; 0.5 A, Nonfloating (6ES5 441-7LA11)



Terminal Assignment

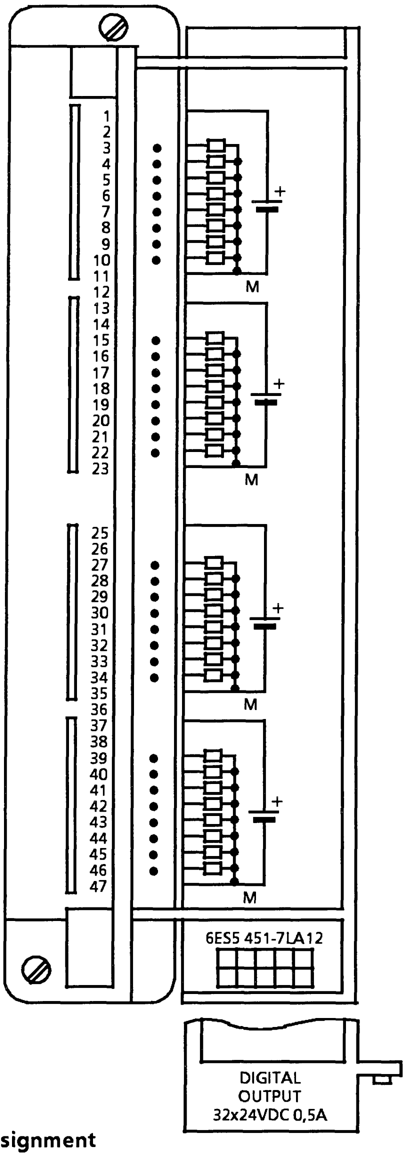


Simplified Schematic

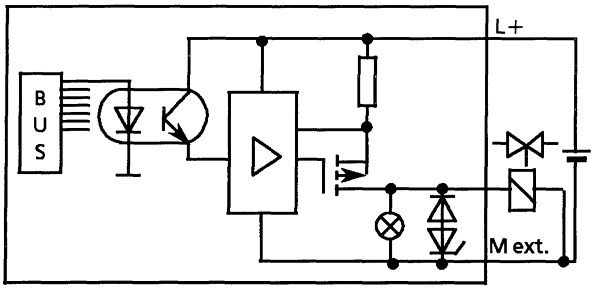
Technical Specifications

Number of outputs		32
Floating		no
Load voltage L+		
- rated value		24 V DC
- permissible range		20 to 30 V
- surge voltage at t ≤0.5 sec.		35 V
Output voltage		
- at "1" signal	minimum	L+ - 2.5 V
Output current at "1" signal		
- rated value		0.5 A
- lamp load	maximum	5 W
Leakage current at "0" signal	maximum	1 mA
Parallel connection of outputs		not possible
Permissible total current of outputs		100 % at 25 °C and 50 % at 55 °C (related to the sum of currents)
Short circuit protection		electronic
Limitation of the voltage induced on circuit interruption		- 15 V
Switching frequency		
- inductive load	maximum	0.5 Hz
- resistive load	maximum	100 Hz
Cable length		
- shielded	maximum	1000 m (3281 ft.)
- unshielded	maximum	600 m (1969 ft.)
Isolation rating		according to VDE 0160
Current consumption		
- from 5 V (internal)		10 mA
- from L+ (without load)		17 mA/per group
Power losses of the module	typically	20 W
Weight	approx.	0.7 kg (1.54 lb.)

Digital Output Modules 32 x 24 V DC; 0.5 A, Floating (6ES5 451-7LA12)



Terminal Assignment

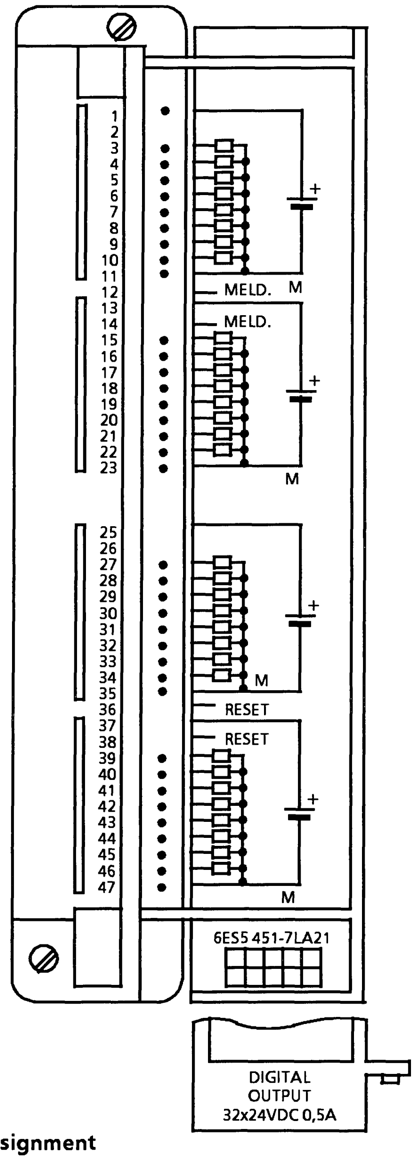


Simplified Schematic

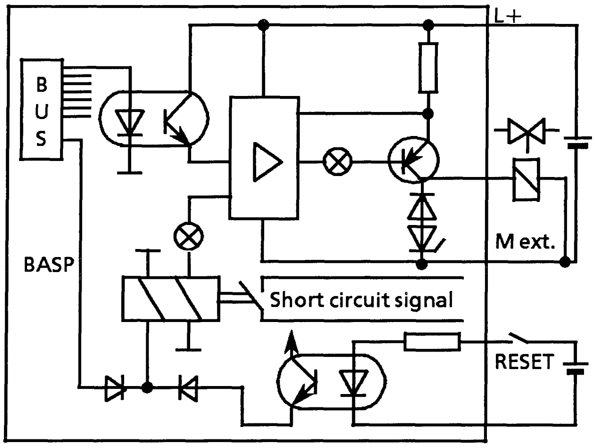
Technical Specifications

Number of outputs		32
Floating		yes (optocoupler)
- in groups of		8
Load voltage L +		
- rated value		24 V DC
- permissible range		20 to 30 V
- surge voltage at t ≤0.5 sec.		35 V
Output voltage		
- at "1" signal	minimum	L + - 2.5 V
Output current for "1" signal		
- rated value		0.5 A
- lamp load	maximum	5 W
Leakage current at "0" signal	maximum	1 mA
Parallel connection of outputs		not possible
Permissible total current of outputs		100 % at 25 °C and 50 % at 55 °C (related to the sum of currents)
Short circuit protection		electronic
Limitation of the voltage induced on circuit interruption		- 15 V
Switching frequency		
- inductive load	maximum	0.5 Hz
- resistive load	maximum	100 Hz
Cable length		
- shielded		1000 m (3281 ft.)
- unshielded		600 m (1969 ft.)
Isolation rating		according to VDE 0160
Rated isolation voltage (between groups)		30 V DC
- isolation group		C
- tested with		500 V AC
Rated isolation voltage (L + to ⏚)		30 V DC
- isolation group		C
- tested with		500 V AC
Current consumption		
- from 5 V (internal)		100 mA
- from L + (without load)		17 mA/per group
Power losses of the module	typically	20 W
Weight	approx.	0.7 kg (1.54 lb.)

Digital Output Module 32 x 24 V DC; 0.5 A, Floating (6ES5 451-7LA21)



Terminal Assignment



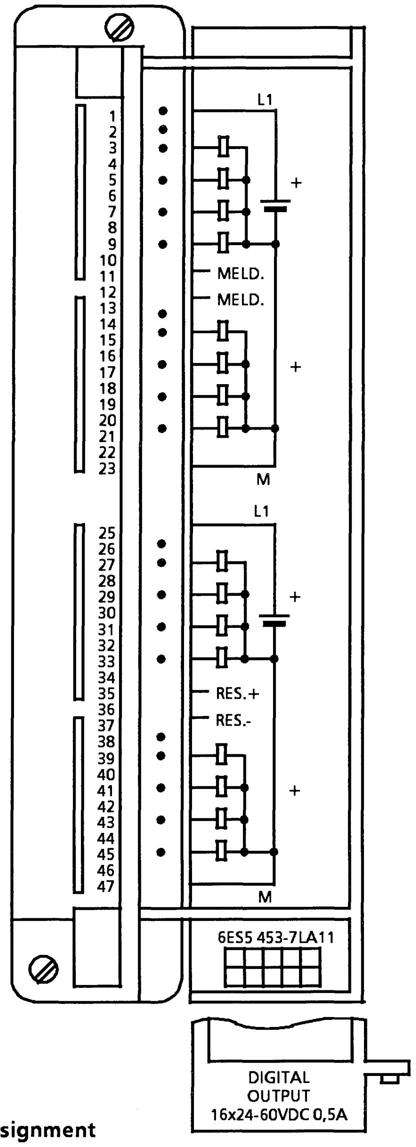
Simplified Schematic

Technical Specifications

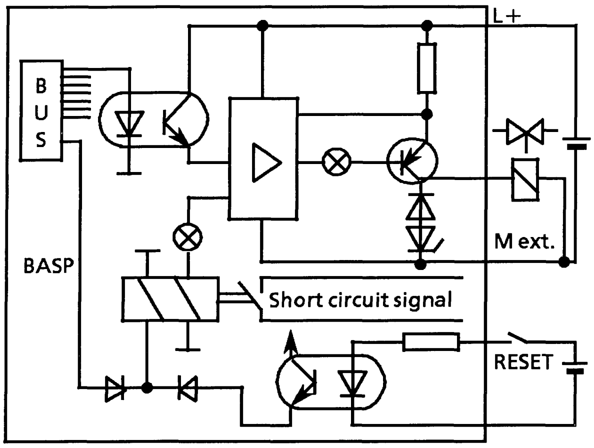
Number of outputs		32
Floating		yes (optocoupler)
- in groups of		8
Load voltage L +		
- rated value		24 V DC
- permissible range		20 to 30 V
- surge voltage at t ≤0.5 sec.		35 V
Output voltage		
- for "1" signal	minimum	L+ - 2.5 V
Output current for "1" signal		
- rated value		0.5 A
- lamp load	maximum	5 W
Leakage current at "0" signal	maximum	1 mA
Parallel connection of outputs		not possible
Permissible total current of outputs		100 % at 25 °C and 50 % at 55 °C (related to the sum of currents)
Short circuit protection		electronic
Short circuit indicator		red LED (per group)
Short circuit signal (latching relay contact)		latching[1]
Type of relay		V23042 B201 B101
- load		30V DC; 0.1 A
- switching capacity		3 W or 6 VA
- reset input		24V DC
Limitation of the voltage induced on circuit interruption		- 15 V
Switching frequency		
- inductive load	maximum	0.5 Hz
- resistive load	maximum	100 Hz
Cable length		
- shielded		1000 m (3281 ft.)
- unshielded		600 m (1969 ft.)
Isolation rating		according to VDE 0160
Rated isolation voltage (between groups)		30 V DC
- isolation group		C
- tested with		500 V AC
Rated isolation voltage (L + to ⏚)		30 V DC
- isolation group		C
- tested with		500 V AC
Current consumption		
- from 5 V (internal)		100 mA
- from L + (without load)		17 mA/per group
Power losses of the module	typically	20 W
Weight	approx.	0.7 kg (1.54 lb.)

[1] Pickup delay: approx. 1 sec. after start of short circuit

Digital Output Module 16 x 24 to 60 V DC; 0.5 A, Floating (6ES5 453-7LA11)



Terminal Assignment



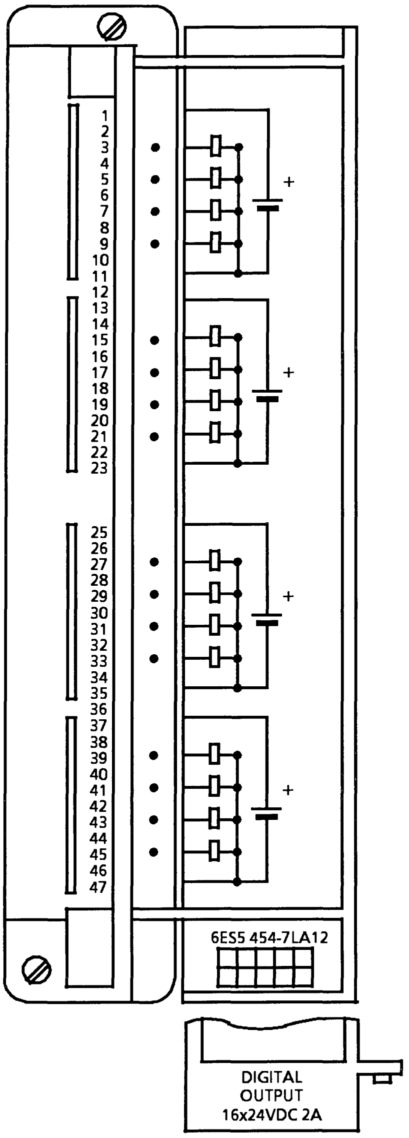
Simplified Schematic

Technical Specifications

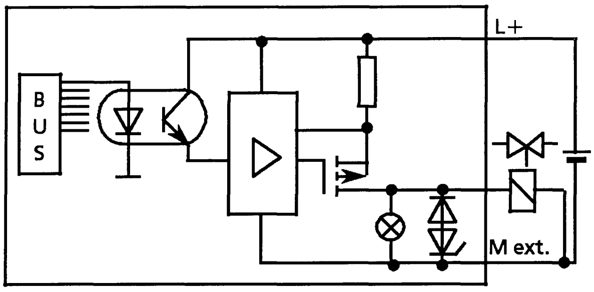
Number of outputs		16
Floating		yes (optocoupler)
- in groups of		8
Load voltage L +		
- rated value		24 to 60 V DC
- permissible range		20 to 75 V
- surge voltage at t ≤0.5 sec.		87 V
Ouput voltage		
- for "1" signal	maximum	L + - 2.5 V
Output current for "1" signal		
- rated value		0.5 A
- lamp load	maximum	5 W
Leakage current at "0" signal	maximum	1 mA
Parallel connection of outputs		not possible
Permissible total current of outputs, per group		100 %
Short circuit protection		electronic
Short circuit indicator		red LED (each group)
Short circuit signal (latching relay contact)		latching [1]
Type of relay		V23042 B201 B101
- load		60 V DC; 0.2 A
- switching capacity		20 W or 35 VA
- reset input		24 to 60 V DC
Limitation of the voltage induced on circuit interruption		- 30 V
Switching frequency		
- inductive load	maximum	0.5 Hz
- resistive load	maximum	100 Hz
Cable length		
- shielded	maximum	1000 m (3281 ft.)
- unshielded	maximum	600 m (1969 ft.)
Isolation rating		according to VDE 0160
Rated isolation voltage (between groups)		75 V DC
- isolation group		C
- tested with		500 V AC
Rated isolation voltage (L + to ⏚)		75 V DC
- isolation group		C
- tested with		500 V AC
Current consumption		
- from + 5 V (internal)		50 mA
- from L + (without load)		50 mA/per group
Power losses of the module	typically	14 W
Weight	approx.	0.7 kg (1.54 lb.)

