SIEMENS

SIMATIC

ET 200S Interface Module IM 151/CPU

Manual

This manual is part of the document package with the order number: **6ES7 151-1AA00-8BA0**

Important Notes, Contents

1 Product Overview

2 Addressing

3 ET 200S in the PROFIBUS Network

4 Commissioning and Diagnostics

5 Functions of the IM 151/CPU

6 Technical Specifications

7 Cycle and Response Times

8 Getting Started

A Configuration Frame and Parameter Assignment Frame for the ET 200S

B Instruction List

C Execution Times of the SFCs and SFBs

D Migration of the IM 151/CPU

Glossary, Index

Edition 05/2000
A5E00058783-01

Safety Guidelines

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

Danger

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

Warning

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

Caution

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

Note

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

Qualified Personnel

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

Correct Usage

Note the following:

Warning

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

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

Trademarks

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

Some of other designations used in these documents are also registered trademarks; the owner's rights may be violated if they are used by third parties for their own purposes.

Siemens AG
Bereich Automatisierungs- und Antriebstechnik (A&D)
Geschäftsgebiet Industrie-Automatisierungsysteme (AS)
Postfach 4848, D- 90327 Nürnberg

Disclaimer of Liability

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

Important Information

Purpose of the Manual

This manual supplements the *ET 200S Distributed I/O System* manual. It describes all the functions of the IM 151/CPU interface module. The manual does not deal with general ET 200S functions. You will find descriptions of these in the *ET 200S Distributed I/O System* manual (see also the section entitled ”Delivery Package”).

The information contained in this manual and in the *ET 200S Distributed I/O System* manual will enable you to operate the ET 200S with the IM 151/CPU interface module as a DP slave on the PROFIBUS-DP or in a stand-alone configuration.

Target Group

The manual describes the hardware of the IM 151/CPU interface module and is aimed at configuration engineers, commissioning engineers and maintenance personnel who use the ET 200S with PLC functionality.

It consists of chapters containing instructions and reference chapters.

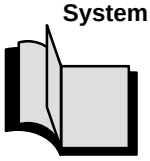
Delivery Package

This delivery package (order number 6ES7 151-1AA00-8BA0) consists of three manuals, with contents as follows:

IM 151/CPU Interface Module

- IM 151/CPU addressing
- ET 200S with IM 151/CPU in the PROFIBUS network
- Commissioning and diagnostics of the IM 151/CPU
- Technical specifications of the IM 151/CPU
- *List of STEP 7* instructions

ET 200S Distributed I/O System



- Installing and wiring the ET 200S
- Commissioning and diagnostics of the ET 200S
- Technical specifications of the IM 151, digital and analog electronic modules, process-related modules
- Order numbers for the ET200S

ET 200S Motor Starters

- Installing and wiring motor starters
- Commissioning and diagnostics of motor starters
- Technical specifications of motor starters
- Order numbers for motor starters

Applicability

This manual is valid for the IM 151/CPU interface module with the order numbers 6ES7 151-7AA00-0AB0 and 6ES7 151-7AB00-0AB0 as well as the components of the ET 200S specified in the *ET 200S Distributed I/O System* manual.

This manual contains a description of the components that were valid at the time the manual was published. We reserve the right to enclose a Product Information bulletin containing up-to-date information about new components and new versions of components.

Standards, Certificates and Approvals

The ET 200S distributed I/O system is based on EN 50170, Volume 2, PROFIBUS. The ET 200S distributed I/O system fulfills the requirements and criteria of IEC 1131, Part 2 and the requirements for obtaining the CE marking. CSA, UL and FM certifications have been obtained for the ET 200S.
Shipbuilding certification has been applied for.

You will find detailed information on these standards, certificates and approvals in the *ET 200S Distributed I/O System* manual.

Position in the Information Landscape

In addition to the ET 200S manuals, you will also need the manual for the DP master used and the documentation for the configuration and programming software used (see the list in Appendix A of the *ET 200S Distributed I/O System* manual).

Note

You will find a detailed list of the contents of the ET 200S manuals in Section 1.2 of this manual.

We recommend that you begin by reading this section so as to find out which parts of which manuals are most relevant to you in helping you to do what you want to do.

Aids to Finding Information

You can quickly access specific information in the manual by using the following aids:

- At the beginning of the manual you will find a comprehensive table of contents and lists of the figures and tables in the manual.
- The sections of the chapters in the manual contain subheadings that allow you to gain a quick overview of the contents of the section.
- You will find a glossary in the appendix at the end of the manual. The glossary contains definitions of the main technical terms used in the manual.
- At the end of the manual you will find a detailed index that enables you to find the information you require quickly and easily.

Contents

1 Product Overview

1.1 What Is the IM 151/CPU Interface Module? 1-2

1.2 Guide to the ET 200S Manuals 1-5

2 Addressing

2.1 Slot-Oriented Addressing 2-2

2.2 User-Defined Addressing 2-5

2.3 Data Transfer with the DP Master 2-6

2.4 Accessing the Intermediate Memory in the IM 151/CPU 2-8

3 ET 200S in the PROFIBUS Network

3.1 ET 200S in the PROFIBUS Network 3-2

3.2 Setting the PROFIBUS Address 3-5

3.3 Network Components 3-7

3.4 Functions via the Programming Device/OP 3-9

3.5 Direct Communication 3-10

4 Commissioning and Diagnostics

4.1 Configuring the IM 151/CPU 4-2

4.2 Resetting the Memory of the IM 151/CPU 4-4

4.3 Commissioning and Startup of the ET 200S 4-7

4.4 Diagnostics Using LEDs 4-9

4.5 Diagnostics via Diagnostic Address with STEP 7 4-11

4.6 Slave Diagnosis 4-14
4.6.1 Station Status 1 to 3 4-16
4.6.2 Master PROFIBUS Address 4-18
4.6.3 Manufacturer ID 4-19
4.6.4 Module Diagnosis 4-20
4.6.5 Module Status 4-21
4.6.6 Structure of the Station Diagnosis 4-22

4.7 System Status List (SSL) 4-24

5 Functions of the IM 151/CPU

5.1 Data for the PROFIBUS-DP 5-2
5.2 The Mode Selector and LEDs 5-4
5.3 Force 5-6
5.4 Memory Module 5-8
5.5 Clock 5-12
5.6 Blocks 5-13
5.7 Parameters 5-16
5.8 Parameterization of the Reference Junction for the Connection of Thermocouples 5-18
5.9 Removal and Insertion of Modules During Operation 5-20
5.10 Switching Power Modules Off and On During Operation 5-22

6 Technical Specifications

6.1 Technical Specifications of the IM 151/CPU 6-2

7 Cycle and Response Times

7.1 Cycle time 7-2
7.2 Response Time 7-6
7.3 Interrupt Response Time 7-9

8 Getting Started

8.1 Step 1: Installing the ET 200S/IM 151/CPU and S7-300 8-4
8.2 Step 2: Wiring the ET 200S/IM 151/CPU and S7-300 8-5
8.3 Step 3: Commissioning the ET 200S/IM 151/CPU 8-6
8.4 Step 4: Configuring the IM 151/CPU for Stand-Alone Operation 8-7
8.5 Step 5: Programming the IM 151/CPU 8-9
8.6 Step 6: Test Run 8-10
8.7 Step 7: Upgrading the IM 151/CPU and Commissioning the S7-300 8-11
8.8 Step 8: Configuring the IM 151/CPU as a DP Slave and the S7-300 as a DP Master 8-12
8.9 Step 9: Programming the IM 151/CPU and the S7-300 8-15
8.10 Step 10: Commissioning and Test Run of the IM 151/CPU and S7-300 . 8-17

A Configuration Frame and Parameter Assignment Frame for the ET 200S

A.1 Structure of the Configuration Frame (SKF) A-2
A.2 Structure of the Configuration Frame (AKF) A-4
A.3 Structure of the Parameter Assignment Frame A-6

B Instruction List

B.1 Address Identifiers and Parameter Ranges B-2
B.2 Abbreviations B-3
B.3 Tab B-3
B.4 Examples of Addressing B-5
B.5 Execution Times with Indirect Addressing B-7
B.5.1 Example of Memory-Indirect, Area-Internal Addressing B-9
B.5.2 Example of Register-Indirect, Area-Internal Addressing B-10
B.5.3 Example of Memory-Indirect, Area-Crossing Addressing B-11
B.5.4 Example of Addressing Via Parameters B-12
B.6 Bit Logic Instructions B-13
B.7 Logic Instructions with Parenthetical Expressions B-15
B.8 ORing of AND Operations B-16
B.9 Logic Instructions with Timers and Counters B-17
B.10 Logic Instructions with the Contents of Accumulator 1 . . . . . B-18
B.11 Logic Instructions with Condition Code Bits B-19
B.12 Edge-Triggered Instructions B-25
B.13 Setting/Resetting Bit Addresses B-26
B.14 Instructions Directly Affecting the RLO B-27
B.15 Timer Instructions B-28
B.16 Counter Instructions B-29
B.17 Load Instructions B-30
B.18 Load Instructions for Timers and Counters B-32
B.19 Transfer Instructions B-33
B.20 Load and Transfer Instructions for Address Registers B-35
B.21 Load and Transfer Instructions for the Status Word B-36
B.22 Load Instructions for DB Numbers and DB Length B-36
B.23 Fixed-Point Math Instructions (16 Bits) B-37
B.24 Fixed-Point Math Instructions (32 Bits) B-38
B.25 Floating-Point Math Instructions (32 Bits) B-39
B.26 Addition of Constants B-39
B.27 Addition via the Address Register B-40
B.28 Comparison Instructions with Integers (16 Bits) B-40
B.29 Comparison Instructions with Integers (32 Bits) B-41
B.30 Comparison Instructions (32-Bit Real Numbers) B-42
B.31 Shift Instructions B-43

B.32 Rotate Instructions B-44
B.33 ACCU Transfer Instructions, Incrementing, Decrementing B-45
B.34 Program Display and Null Operation Instructions B-45
B.35 Data Type Conversion Instructions B-46
B.36 Complement Formation B-47
B.37 Block Call Instructions B-48
B.38 Block End Instructions B-49
B.39 Exchanging Data Blocks B-49
B.40 Jump Instructions B-50
B.41 Instructions for the Master Control Relay (MCR) B-52

C Execution Times of the SFCs and SFBs

C.1 System Functions (SFCs) C-1
C.2 System Function Blocks (SFBs) C-3

D Migration of the IM 151/CPU

D.1 Differences to Selected S7-300 CPUs D-2
D.2 Porting the User Program D-3

Glossary

Index

Figures

1-1	View of the ET 200S Distributed I/O System with the IM 151/CPU	1-3
1-2	Components and the Manuals Required for Them	1-5
2-1	Structure of the Default Address Area	2-2
2-2	Slots on the ET 200S	2-2
2-3	Example of Address Assignment for I/O Modules	2-4
2-4	Structure of the Address Area for User-Defined Addressing	2-5
2-5	Principles of Data Transfer Between the DP Master and the ET 200S with the IM 151/CPU	2-6
3-1	Example of a PROFIBUS Network	3-2
3-2	The PG/OP Accesses the ET 200S via the DP Interface in the DP Master	3-4
3-3	The Programming Device Accesses the ET 200S on a Stand-Alone Basis	3-4
3-4	Setting the PROFIBUS Address	3-6
3-5	Connecting the DP Network	3-8
3-6	Direct Communication with the IM 151/CPU	3-10
4-1	How to Use the Mode Selector to Reset the Memory	4-5
4-2	Diagnostic Addresses for the DP Master and ET 200S	4-11
4-3	Format of the Slave Diagnostic Data	4-15
4-4	Structure of the Module Diagnosis	4-20
4-5	Structure of the Module Status	4-21
4-6	Structure of the Station Diagnosis (1)	4-22
4-7	Structure of the Station Diagnosis (2)	4-23
5-1	Mode selector	5-4
5-2	Principle Behind Forcing	5-7
5-3	Position of the Memory Card Slot for the MMC on the IM 151/CPU	5-9
5-4	Example of a Parameterization Dialog Box for the CPU Module Data in STEP 7 V5.1	5-19
6-1	Basic Circuit Diagram for the IM 151/CPU	6-4
6-2	Basic Circuit Diagram for the IM 151/CPU FO	6-4
7-1	Component Parts of the Cycle Time	7-2
7-2	Shortest Response Time	7-7
7-3	Longest Response Time	7-8
A-1	Description of Byte 0 of the Address Area Identifiers	A-3
A-2	Description of Byte 1 of the Address Area Identifiers	A-3
A-3	Standard Section of the Parameter Assignment Frame	A-6
A-4	General Section of the Parameter Assignment Frame for the IM 151/CPU	A-7
D-1	Example: FB with Unpacked Addresses	D-3
D-2	Example: FB with Packed Addresses	D-4
D-3	Example: Rewiring of the Signals	D-5

Tables

1-1	Topics of the Manuals in the ET 200S Manual Package	1-6
2-1	Addresses of the ET 200S I/O Modules	2-3
2-2	Examples of the Maximum Configuration	2-7
2-3	Accessing the Address Areas	2-8
2-4	Addressing Interface in STEP 7 V5.1 (Extract)	2-9
3-1	Network Components	3-7
4-1	Configuration Options	4-2
4-2	Ways to Reset the Memory	4-4
4-3	Internal CPU Events at Memory Resetting	4-5
4-4	LEDs for PROFIBUS-DP	4-10
4-5	Responses to Operating Mode Changes and Interruptions in User Data Transfer in the DP Master and the ET 200S with the IM 151/CPU	4-12
4-6	Evaluation of RUN/STOP Transitions in the DP Master/ET 200S	4-13
4-7	Structure of Station Status 1 (Byte 0)	4-16
4-8	Structure of Station Status 2 (Byte 1)	4-17
4-9	Structure of Station Status 3 (Byte 2)	4-17
4-10	Structure of the Master PROFIBUS Address (Byte 3)	4-18
4-11	Structure of the Manufacturer Identification (Bytes 4 and 5)	4-19
4-12	Additional Interrupt Information and Diagnostic Data	4-23
4-13	SSL Sublists of the IM 151/CPU	4-24
5-1	Attributes of the Device Database (DDB) File	5-2
5-2	Positions of the Mode Selector	5-4
5-3	LEDs for CPU Functionality	5-5
5-4	Available MMCs	5-8
5-5	Firmware Update with MMC	5-10
5-6	Backing Up the Operating System	5-11
5-7	Features of the Clock	5-12
5-8	Overview of the Blocks	5-13
5-9	OBs for Cycle and Restart	5-14
5-10	OBs for Interrupts	5-14
5-11	OBs for Error/Fault Responses	5-15
5-12	Parameter Blocks, SetParameters and Their Ranges for the IM 151/CPU	5-16
5-13	Parameterization of the Reference Junction	5-18
5-14	Result of the Preset/Actual Comparison in Modules that Cannot be Parameterized	5-21
5-15	Result of the Preset/Actual Comparison in the Case of Parameterizable Modules with the Power Module Switched on	5-21
5-16	Result of the Preset/Actual Comparison in the Case of Parameterizable Modules with the Power Module Switched off	5-21
6-1	Terminal Assignment of the IM 151/CPU Interface Module	6-3
7-1	Extending the Cycle by Nesting Interrupts	7-3
7-2	Process Image Updating	7-3
7-3	Dependency of the User Program Scanning Time	7-4
7-4	Updating the S7 Timers	7-4
7-5	Extending the Cycle by Nesting Interrupts	7-5
7-6	Interrupt Response Times of the IM 151/CPU (Without Communication)	7-9

A-1 Structure of the Configuration Frame in the special identifier format (SKF) A-2
A-2 Identifiers for the Address Areas of the Intermediate Memory A-3
A-3 Structure of the Configuration Frame in the Normal Identifier Format (AKF) A-4
A-4 Structure of the Configuration Frame with Default Setting for the Address Areas (Normal Identifier Format in Accordance with AKF) A-5
D-1 Differences to Selected S7-300 CPUs D-2
D-2 Example: Replacements under Options! Rewire D-4

Product Overview

1

In This Chapter

The product overview provides information about

- The role of the IM 151/CPU interface module within the ET 200S distributed I/O system
- Which manuals in the ET 200S manual package contain what information.