[1] Pickup delay: approx. 1 sec. after start of short circuit

Digital Output Module 16 x 24 V DC; 2 A, Floating (6ES5 454-7LA12)



Terminal Assignment

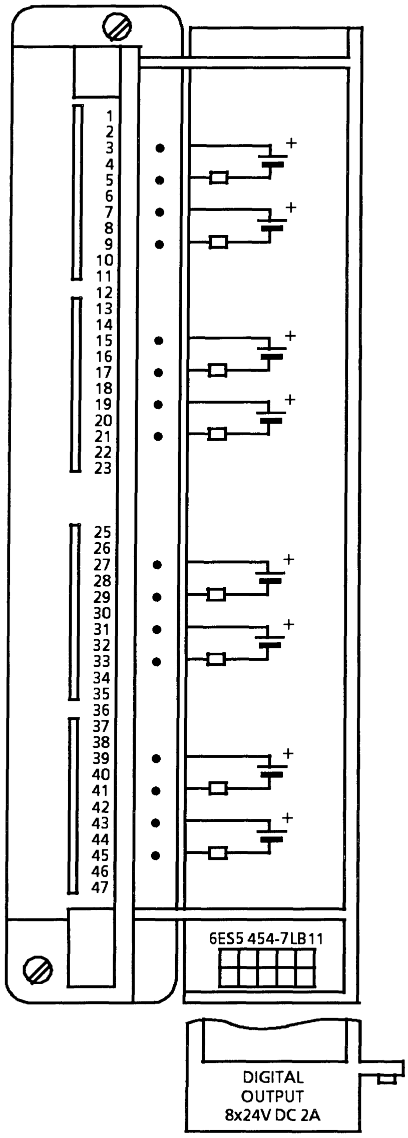


Simplified Schematic

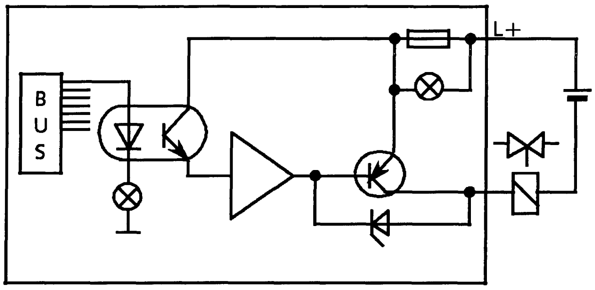
Technical Specifications

Number of outputs		16
Floating		yes (optocoupler)
- in groups of		4
Load voltage L+		
- rated value		24 V DC
- permissible range		20 to 30 V
- surge voltage at t ≤0.5 sec.		35 V
Output voltage		
- for "1" signal	minimum	L+ - 3 V
Output current at"1" signal		
- rated value		2 A
- lamp load	maximum	10 W
Leakage current at "0" signal	maximum	1 mA
Parallel connection of outputs		not possible
Permissible total current of outputs, per group		50 % (related to the sum of the currents of a group)
Short circuit protection		electronic
Limitation of the voltage induced on circuit interruption		- 15 V
Switching frequency		
- inductive load	maximum	0.27 Hz
- resistive load	maximum	100 Hz
Cable length		
- shielded	maximum	1000 m (3281 ft.)
- unshielded	maximum	600 m (1969 ft.)
Isolation rating		according to VDE 0160
Rated isolation voltage (between groups)		30 V DC
- isolation group		C
- tested with		500 V AC
Rated isolation voltage (L+ to ⏚)		30 V DC
- isolation group		C
- tested with		500 V AC
Current consumption		
- from 5 V (internal)		50 mA
- from L+ (without load)		8.5 mA/per group
Power losses of the module	typically	20 W
Weight	approx.	1.1 kg (2.43 lb.)

Digital Output Module 8 x 24 V DC; 2 A, Floating (6ES5 454-7LB11)



Terminal Assignment

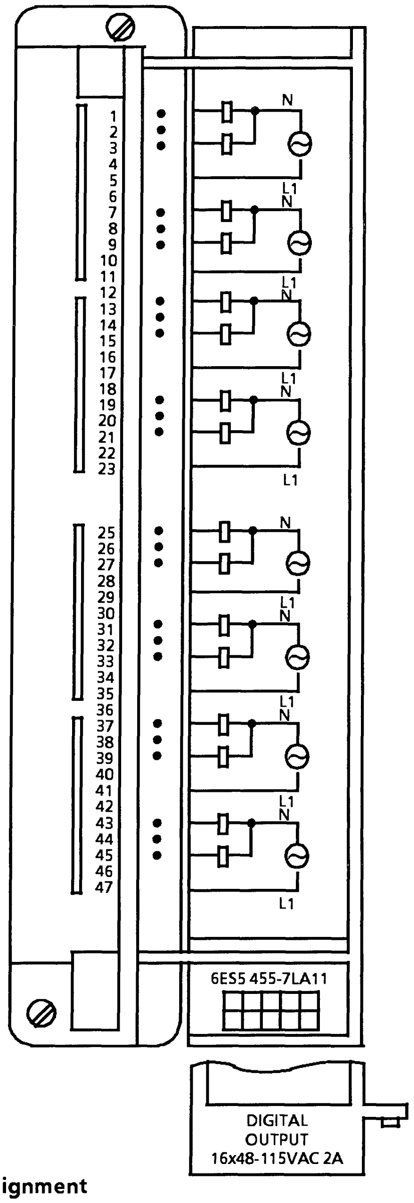


Simplified Schematic

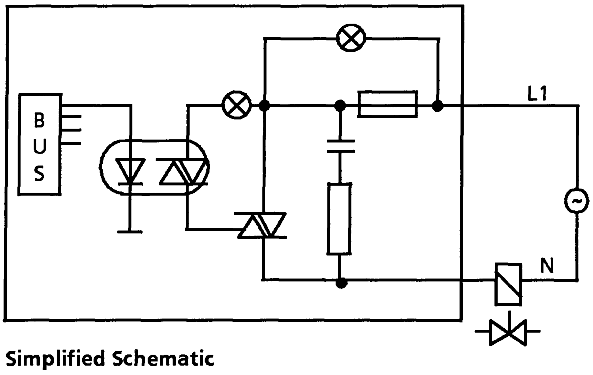
Technical Specifications

Number of outputs		8
Floating		yes (optocoupler)
- in groups of		1
Load voltage L +		
- rated value		24 V DC
- permissible range		20 to 30 V
- surge voltage at t ≤0.5 sec.		35 V
Output voltage		
- for "1" signal	minimum	L + - 3 V
Output current		
- rated value		2 A
- lamp load	maximum	10 W
Leakage current at "0" signal	maximum	1 mA
Parallel connection of outputs		possible
- maximum current		1 x rated current
Permissible total current of outputs, per group		100 % at 25 °C and 50 % at 55 °C (related to the sum of the currents of a group)
Short circuit protection (for each group)		with F 2.5 A fuse (e.g. Wickmann 19340)
Limitation of the voltage induced on circuit interruption	typically	- 23 V
Switching frequency		
- inductive load	maximum	0.27 Hz
- resistive load	maximum	100 Hz
Cable length		
- shielded	maximum	1000 m (3281 ft.)
- unshielded	maximum	600 m (1969 ft.)
Isolation rating		according to VDE 0160
Rated isolation voltage (between groups)		30 V DC
- isolation group		C
- tested with		500 V AC
Rated isolation voltage (L + to ⏚)		30 V DC
- isolation group		C
- tested with		500 V AC
Current consumption		
- from + 5 V (internal)	maximum	50 mA
Power losses of the module	typically	20 W
Weight	approx.	0.8 kg (1.76 lb.)

Digital Output Module 16 x 48 to 115 V AC;2 A, Floating (6ES5 455-7LA11)



Terminal Assignment

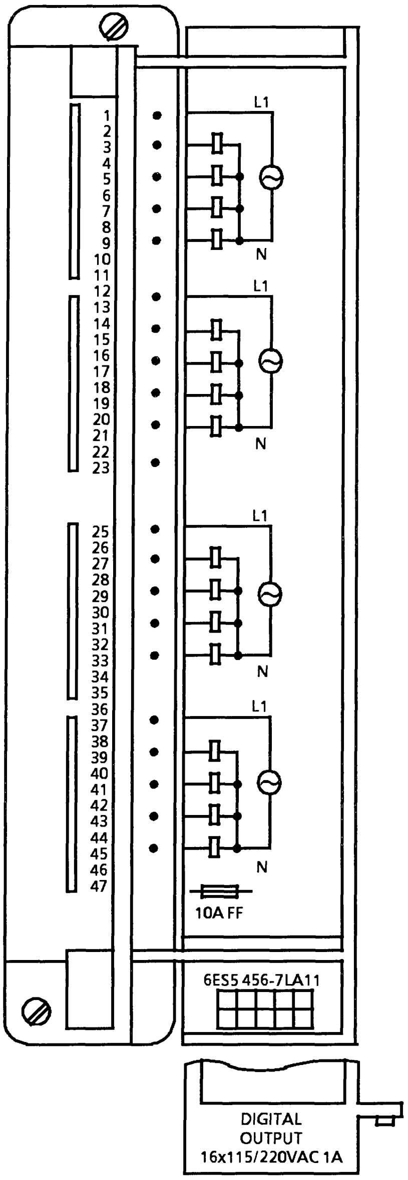


Simplified Schematic

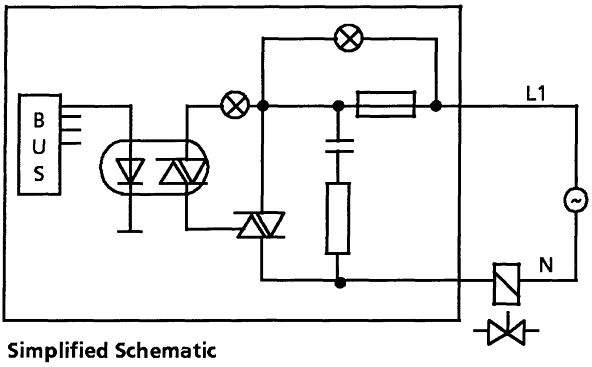
Technical Specifications

Number of outputs		16
Floating		yes (optocoupler)
- in groups of		2
Load voltage L1		
- rated value		48/115 V AC
- frequency		47 to 63 Hz
- permissible range		40 to 140 V
Output voltage		
- for "1" signal	minimum	L1 - 7 V
Output current at "1" signal		
- rated value		2 A/per group
- permissible range		40 mA to 2 A
- lamp load	maximum	50/100 W/per group
Leakage current at "0" signal	maximum	1/3 mA
Parallel connection of outputs		not possible
Making capacity		depends on the size of the fuse
Permissible total current of outputs		100 %
Short circuit protection (for each group)		with Gould GAB4 fuse or Bussmann ABC4
Fault indication (red LED for each group)		defective fuse
Switching frequency	maximum	10 Hz
Cable length		
- shielded	maximum	1000 m (3281 ft.)
- unshielded	maximum	300 m (984 ft.)
Isolation rating		according to VDE 0160
Rated isolation voltage (between groups)		250 V AC
- isolation group		C
- tested with		1500 V AC
Rated isolation voltage (L1 to ⏚)		250 V AC
- isolation group		C
- tested with		1500 V AC
Current consumption		
- from 5 V (internal)	maximum	175 mA
Power losses of the module	typically	16 W
Weight	approx.	1.1 kg (2.43 lb.)

Digital Output Module 16 x 115 to 230 V AC; 1 A, Floating (6ES5 456-7LA11)



Terminal Assignment



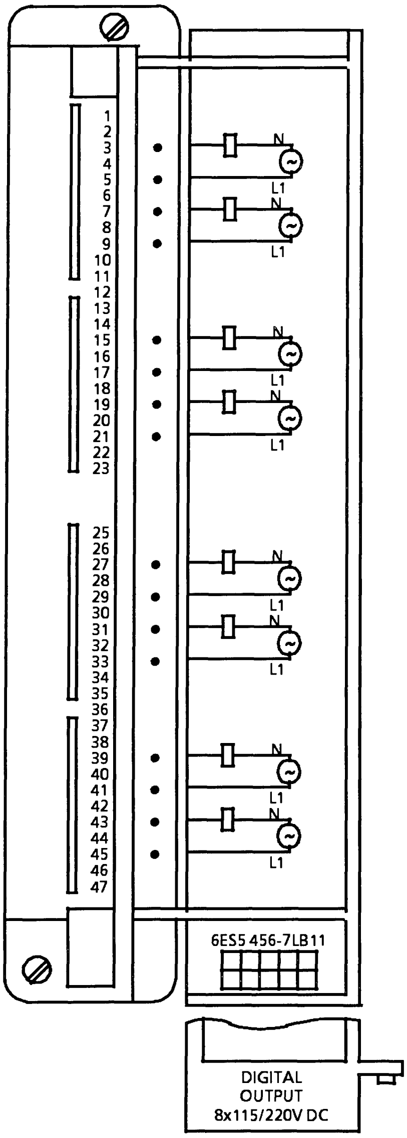
Simplified Schematic

Technical Specifications

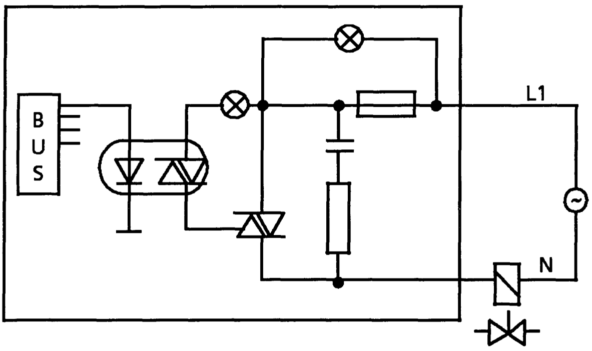
Number of outputs		16
Floating		yes (optocoupler)
- in groups of		4
Load voltage L1		
- rated value		115/230 V AC
- frequency		47 to 63 Hz
- permissible range		89 to 264 V
Output voltage		
- at "1" signal	minimum	L1 - 7 V
Output current at "1" signal		
- rated value		1 A
- permissible range		40 mA to 1 A
- lamp load		25/50 W
Leakage current at "0" signal	typically	3/5 mA [1]
Parallel connection of outputs		not possible
Making capacity		depends on the size of the fuse
Permissible total current of outputs		100 %
Short circuit protection (for each group)		fuse (10 A FF) (e.g. Wickmann 19231)
Fault indication (red LED for each group)		defective fuse
Switching frequency	maximum	10 Hz
Cable length		
- shielded		1000 m (3281 ft.)
- unshielded		300 m (984 ft.)
Isolation rating		according to VDE 0160
Rated isolation voltage (between groups)		250 V AC
- isolation groups		C
- tested with		1500 V AC
Rated isolation voltage (L1 to ⏚)		250 V AC
- isolation groups		C
- tested with		1500 V AC
Current consumption		
- from 5 V (internal)	maximum	70 mA
Power losses of the module	typically	16 W
Weight	approx.	1.1 kg (2.43 lb.)