Chapter Overview

In Section	Contents	Page
1.1	What Is the IM 151/CPU Interface Module?	1-2
1.2	Guide to the ET 200S Manuals	1-5

1.1 What Is the IM 151/CPU Interface Module?

What Is the IM 151/CPU?

The IM 151/CPU is a component of the ET 200S distributed I/O system with IP 20 protection. The IM 151/CPU interface module is an intelligent preprocessing unit (intelligent slave). It enables you to decentralize control tasks.

An ET 200S with an IM 151/CPU can therefore exercise full and, if necessary, independent control over a process-related functional unit and can be used as a stand-alone CPU. The use of the IM 151/CPU leads to further modularization and standardization of process-related functional units and simple, clear machine concepts.

How Is the IM 151/CPU Integrated in the ET 200S?

The IM 151/CPU interface module is integrated in the ET 200S in the same way as any other module. In other words, its configuration concept, installation and expansion capability are the same.

View

The figure below shows a sample configuration of an ET 200S with an IM 151/CPU.

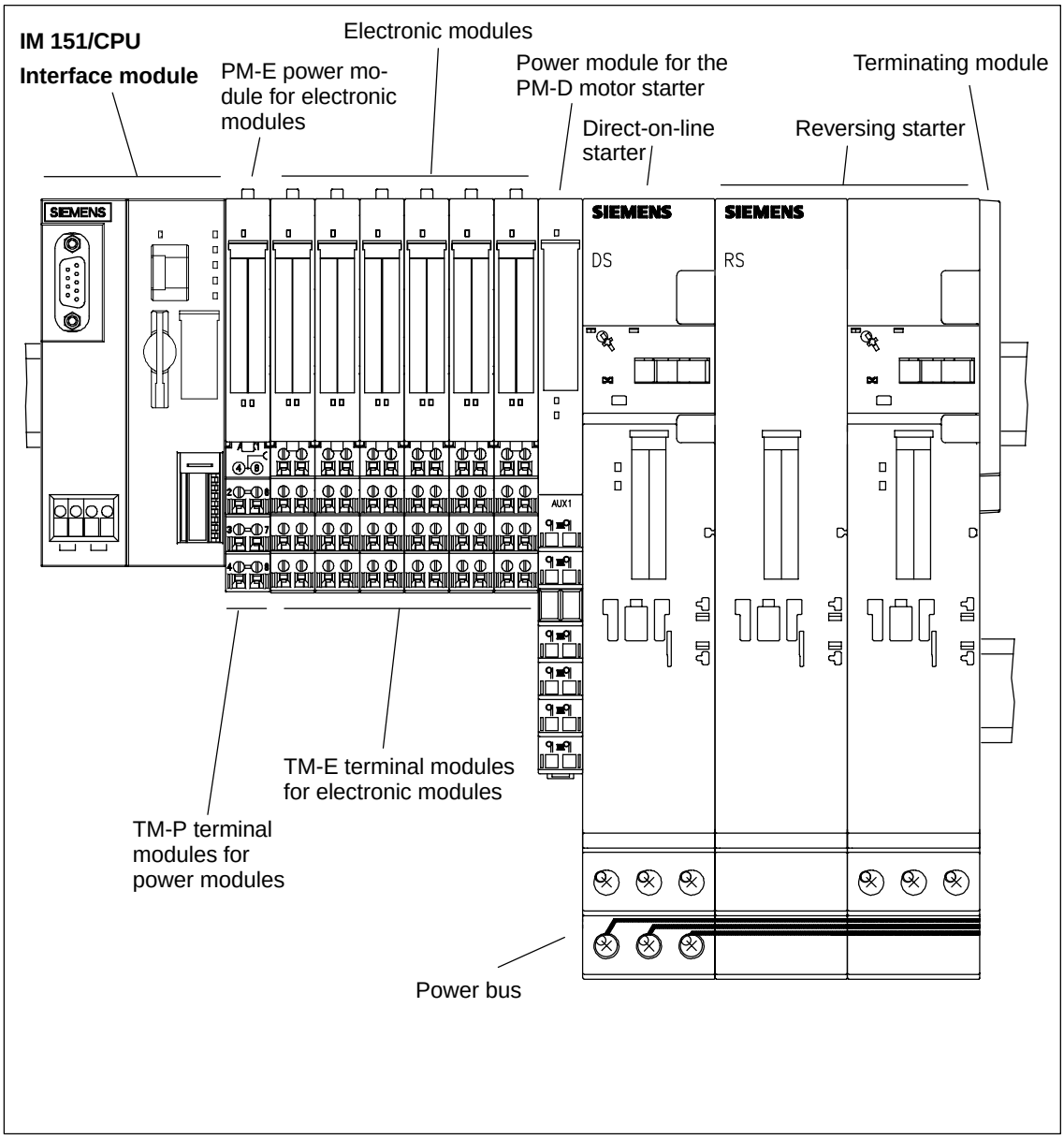


Figure 1-1 View of the ET 200S Distributed I/O System with the IM 151/CPU

Features of the IM 151/CPU Compared to Other Modules

The IM 151/CPU interface module has the following special features:

- The interface module has PLC functionality (integrated CPU with 24 KB working memory and 40 KB RAM load memory).
- The interface module can be enhanced with up to 63 I/O modules from the ET 200S range.
- The interface module has an operating mode switch with positions for RUN-P, STOP and MRES.
- There are 6 LEDs on the front of the interface module to indicate the following:
  - ET 200S faults (SF)
  - Bus faults (BF)
  - Supply voltage for electronic components (ON)
  - Force requests (FRCE)
  - Operating mode of the IM 151/CPU (RUN and STOP)
- Variants for connection to the PROFIBUS-DP via RS485 and fiber-optic cables (FO variant)

How Is the ET 200S Configured with the IM 151/CPU?

To configure the ET 200S with the IM 151/CPU (configuration and parameter assignment), you require *HWCONFIG*, which is part of the configuration software *STEP 7*, as of V 5.1. You can find out how to configure the ET 200S with the IM 151/CPU in Section 4.1 of this manual.

How Is the IM 151/CPU Programmed?

To program the IM 151/CPU, you require the *STEP 7* configuration software, as of V 5.1. You can find the *STEP 7* instruction set for programming the IM 151/CPU in Appendix B.

1.2 Guide to the ET 200S Manuals

You are Using the Following Components ...

The components of the ET 200S are described in various manuals in the ET 200S package. The figure below shows possible ET 200S configurations and the manuals required for them.

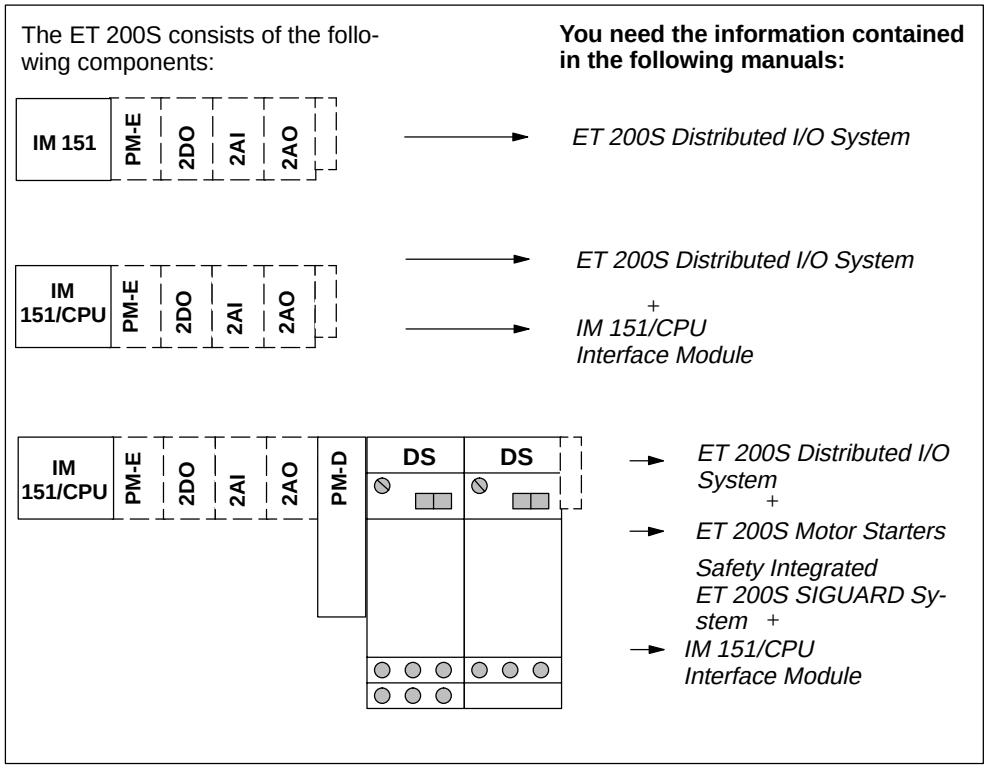


Figure 1-2 Components and the Manuals Required for Them

Where Do You Find What Information?

The following table will help you find the information you require quickly. It tells you which manual you need to refer to and which chapter deals with the topic you are interested in.

Table 1-1 Topics of the Manuals in the ET 200S Manual Package

Contents	Manual			Chapter/ Appendix
	ET 200S Distributed I/O System	*IM 151/CPU Interface Module*	*ET 200S Motor Starters*	
ET 200S components	x			1.2
ET 200S motor starter components			x	1
ET 200S configuration options	x			3
ET 200S motor starters configuration options			x	1
Installing the ET 200S; setting the PROFIBUS address	x			4
Installing ET 200S motor starters			x	2
IM 151/CPU addressing		x		2
Electrical design and wiring of the ET 200S	x			5
The ET 200S with the IM 151/CPU on the PROFIBUS network		x		3
Commissioning and diagnostics of the ET 200S	x			6
Commissioning and diagnostics of the ET 200S with motor starters			x	3
Commissioning and diagnostics of the ET 200S with the IM 151/CPU		x		4
Functions of the IM 151/CPU		x		5
General technical specifications of the ET 200S (standards, certificates and approvals, EMC, environmental conditions, etc.)	x			7
Technical specifications of interface modules, terminal modules, power and electronic modules	x			8, 9, 10, 11, 12
Technical specifications of the ET 200S motor starter			x	4
General technical specifications of the IM 151/CPU		x		6
Safety integrated ET 200S SIGUARD system			x	9
Order numbers for the ET 200S	x			Q
Order numbers for the ET 200S motor starters			x	Q
IM 151/CPU cycle and response times		x		7
Configuration and parameter assignment frame for the IM 151/CPU		x		Q
List of STEP 7 instructions		x		B
Execution times of SFCs		x		C
Glossary	x			Glossary

Addressing

2

Principle of Data Transfer Between the DP Master and the ET 200S

This chapter contains information on the addressing of I/O modules and data transfer between the DP master and the IM 151/CPU.

The following alternatives are available for addressing the I/O modules:

- Slot-oriented address allocation:
  Slot-oriented address allocation is the default form of addressing, in which STEP 7 allocates a fixed module base address to each slot number.
- User-oriented address allocation:
  You can allocate each module any address within the available IM 151/CPU address area.

For information on the addressing of the IM 151/CPU on the PROFIBUS-DP, see Section 3.2.

In This Chapter

In Section	Contents	Page
2.1	Slot-Oriented Addressing	2-2
2.2	User-Defined Addressing	2-4
2.3	Data Transfer with the DP Master	2-5
2.4	Accessing the Intermediate Memory in the IM 151/CPU	2-7

2.1 Slot-Oriented Addressing

Slot-Oriented Address Allocation

In slot-oriented addressing (default addressing) each slot number in a module is allocated an address area in the IM 151/CPU.

Depending on the type of the I/O module, the addresses are digital or analog (see Table 2-1). The address allocation is not fixed and can be changed, but there is a default address area.

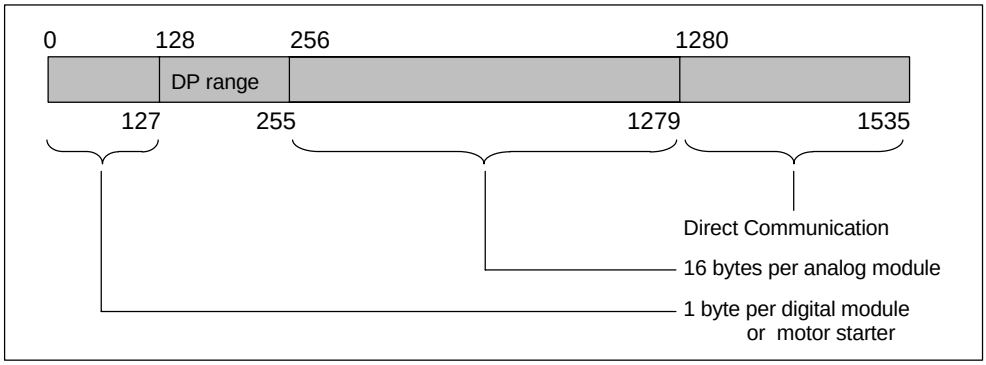


Figure 2-1 Structure of the Default Address Area

Slot Assignment

The figure below shows an ET 200S configuration with digital electronic modules, analog electronic modules, process-related modules and the slot assignment.

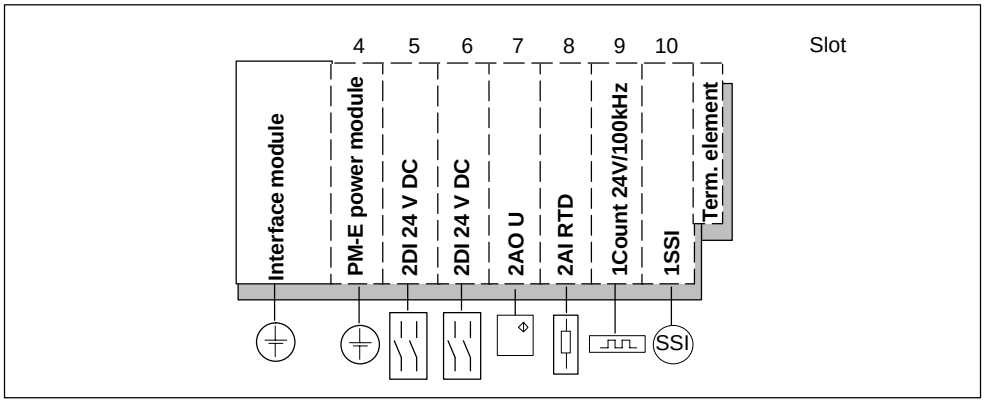


Figure 2-2 Slots on the ET 200S

Address Assignment

Depending on the slot, 1 byte is reserved for digital I/Os and 16 bytes are reserved for analog I/Os in the address areas of the IM 151/CPU for each I/O module (maximum of 63).

The table below indicates the default address assignment for analog and digital modules per slot. The address areas of the I/O modules are ”visible” only to an IM 151/CPU in the ET 200S, not to the associated DP master. The DP master has no direct access to the I/O modules.

Table 2-1 Addresses of the ET 200S I/O Modules

Reserved Address Area	Slot Number									
	1	2	3	4	5	6	7	8	...	66
Digital modules/ motor starters	IM 151/CPU			-	1	2	3	4	...	62
Analog modules, process-related modules				-	272 to 287	288 to 303	304 to 319	320 to 335	...	1248 to 1263
Power modules				256	272	288	304	320		1248

The unassigned addresses in the range 64 to 127 are in the process image in default addressing and can be used any way you choose in the user program. If 2 bits in a byte are already occupied by a digital module, the remaining 6 bits cannot be used as you choose (e.g. the bits 1.4 to 1.7 in Figure 2-3).

You can use the bytes in the address areas that are not used by modules in any way you choose in your user program. In the configuration in Figure 2-3, for example, bytes 2 and 3 can be used as you choose.

Example of Slot-Oriented Address Assignment for I/O Modules

The figure below illustrates a sample ET 200S configuration, showing an example of the address allocation for I/O modules. The addresses for the I/O modules are predefined in default addressing.

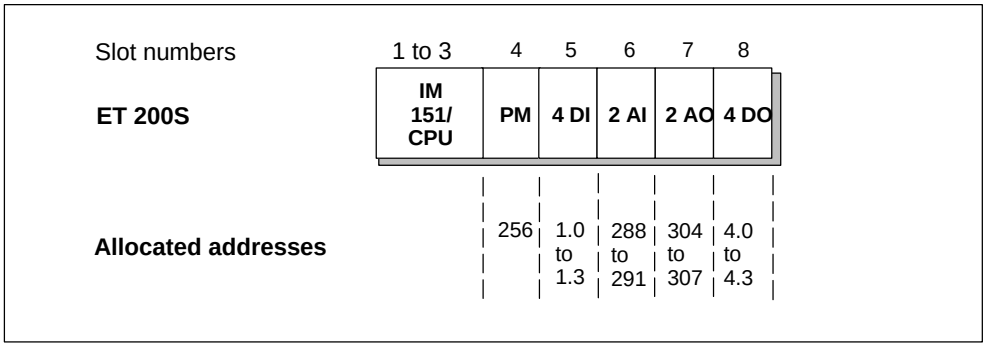


Figure 2-3 Example of Address Assignment for I/O Modules

2.2 User-Defined Addressing

User-Oriented Address Allocation

User-oriented address allocation means you can select the following in units of 1 byte and independent of one another within the range 0 to 1535:

- Input addresses of modules
- Output addresses of modules

The addresses 0 to 127 are in the process image. Assign the addresses in *STEP 7*. When you do this, you define the base address of the module, on which all the addresses of the module depend.

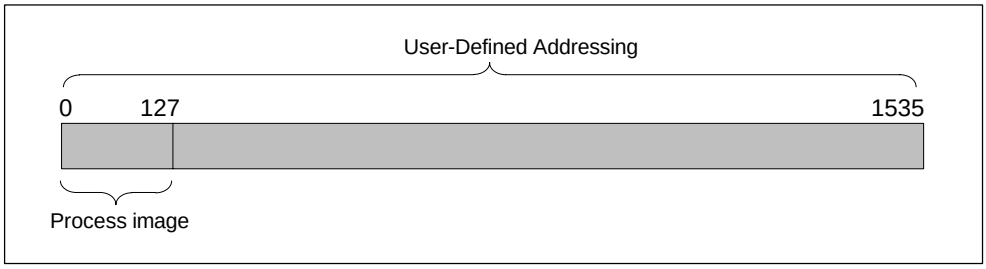


Figure 2-4 Structure of the Address Area for User-Defined Addressing

Note

Bit-specific addressing is not possible in user-defined address allocation, and compression of digital channels is therefore not supported. It is not possible to compress addresses.

Advantages

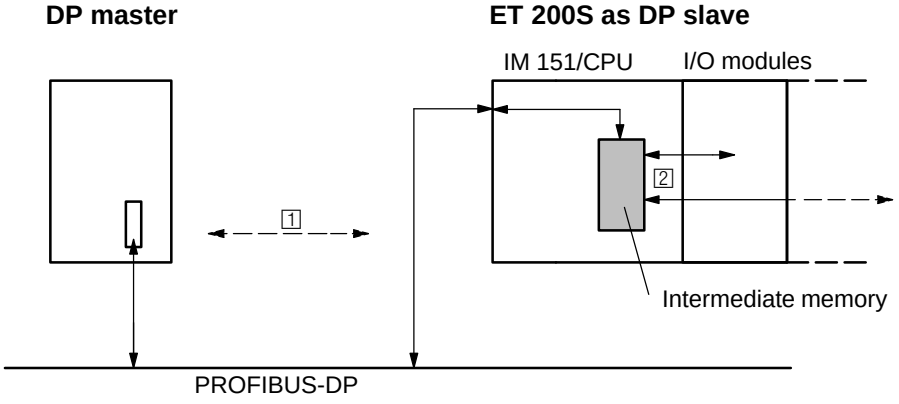
Advantages of user-defined address allocation:

- Optimum utilization of the address areas available, since ”address gaps” between the modules do not occur.
- When creating standard software, you can specify addresses that are independent of the configuration of the ET 200S station.

2.3 Data Transfer with the DP Master

User Data Transfer Via an Intermediate Memory

The user data is located in an intermediate memory in the IM 151/CPU. This intermediate memory is always used when user data is transferred between the IM 151/CPU and the DP master. The intermediate memory consists of a maximum of 32 address areas.



1 Data transfer between DP master and the ET 200S via an intermediate memory in the IM 151/CPU
2 Data transfer between the IM 151/CPU and I/O modules

Figure 2-5 Principles of Data Transfer Between the DP Master and the ET 200S with the IM 151/CPU

Address Areas for User Data Transfer with the DP Master

The ET 200S provides the PROFIBUS-DP with a maximum of 64 bytes of input data and 64 bytes of output data. This data can be addressed in the intermediate memory of the IM 151/CPU in up to 32 address areas.

An address area contains a maximum of 32 bytes. A maximum of 64 bytes is available for input and output data.

The address areas start at 128 by default. The data is entered without a gap as of address 128.

Table 2-2 Examples of the Maximum Configuration

	Example 1	**Example 2**	**Example 3**	**Example 4**
Input address areas Bytes per address area	16 4	2 32	16 4	0 0
Output address areas Bytes per address area	16 4	2 32	16 2	32 2
Total no. of address areas Total no. of bytes for inputs Total no. of bytes for outputs	**32** **64** **64**	4 **64** **64**	**32** **64** 32	**32** 0 **64**

Data consistency

You define data consistency as byte, word, or overall consistency per address area. Consistency can amount to up to 32 bytes/16 words per address area.

DP Diagnostic Address in *STEP 7*

When the ET 200S is configured with *STEP 7*, a diagnostic address is set. The ET 200S receives information on the status of the DP master or on a bus interruption by means of this diagnostic address (see Section 4.5). In DP slave operation the default diagnostic address is 1534.

2.4 Accessing the Intermediate Memory in the IM 151/CPU

Access in the User Program

The following table tells you how to access the intermediate memory in the ET 200S from the user program:

Table 2-3 Accessing the Address Areas

Access Dependent on Data Consistency	The Following Applies
1-, 2- or 4-byte data consistency with load/transfer instructions	All areas parameterized with “unity” consistency can be accessed. You can address a maximum of 64 bytes of input data using load instructions and a maximum of 64 bytes of output data using transfer instructions (L PIB/PIW/PID; T PQB/PQW/PQD; see also Appendix B). The data consistency for word addressing is 2 bytes; for double-word addressing it is 4 bytes. Access is also possible via the process image.
1- to 32-byte data consistency on the PROFIBUS-DP with SFC 14 and SFC 15	If you want to access data in the intermediate memory, you have to read the input data with SFC 14 ”DPRD_DAT” and write the output data with SFC 15 ”DPWR_DAT”. These SFCs have data consistency of 1 to 32 bytes. You can only copy the input data read with SFC 14 as a block of 1 to 32 bytes to a memory marker address area, for example, where it can be addressed with A M x.y. You can also write only one block of 1 to 32 bytes as output data with SFC 15 (see also the *System and Standard Functions*) Reference Manual. If you access areas with “whole length” consistency, the length in the SFC must correspond to the length of the parameterized area.

Access to Free Areas in the Process Image

If you access available but unconfigured process image areas, no process image errors will be generated. You can therefore use inputs and outputs in the process image to which no I/O modules are allocated as markers.

Rules for Address Allocation

You must comply with the following rules when allocating addresses for the ET 200S with the IM 151/CPU:

- Assignment of the address areas:
  - Input data for the ET 200S is **always** output data for the DP master
  - Output data for the ET 200S is **always** input data for the DP master
- You access the data in the user program using load/transfer instructions or SFCs 14 and 15.
- The length, unit and consistency of the associated address areas for the DP master and the DP slave must be identical.
- Addresses for the master and the slave can be different in the logically identical intermediate memory (mutually independent logical I/O address areas in the master and the slave CPU)

When the IM 151/CPU is configured with *STEP 7* for operation in the S5 or in non-Siemens systems, it is clear that only the logical addresses within the slave CPU are allocated. The addresses are then assigned in the master system using the specific configuration tool of the master system.

Addressing Interface in *STEP 7*

The following table illustrates the principles of address allocation. You will also find this table in the *STEP 7* interface. You must set the mode ”MS” (for master slave) or ”DX” (direct connection) in *STEP 7* (see Section 3.5).

Table 2-4 Addressing Interface in *STEP 7 V5.1 (Extract)*

	Mode	**Master**		**PROFIBUS-DP Partner**		**Parameters**		
		I/O	Address	I/O	Address	Length	Unit	Consistency
1	MS	Q	200	I	128	4	Byte	Unit
2	MS	Q	300	I	132	8	Byte	Total length
3	MS	I	700	Q	128	4	Word	Unit
4	MS	I	50	Q	136	4	Byte	Unit
	MS: Master slave	Address areas in the IM 151/CPU		Address areas in the DP master CPU		These address area parameters must be identical for the DP master and the IM 151/CPU		

Default Setting for Address Areas

If, when configuring the ET 200S, you do not parameterize any address areas for data transfer with the DP master, the ET 200S starts up on the PROFIBUS-DP with a default setting.

The default setting is:

- 16 words of input data; unit consistency (i.e. word)
- 16 words of output data; unit consistency (i.e. word)

If the IM 151/CPU is configured for stand-alone operation (“no DP” operating mode), there is no default setting for the address areas because an intermediate memory is not configured in stand-alone operation.

Sample Program

Below you will see a sample program for data interchange between the DP master and the DP slave.

You can find the addresses in Table 2-4.

SFCs 14 and 15 are called by specifying the logical address in hexadecimal format.

In the IM 151/CPU			
Data preprocessing in the DP slave:			
L	2		Load actual value 2 and
T	MB	6	transfer to memory byte 6.
L	IB	0	Load input byte 0 and
T	MB	7	transfer to memory byte 7.
Forward data to DP master			
L	MW	6	Load memory word 6 and
T	PQW	136	transfer to peripheral output word 136
In the DP Master CPU			
Postprocess received data in the DP master:			
L	PIB	50	Load peripheral input byte 50 and
T	MB	60	transfer to memory byte 60.
L	PIB	51	Load peripheral input byte 51 and
L	B#16#3		load byte 3;
+	I		add the values as integer data type and
T	MB	61	transfer the result to memory byte 61.
Data preprocessing in the DP master:			
L	10		Load actual value 10 and
+	3		add 3,
T	MB	67	transfer the result to memory byte 67.
Send the data (memory bytes 60 to 67) to the DP slave:			
CALL	SFC	15	Call system function 15:
LADDR:= W#16#12C			Write the data to the output address area as of
RECORD:= P#M60.0 Byte8			address 300 (12C hexadecimal) with a length of 8
RET_VAL:=MW 22			bytes as of memory byte 60.
In the IM 151/CPU			
Receive data from the DP master (stored in MB 30 to 37):			
CALL	SFC	14	Call system function 14:
LADDR:= W#16#84			Write the data from the input address area as of
RET_VAL:=MW 20			address 132 (84 hexadecimal) with a length of 8
RECORD:=P#M30.0 Byte8			bytes to memory byte 30.
Postprocess received data:			
L	MB	30	Load memory byte 30 and
L	MB	37	load memory byte 37;
+	I		add the values as integer data type and
T	MW	100	transfer the result to memory byte 100.

User Data Transfer in STOP Mode

The user data in the intermediate memory is processed differently depending on whether the DP master or the DP slave (IM 151/CPU) goes into STOP mode.

- If the IM 151/CPU goes into STOP: The data in the intermediate memory (outputs only from the slave's viewpoint) of the IM 151/CPU are overwritten with "0"; i.e. the DP master or a recipient in direct communication reads "0".
- If the DP master goes into STOP: The current data in the intermediate memory of the IM 151/CPU (inputs in the slave, outputs in the master) are retained and can be read out in the user program of the IM 151/CPU.

ET 200S in the PROFIBUS Network

3

Introduction

You can integrate the ET 200S with the IM 151/CPU as a node in a PROFIBUS network. This chapter contains a description of a typical network configuration with the ET 200S and the IM 151/CPU. It also tells you which functions can be executed via the programming device or OP on the ET 200S and which options are available for direct connection.