[1] Please note the max. dropout capacity of the load connected (contactors of the 3TJ1 to 3TJ5 series and contactors of the SIMICOMT series cannot be triggered)!

Digital Output Module 8 x 115 to 230 V AC; 2 A (6ES5 456-7LB11)



Terminal Assignment



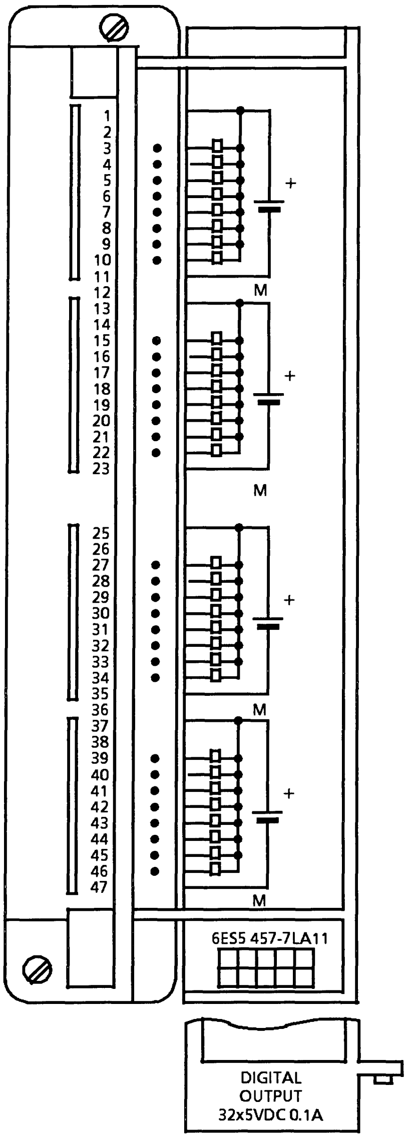
Simplified Schematic

Technical Specifications

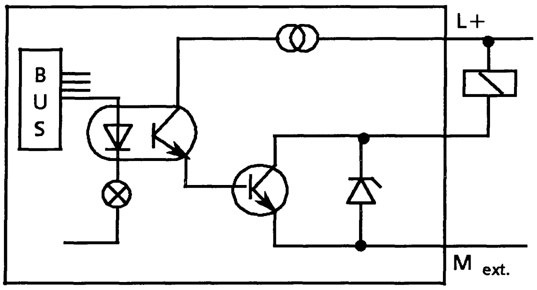
Number of outputs		8
Floating		yes (optocoupler)
- in groups of		1
Load voltage L1		
- rated value		115 to 230 V AC
- frequency		47 to 63 Hz
- permissible range		89 to 264 V
Output voltage		
- at "1" signal	minimum	L1 - 7 V
Output current at "1" signal		
- rated value		2 A
- permissible range		40 mA to 2 A
- lamp load		25/50 W
Output current at "0" signal	typically	3 to 5 mA [1]
Parallel connection of outputs		not possible
Making capacity		depends on the size of the fuse
Permissible total current of outputs		100 %
Short circuit protection (for each group)		fuse (6.3 A FF) (e.g. Wickmann 19231)
Fault indication (red LED for each group)		defective fuse
Switching frequency	maximum	10 Hz
Cable length		
- shielded		1000 m
- unshielded		300 m
Isolation rating		according to VDE 0160
Rated isolation voltage (between groups)		250 V AC
- isolation group		C
- tested with		2700 V AC
Rated isolation voltage (L1 to ⏚)		250 V AC
- isolation group		C
- tested with		2700 V AC
Current consumption		
- from 5 V (internal)	maximum	35 mA
Power losses of the module	typically	16 W
Weight	approx.	1.1 kg (2.43 lb.)

[1] Please note the max. dropout capacity of the load connected (contactors of the 3TJ1 to 3TJ5 series and contactors of the SIMICOMT series cannot be triggered)!

Digital Output Module 32 x 5 to 24 V DC; 0.1 A, Floating (6ES5 457-7LA11)



Terminal Assignment



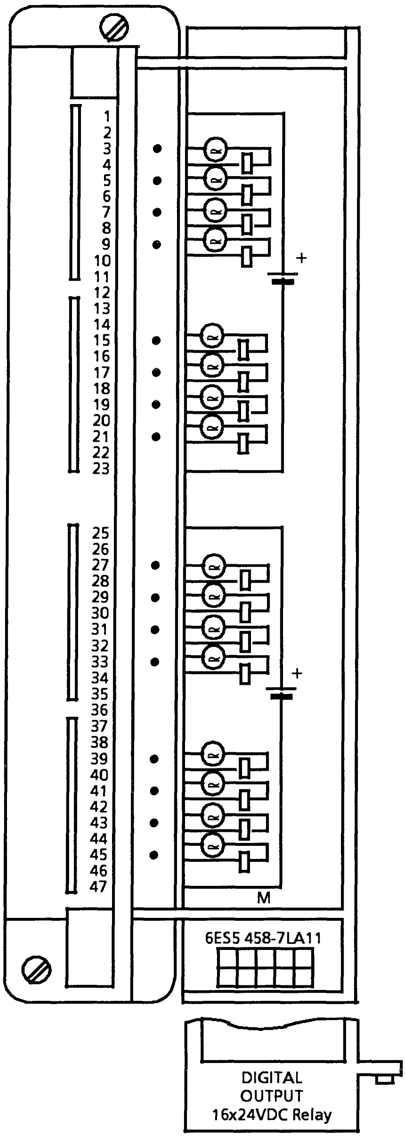
Simplified Schematic

Technical Specifications

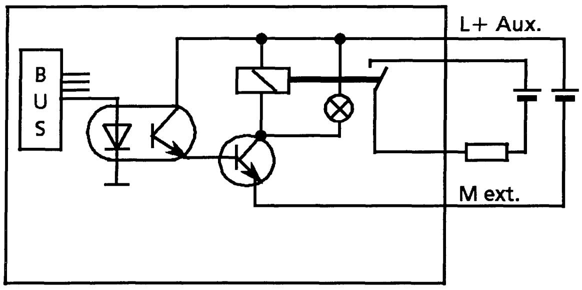
Number of outputs		32
Floating		yes (optocoupler)
- in groups of		8
Load voltage L1		
- rated value		5 to 24 V DC
- permissible range		4.75 to 30 V
Output voltage [1]		TTL compatible
Output current for "1" signal	maximum	100 mA
Parallel connection of outputs		possible
Permissible total current of outputs		100 %
Short circuit protection		not available
Limitation of the voltage induced on circuit interruption (for $V_p = 30$ V)		- 10 V
Switching frequency		
- inductive load		2 Hz
- resistive load		10 Hz
Cable length		
- shielded		1000 m (3281 ft.)
- unshielded		300 m (984 ft.)
Isolation rating		according to VDE 0160
Rated isolation voltage (between groups)		30 V
- isolation group		C
- tested with		500 V
Rated isolation voltage (5 V to ⏚)		30 V
- isolation group		C
- tested with		500 V
Current consumption		
- from 5 V (internal)	maximum	100 mA
- from L1 (without load)	maximum	4 mA
Power losses of the module	typically	6 W
Weight	approx.	0.7 kg (1.54 lb.)

[1] Transistor with open collector - current sinking

Relay Output Module for Measuring Currents 16 x 24 V DC (6ES5 458-7LA11)



Terminal Assignment

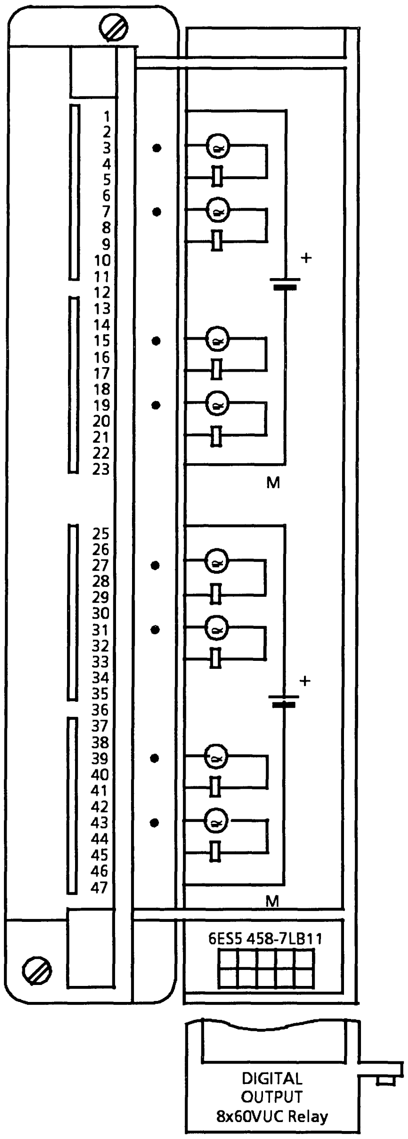


Simplified Schematic

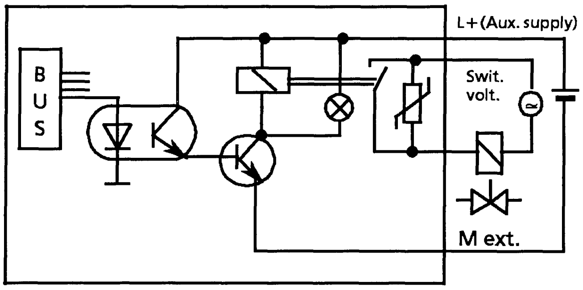
Technical Specifications

Number of outputs		**16**
- contact bridge		**no**
- galvanic isolation		**yes**
- in groups of		**1**
- relay type		**3700-2501-011 (Günther)**
Limiting continuous current per contact		**0.5 A**
Parallel connection of outputs		**possible**
Permissible total current of outputs		**100 %**
Switching frequency		
- resistive load	**maximum**	**60 Hz**
- inductive load		**not permissible**
Switching voltage	**maximum**	**30 V DC**
Switching capacity of the contacts		
- resistive load		**10 W at 0.5 A;**
- inductive load		**not permissible**
Number of contact operating cycles according to VDE 0660, Part 200		
- DC 11		**1 x 10^9**
Supply voltage L + (for relay)		
- rated value		**24 V DC**
- permissible range		**20 to 30 V**
- surge voltage at t<0.5 sec.		**35 V**
- ripple	**maximum**	**3.6 V**
Cable length		
- shielded		**1000 m (3281 ft.)**
- unshielded		**300 m (984 ft.)**
Isolation rating		**according to VDE 0160**
Rated isolation voltage (between contacts)		**30 V DC**
- isolation group		**C**
- tested with		**500 V AC**
Rated isolation voltage (contacts to L +)		**30 V DC**
- isolation group		**C**
- tested with		**500 V AC**
Rated isolation voltage (contacts to ⏚)		**30 V DC**
- isolation group		**C**
- tested with		**500 V AC**
Current consumption		
- from 5 V (internal)	**maximum**	**50 mA**
- fromL + (for relay)		**240 mA**
Power losses of the module	**typically**	**5W**
Weight	**approx.**	**0.8 kg (1.76 lb.)**

Relay Output Module 8 x 30 V DC/230 V AC (6ES5 458-7LB11)



Terminal Assignment

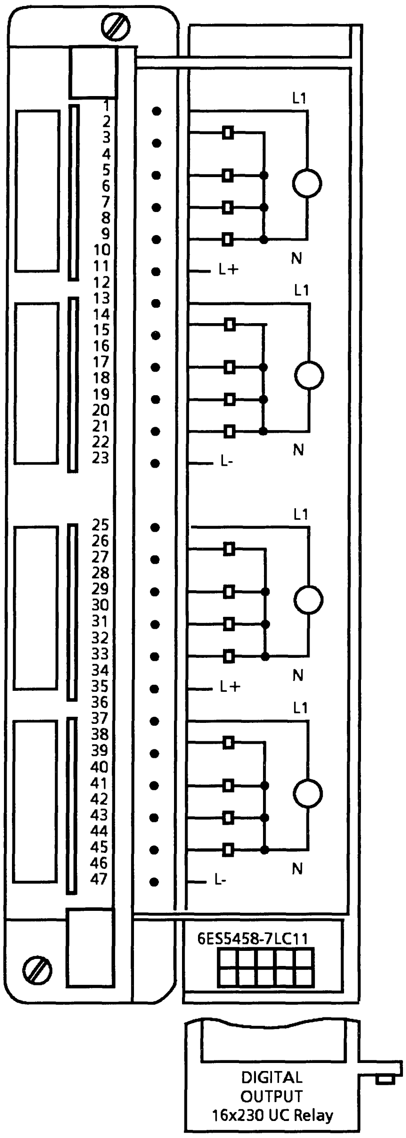


Simplified Schematic

Technical Specifications

Number of outputs		8
- contact bridge		varistor SIOV-S07-K275
- floating		yes
- in groups of		1
- relay type		V23057-A006-A402 (Siemens)
Limiting continuous current per contact		5 A
Parallel connection of outputs		possible
Permissible total current of outputs		100 %
Switching capacity of the contacts		
- resistive load		5 A at 250 V AC 2.5 A at 30 V DC
- inductive load		1.5 A at 250 V AC 0.5 A at 30 V DC
Switching frequency	maximum	10 Hz
Number of contact operating cycles according to VDE 0660, Part 200		
- AC 11		1.5×10^6
- DC 11		0.5×10^6
Supply voltage L+ (for relay)		
- rated value		24 V DC
- permissible range		20 to 30 V
- surge voltage at t	0.5 sec.	35 V
- ripple	maximum	3.6 V
Isolation rating		according to VDE 0160
Rated isolation voltage (between contacts)		250 V AC
- isolation group		C
- tested with		1500 V AC
Rated isolation voltage (between contacts L+)		250 V AC
- isolation group		C
- tested with		1500 V AC
Rated isolation voltaqe (between contacts ⏚)		250 V AC
- isolation group		C
- tested with		1500 V AC
Current consumption		
- from 5 V (internal)	maximum	50 mA
- from L+ (for relay)		200 mA
Power losses of the module	typically	4 W
Weight	approx.	0.8 kg (1.76 lb.)