Chapter Overview

In Section	Contents	Page
3.1	ET 200S in the PROFIBUS Network	3-2
3.2	Setting the PROFIBUS Address	3-5
3.3	Network Components	3-7
3.4	Functions via the Programming Device/OP	3-9
3.5	Direct Communication	3-10

More Information

You will find more information on the structure of networks in the manual for the DP master.

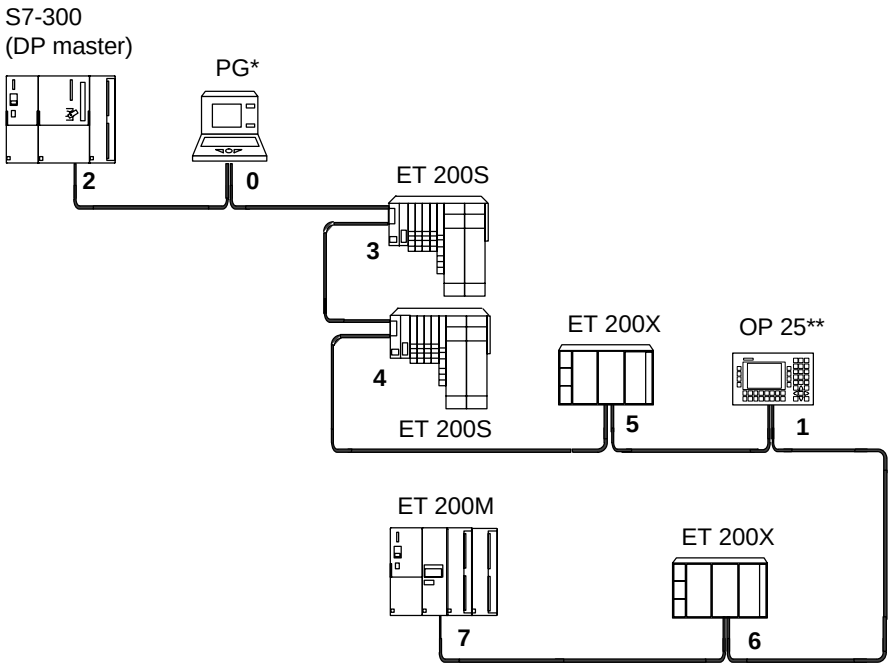
Connecting Fiber-Optic Cables to the IM 151/CPU FO

You can find information on connecting fiber-optic cables to the IM 151/CPU FO in the *ET 200S Distributed I/O Device* manual in the chapter entitled *Wiring and Fitting*. The information it contains for the IM 151 FO also applies to the IM 151/CPU FO.

3.1 ET 200S in the PROFIBUS Network

Structure of a PROFIBUS Network

The figure below illustrates the basic structure of a PROFIBUS network with one DP master and several DP slaves.



* **The ET 200S can be configured and programmed from this programming device**

** **Operating and monitoring functions can be executed on the ET 200S**

0 ... 7 PROFIBUS addresses of the nodes

Figure 3-1 Example of a PROFIBUS Network

Hardware Prerequisites in the Programming Device/OP for Accessing the ET 200S

Before you can access an IM 151/CPU from a programming device/operator panel, the programming device/operator panel must fulfill the following requirements:

- it must have an integrated PROFIBUS-DP interface or DP card; or
- it must have an integrated MPI interface or MPI card.

Access to the ET 200S

The IM 151/CPU is a passive bus node. The programs and configuration of the IM 151/CPU can be transferred to the IM151/CPU by choosing ”Load PLC” from the programming device in SIMATIC Manager. All the other diagnostic and test functions are also possible with the programming device.

If the programming device is currently the only active bus node, this must be set beforehand in SIMATIC Manager by choosing the ”Setting the PG/PC Interface” menu command (see Section 3.4).

However, you can still install OPs/OSs (operator panels/operator stations) as fixed components of the PROFIBUS network for operating and monitoring functions.

You cannot access an ET 200S from more than five devices in parallel:

- 1 connection is reserved for the programming device.
- 1 connection is reserved for an operator panel or an operator station.
- 3 connections are available as desired for programming devices, operator panels/operator stations and CPUs

We recommend that you allocate a PROFIBUS address to the programming device/operator panel in the same way as for other network nodes (see Figure 3-1).

Maximum Data Transfer Rate with a Programming Device Connecting Cable

You can obtain a maximum data transfer rate of 1.5 Mbps using the programming device connecting cable.

Examples for the Connection of the Programming Device/OP on the ET 200S

- The programming device/OP is connected to the PROFIBUS-DP interface of the DP master, but can be connected just as well to any other station in the DP network, including the ET 200S.

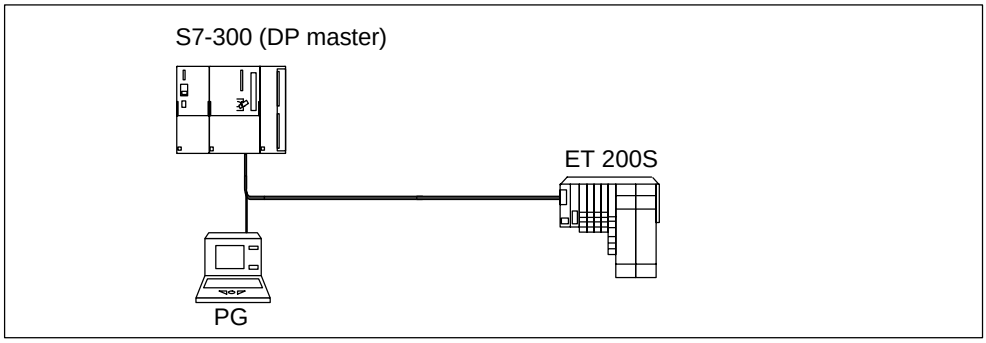


Figure 3-2 The PG/OP Accesses the ET 200S via the DP Interface in the DP Master

- The programming device is connected to the ET 200S on a stand-alone basis for commissioning (you don't add the ET 200S to the PROFIBUS network until later).
  Note: A special setting must be made in *STEP 7* for the stand-alone operation of the ET 200S with the IM 151/CPU if there is no active PROFIBUS node on the bus except the programming device (see Section 3.4).

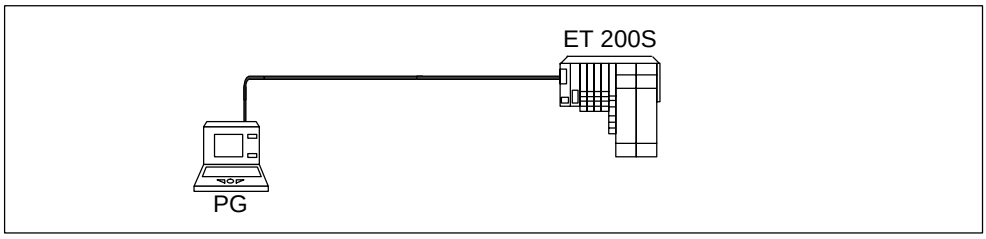


Figure 3-3 The Programming Device Accesses the ET 200S on a Stand-Alone Basis

- The programming device can also be a direct DP node, although a spur line (e.g. programming device connecting cable) is not permissible with a transmission rate of 12 Mbps.

3.2 Setting the PROFIBUS Address

Features

Use the PROFIBUS address to specify the address at which the IM 151/CPU is contacted on the PROFIBUS-DP.

Prerequisites

- The PROFIBUS-DP address for the IM 151/CPU is set via a DIP switch. The DIP switches are on the front of the module.
- Permissible PROFIBUS-DP addresses are 1 to 125.
  If you set an invalid address, the IM 151/CPU will not start up. It can then be reached on the PROFIBUS bus system at address 126.
- Each address can be allocated only once on the PROFIBUS-DP.
- The PROFIBUS address configured in STEP 7 must be identical to the DIP switch setting.
  If the setting does not match, the IM 151/CPU will not start up. It can then be reached on the PROFIBUS bus system at the address set on the DIP switch.
- At startup without STEP 7 configuration, only the setting on the DIP switch is relevant.

Setting the PROFIBUS Address

The DIP switch has 2 functions:

- Switches 1-7:
  These are used to set the PROFIBUS addresses 1-125.
- Switch 8:
  If the IM 151/CPU is not configured, you can use this switch to toggle between stand-alone and DP slave operation in the case of a default startup.
  ON: stand-alone operation.
  OFF: DP slave operation

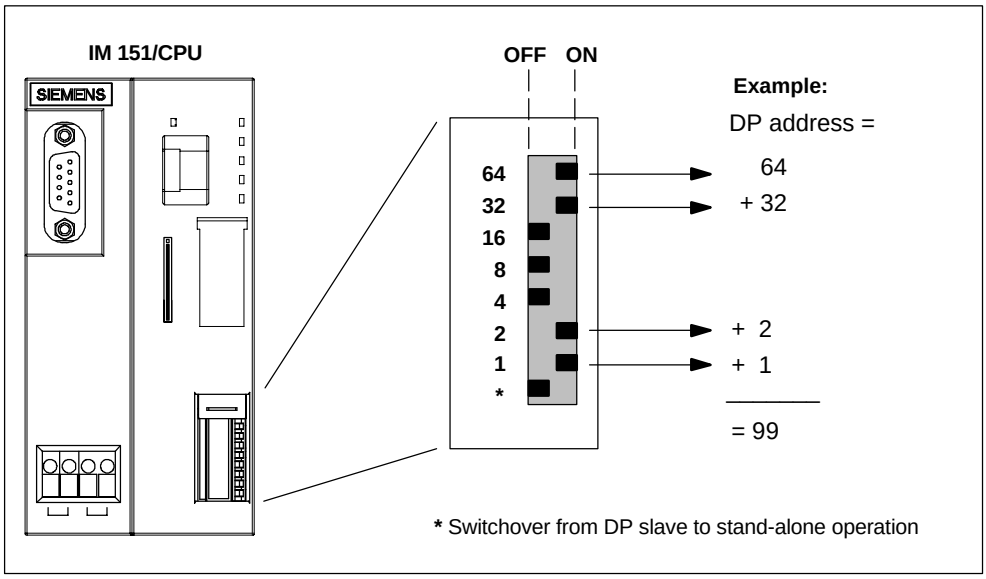


Figure 3-4 Setting the PROFIBUS Address

Changing the PROFIBUS-DP

You change the PROFIBUS-DP address in exactly the same way you set it. Any changes made to the PROFIBUS-DP address are valid for the ET 200S after the interface module is powered up.

The PROFIBUS address configured in STEP 7 must remain identical to the DIP switch setting. To ensure the validity of the changed address, the existing configuration in STEP 7 must be altered accordingly.

3.3 Network Components

To connect the ET 200S to the PROFIBUS-DP network, you need the following network components:

Table 3-1 Network Components

Purpose	Network Components	Order Numbers
To set up the network	Cables (e.g. 2-core, shielded or 5-core, unprepared)	6XV1 830-0AH10 (2-core) 6XV1 830-0BH10 (2-core with PE sheath) 6XV1 830-3CH10 (2-core, for festoon attachment) 6XV1 830-3BH10 (drum cable) 6XV1 830-3AH10 (direct-buried cable) 6ES7 194-1LY00-0AA0-Z (5-core with PVC sheath) 6ES7 194-1LY10-0AA0-Z (5-core; oil-resistant, can be dragged, conditionally resistant to welding; with PUR sheath)
To connect the programming device and the ET 200S on the PROFIBUS-DP network	Bus connector without a programming device socket (up to 12 Mbps)	6ES7 972-0BA10-0XA0 (with a straight outgoing cable unit) 6ES7 972-0BA40-0XA0 (with a slanted outgoing cable unit)
To make a dual connection – for the programming device and the DP master on the PROFIBUS-DP network, for example – via a DP interface (see Figure 3-5)	Bus connector with a programming device socket (up to 12 Mbps)	6ES7 972-0BB10-0XA0 (with a straight outgoing cable unit) 6ES7 972-0BB40-0XA0 (with a slanted outgoing cable unit)
To connect the programming device to the bus connector with the programming device socket	Programming device connecting cable (up to 1.5 Mbps)	6ES7 901-4BD00-0XA0

Example of the Use of Network Components

The figure below shows the example from Figure 3-2 with the use of the network components. Connecting the bus cable to the bus connector is described in the Product Information document for the bus connector.

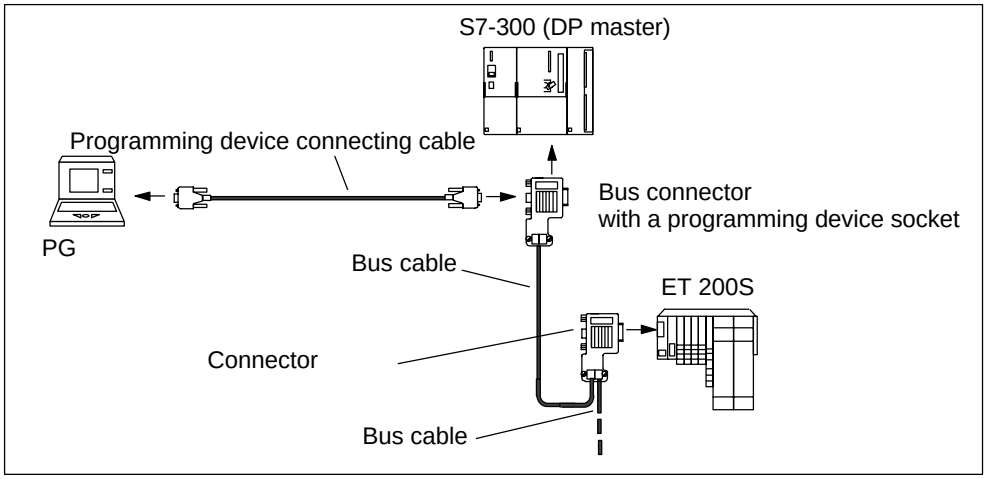


Figure 3-5 Connecting the DP Network

Connecting the IM 151/CPU FO

You can find information on the connection and wiring of fiber-optic cables in the chapter on wiring and fitting in the *ET 200S Distributed I/O Device manual.*

3.4 Functions via the Programming Device/OP

You can use the programming device to:

- Configure the IM 151/CPU with ET 200S modules and put them into operation on the PROFIBUS-DP
- Program the CPU part of the IM 151/CPU
- Execute test functions such as “Monitor/Modify Variables” and “Program Status”
- Display the module status (i.e. for the CPU part, for example, you can display the utilization of the load and working memory, stack contents and diagnostic buffer contents)

You can use the OP to:

- Operate and monitor

You will find a detailed description of the functions in the *STEP 7* online help system.

Running the ET 200S on a Stand-Alone Basis with the Programming Device – Required Settings in *STEP 7*

If you connect an ET 200S to a programming device on a stand-alone basis, you must specify a setting for the programming device interface in *STEP 7* for the execution of online functions on the IM 151/CPU (e.g. downloading a configuration or user program to the IM 151/CPU, or reading out information from the IM 151/CPU online). Proceed as follows:

1. In *STEP 7*, choose the “Setting the PG/PC Interface” tool (**Start > STEP 7 > Setting the PG/PC Interface**).
2. Set the interface of your programming device to PROFIBUS.
3. Call the properties of the PROFIBUS network.
4. Set the properties so that the programming device/PC is the only active master on the bus.