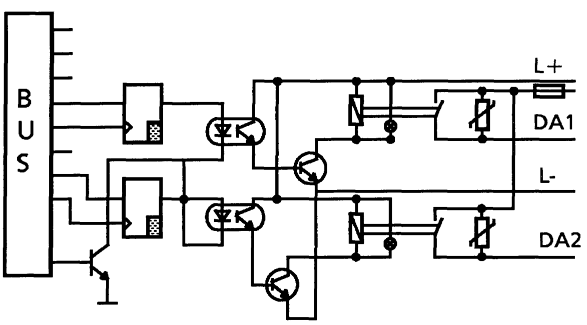
Relay Output Module 16 x 230 V UC (6ES5 458-7LC11)



Technical Specifications

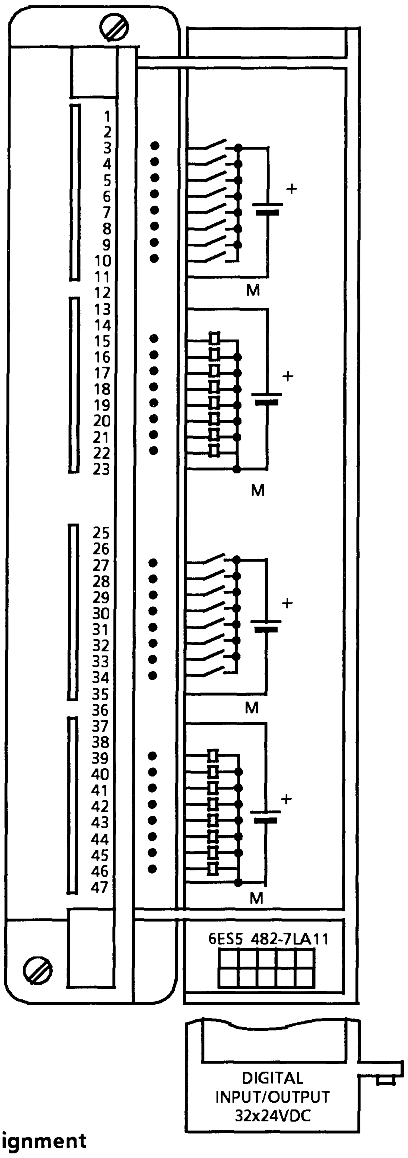
Number of outputs		16
- contact bridge		varistor SIOV-S07-K275
- galvanic isolation		yes
- in groups of		4 outputs
- relay type		MSR V23061-B1007-A401
Switching capacity of the contacts		
- resistive load		5.0 A at 250 V AC 5.0 A at 30 V DC 0.4 A at 110 V DC
- inductive load		1.5 A at 250 V AC 1.0 A at 30 V DC 0.08 A at 110 V DC
Switching frequency		
- resistive load	maximum	10 Hz
- inductive load		2 Hz
Current load of the relay contacts		
- one relay per root		5 A/contact
- two relays " "		4 A/contact
- three " " "		2.5 A/contact
- four " " "		2 A/contact
Number of contact operating cycles according to VDE 0660, Part 200		
- DC 11		1.5 x 10^6 (AC) 0.5 x 10^6 (DC)
Supply voltage L+/L- (for relay)		
- rated value		24 V DC
- permissible range		20 to 30 V
- surge voltage at t≤0.5 sec.		35 V
- ripple	maximum	3.6 V
Cable length		
- shielded		1000 m (3281 ft.)
- unshielded		300 m (984 ft.)
Isolation rating		according to VDE 0160
Rated isolation voltage (contacts to L+)		250 V AC
- isolation group		C
- tested with		1500 V AC
Rated isolation voltage (contacts to ⏚)		250 V AC
- isolation group		C
- tested with		1500 V AC
Current consumption		
- from 5 V (internal)		2 mA (+4 mA per active channel)
- from L+ (for relay)		16 mA per active channel
Power losses of the module	typically	6.5 W
Weight	approx.	0.8 kg (1.76 lb.)

Terminal assignment

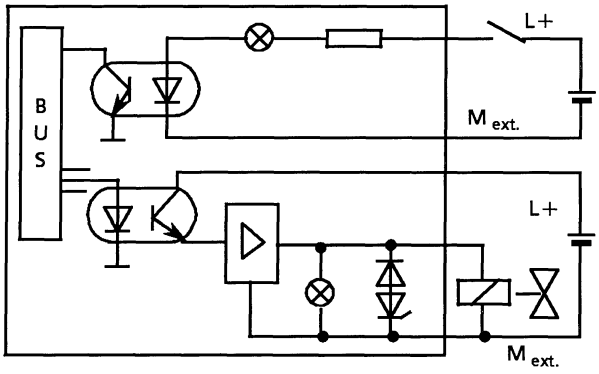


15.2.6 Digital Input/Output Module

Digital Input/Output Module 32 x 24 V DC; 0.5 A (6ES5 482-7LA11)



Terminal Assignment



Simplified Schematic

Technical Specifications

Number of inputs		16
Floating		yes (optocoupler)
- in groups of		8
Input voltage		
- rated value		24 V DC

The technical specifications for the inputs correspond to those of the 6ES5 430-7LA11 digital input module.

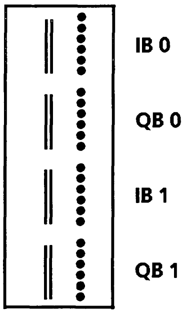
Number of outputs		16
Floating		yes (optocoupler)
- in groups of		8
Output current at "1" signal		
- rated value		0.5 A

The technical specifications for the outputs correspond to those of the 6ES5 451-7LA11 digital output module.

Outputs	0 to 3 and 4 to 7 8 to 11 and 12 to 15	can be connected in parallel
Parallel current		≤0.8 x I $_{rated}$
Permissible current of outputs		100 % at 35 °C and 50 % at 55 °C (related to the sum of the currents of a group)
Current consumption		
- from 5 V (internal)	maximum	50 mA
Power loss	typically	18 W
Weight	approx.	0.7 kg (1.54 lb.)
Setting on IM 306		16-channel

The inputs and outputs are referenced under the same address (e.g. I 0.0 to I 1.7 and Q 0.0 to Q 1.7).

Example: Addressing must be as follows for module start address "0":



15.2.7 Analog Input Modules

Analog Input Module 8 x I/V/PT 100, Floating (6ES5 460-7LA12)

Technical Specifications

Number of inputs		8 voltage/current inputs or 8 inputs for PT 100
Floating		yes (not for PT 100)
Input ranges (rated values)		±50 mV; ±500 mV; PT 100; ±1 V; ±5 V; ±10 V; ±20 mA; +4 to 20 mA (can be selected for four channels at a time using range cards)
Input resistance		±50 mV: ≥10 MΩ ±500 mV: ≥10 MΩ PT 100: ≥10 MΩ ±1 V: 90 kΩ; 2 ‰ ±5 V: 50 kΩ; 2 ‰ ±10 V: 50 kΩ; 2 ‰ ±20 mA: 25 Ω; 1 ‰ ±4 to 20 mA:31.25 Ω; 1 ‰
Type of connection for sensors		Two-wire connection; four-wire connection for PT 100
Digital representation of the input signal		12 bits plus sign or 13 bits two's complement (2048 units = rated value)
Measuring principle		integrating
Conversion principle		voltage-time conversion (dual-slope)
Integration time (adjustable for optimum noise suppression)		20 msec. at 50 Hz 16.6 msec. at 60 Hz
Coding time (Single coding for 2048 units)	maximum	60 msec. at 50 Hz 50 msec. at 60 Hz
Cycle time for - 8 inputs		0.48 sec. at 50 Hz
Permissible voltage between inputs and between inputs and central grounding point (destruction limit)	maximum	18 V or 75 V for max. 1 msec. and a duty cycle 1 : 20
Permissible voltage between the reference potential of a nonfloating sensor and the central grounding point	maximum	75 V DC /60 V AC

Error indication for - overranging		yes (exceeding 4095 units)
- wire break of the sensor line		can be designed for the ranges 50 mV, 500 mV and PT 100 (only measuring leads)
Wire-break test current (disconnectable)		configurable
Noise suppression for f = n x (50/60 Hz ± 1 %) n = 1, 2, ...		
- common mode noise (V_s < 1 V)	min.	100 dB
- series mode noise (peak noise value < rated value of the range)	min.	40 dB
Basic error limits		±50 mV : ±2 ‰ ±500 mV : ±1.5 ‰ PT 100 : ±2 ‰ ±1 V : ±3.5 ‰ ±5 V : ±3.5 ‰ ±10 V : ±3.5 ‰ ±20 mA : ±2.5 ‰ +4 to 20 mA : ±2.5 ‰
Operational error limits (0°C to 55°C)		±50 mV : ±5 ‰ ±500 mV : ±4.5 ‰ PT 100 : ±5 ‰ ±1 V : ±7.7 ‰ ±5 V : ±7.7 ‰ ±10 V : ±7.7 ‰ ±20 mA : ±6.7 ‰ +4 to 20 mA : ±6.7 ‰
Cable length - shielded	maximum	200 m (656 ft.); 50 m for ± 50 mV
Front connector		46 pins
Isolation rating		according to VDE 0160
Rated isolation voltage (channel to channel) - tested with		500 V
Rated isolation voltage (channel to ⏚) - tested with		500 V
Current consumption - rated value		24 V DC
- ripple V_{PP}		3.6 V
- permissible range (including ripple)		20 to 30 V
Current consumption - from 5 V (internal)	typically	0.15 A
- from 24 V (external)	typically	0.1 A
Power losses of the module	typically	3 W
Weight	approx.	0.4 kg (0.88 lb.)

Analog Input Module 8 x I/V/PT 100, Floating (6ES5 460-7LA13)

Technical Specifications

Number of inputs		8 voltage/current inputs or 8 inputs for PT 100
Floating		yes
Input ranges (rated values)		± 50 mV; ± 500 mV; PT 100; ± 1 V; ± 5 V; ± 10 V; ± 20 mA; + 4 to 20 mA (can be selected for four channels at a time using range cards)
Input resistance		± 50 mV: ≥10 MΩ ± 500 mV: ≥10 MΩ PT 100: ≥10 MΩ ± 1 V: 90 kΩ; 2 ‰ ± 5 V: 50 kΩ; 2 ‰ ± 10 V: 50 kΩ; 2 ‰ ± 20 mA: 25 Ω; 1 ‰ ± 4 to 20 mA: 31.25 Ω; 1 ‰
Type of connection for sensors		Two-wire connection; four-wire connection for PT 100
Digital representation of the input signal		12 bits + sign or 13 bits two's complement (2048 units= rated value)
Measuring principle		integrating
Conversion principle		Voltage-time-conversion
Integration time (adjustable for optimum noise suppression)		20 ms at 50 Hz 16.6 ms at 60 Hz
Coding time (Single coding for 2048 units)	max.	60 ms at 50 Hz 50 ms at 60 Hz
Cycle time for - 8 inputs		0.48 s at 50 Hz
Permissible voltage between inputs and central grounding point (destruction limit)	max.	18 V or 75 V for max. 1 ms and a duty cycle of 1 : 20
Permissible voltage	max.	75 V DC/60 V AC
Error indication for		
- overranging		yes (exceeding 4095 units)
- wire break of sensor line		Configurable in the range 50 mV, 500 mV and PT 100 (only measuring leads)

Disconnectable wirebreak test current		Configurable	
Noise suppression for f=n x (50/60 Hz ± 1%) n=1, 2, ...			
- Common mode noise (V_s<1 V)	min.	100 dB	
- Series mode noise (peak noise value < rated value of range)	min.	40 dB	
Basic error limits		± 50 mV :	± 2‰
		± 500 mV :	± 1.5 ‰
		PT 100 :	± 2‰
		± 1 V :	± 3,. ‰
		± 5 V :	± 3.5 ‰
		± 10 V :	± 3.5 ‰
		± 20 mA :	± 2.5 ‰
		+ 4 to 20 mA:	± 2.5 ‰
Operational error limits (0°C to 55°C)		± 50 mV :	± 5 ‰
		± 500 mV :	± 4.5 ‰
		PT 100 :	± 5 ‰
		± 1 V :	± 7.7 ‰
		± 5 V :	± 7.7 ‰
		± 10 V :	± 7.7 ‰
		± 20 mA :	± 6.7 ‰
		+ 4 to 20 mA:	± 6.7 ‰
PT100 climatic measuring range			
- Measuring range incl. override range	approx.	-100°C to +100°C	
- Resolution		0.025 °C per unit	
- Basic error at T= 25°C	max.	+/- 0.2 °C	
- Operational error limit at T=0 to 65 °C	max.	+/- 0.5 °C	
- Range card		6ES5 498-1AAA11	
Cable length			
- shielded	max.	200 m; 50 m at ± 50 mV	
Front connector		46-pole	
Isolation rating		to VDE 0160	
Rated isolation voltage (channel to ⏚)			
- tested with		500 V	
Current consumption			
- Rated value		24 V DC	
- Ripple Vpp		3.6 V	
- Permissible range (including ripple)		20 to30 V	
Current consumption			
- from 5 V (internal)	typ.	70 mA	
- from 24 V (external)	typ.	50 mA	
Power losses of the module	typ.	3 W	
Weight	approx.	0.7 kg (1.54 lb)	

Analog Input Module 4 x I/V, Floating (6ES5 463-4UA11/-4UA12/-4UB11/-4UB12)

Technical Specifications	
Rated input ranges (adjustable at front connector)	-0,05 to +1V -0,5 to +10V -1 to +20 mA +4 to 20 mA for 2-wire transducer +4 to 20 mA for 4-wire transducer
Number of inputs	4 voltage/ current inputs
Measuring value representation	11 bits (two's complement) 1024 units in the rated range
Override range	100% (at full accuracy)
Error message at overflow	yes (from 50 % override onwards)
Measuring principle	integrating
Conversion principle	Voltage-frequency conversion
Floating	yes; 4 inputs against M and against one another
Permissible potential difference between the reference potentials of sensors and the module (V_{CM}) and between the sensors (channels)	max. 25 V AC / 60 V DC
Current consumption	
- Digital section of system bus	5V± 5%; typ. 150 mA
- Analog section of load voltage	24V; approx. 150 mA
- Enabling of module F+/F-	24V; approx. 7 mA
Integration time	
- 463-4UA11/-4UA12	20 ms at 50 Hz
- 463-4UB11/-4UB12	16 $^2/_3$ ms at 60 Hz
Coding time per measuring value	
- 463-4UA11/-4UA12	20 ms at 50 Hz
- 463-4UB11/-4UB12	16 $^2/_3$ ms at 60 Hz
Cycle time for 4 measuring values (max. delay for measuring value acquisition)	
- 463-4UA11/-4UA12	20 ms at 50 Hz
- 463-4UB11/-4UB12	16 $^2/_3$ ms at 60 Hz