Note: When you operate the ET 200S on a stand-alone basis and the programming device/PC is not set as the only master, an online connection to the IM 151/CPU is not possible.

If you subsequently configure a DP master for the network and want to go online, you should cancel these settings; additional security functions are thus activated against bus faults.

3.5 Direct Communication

You can configure the IM 151/CPU as an intelligent slave with *STEP 7* V 5.1 for direct communication. Direct communication is a special communication relationship between PROFIBUS-DP nodes.

Principle

Direct communication is characterized by the fact that the PROFIBUS-DP nodes ”listen in” to find out which data a DP slave is sending back to its DP master. Using this function, the eavesdropper (recipient) can directly access changes to the input data of remote DP slaves.

During configuration in *STEP 7*, you set via the relevant I/O input addresses the address area of the recipient at which the required data of the sender is to be read.

Example

Figure 3-6 gives you an example of the relationships you can configure in direct communication in *STEP 7* V 5.1 with an IM 151/CPU. Other DP slaves can only be senders here.

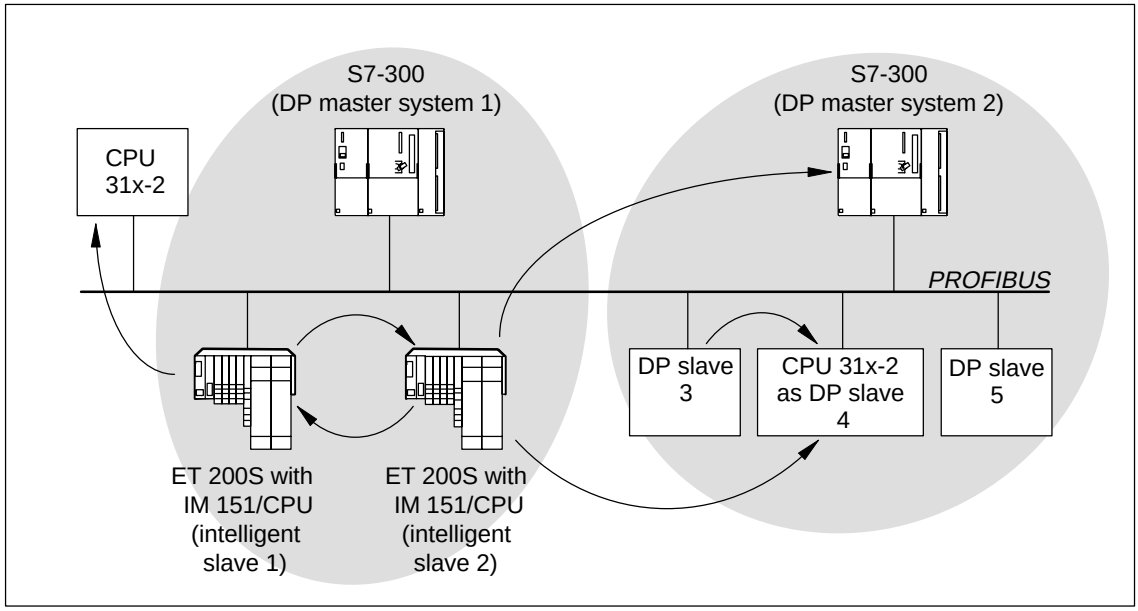


Figure 3-6 Direct Communication with the IM 151/CPU

Functionality in Direct Communication

The IM 151/CPU offers the following functionality in direct communication:

- Passive sender:
  When requested by the DP master, the IM 151/CPU, as the DP slave, sends the process outputs configured for direct communication as a broadcast frame to all bus nodes. Other recipients filter the relevant data from this broadcast frame.
- Recipient:
  Filtering of the data (1 to 32 bytes) from the broadcast frame from a maximum of 8 senders configured by means of *STEP 7* as relevant for direct communication. One pick-up is possible per sender.

Note

The system load increases with the number of configured recipients. The total system load is specified as a multiplying factor for the run times of the individual statements.

Diagnostics in Direct Communication

Only the results of connection monitoring can be used in the diagnostics of the DP slaves configured for direct communication, because diagnostic messages of the DP slaves that have been listened in on are only reported to the DP master.

The asynchronous OB 86 is called in the event of station failure and reintegration. If data is accessed during a station failure of the sender, an I/O access error is detected and OB 122 is called. Only the identifiers “module plugged” and “module available” are relevant for the module status data.

Commissioning and Diagnostics

4

Configuring the IM 151/CPU with *STEP 7*

This chapter outlines how to configure an ET 200S for the IM 151/CPU with *STEP 7*.

Resetting the Memory of the IM 151/CPU

Under certain circumstances you have to reset the memory of the CPU component of the IM 151/CPU. This chapter describes these circumstances and the procedure for resetting the memory of the CPU component.

Diagnostic Options

The ET 200S distributed I/O system is designed to make handling and commissioning as simple as possible. If a fault or an error should occur in spite of this, you can analyze it using the LEDs, the slave diagnosis and the diagnostic options in *STEP 7*.

Interrupt Evaluation

To help you evaluate the **interrupts** of the ET 200S, we will examine the difference between these and the interrupts of the S7/M7 DP master and other DP masters.

Chapter Overview

In Section	Contents	Page
4.1	Configuring the IM 151/CPU	4-2
4.2	Resetting the Memory of the IM 151/CPU	4-4
4.3	Commissioning and Startup of the ET 200S	4-7
4.4	Diagnostics Using LEDs	4-9
4.5	Diagnostics via the Diagnostic Address with *STEP 7*	4-11
4.6	Slave Diagnosis	4-14
4.7	System Status List (SSL)	4-24

4.1 Configuring the IM 151/CPU

Configure the IM 151/CPU interface module as a DP slave or as a stand-alone module.

The IM 151/CPU is presented to the user in *STEP7* as an S7-300 module that is always created together with a rack in an S7-300 station. Similarly, the module can only be deleted with the rack!

Expansion racks cannot be configured in an S7-300 station that contains an IM151/CPU. The IM 151/CPU is positioned at slot 2 and receives a DP submodule. This configuration applies both to the variant with RS 485 and the variant with a fiber-optic cable connection. The first plug-in modules can be configured as of slot 4.

The following configuration options are available:

Table 4-1 Configuration Options

Configuration Environment	Configuration Tool	Configurable Operating Mode
SIMATIC S7	*STEP 7* (HWCONFIG) as of V5.1	• Stand-alone • IM 151/CPU as S7 slave
SIMATIC S5	COM PROFIBUS	Fully configured and programmed IM 151/CPU, integrated as a standard intelligent slave in COM PROFIBUS
Non-Siemens systems	Non-Siemens tool	Fully configured and programmed IM 151/CPU, integrated as a standard intelligent slave in a non-Siemens tool

Prerequisite

You must have *STEP 7* (as of V 5.1) open and be working in *STEP 7* SIMATIC Manager.

Procedure

Proceed as follows:

1. Configure the ET 200S (with the IM 151/CPU) as an S7-300 station.
   - Create a new station of the type **S7-300** (**Insert** → **Station**) menu command).
   - Change to the hardware configuration window for this station.
   - In the “Hardware Catalog” window, select the PROFIBUS-DP/ET 200S/IM 151/CPU folder.
   - Drag and drop the “IM 151/CPU” object in the empty station window.
   - Configure the ET 200S with the required I/O modules.
   - Save the station (i.e. the ET 200S).
2. Configure a DP master (e.g. CPU with integrated PROFIBUS-DP interface or CP 342-5 with PROFIBUS-DP interface as of 6GK7 342-5DA01-0XE0, version 2) in another station in the same project.
3. Drag the ET 200S (with the IM 151/CPU) from the “Hardware Catalog” window (from the **configured stations**) container and drop it on the icon for the DP master system.
4. Double-click the intelligent DP slave icon, and select the “Interconnecting” tab. Specify on this tab which station is to represent the intelligent DP slave.
5. Select the intelligent DP slave, and click the “Interconnect” button.
6. Select the (slave) configuration tab, and assign the master and slave addresses.
7. Click ”OK” to accept the settings.
8. The two stations must then be reloaded to start master-slave communication.

Configuration in a Non-Siemens System

Using the DDB file you can also integrate the IM 151/CPU in non-Siemens systems as a DP standard slave. In this case the diagnostic frame consists of the following:

- Station Status
- Master PROFIBUS Address
- Manufacturer ID
- Module Diagnostics
- Module Status

4.2 Resetting the Memory of the IM 151/CPU

When Do You Reset the Memory of the IM 151/CPU?

You must reset the memory of the IM 151/CPU if an entire new user program is to be transferred to the CPU or if the STOP indicator is flashing at 1-second intervals.

The following are possible reasons for the MRES request:

- The ET 200S is starting up for the first time.
- The backup memory is inconsistent.
- The user memory is inconsistent.
- The memory module has been replaced.

How Do you Reset the Memory?

There are two ways of resetting the IM 151/CPU:

Table 4-2 Ways to Reset the Memory

Resetting the Memory with the Mode Selector	Resetting the Memory with the Programming Device
Described in this chapter.	Only possible during CPU STOP (see the programming device manuals and the *STEP 7 online help* system)

Resetting the Memory of the IM 151/CPU with the Mode Selector

To reset the memory of the IM 151/CPU using the mode selector, proceed as follows (see also Figure 4-1):

1. Switch the mode selector to the STOP position.
2. Depress the mode selector in the MRES position. Leave it in this position until the STOP LED comes on for the second time (corresponds to 3 seconds), and then let it snap back to the STOP position again.
3. Within 3 seconds, you must press the mode selector back to the MRES position and hold it in this position until the STOP LED flashes (at 2 Hz). When the IM 151/CPU has completed the memory reset, the STOP LED stops flashing and remains on.

   The IM 151/CPU has reset the memory.

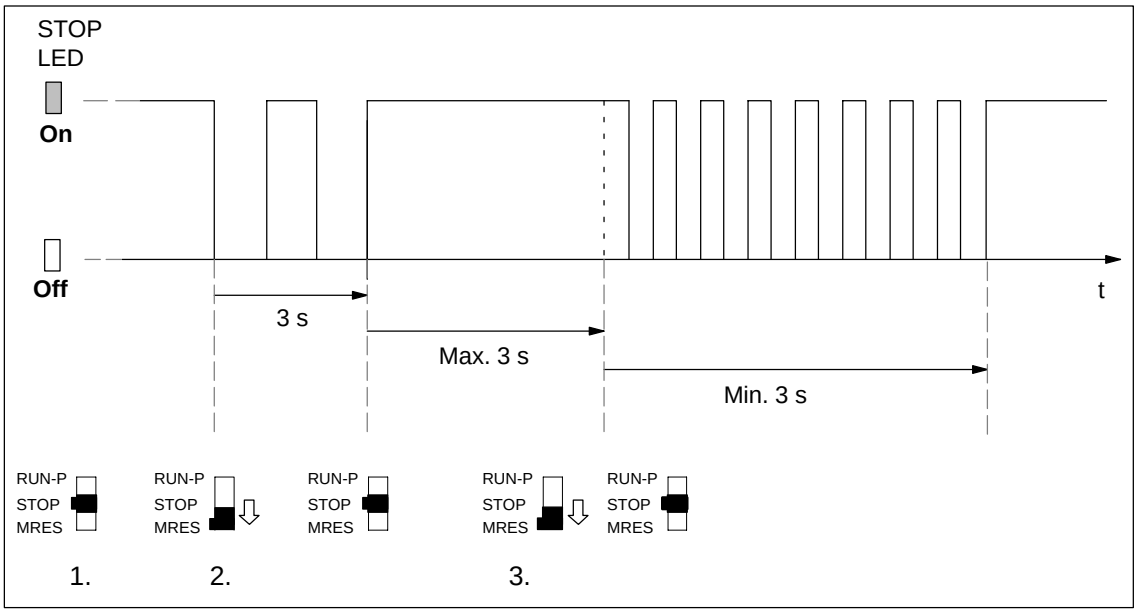


Figure 4-1 How to Use the Mode Selector to Reset the Memory

Is the STOP LED Not Flashing at Memory Resetting?

Does the STOP LED not flash during memory reset or do other indicators come on? You must repeat steps 2 and 3. If the IM 151/CPU still doesn't execute a memory reset, you have to read out the diagnostic buffer of the CPU component with the programming device (see the *STEP 7 user manual*).

What Happens in the CPU Component of the IM 151/CPU?

Table 4-3 Internal CPU Events at Memory Resetting

Event	Response of the CPU in the IM 151/CPU
Sequence of operations in the CPU component of the IM 151/CPU	1. The CPU deletes the entire user program in working memory and the load memory. 2. The CPU deletes the backup memory (retentive data). 3. The CPU tests its own hardware. 4. If you have inserted a memory module (micro memory card = MMC), the CPU copies the relevant contents of the module to the working memory.
Memory contents after reset	The user program from the memory module is transferred to the working memory again.
What's left?	The contents of the diagnostic buffer and the runtime meter.

Note

If the CPU cannot copy the contents of the memory module and requests a memory reset:

- Remove the memory module.
- Reset the CPU memory.
- Read out the diagnostic buffer.

You can read out the diagnostic buffer with the programming device (see the *STEP 7 online help system*).

4.3 Commissioning and Startup of the ET 200S

Commissioning the ET 200S

Commission the ET 200S distributed I/O system as follows:

1. Install the ET 200S distributed I/O system
   (see the *ET 200S Distributed I/O System* manual).
2. Set the PROFIBUS address on the IM 151/CPU if you don't require stand-alone operation
   (see the *ET 200S Distributed I/O System* manual).
3. Wire the ET 200S distributed I/O system
   (see the *ET 200S Distributed I/O System* manual).
4. During configuration as a DP slave, specify in the configuration software the address areas in the IM 151/CPU via which data exchange with the DP master is to take place (or use the ET 200S default setting; see Section 2.4).
5. Switch on the sensor supply voltage for the ET 200S.
6. If necessary, switch on the load voltage and the supply voltage for the motor starters.
7. If necessary, switch the CPU component of the IM 151/CPU to STOP mode.
8. Download the configuration for the IM 151/CPU to the ET 200S.
9. Switch the IM 151/CPU to RUN mode.

Backing Up the User Program

When the ET 200S is commissioned, the user program is still not secure against a power failure, because it is only in the RAM load memory. To store the user program in a powerfail-proof location, you have the following options:

- Copy RAM-to-ROM:
  Insert an empty memory module before the program is downloaded, and acknowledge the memory reset request this causes. RAM to ROM copying is initiated via the programming device.
- Download the user program:
  The program is downloaded from the programming device to the memory module inserted in the CPU by means of the “Load User Program” function.
- Transfer by means of the memory module:
  The user program is transferred to the memory module on the programming device. The memory module is then inserted in the CPU and the memory reset request acknowledged.