Input resistance for input range	
- 1 V	>10 MΩ
- 10 V	90 kΩ
- 20 mA	50 Ω
- 4 to 20 mA	62.5 Ω
Max. permissible input voltage without causing destruction	± 30V; 75V for max. 1 ms and a duty cycle of 1:10
Noise suppression	
- Common mode noise	> 80 dB (f=0 to 50 kHz)
- Series mode noise	> 40 dB ($V_S \leq 0.1 \times U_N$)
Error referred to rated value	
- Linearity	± 5×10^{-4}
- Tolerance	± 1 unit
- Temperature error	0.6 x 10^{-4}/K
Isolation voltage to VDE 0160	Inputs against earthing point and input against input; tested at 500 V AC
Surge withstand capability tested to IEC 266-4	Inputs against L-: V_S=1 kV; 1.2/50 µs

Analog Input Module 16 x I/V or 8 x PT 100, Nonfloating (6ES5 465-7LA13)

Technical Specifications

Number of inputs		16 voltage/current inputs or 8 inputs for PT 100
Floating		no
Input ranges (rated values)		±50 mV; ±500 mV; PT 100; ±1 V; ±5 V; ±10 V; ±20 mA; +4 to 20 mA (can be selected for four channels at a time using range cards)
Input resistance		±50 mV: ≥10 MΩ ±500 mV: ≥10 MΩ PT 100: ≥10 MΩ ±1 V: 90 kΩ; 2 ‰ ±5 V: 50 kΩ; 2 ‰ ±10 V: 50 kΩ; 2 ‰ ±20 mA: 25 Ω; 1 ‰ ±4 to 20 mA:31.25 Ω; 1‰
Type of connection for sensors		Two-wire connection; four-wire connection for PT 100
Digital representation of the input signal		12 bits + sign or 13 bits two's complement (2048 units = rated value)
Measuring principle		integrating
Conversion principle		voltage-time conversion (dual-slope)
Integration time (adjustable for optimum noise suppression)		20 msec. at 50 Hz 16.6 msec. at 60 Hz
Coding time (single coding for 2048 units)	maximum	60 msec. at 50 Hz 50 msec. at 60 Hz
Coding time - 8 inputs - 16 inputs		 0.48 sec. at 50 Hz 0.96 sec. at 50 Hz
Permissible voltage between inputs and between inputs and central grounding point (destruction limit)	maximum	18 V or 75 V for max. 1 msec. and a duty cycle of 1 : 20
Permissible voltage between the reference potential of a nonfloating sensor and the central grounding point	maximum	±1 V
Error indication for - overranging - wire break of the sensor line		 yes (exceeding 4095 units) can be designed for the ranges 50 mV, 500 mV (PT 100)
Wire break test current disconnectable		configurable

Noise suppression for f = n x (50/60 Hz ± 1%) n = 1, 2, ... - common mode noise (V_p < 1 V) - series mode noise (peak noise value < rated value of the range)	 min. min.	 86 dB 40 dB
Basic error limits		±50 mV : ±2 ‰ ±500 mV : ±1.5 ‰ PT 100 : ±2‰ ±1 V : ±3.5 ‰ ±5 V : ±3.5 ‰ ±10 V : ±3.5 ‰ ±20 mA : ±2.5 ‰ +4 to 20 mA : ±2.5 ‰
Operational error limits (0°C to 55°C)		±50 mV : ±5 ‰ ±500 mV : ±4,5 ‰ PT 100 : ±5 ‰ ±1 V : ±7.7 ‰ ±5 V : ±7.7 ‰ ±10 V : ±7.7 ‰ ±20 mA : ±6.7 ‰ +to 20 mA : ±6.7 ‰
Cable length - shielded	 maximum	 200 m; 50 m for ± 50 mV
Front connector		46 pin
Power supply - rated value - ripple V_{PP} - permissible range (including ripple)		 24 V DC [1] 3.6 V 20 to 30 V
Current consumption - from 5 V (internal) - from 24 V	 typically maximum	 0.15 A 20 mA/transducer
Power losses of the module	typically	0.75 W
Weight	approx.	0.4 kg (0.88 lb.)

[1] only required for two-wire transducers or for disconnecting the wire break test current

Analog Input Module 16 x I/V or 8 x I/V, Floating (6ES5 466-3LA11)

Technical Specifications

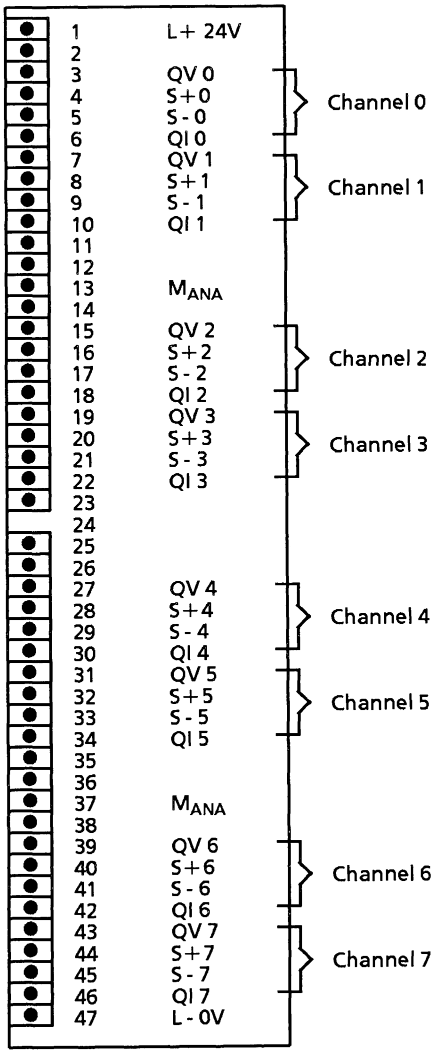
Number of inputs	16 individual or 8 differential inputs in groups of 4 or 2 channels (switchable) voltage measurement or current measurement
Floating	yes
Input ranges	0 to 20 mA, 4 to 20 mA, ±20 mA, 0 to 1.25 V, 0 to 2.5 V, 0 to 5 V, 1 to 5 V, 0 to 10 V, ±1.25 V, ±2.5 V, ±5 V, ±10 V
Input resistance	
- Voltage measuring range	≥10 MΩ
- Current measuring range	125 Ω
Type of connection for sensors	Two-wire connection
Digital representation of the input signal	any of the following representations - 12 bits two's complement - 11 bits + sign - 12 bits binary
Measuring principle	Momentary value decoding
Conversion principle	successive approximation
Conversion time typically	25 µsec. (per channel)
Coding time (per measured value)	250 µsec.
Duration of cyclic sampling (scan time)	
- for 8 measured values	maximum 2 msec.
- for 8 measured values	maximum 4 msec.
Max. permissible input voltage (without destruction)	maximum ±30V (static) or ± 75V (Pulse for max. 1 msec. and a duty cycle 1:20)
Permissible isolation voltage between the reference potential and the central grounding point	maximum 60 V AC/75 V DC
Error indication for	
- Overflow	yes (overflow bit set)
- Internal errors	yes (error bit (= E bit) set)
Noise suppression common mode noise (V_{PP}=1 V)	minimum 70 dB

Basic error limits	
- Voltage ranges outside 0 to 1.25 V, +1.25 V	0.1 %
- Current ranges and 0 to 1.25 V, +1.25 V	0.2 %
Operational error limits (0 °C to 60 °C)	
- Voltage ranges outside 0 to 1.25 V, +1.25 V	0.2 %
- Current ranges and 0 to 1.25 V, +1.25 V	0.2 %
Individual errors	
- Linearity	0.02 %
- Tolerance	0.05 %
- Polarity error	0.05 %
Temperature error	0.005 %/K
Cable length	
- shielded	maximum 200 m (656 ft.)
Front connector	43 pins
Isolation rating	to VDE 0160
Rated isolation voltage (channels to grounding point) tested with	500 V
Supply voltage	
- internal	+5 V +/- 5 %
- external	none
Internal current consumption	typically 0.7 A
Power losses of the module	typically 3.5 W
Weight	approx. 0.4 kg
Design	ES 902

15.2.8 Analog Output Modules

Analog Output Modules 8 x ± 10 V; 0 to 20 mA; Floating (6ES5 470-7LA12)

Terminal Assignment of the Front Connectors



1	L+ 24V	
2		
3	QV 0	Channel 0
4	S+0	
5	S - 0	
6	QI 0	
7	QV 1	Channel 1
8	S+1	
9	S - 1	
10	QI 1	
11		
12		
13	M_{ANA}	
14		
15	QV 2	Channel 2
16	S+2	
17	S - 2	
18	QI 2	
19	QV 3	Channel 3
20	S+3	
21	S - 3	
22	QI 3	
23		
24		
25		
26		
27	QV 4	Channel 4
28	S+4	
29	S - 4	
30	QI 4	
31	QV 5	Channel 5
32	S+5	
33	S - 5	
34	QI 5	
35		
36		
37	M_{ANA}	
38		
39	QV 6	Channel 6
40	S+6	
41	S - 6	
42	QI 6	
43	QV 7	Channel 7
44	S+7	
45	S - 7	
46	QI 7	
47	L - 0V	

M_{ANA}	=	common reference point of all current and voltage channels
QV x	=	voltage output channel x
QI x	=	current output channel x
S+x	=	sensor line+ channel x
S - x	=	sensor line - channel x

Analog Output Module 8 x ± 10 V; 0 to 20 mA; Floating (6ES5 470-7LA12)

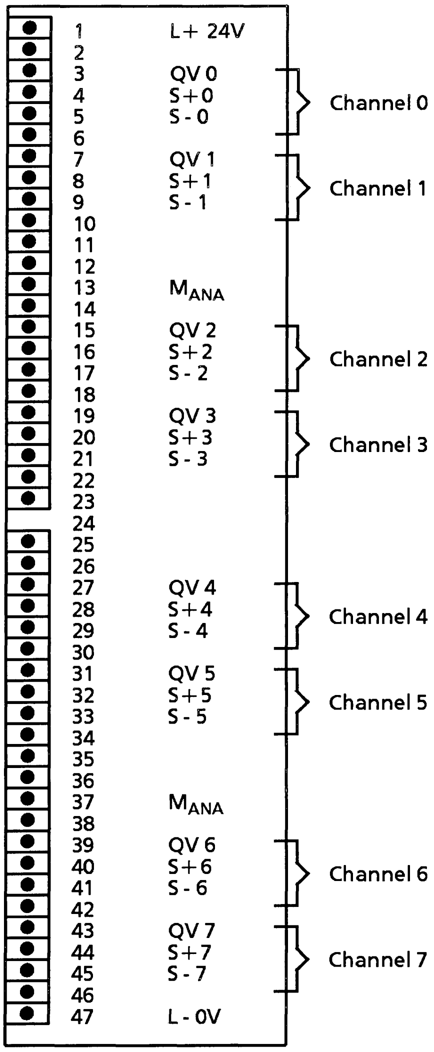
Technical Specifications

Number of outputs		8 voltage and current outputs
Type of outputs		floating (not between the inputs)
Output ranges (rated values)		± 10 V; 0 to 20 mA
Load resistance		
- for voltage outputs	minimum	3.3 kΩ
- for current outputs	maximum	300 Ω
Load connection		Load to M_{ANA} - terminal
Digital representation of the output signal		11 bits plus sign (1024 units = rated value)
Conversion time		1 msec.
Permissible overload capability	approx.	25 % (up to 1280 units)
Short circuit protection		yes
Short circuit current	approx.	25 mA (for a voltage output)
Open circuit voltage	approx.	18 V (for current output)
Voltage between the reference potential of the load (M_{ANA} terminal) and the housing	maximum	60 V AC/75 V DC
Linearity in the rated range		±2.5 ‰ ± 3 units
Operational error limits (0 °C to 55 °C)		±6 ‰
Cable length		
- shielded	maximum	200 m (656 ft.)
Front connector		46 pin
Isolation rating		according to VDE 0160
Rated isolation voltage (outputs to ⏚)		
- tested with		500 V
Capacitive load of the outsputs:	max.	1 µF (per channel)

Power supply		
- rated value		24 V DC
- ripple V_{PP}		3.6 V
- permissible range (ripple included)		20 to 30 V
Current consumption		
- from 5 V (internal)	typically	0.25 A
- from 24 V (external)	typically	0.3 A
Power losses of the module	typically	8.5 W
Weight	approx.	0.4 kg (0.88 lb.)