See Section 5.4, ”Memory Module”.

Tip: Program OB 82 and 86 During Commissioning

Always program OB 82 and 86 when carrying out commissioning as a DP slave with *STEP 7* in the DP master and the DP slave. This allows you to detect and evaluate operating modes or interruptions in user data transfer (see Tables 4-5 and 4-6).

Note

Without configuration, a default startup is possible if the power modules are switched on and all the modules are inserted.

4.4 Diagnostics Using LEDs

LEDs

The RUN, STOP, ON, BF, SF and FRCE LEDs display important information on the states of the IM 151/CPU to the user.

The IM 151/CPU has the following 6 LEDs:

- ”SF” LED (**S**ystem **F**ault) for indicating the presence of a fault in the ET 200S
- ”BF” LED (**B**us **F**ault) for indicating faults on the PROFIBUS-DP
- ”ON” LED for indicating that the ET 200S is connected to a power supply
- ”FRCE” LED comes on if a force request is active.
- “RUN” LED for indicating that the CPU component of the IM 151/CPU is in RUN mode
- “STOP” LED for indicating that the CPU component of the IM 151/CPU is in STOP mode

The meaning of the LEDs for CPU functionality is described in detail in Section 5.2.

“ON” LED Is Off

If the “ON” LED is off, either no supply voltage or insufficient supply voltage is being applied to the electronic components/sensors of the ET 200S. The cause is likely to be a defective fuse or inadequate or nonexistent system voltage.

Diagnosis of DP Functionality Using the “BF” and “SF” LEDs

If the “BF” and “SF” LEDs light up or flash, the ET 200S is not configured correctly. The table below shows you the possible error indications together with their meanings and the necessary action.

The table below shows the LED states for DP slave operation. DP functionality is irrelevant in stand-alone operation, and a BF LED is not activated (there is no LED for transmission rate detection).

Table 4-4 LEDs for PROFIBUS-DP

”BF” LED	”SF” LED	Description	Cause	Error handling
On	On	No connection to the DP master	• Bus connection interrupted • Master does not exist or is switched off SF is on due to station failure	• Check that the connector for the PROFIBUS-DP is inserted correctly. • Check whether the bus cable to the DP master is defective.
LED flashes	On	Parameter assignment error; there is no data exchange	• Slave not configured or incorrectly configured • Incorrect but permissible station address set • Configured address areas of the actual configuration not identical to the target configuration • Station failure of a configured sender in direct data communication • DP master does not exist/is switched off, but a connection to the programming device/OP exists	• Check the hardware of the ET 200S. • Check the configuration and parameterization of the ET 200S. • Check the setting for the configured address areas for the master and slave.
*	On	Slave error: impermissible station address	• Impermissible station address set (0, 126,127,...) • Set station address differs from the configured address	Set the PROFIBUS address in the correct range using the DIL switch block in the basic module.
LED off	On	Slave error: diagnostic interrupt	Master in STOP	Switch the DP master to RUN mode.
LED off	LED off	Data exchange taking place	The target configuration and actual configuration of the ET 200S match.	

* The status is not relevant.

4.5 Diagnostics via Diagnostic Address with *STEP 7*

Malfunctions that occur in the ET 200S are indicated by the “SF” LED, and the cause is entered in the diagnostic buffer of the IM 151/CPU. Either the CPU component of the IM 151/CPU goes into STOP mode, or you can respond to errors by means of error or interrupt OBs in the user program.

To enable a response to be made, it must be possible to identify whatever caused the problem by means of a diagnostic address.

Diagnostic Addresses

If you run the ET 200S with a DP master from the SIMATIC S7 range on the PROFIBUS-DP, diagnostic addresses are assigned in *STEP 7* as follows:

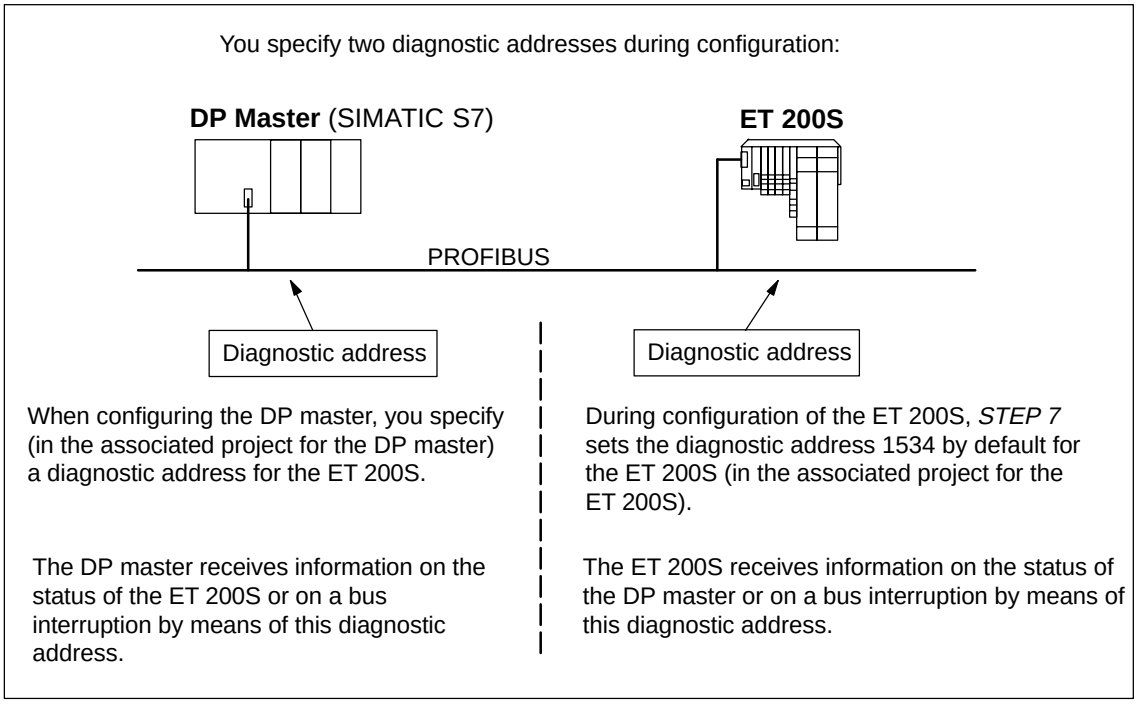


Figure 4-2 Diagnostic Addresses for the DP Master and ET 200S

Event Identification

The following table indicates how the DP master or the IM 151/CPU of the ET 200S identifies changes in operating mode and interruptions in user data transfer.

Table 4-5 Responses to Operating Mode Changes and Interruptions in User Data Transfer in the DP Master and the ET 200S with the IM 151/CPU

Event	What Happens ...	
	In the DP Master	**In the IM 151/CPU**
Bus interruption (short-circuit, connector removed)	• OB 86 is called with the message *Station failure* (incoming event; diagnostic address of the IM 151/CPU) • During I/O access to the transfer area: OB 122 is called (I/O access error)	• OB 86 is called with the message *Station failure* (incoming event; diagnostic address of the IM 151/CPU) • During I/O access to the transfer area: OB 122 is called (I/O access error)
ET 200S: RUN → STOP	• OB 82 is called with the message *Faulty module* (incoming event; diagnostic address of the IM 151/CPU; variable OB82_MDL_STOP=1)	–
ET 200S: STOP → RUN	• OB 82 is called with the message *Module ok* (outgoing event; diagnostic address of the IM 151/CPU; variable OB82_MDL_STOP=0)	–
DP master: RUN → STOP	–	• OB 82 is called with the message *Faulty module* (incoming event; diagnostic address of the IM 151/CPU; variable OB82_MDL_STOP=1)
DP master: STOP → RUN	–	• OB 82 is called with the message *Module ok* (outgoing event; diagnostic address of the IM 151/CPU; variable OB82_MDL_STOP=0)

Evaluation in the User Program

The table below shows you how to evaluate RUN/STOP transitions in the DP master (SIMATIC S7) or ET 200S, for example.

Table 4-6 Evaluation of RUN/STOP Transitions in the DP Master/ET 200S

In the DP Master		In the ET 200S (IM 151/CPU)
Diagnostic addresses: (example) Master diagnostic address=1023 Slave diagnostic address in the master system=**1022**		Diagnostic addresses: (example) Slave diagnostic address=**1534** Master diagnostic address=not relevant
The CPU calls OB 82 with the following information: • OB 82_MDL_ADDR:=**1022** • OB82_EV_CLASS:=B#16#39 (incoming event) • OB82_MDL_DEFECT:=Module fault Tip: This information is available in the diagnostic buffer of the CPU In the user program, you should also program SFC 13 (“DPNRM_DG”) to read out the slave diagnosis.	←	CPU in IM 151/CPU: RUN → STOP The CPU generates a diagnostic frame (slave diagnosis; see the *ET 200S Distributed I/O System* manual).
CPU: RUN → STOP	→	The CPU component in the IM 151/CPU calls OB 82 with information including the following: • OB 82_MDL_ADDR:=**1534** • OB82_EV_CLASS:=B#16#39 (incoming event) • OB82_MDL_DEFECT:=Module fault Tip: This information is also contained in the diagnostic buffer of the CPU component

4.6 Slave Diagnosis

Interrupts with *STEP 7* and an S7/M7 DP Master

You can trigger a process interrupt for the DP master from the user program of the IM 151/CPU.

When you call SFC 7 (“DP_PRAL”), you activate an OB 40 in the user program of the DP master. SFC 7 allows you to forward interrupt information in a double word to the DP master; this information can then be evaluated in OB 40 in variable OB40_POINT_ADDR. You can program the interrupt information as desired. A detailed description of SFC 7 “DP_PRAL” can be found in the reference manual entitled *System Software S7-300/400 - System and Standard Functions*.

Interrupts with *STEP 7* and Another DP Master

If you run the IM 151/CPU with another DP master, interrupts are appear in the station diagnosis of the IM 151/CPU (see Section 4.6.6). You must continue processing the results of the diagnosis in the user program of the DP master.

Note

Note the following in order to be able to evaluate the diagnostic interrupt and process interrupt via the station diagnosis when using a different DP master:

- The DP master should be able to store diagnostic messages; this means that diagnostic messages should be stored within the DP master in a ring buffer store. If the DP master cannot store the diagnostic messages, only the most recently received diagnostic message, for example, is stored.
- With an IM 308-C as the DP master you cannot use process interrupts in the station diagnosis.

Reference

You can find more information on the generally applicable slave diagnosis for the ET 200S in the relevant section of the *ET 200S Distributed I/O System manual.* You can also find out there which blocks you can use to request the slave diagnosis.

Structure of the Slave Diagnostic Data

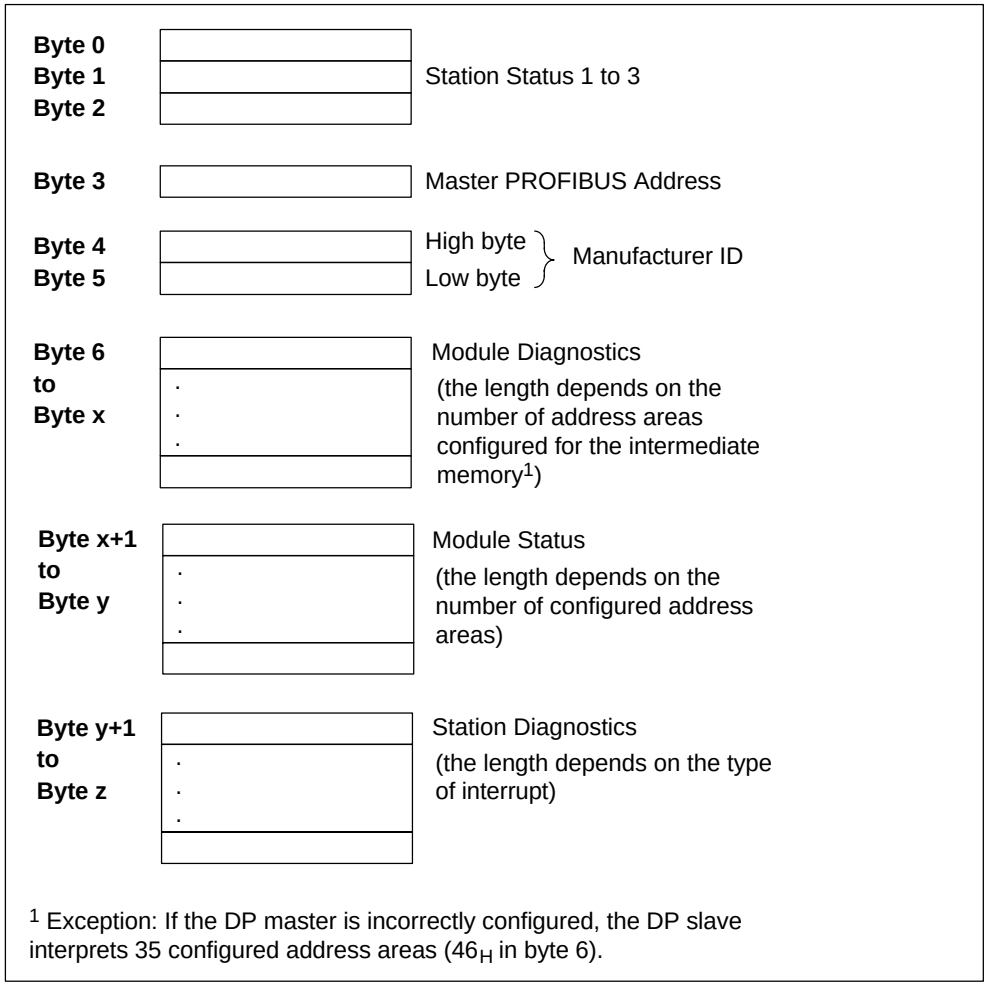


Figure 4-3 Format of the Slave Diagnostic Data

4.6.1 Station Status 1 to 3

Definition

Station status 1 to 3 provides an overview of the status of a DP slave.

Station Status 1

Table 4-7 Structure of Station Status 1 (Byte 0)

Bit	Description	Remedy
0	1: DP slave **cannot** be addressed by DP master.	• Is the correct DP address set on the DP slave? • Is the bus connector inserted? • Does the DP slave have power? • Is the RS 485 repeater correctly set? • Execute a Reset on the DP slave.
1	1: DP slave is not ready for data interchange.	• Wait; the DP slave is still doing its run-up.
2	1: The configuration data which the DP master sent to the DP slave do not correspond with the DP slave's actual configuration.	• Was the software set for the right station type or the right DP slave configuration?
3	1: Diagnostic interrupt, generated by a RUN/STOP transition on the CPU 0: Diagnostic interrupt, generated by a STOP/RUN transition on the CPU	• You can read out the diagnostic data.
4	1: Function is not supported, for instance changing the DP address at the software level.	• Check the configuring data.
5	0: This bit is always "0".	–
6	1: DP slave type does not correspond to the software configuration.	• Was the software set for the right station type? (parameter assignment error)
7	1: DP slave was parameterized by a different DP master to the one that currently has access to it.	• Bit is always "1" when, for instance, you are currently accessing the DP slave via the PG or a different DP master. The DP address of the master that parameterized the slave is located in the "Master PROFIBUS address" diagnostic byte.

Station Status 2

Table 4-8 Structure of Station Status 2 (Byte 1)

Bit	Description
0	1: DP slave must be parameterized again and reconfigured.
1	1: A diagnostic message has arrived. The DP slave cannot continue operation until the error has been rectified (static diagnostic message).
2	1: This bit is always “1” when there is a DP slave with this DP address.
3	1: The watchdog monitor has been activated for this DP slave.
4	0: This bit is always “0”.
5	0: This bit is always “0”.
6	0: This bit is always “0”.
7	1: DP slave is deactivated, that is to say, it has been removed from the scan cycle.

Station Status 3

Table 4-9 Structure of Station Status 3 (Byte 2)

Bit	Description
0 to 6	0: These bits are always “0”.
7	1: • More diagnostic messages have arrived than the DP slave can buffer. • The DP master cannot enter all the diagnostic messages sent by the DP slave in its diagnostic buffer.

4.6.2 Master PROFIBUS Address

Definition

The DP address of the DP master is stored in the master PROFIBUS address diagnostic byte:

- The master that parameterized the DP slave
- The master that has read and write access to the DP slave

Master PROFIBUS Address

Table 4-10 Structure of the Master PROFIBUS Address (Byte 3)

Bit	Description
0 to 7	DP address of the DP master that parameterized the DP slave and has read/write access to that DP slave.
	FF_H: DP slave was not parameterized by a DP master.

4.6.3 Manufacturer ID

Definition

The manufacturer identification contains a code specifying the DP slave's type.

Manufacturer ID

Table 4-11 Structure of the Manufacturer Identification (Bytes 4 and 5)

Byte 4	Byte 5	Manufacturer Identification for
80_H	$6C_H$	IM 151/CPU
80_H	$6D_H$	IM 151/CPU FO

4.6.4 Module Diagnosis

Definition

The module diagnosis indicates for which of the configured address areas of the intermediate memory an entry has been made. The figure below shows the structure of the module diagnosis.

Structure

The figure below shows the structure of the module diagnosis.

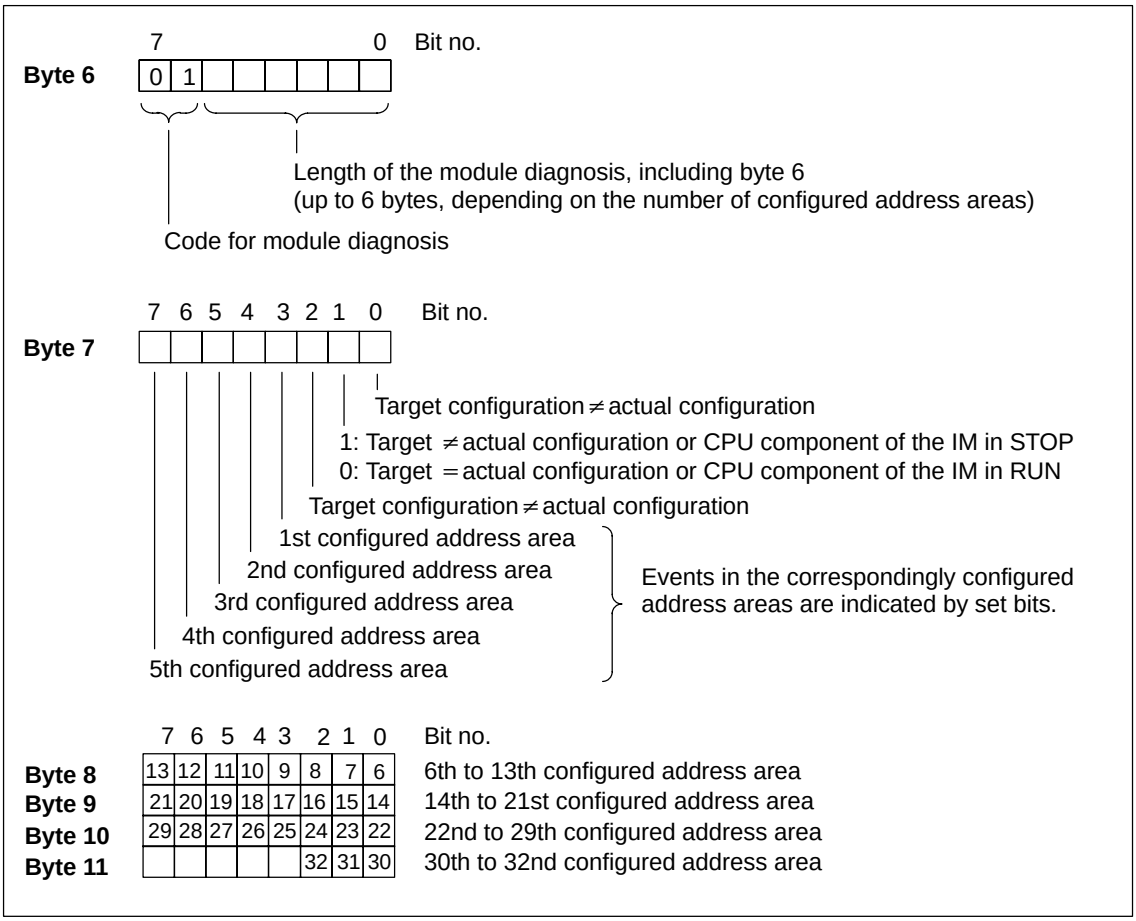


Figure 4-4 Structure of the Module Diagnosis

4.6.5 Module Status

Definition

The module status, as a particular form of module diagnosis, indicates the status of the configured address areas of the intermediate memory and expands on the module diagnosis. The module status starts after the module diagnosis and varies in length according to the number of configured address areas.

Structure

The module status of IM 151/CPU is structured as follows:

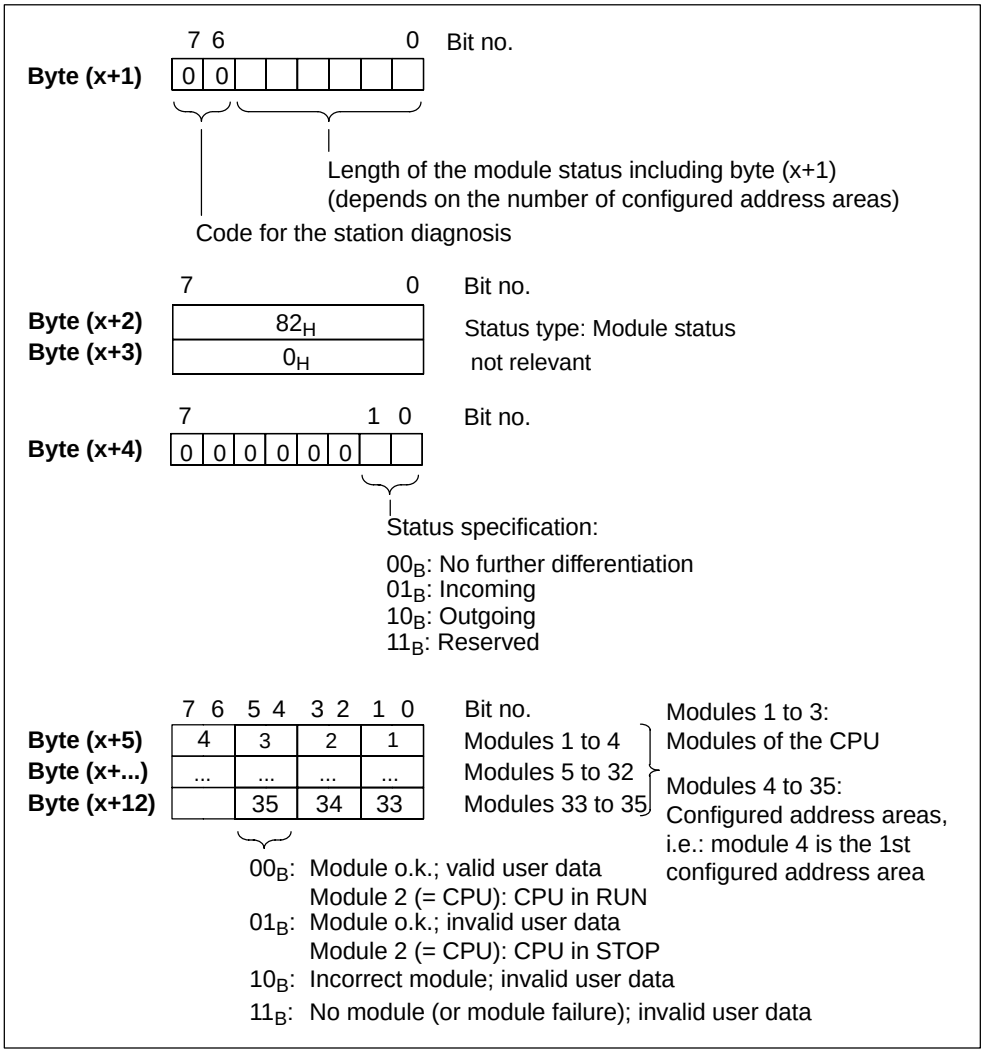


Figure 4-5 Structure of the Module Status

4.6.6 Structure of the Station Diagnosis

The device-specific diagnosis is **only** issued in S7 slave operating mode.

Definition

The station diagnosis provides detailed information on a DP slave. The station diagnosis starts after the module status and contains a maximum of 20 bytes for the IM 151/CPU, depending on the slave operating mode and the parameter assignment frame.

The following generally applies: If an error occurs, the corresponding bit is set to “1”.

Structure

The figure below illustrates the structure and contents of the bytes for the IM 151/CPU.

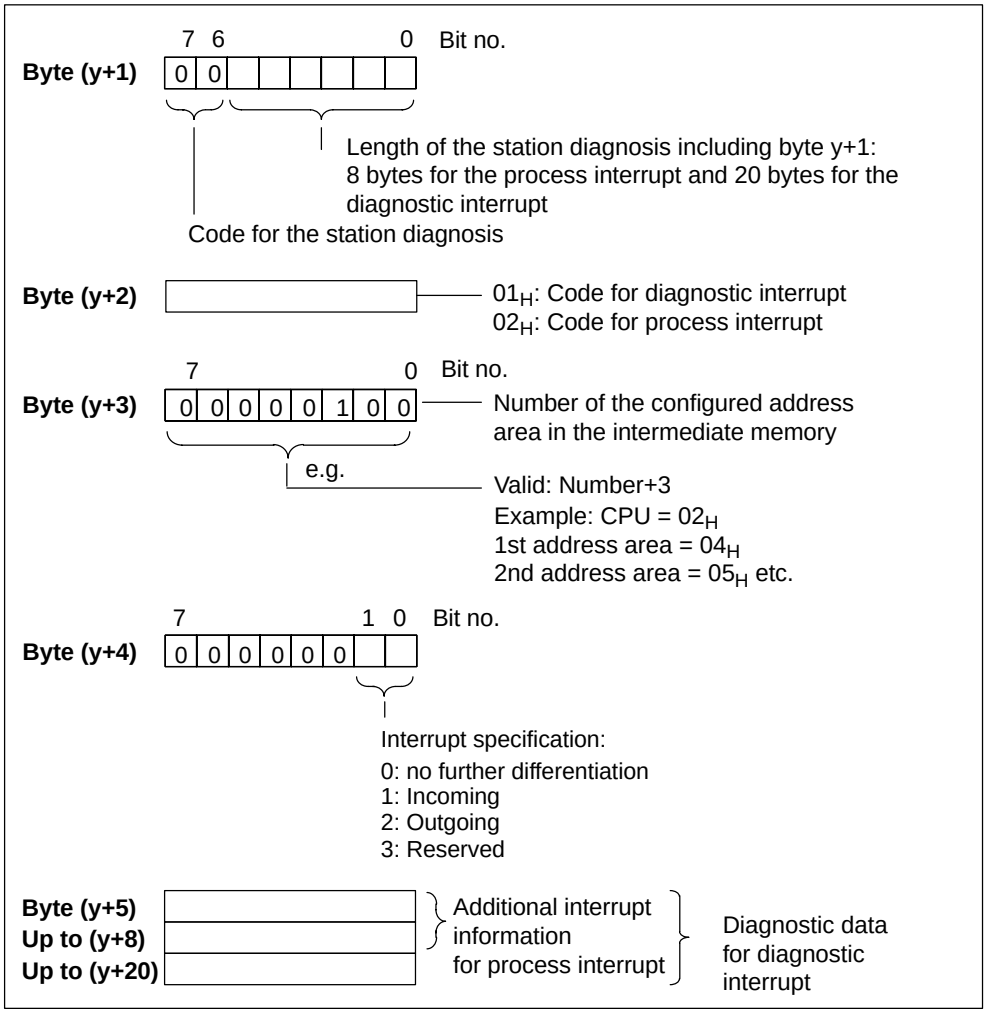


Figure 4-6 Structure of the Station Diagnosis (1)

Additional Interrupt Information and Diagnostic Data

The meaning of the bytes as of byte (y+5) depends on byte (y+2):

Table 4-12 Additional Interrupt Information and Diagnostic Data

Byte (y+2) Contains the Code for...	
Diagnostic Interrupt (01_H)	**Process Interrupt (02_H)**
The diagnostic data contains the 16 bytes of status information of the CPU component of the IM 151/CPU. Figure 4-7 shows the contents of the first four bytes of the diagnostic data. The next 12 bytes are always 0.	For the process interrupt, you can program four bytes of interrupt information. You transfer these 4 bytes to the DP master in *STEP 7* using SFC 7 (“DP_PRAL”) (see Section 4.5).

Diagnostic Data

The figure below shows the structure and contents of the bytes for the diagnostic interrupt. Bytes (y+8) to (y+20) are set to “0”.