Analog Output Module 8 x ± 10 V; Floating (6ES5 470-7LB12)

Terminal Assignment of the Front Connector



1	L+ 24V	
2		
3	QV 0	Channel 0
4	S+0	
5	S - 0	
6		
7	QV 1	Channel 1
8	S+1	
9	S - 1	
10		
11		
12		
13	M_{ANA}	
14		
15	QV 2	Channel 2
16	S+2	
17	S - 2	
18		
19	QV 3	Channel 3
20	S+3	
21	S - 3	
22		
23		
24		
25		
26		
27	QV 4	Channel 4
28	S+4	
29	S - 4	
30		
31	QV 5	Channel 5
32	S+5	
33	S - 5	
34		
35		
36		
37	M_{ANA}	
38		
39	QV 6	Channel 6
40	S+6	
41	S - 6	
42		
43	QV 7	Channel 7
44	S+7	
45	S - 7	
46		
47	L - 0V	

M_{ANA} = common reference point of all current and voltage channels
QV x = voltage output channel x
S+x = sensor line + channel x
S - x = sensor line - channel x

Analog Output Module 8 x ± 10 V; Floating (6ES5 470-7LB12)

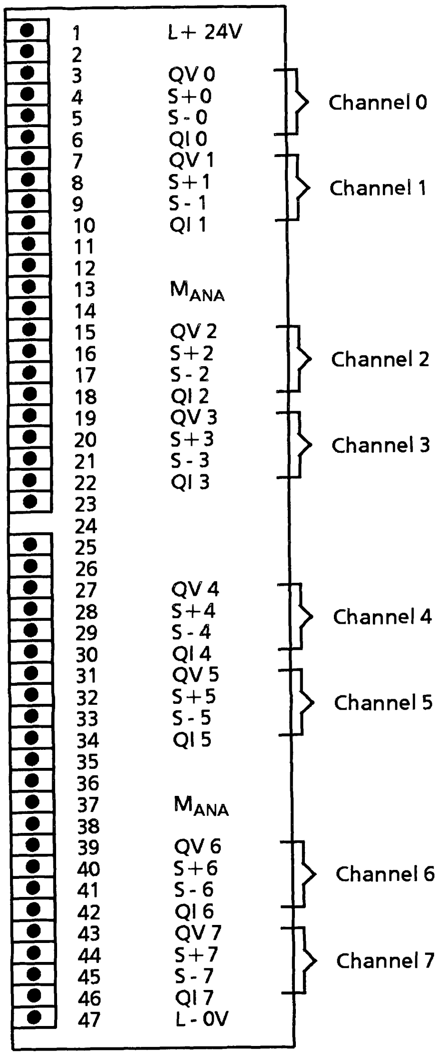
Technical Specifications

Number of outputs		8 voltage and current outputs
Floating		yes (not between the inputs)
Output ranges (rated value)		±10 V
Load resistance	minimum	3.3 kΩ
Load connection		Load to M_{ANA}-terminal
Digital representation of the output signal		11 bits sign (1024 units = rated value)
Conversion time		1 msec.
Permissible overload capability	approx.	25 % (up to 1280 units)
Short circuit protection		yes
Short circuit current	approx.	25 mA
Voltage between the reference potential of the load (M_{ANA}-terminal) and the housing	maximum	60 V AC/75 V DC
Linearity in the rated range		±2,5 ‰ ±3 units
Operational error limits (0 °C to 55 °C)		±6 ‰
Cable length - shielded	maximum	200 m (656 ft.)
Front connector		46 pin
Isolation rating		according to VDE 0160
Rated isolation voltage (outputs to ⏚) - tested with		500 V
Capacitive load of the outputs:	max.	1 µF (per channel)

Power supply		
- rated value		24 V DC
- ripple V_{PP}		3.6 V
- permissible range (ripple included)		20 to 30 V
Current consumption		
- from 5 V (internal)	typically	0.25 A
- from 24 V (external)	typically	0.3 A
Power losses of the module	typically	8.5 W
Weight	approx.	0.4 kg (0.88 lb.)

Analog Output Module 8 x + 1 to 5 V; + 4 to 20 mA; Floating (6ES5 470-7LC12)

Terminal Assignment of the Front Connector



1	L+ 24V	
2		
3	QV 0	Channel 0
4	S+0	
5	S - 0	
6	QI 0	
7	QV 1	Channel 1
8	S+1	
9	S - 1	
10	QI 1	
11		
12		
13	M_{ANA}	
14		
15	QV 2	Channel 2
16	S+2	
17	S - 2	
18	QI 2	
19	QV 3	Channel 3
20	S+3	
21	S - 3	
22	QI 3	
23		
24		
25		
26		
27	QV 4	Channel 4
28	S+4	
29	S - 4	
30	QI 4	
31	QV 5	Channel 5
32	S+5	
33	S - 5	
34	QI 5	
35		
36		
37	M_{ANA}	
38		
39	QV 6	Channel 6
40	S+6	
41	S - 6	
42	QI 6	
43	QV 7	Channel 7
44	S+7	
45	S - 7	
46	QI 7	
47	L - 0V	

M_{ANA} = common reference point of all current and voltage channels

QV x = voltage output channel x

QI x = current output channel x

S+x = sensor line + channel x

S - x = sensor line - channel x

Analog Output Module 8 x + 1 to 5 V; + 4 to 20 mA; Floating (6ES5 470-7LC12)

Technical Specifications

Number of outputs		8 voltage and current outputs
Floating		yes (not between the inputs)
Output ranges (rated values)		+1 to 5 V; +4 to 20 mA
Load resistance		
- for voltage outputs	minimum	3.3 kΩ
- for current outputs	maximum	300 Ω
Load connection		Load to M_{ANA}-terminal
Digital representation of the output signal		11 bits sign (1024 units = rated value)
Conversion time		1 msec.
Permissible overload	approx.	25 % (up to 1280 units)
Short circuit protection		yes
Short circuit current	approx.	25 mA (for a voltage output)
Open circuit voltage	approx.	18 V (for current output)
Voltage between the reference potential of the load (M_{ANA}-terminal) and the housing	maximum	60 V AC/75 V DC
Linearity in the rated range		±2.5 ‰ ±3 units
Operational error limits (0 °C to 55 °C)		±6 ‰
Cable length		
- shielded	maximum	200 m (656 ft.)
Front connector		46 pins
Isolation rating		according to VDE 0160
Rated isolation voltage (outputs to ⏚)		
- tested with		500 V
Capacitive load of the outputs:	max.	1 µF (per channel)

Power supply		
- rated value		24 V DC
- ripple V_{pp}		3.6 V
- permissible range (ripple included)		20 to 30 V
Current consumption		
- from 5 V (internal)	typically	0.25 A
- from 24 V (external)	typically	0.3 A
Power losses of the module	typically	8.5 W
Weight	approx.	0.4 kg (0.88 lb.)

15.2.9 Signal Preprocessing Modules

The following Signal Preprocessing Modules can be used in the S5-115U programmable controller with CPU 945.

Table 15.1 Overview of Signal Preprocessing Modules

Signal Preprocessing Modules*	Current consumption (internal at 5V)	Fan required?	Adapter casing required ?
IP 240 Counter and pos. decoder mod.	0.6** A	No	Yes
IP 241 Digital position decoder module	1 A	Yes	Yes
IP 241 USW Ultrasonic pos. decoder mod.	1.2 A	Yes	Yes
IP 242B Counter module	1.2 A	No	Yes
IP 243 Analog module	1.2 A	No	Yes
IP 244 Temperature control module	0.8 A	No	Yes
IP 246 Positioning module	1.0 A	No	Yes
IP 247-4UA11 IP 247-4UA21 Positioning module	0.8 A	Yes No	Yes
IP 252 Closed-loop control module	2.3 A	No	Yes
IP 281 Counter module	0.6 A	No	Yes
IP 288 Positioning module/ Cam controller	0.8 A	No	Yes
WF 705 Position decoder	0.5 A	No	Yes
WF 706 Positioning module	0.75 A (3-channel) 1.5 A (6-channel)	No	Yes
WF 707 Cam controller	0.5 A	No	Yes
WF 721 Positioning module	1.0 A	No	Yes
WF 723 Positioning module	1.3 A	No	Yes

* For Order Nos. of the modules or relevant Manuals, please refer to the Catalog

** Without sensor power supply

15.2.10 Communications Processors

The following communications processors can be used in the S5-115U programmable controller with CPU 945:

Table 15.2 Overview of Communications Processors

Communications Processors*	Current Consumption (internal at 5V)	Fan required ?	Adapter casing required ?
CP 516 Memory module	0.8 A	No	Yes
CP 524/544 Computer link	1.5 A	Yes	Yes
CP 525 Listing/computer link	1.8 A	Yes	Yes
CP 526 Listing/computer link	2.2 A	Yes	Yes
CP 530A Configuring a SINEC L1 LAN	1.0 A	Yes	Yes
CP 530 Configuring a SINEC L1 LAN	1.0 A	No	No
CP 5430/5431 Configuring a SINEC L2 LAN	0.45 A	No[1]	Yes
CP 143-0AB.. Configuring a SINEC H1 LAN	2. 5 A	Yes	Yes
CP 523 Serial input/output	0.13 A	No	Yes
CP 527/528 - for monochrome VDUs - for colour VDUs	1.5 A/1.8 A	No	Yes
CP 552-1 CP 552-2 Diagnostics processor	1.8 A 3.2 A	No No	Yes
CP 580 CP 581	5.5 A 1.8 A**	Yes ggf.	Yes[3] Yes[4]

* For Order Nos. of the modules and the relevant manuals, please refer to the Catalog

** Basic board

[1] see Section 3 "Installation Guidelines"

[2] can only be plugged in with adapter casing 6ES5 491-0LC11

[3] can only be plugged in with adapter casing 6ES5 491-0LD11

[4] can only be plugged in with adapter casing 6ES5 491-0LD11 (if required)

15.2.11 Interface Modules

The following interface modules can be used in the S5-115U programmable controller:

Table 15.3 Overview of Interface Modules

Interface Module	Application	Technical Specifications
IM 305 (6ES5 305-7LA11)	The IM 305 interface module is used for the centralized connection of an expansion unit (EU) to a central controller (CC) (see Section 3.3).	Power supply for EU max. 1 A Current consumption (5 V; internal requirement) 0.1 A Cable length 0.5 m Weight (total) approx. 0.6 kg
IM 306 (6ES5 306-7LA11)	The IM 306 interface module is used for the centralized connection of up to 3 expansion units (EUs) to a central controller (CC) (see Section 3.3).	Power supply for EU max. 2 A Current consumption (5 V; internal requirement) 0.5 A Weight approx. 0.6 kg **Accessories** 705 connecting cable (see Catalog ST 52.3)
IM 304 (6ES5 304-3UB11)	The IM 304 interface module is used in connection with the IM 314 interface module to connect expansion units (EUs) over a distance of max. 600 m (1900 ft) to a central controller (CC) in a distributed configuration (see Section 3.3).	Current consumption (5 V; internal requirement) 1.5 A Weight approx. 0.3 kg **Accessories** Adapter casing 6ES5 491-0L
IM 314 (6ES5 314-3UA11)	The IM 314 interface module is used in connection with the IM 304 interface module to connect expansion units (EUs) over a distance of max. 600 m (1900 ft) to a central controller (CC) in a distributed configuration (see Section 3.3).	Current consumption (5 V; internal requirement) 1 A Weight approx. 0.3 kg **Accessories** Adapter casing 6ES5 491-0LB12 Terminator for IM 314 6ES5 760-1AA11 721 connecting cable (see Catalog ST 52.3)

Table 15.3 Overview of Interface Modules (Continued)

Interface Module	Application	Technical Specifications
IM 307 (6ES5 307-3UA11)	The IM 307 interface module is used in connection with the IM 317 interface module for the distributed connection of expansion units (EUs) to a central controller (CC) via fiber optic cables.	Current consumption (5 V; internal requirement) 1 A Weight approx. 0.4 kg **Accessories** Adapter casing 6ES5 491-0LB12
IM 317 (6ES5 317-3UA11)	The IM 317 interface module is used in connection with the IM 307 interface module for the distributed connection of expansion units (EUs) to a central controller (CC) via fiber optic cables.	Current consumption (5 V; internal requirement) 1 A Weight approx. 0.4 kg **Accessories** Adapter casing 6ES5 491-0LB12 722 connecting cable (see Katalog ST 52.3)
IM 308 (6ES5 308-3UA12)	The IM 308-B interface module is used for linking the ET 100 with the S5-155U programmable controller.	Current consumption (5 V; internal requirement) max. 0.5 A Weight approx. 0.4 kg **Accessories** Adapter casing 6ES5 491-0LB12
IM 308-B (6ES5 308-3UB11)	The IM 308-B interface module is used for connecting the S5-115U programmable controller to the SINEC L2 LAN (with DP protocol)	Current consumption (5 V; internal requirement) max. 0.6 A Weight approx. 0.5 kg **Accessories** Adapter casing 6ES5 491-0LB12
IM 318 (6ES5 318-3UA11)	The IM 318 interface module is used in connection with the IM 308 interface module for the distributed connection - over max. 3000 m (9900 ft) of expansion units (EUs) to a central controller (CC).	Current consumption (5 V; internal requirement) max. 0.3 A Weight approx.0.34 kg **Accessories** Adapter casing 6ES5 491-0LB12

15.2.12 The 313 Watchdog Module (6ES5 313-3AA11)

313 Watchdog Module **(6ES5 313-3AA11)**

The 313 Watchdog module monitors signals on the S5 bus. It can be used to check the connection between a central controller and an expansion unit for open circuits and short circuits.

Technical Specifications

Current consumption (at 5 V)	maximum 0,4 A

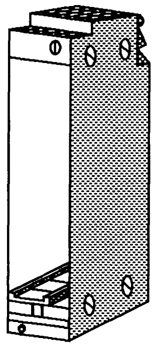
For further technical specifications
see Manual C 79 000-B85000-C266-1

Accessories

Adapter casing	6ES5 491-0LA12

15.3 Accessories

Adapter Casing for Two Printed Circuit Boards (6ES5 491-0LB12)

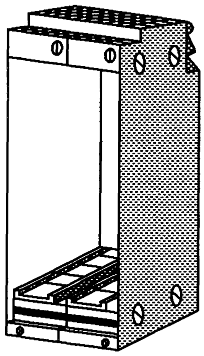


Technical Specifications

Dimensions (w×h×d) in mm	43 x 303 x 187
Weight	approx. 0.9 kg (2 lb.)

Even modules which are not of the block type can be used in the S5-115U, provided an adapter casing is available.
The adapter casing can take one module or, in the case of the CR 700-3 subrack, two modules, but only one double-width module IP241, IP 245, IP 246 and IP 247 (self-ventilated model) IP 252 and CP 535.

Adapter Casing for SIMATIC S5 CP 580/581 or for up to 4 Printed Circuit Boards (6ES5 491-0LD11)

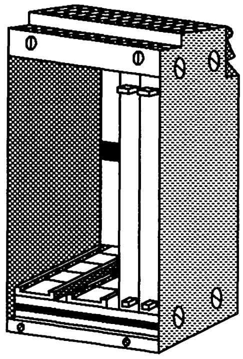


Technical Specifications

Dimensions (w×h×d) in mm	86 x 303 x 187
Weight	approx. 0,8 kg (1.8 lb.)

Even modules which are not of the block type can be used in the S5-115U, provided an adapter casing is available.
The adapter casing can take two double-width or 4 single-width modules; the SIMATIC S5 CP 580/581 can be used in the S5-115U programmable controller with the help of this adapter casing.

Adapter Casing for CP 551 Bulk Storage Memory or for up to 6 Printed Circuit Boards (6ES5 491-0LC11)

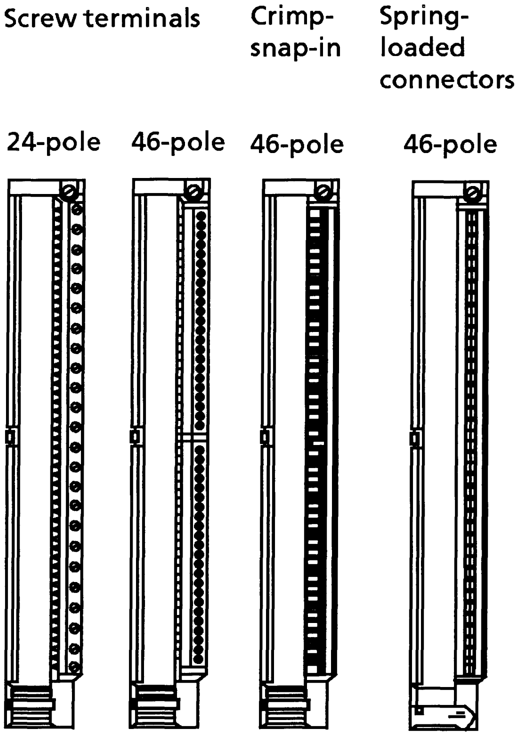


Technical Specifications

Dimensions (w×h×d) in mm		129 x 303 x 187
Weight	approx.	1.8 kg (3.97

Even modules which are not of the block type can be used in the S5-115U, provided an adapter casing is available.
The adapter casing can take three double-width or six single-width modules; the CP 551 hard disk drive can also be used in the S5-115U subracks with the help of this adapter casing.

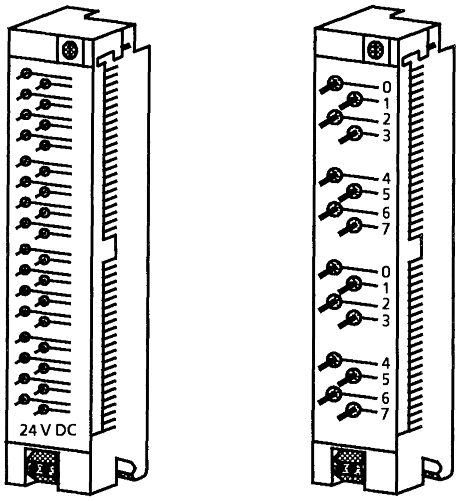
490 Front Connector



Technical Specifications
see Catalog ST 52.3

490 Front Connector	
- for screw-type terminals	
- 24-pole	**6ES5 490-7LB11**
- 46-pole	**6ES5 490-7LB21**
763 jumper comb (for use with screw-terminal front connectors)	6ES5 763-7LA11
- for crimp snap-in connections 46-pole	
- without crimp contacts	**6ES5 490-7LA21**
- with 50 crimp contacts	**6ES5 490-7LA11**
Crimp contacts (250)	6XX3 070
Crimping tool for crimping the crimp contacts	6XX3 071
Extraction tool for crimp contacts	6ES5 497-8MA11
- for spring-loaded connectors	
- 46-pole	**6ES5 490-7LC11**

Simulator



Technical Specifications
see Catalog ST 52.3

Simulator

- 32 switches/buttons	**6ES5 490-7SA11**
24 V DC	
can be plugged into	6ES5 420-7LA11
	6ES5 430-7LA11
- 16 switches/buttons	**6ES5 490-7SA21**
24/48/60 V AC/DC	
can be plugged into	6ES5 431-7LA11
	6ES5 432-7LA11
	6ES5 435-7LA11
	6ES5 435-7LB12
	6ES5 436-7LA11
	6ES5 436-7LB12

Fan Subassembly

If the 6ES5 951-7LD11 or 6ES5 951-7ND11 power supply modules carry a load of more than 7 A, or if modules with a high power consumption are used, a fan subassembly is necessary.

Technical Specifications (6ES5 981-0HA11 and 6ES5 981-0HB11)

Fan	6ES5 981-0HA11	6ES5 981-0HB11
Input voltage		
- rated value	230/115 V AC	230/115 V AC
- tolerance	- 10 % to + 10 %	- 10 % to + 10 %
Network frequency		
- Rated value	50/60 Hz	50/60 Hz
Input current	typically 420 mA	typically 420 mA
Contact rating		
- with ohmic load	5.0 A at 230 V AC	5.0 A at 230 V AC
2.5 A	at 30 V DC	2.5 A at 30 V DC
- with inductive load	1.5 A at 230 V AC	1.5 A at 230 V AC
0.5 A	at 30 V DC	0.5 A at 30 V DC
- Life span		
Operating cycles	$1.5 \cdot 10^6$ AC11	$1.5 \cdot 10^6$ AC11

Fan Subassembly (Continued)

Technical Specifications (6ES5 981-0HA21 and 6ES5 981-0HB21)

Fan	6ES5 981-0HA21	6ES5 981-0HB21
Degree of protection	IP20 to DIN 40 050	IP20 to DIN 40 050
Radio interference		
suppression level	A to VDE 0871	A to VDE 0871
Dimensions	423 x 110 x 135	294 x 110 x 135
w x h x d (mm)	16.6 x 3.93 x 5.31 in.	11.5 x 3.93 x 5.31 in.
Weight	1.5 kg (3.3 lb.)	1.4 kg (3 lb.)
Accessories		
Installation parts	6ES5 981-0GA11	6ES5 981-0GB11
Filter mat unit	6ES5 981-0JA11	6ES5 981-0JB11
Input voltage		
- rated value	24 V DC	24 V DC
- permissible range	+20 V to +30 V	+20 V to +30 V
(including ripple)		
Input current	typically 800 mA	typically 800 mA
Contact rating		
- with ohmic load	5.0 A at 230 V AC	5.0 A at 230 V AC
	2.5 A at 30 V DC	2.5 A at 30 V DC
- with inductive load	1.5 A at 230 V AC	1.5 A at 230 V AC
	0.5 A at 30 V DC	0.5 A at 30 V DC
- Life span		
Operating cycles	1.5·106 DC11	1.5·106 DC11
Degree of protection	IP20 to DIN 40 050	IP20 to DIN 40 050
Radio interference		
suppression level	A to VDE 0871	A to VDE 0871
Dimensions	423 x 110 x 135	294 x 110 x 135
w x h x d (mm)	16.6 x 3.93 x 5.31 in.	11.5 x 3.93 x 5.31 in.
Weight	1.5 kg (3.3 lb.)	1.4 kg (3 lb.)
Accessories		
Installation parts	6ES5 981-0GA11	6ES5 981-0GB11
Filter mat unit	6ES5 981-0JA11	6ES5 981-0JB11

Back-Up Battery **(6EW1 000-7AA)**

Technical Specifications

Lithium battery (3.4 V/5.2 Ah)		
- back-up time (at 25 °C and constant backup of the CPU with memory submodule)	approx.	2 years
- service life (at 25 °C)	approx.	5 years
- external battery backup		3.4 to 9 V

Types of Fuses

Wickmann 19231	**2,5 A FF**	**6ES5 980-3BC21**
	4 A FF	**6ES5 980-3BC51**
	10 A FF	**6ES5 980-3BC41**
Gould GAB4		
Bussmann ABC4		

Types of Relays

Siemens V23042 B201 B101

Günther 3700-2501-011

Siemens V23157-006-A402

Appendices

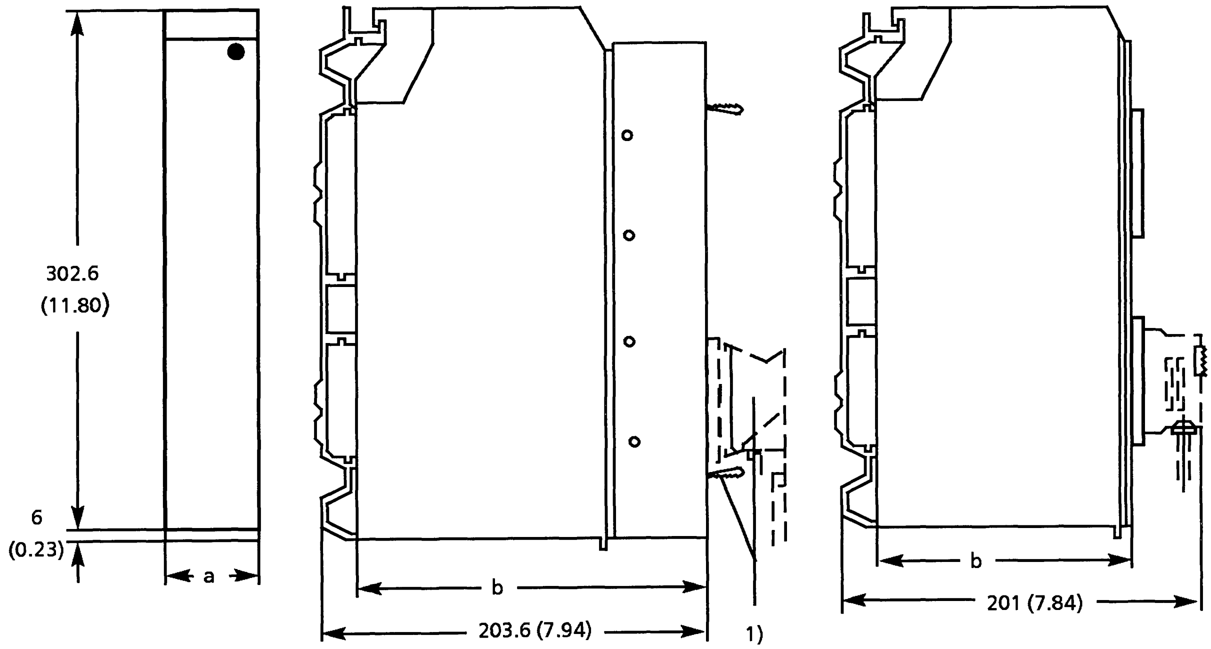
A Dimension Drawings

Figures

Table

A Dimension Drawings

A.1 Dimensions of the Modules



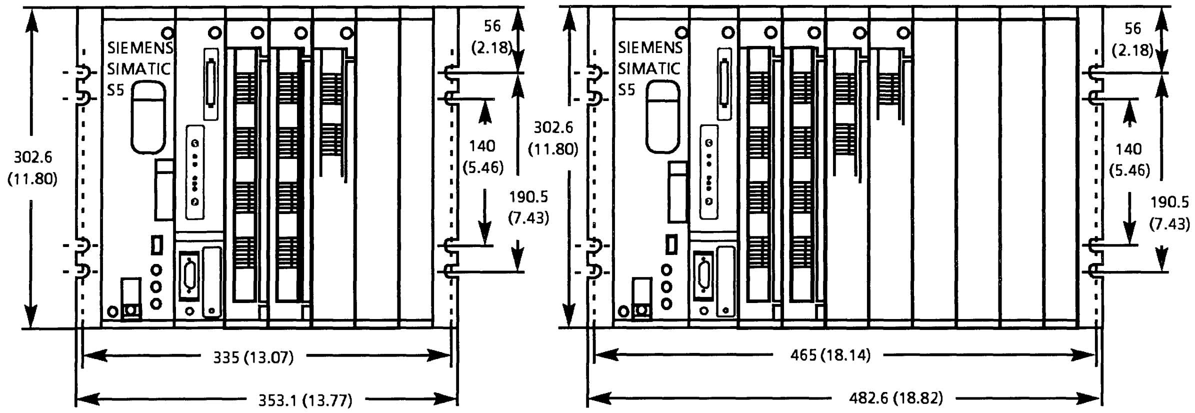
1 Operator control and display elements (e.g. when using an adapter casing) protrude from the front (e.g. CP 525)

Figure A.1 Dimension Drawings of the Modules

Table A.1 Module Dimensions

	a mm (in.)	b mm (in.)	Mechnical Slot Coding
Power Supply Module	65 (2.54)	187 (7.29)	---
Central Processing Unit	43 (1.68)	187 (7.29)	---
Digital and Analog Modules			built in
Adapter Casing			
Interface Module	25 (0.98)	133 (5.19)	---

A.2 Dimension Drawings of the Subracks



CR 700-0

CR 700-2F; ER 701-1 ... 3

Figure A.2 Dimension Drawings of the Subracks

A.3 Dimension Drawings for Cabinet Installation

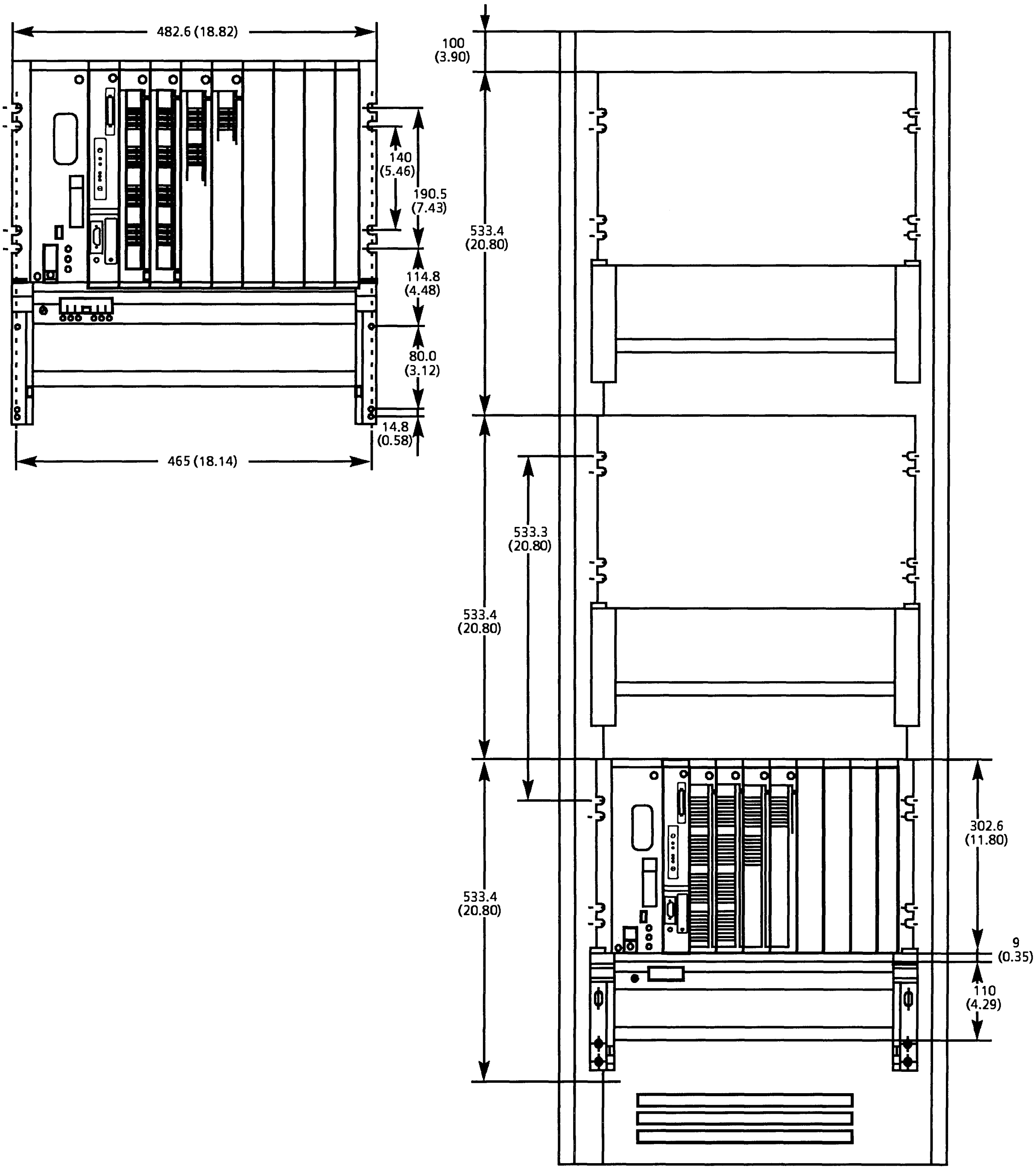


Figure A.3 Dimensions for Installation in a 19-Inch Cabinet

Important

The 533.4 (20.80) spacing must be maintained even if no fan is used.

B Maintenance

Figures

B Maintenance

Proper functioning of the programmable controller can only be guaranteed if the electronic components have not been interfered with.
This appendix describes the maintenance jobs you can perform on your programmable controller. These are

- Changing the fuses
- Installing and changing the batteries
- Changing the fan filter
- Replacing the fan motor

B.1 Changing Fuses

For the output modules with red LED indicators for fuse failure, you can remove the fuses with a screwdriver (maximum width 3 mm). Swing the front connectors out to access the fuses. Fuse specifications are noted on the inside of the front doors.

B.2 Installing or Changing Battery

The power supply modules 6ES5 951-7LB21/7NB21 are provided with a backup battery. Use a 3.4 V / 5 Ah lithium battery (Order No. 6EW1 000-7AA; size C) for backup in those power supply modules. Its service life for continuous backup is at least two years (one year when using CPs).
The power supply modules 6ES5 951-7LD21/7ND41/7ND51 are provided with two backup batteries. Use a 3.6 V / 1.75 Ah lithium battery each (Order No. 6ES5 980-0AE11; size AA) for backup in those power supply modules. Its service life for continuous backup is at least one year.

Changing Battery in Power Supply Modules with 2 Backup Batteries

- Since the second battery takes over the backup function you can change the discharged battery without any problems.
- After changing of the battery, the backup function remains with the second battery. Only when this battery is discharged, the new battery takes over the backup functions.

Note

If you install or change a battery when the PLC is shut off and there is no external voltage supply, perform an Overall Reset on the CPU afterwards. Otherwise, the CPU cannot go into the RUN mode.

B.2.1 Removing the Battery

Proceed as follows:

1. Open the battery compartment as follows (see Figure B.1)
   ① Press the slide down.
   ② Swing the battery compartment door out and down.
2. Removing the battery
   Remove the battery by pulling the end of the plastic ribbon out. The battery slides out of its clamp and falls out.
3. Closing the battery compartment door
   Close the battery compartment door by swinging it back into place. Latch it with the slide.

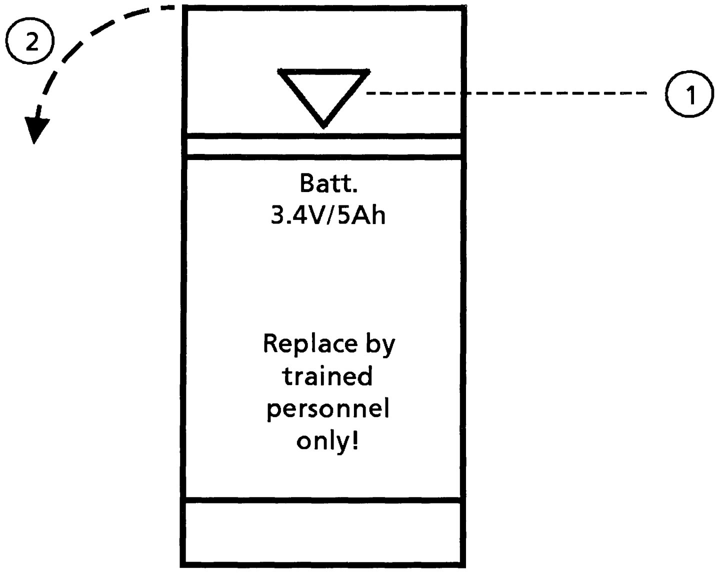


Figure B.1 Opening the Battery Compartment

B.2.2 Installing the Battery

To install a battery, proceed as described below:

1. Open the battery compartment door (see Figure B.1)
   ① Press the slide down and
   ② swing the battery compartment door out and down.

Install the battery after noting the following points:

- The poles are indicated on the back of the battery compartment.
- The plastic ribbon should be to the left of the battery so that its end stays in a freely accessible position.
- Before using a lithium battery, you should depassivate it by loading it with 100 ohms for two hours.

2. Install the battery
3. Close the battery compartment door.
   Close the door of the battery compartment by swinging it back into place. Latch it with the slide.

B.2.3 Battery Disposal

Note the warning below and dispose of used batteries carefully!

Warning

Improper handling can cause a lithium battery to catch fire and explode. Do not recharge or disassemble a lithium battery.
Keep it away from water and open flame. Do not expose it to temperatures greater than 100 ° C!

B.3 Changing the Fan Filter

Under the fan is a filter (Order No. 6ES5 981-0JA11) to keep the electronic components and the printed circuit boards in the modules clean. As preventive maintenance, change this filter regularly according to the degree of air pollution in the PLC's environment.

To change the filter, proceed as described below (see Figure B.2):

1. Pull out the dirty filter using the two handles ①.
2. Place the new filter in the guide tracks ② and push it back.

Note

You can change the filter while the PLC is operating.

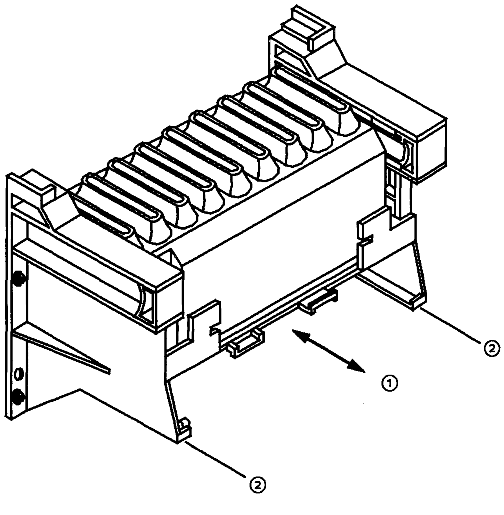


Figure B.2 Changing the Fan Filter

B.4 Replacing the Fan Motor

The fan motors can be exchanged in all fan subassemblies of the S5-115U programmable controller.
For this purpose, Siemens offers a fan replacement package (Order No. 6ES5 988-7NA11).

This package contains the following:

- Fan motor
- Plug-in connector
- Repair instructions

Since the repair instructions form part of the fan replacement package, the removal and installation of the fan motor is not described in this section.

C Guidelines for Handling Electrostatic Sensitive Devices (ESD)

Figures

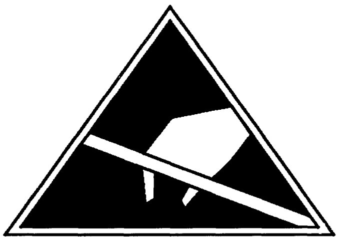
C Guidelines for Handling Electrostatic Sensitive Devices (ESD)

What is ESD?

All electronic modules are equipped with large-scale integrated ICs or components. Due to their design, these electronic elements are very sensitive to overvoltages and thus to any electrostatic discharge.

These **E**lectrostatic **S**enstive **D**evices are commonly referred to by the abbreviation **ESD.**

Electrostatic sensitive devices are labelled with the following symbol:



⚠ **Caution**

Electrostatic sensitive devices are subject to voltages that are far below the voltage values that can still be perceived by human beings. These voltages are present if you touch a component or module without previously being electrostatically discharged. In most cases, the damage caused by an overvoltage is not immediately noticeable and results in total damage only after a prolonged period of operation.

Electrostatic charging of objects and persons

Every object with no conductive connection to the electrical potential of its surroundings can be charged electrostatically. In this way, voltages up to 15000 V can build up whereas minor charges, i.e. up to 100 V, are not relevant.

Examples:

● Plastic covers	up to 5000 V
● Plastic cups	up to 5000 V
● Plastic-bound books and notebooks	up to 8000 V
● Desoldering device with plastic parts	up to 8000 V
● Walking on plastic flooring	up to 12000 V
● Sitting on a padded chair	up to 15000 V
● Walking on a carpet (synthetic)	up to 15000 V

Limits for perceiving electrostatic discharges

An electrostatic discharge is

- perceptible from 3500 V
- audible from 4500 V
- visible from 5000 V

A fraction of these voltages is capable of destroying or damaging electronic devices.

Carefully note and apply the protective measures described below to protect and prolong the life of your modules and components.

General protective measures against electrostatic discharge damage

- Keep plastics away from sensitive devices. Most plastic materials have a tendency to build up electrostatic charges easily.

- Make sure that the personnel, working surfaces and packaging are sufficiently grounded when handling electrostatic sensitive devices.

- If possible, avoid any contact with electrostatic sensitive devices. Hold modules without touching the pins of components or printed conductors. In this way, the discharged energy cannot affect the sensitive devices.

Additional precautions for modules without housings

Note the following measures that have to be taken for modules that are not protected against accidental contact:

- Touch electrostatical sensitive devices only
  - if you wear a wristband complying with ESD specifications or
  - if you use special ESD footwear or ground straps when walking on an ESD floor.

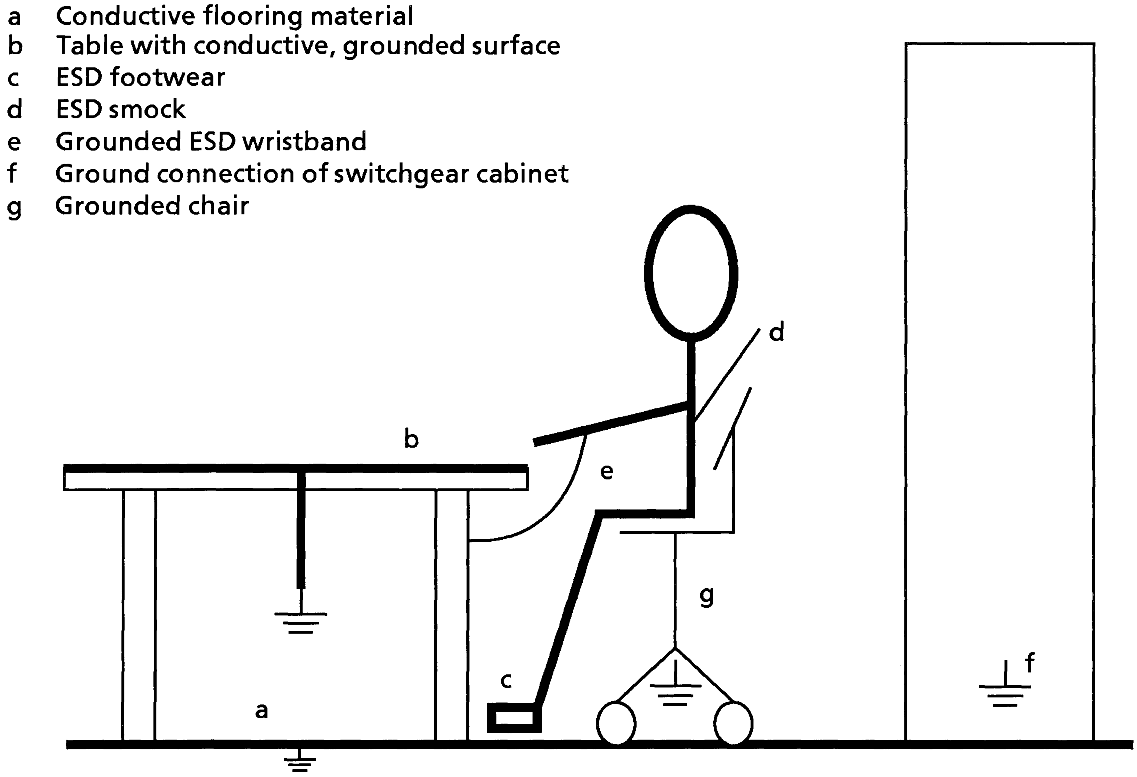
- Persons working on electronic devices should first discharge their bodies by touching grounded metallic parts (e.g. bare metal parts of switchgear cabinets, water pipes, etc.).

- Protect the modules against contact with chargeable and highly insulating materials, such as plastic foils, insulating table tops or clothes made of plastic fibres.

- Place electrostatic sensitive devices only on conductive surfaces:
  - Tables with ESD surface
  - Conductive ESD foam plastic (ESD foam plastic is mostly coloured black)
  - ESD bags

- Avoid direct contact of electrostatic sensitive devices with visual display units, monitors or TV sets (minimum distance to screen > 10 cm).

The following Figure once again illustrates the precautions for handling electrostatically sensitive devices.



a Conductive flooring material
b Table with conductive, grounded surface
c ESD footwear
d ESD smock
e Grounded ESD wristband
f Ground connection of switchgear cabinet
g Grounded chair

Figure C.1 ESD Measures

Taking measurements and working on ESD modules

Measurements may be taken on electrostatic sensitive devices only if

- the measuring device is grounded (e.g. via protective conductor) or
- the tip of the isolated measuring tool has previously been discharged (e.g. by briefly touching grounded metal parts).

List of Abbreviations

List of Abbreviations

List of Abbreviations

Abbreviation	Explanation
ACCU 1	Accumulator 1
ACCU 2	Accumulator 2
AI	Analog input module
AQ	Analog output module
ASCII	American Standard Code for Information Interchange
BASP	Command output disable
BR	Base address register
CBR	"Receive" coordination byte
CBS	"Send" coordination byte
CC 0 /CC 1	Condition code bit 0 / condition code bit 1
CP	Communications processor
CPU	Central processing unit
CSF	STEP 5 control system flowchart method of representation
DB	Data block
DBL	Length register of data block
DBS	Start address register of data block
DX	Data block (extension)
EMC	Electromagnetic compatibility
FB	Function block
FX	Function block (extension)
IP	Intelligent I/O
LAD	STEP 5 ladder diagram method of representation
NST	Nesting stack
NSTP	Nesting stack pointer
OB	Organization block
OP	Operator panel
OS	Latching overflow, overflow flag
OV	Overflow, overflow flag
PB	Program block
PG	Programmer
PII/PII'	Process image of the inputs
PIQ/PIQ'	Process image of the outputs
PLC	Programmable controller
PS	Power supply

List of Abbreviations

Abbreviation	Explanation
QVZ	Timeout
RLO	Result of logic operation
RS	System data area
RT	Extended system data area
SAC	Step address counter
SB	Sequence block
STL	STEP 5 statement list method of representation

Index

Index