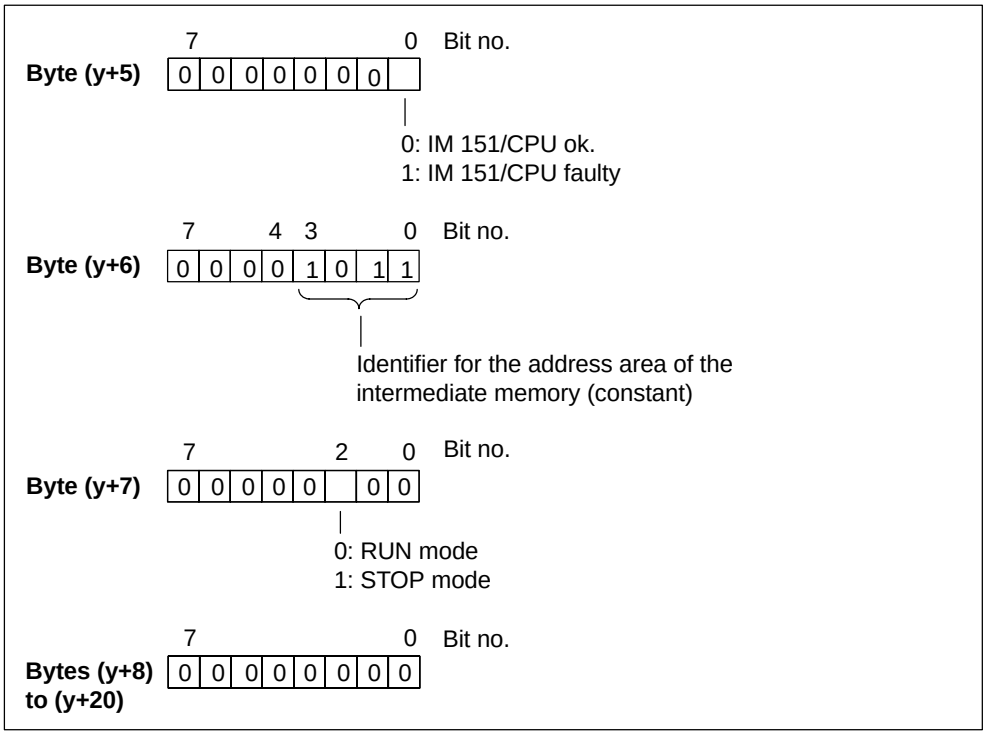


Figure 4-7 Structure of the Station Diagnosis (2)

4.7 System Status List (SSL)

Possible System Status Sublists

The following table lists all the possible sublists with the associated sublist extracts and the SSL IDs.

Details on how to read out the SSL using, for example, SFC 51, and more information on the contents can be found in the following:

- In the *System Software for S7-300/400* reference manual, in the chapter entitled *System and Standard Functions, System Status List SSL*
- In the STEP 7 online help system, context-sensitive help on the SFBs/SFCs.

Table 4-13 SSL Sublists of the IM 151/CPU

Sublist	SSL ID Sublist	Sublist Extract	SSL ID Extract
Module identification	W#16#xy11	CPU identification	W#16#0111
CPU features	W#16#xy12	All features Features of a group SSL sublist header information only	W#16#0012 W#16#0112 W#16#0F12
User memory areas	W#16#xy13	Records of all user memory areas	W#16#0013
System areas	W#16#xy14	Records of all system areas	W#16#0014
Block types	W#16#xy15	Records of all block types Records of all OBs	W#16#0015 W#16#0115
Status of the module LEDs	W#16#xy19	Status of all LEDs SSL sublist header information only	W#16#0019 W#16#0F19
Interrupt status	W#16#xy22	Record for the specified interrupt	W#16#0222
Communication: status data	W#16#xy32	Status data for a communication component Status data for a communication component	W#16#0132 W#16#0232
Status of the module LEDs	W#16#xy74	Status of all LEDs Status of an LED SSL sublist header information only	W#16#0074 W#16#0174 W#16#0F74

Table 4-13 SSL Sublists of the IM 151/CPU

Sublist	SSL ID Sublist	Sublist Extract	SSL ID Extract
Module status information	W#16#xy91	Module status information of all inserted modules	W#16#0A91
		Module status information of a module in a central configuration or at a integrated DP interface modules	W#16#0C91
		Module status information of all modules in the specified mounting rack/DP station	W#16#0D91
Mounting rack/station status information	W#16#xy92	Preset status in the central configuration of the master system	W#16#0092
		Actual status in the central configuration of the master system	W#16#0292
		OK status of the expansion units of the master system	W#16#0692
		SSL sublist header information only	W#16#0F92
Diagnostic buffer	W#16#xyA0	All the entries that can be delivered in the operating mode	W#16#00A0
		The most recent entries	W#16#01A0
Module diagnostic information (DS 0)	W#16#00B1	–	–
Module diagnostic information (DS 1), physical address	W#16#00B2	–	–
Module diagnostic information (DS 1), logical address	W#16#00B3	–	–

Functions of the IM 151/CPU

5

In This Chapter

In this chapter you will find:

- Important features of the IM 151/CPU for the PROFIBUS-DP
- A list of the CPU functions of the IM 151/CPU that you can call with *STEP 7*, such as the integrated clock, blocks for the user program and parameters that can be set

Chapter Overview

In Section	Contents	Page
5.1	Data for the PROFIBUS-DP	5-2
5.2	The Mode Selector and LEDs	5-4
5.3	Force	5-6
5.4	Memory Module	5-8
5.5	Clock	5-12
5.6	Blocks	5-13
5.7	Parameters	5-16
5.8	Parameterization of the Reference Junction for the Connection of Thermocouples	5-18
5.8	Removal and Insertion of Modules During Operation	5-20
5.10	Switching Power Modules Off and On During Operation	5-22

5.1 Data for the PROFIBUS-DP

DDB (Device Database) File

A DDB file contains all the slave-specific properties. The structure of the DDB file is defined in IEC 61158/EN 50170, Volume 2, PROFIBUS.

You only need the DDB file if:

- You are using the ET 200S with a DP master from the SIMATIC S5 range (configuration with COM PROFIBUS).
- You are using the ET 200S with a non-SIMATIC DP master (configuration with a non-Siemens tool).

You can download the DDB file from the Internet. You will find all the DDB files under “Downloads” on the SIMATIC Customer Support web site:

- http://www.ad.siemens.de/support/html_00/index.shtml
- http://www.ad.siemens.de/csi/gsd

Alternatively, you can obtain the DDB file via modem on +49 (0)911-737972 or at CompuServe in AUTFORUM (`GO AUTFORUM`).

Important Features

If you do not have the DDB file to hand, the following table lists the most important features of the IM 151/CPU.

Tabelle 5-1 Attributes of the Device Database (DDB) File

Feature	DP Keyword to IEC 61158/EN 50170, Volume 2, PROFIBUS	IM 151/CPU
Manufacturer ID	Ident_Number	$806C_H$ $806D_H$ (FO)
Supports FMS	FMS_supp	No
Supports 9.6 kbps	9.6_supp	Yes
Supports 19.2 kbps	19.2_supp	Yes
Supports 45.45 kbps	45.45_supp	Yes
Supports 93.75 kbps	93.75_supp	Yes
Supports 187.5 kbps	187.5_supp	Yes
Supports 500 kbps	500_supp	Yes
Supports 1.5 Mbps	1.5M_supp	Yes
Supports 3 Mbps	3M_supp	Yes No (FO)
Supports 6 Mbps	6M_supp	Yes No (FO)

Tabelle 5-1 Attributes of the Device Database (DDB) File

Feature	DP Keyword to IEC 61158/EN 50170, Volume 2, PROFIBUS	IM 151/CPU
Supports 12 Mbps	12M_supp	Yes
Supports the FREEZE control command	Freeze_Mode_supp	Yes
Supports the SYNC control command	Sync_Mode_supp	Yes
Supports automatic baud rate detection	Auto_Baud_supp	Yes
PROFIBUS address modifiable by software	Set_Slave_Add_supp	No
Length of user-specific parameter assignment data	User_Prm_Data_Len	3 bytes
User-specific parameter assignment data	User_Prm_Data	Yes
Minimum interval between two slave list rotations	Min_Slave_Intervall	1(100µs)
Modular device	Modular_Station	1
Maximum number of modules	Max_Module	35
Maximum number of inputs in bytes	Max_Input_Len	64
Maximum number of outputs in bytes	Max_Output_Len	64
Maximum combined number of inputs and outputs in bytes	Max_Data_Len	128
Central display of vendor-specific status and error messages	Unit_Diag_Bit	Via “ON” LED
Allocation of values in the station diagnostic field to texts	Unit_Diag_Area	Unassigned
Identifiers of all address areas for PROFIBUS	Module, End_Module	Yes
Allocation of vendor-specific error types in channel-specific diagnostic field to texts	Channel_Diag	No
Maximum length of the diagnostic data	Max_Diag_Data_Len	45 bytes

5.2 The Mode Selector and LEDs

Mode Selector

The mode selector of the IM 151/CPU is designed as a 3-step toggle switch (see below):

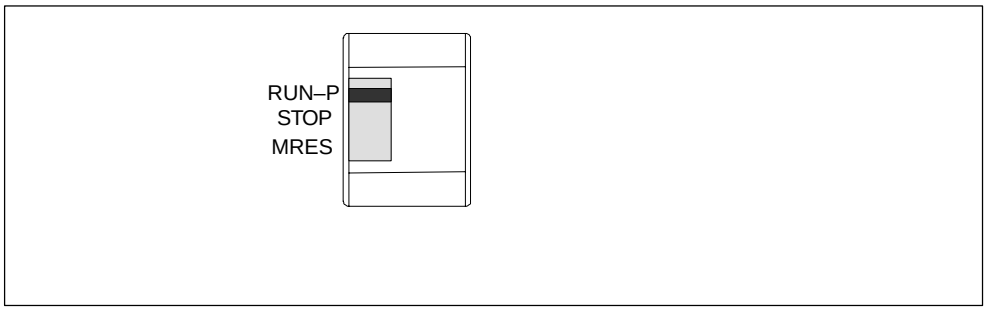


Figure 5-1 Mode selector

Positions of the Mode Selector

The positions of the mode selector are explained in the order in which they are arranged on the IM 151/CPU.

Table 5-2 Positions of the Mode Selector

Position	Description	Description
RUN-P	RUN-PROGRAM mode	The CPU processes the user program.
STOP	STOP mode	The CPU does not process the user program. Programs can: • Be read out from the CPU using a programming device (CPU → PG) • Transferred to the CPU (PG → CPU)
MRES	Reset CPU memory	Momentary-contact position of the mode selector for resetting the CPU memory. You must adhere to a specific sequence when resetting the CPU memory using the mode selector (see Section 4.2)

Meanings of the LEDs for CPU Functionality

For the CPU component of the IM 151/CPU there are 2 separate LEDs that indicate the operating modes of the CPU:

- RUN
- HALT
- STOP

You can obtain information on the power supply of the CPU, on force requests and on general errors via 3 additional LEDs.

Table 5-3 LEDs for CPU Functionality

LED	Description	Description
ON (green)	Power up	**Comes on** when the supply voltage is applied to the CPU
RUN (green)	RUN mode	**Shines continuously** when the CPU component is processing the user program. **Flashes** at 2 Hz during startup of the CPU component: • For at least 3 secs, but the startup of the CPU component can also be shorter. • During the startup of the CPU component, the STOP LED also lights up; when the STOP LED goes off, the outputs are enabled. **Flashes** at 0.5 Hz when the CPU has reached a breakpoint you have set. At the same time the STOP LED comes on
STOP (yellow)	STOP mode	**Comes on** when the CPU component: • Is not processing a user program • Has reached a breakpoint you have set At the same time the RUN LED flashes at 0.5 Hz **Flashes** at 0.5Hz, when the CPU component requests a memory reset (see Section 4.2)
FRCE (yellow)	Force job active	**Lights up** when a force request is active
SF (red)	Group error	**Lights up** in the event of • Programming errors • Parameter assignment errors • Calculation errors • Timing errors • I/O errors • Hardware errors • Firmware errors To determine the exact nature of the error/fault, you have to use a programming device and read out the contents of the diagnostic buffer.

Meanings of Other LEDs

The LEDs “SF” (from the PROFIBUS-DP viewpoint) and “BF” are described in Section 4.4.

5.3 Force

Force Test Function

In the case of the IM 151/CPU, you can preset the inputs and the outputs in the process image with fixed values using the “Force” function.

The values (force values) you have preset can still be controlled in the IM 151/CPU by the user program and by programming device/operator panel functions. This is shown in Figure 5-2.

You can force a maximum of 10 variables with the IM 151/CPU.

Caution

The force values in the process-image input table can be overwritten by write commands (for example T IB x, = I x.y, copy with SFC, etc.) as well as by I/O read commands (L PIW x, for example) in the user program or by PG/OP write functions.

Outputs preset with force values only return the force value provided the user program does not execute any write accesses to the outputs using I/O write commands (e.g. T PQB x) and provided no PG/OP functions write to these outputs.

It is essential to note that force values in the process-image input/output table cannot be overwritten by the user program or by PG/OP functions.

Principle Behind Forcing with the IM 151/CPU

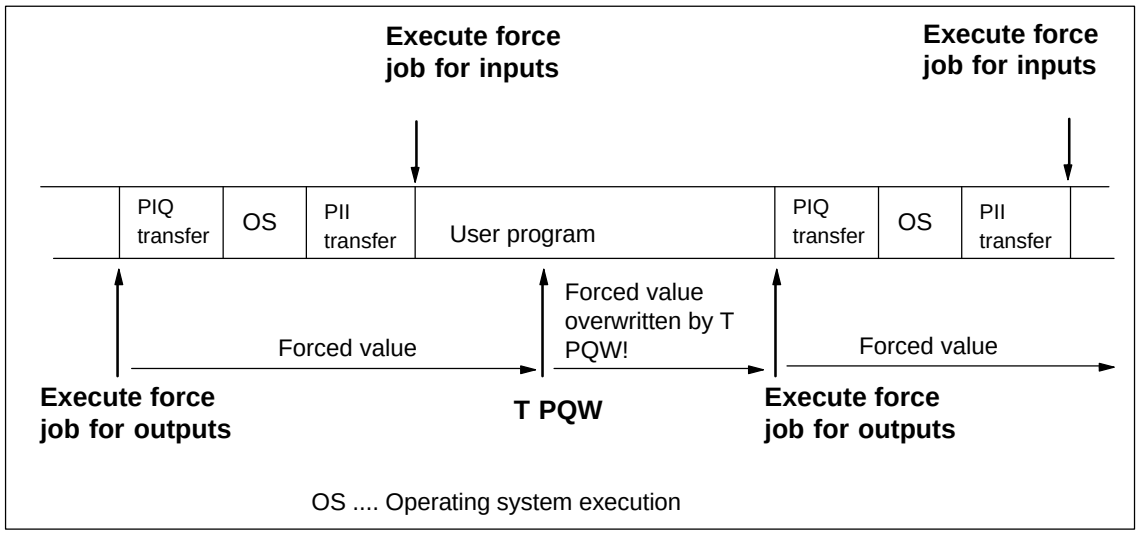


Figure 5-2 Principle Behind Forcing

Note

Due to insufficient battery backup, force requests do not survive after power off/power on.

Application Example

Prerequisite: There is no direct I/O access in your user program.

If, for example, an enable sensor (f) in your system is defective and it continually indicates a logical 0 to your user program, for example, via input 1.2, you can bridge this sensor by forcing the input to 1, ensuring that your system continues to operate.

However, because the sensor is out of operation, you must monitor the functionality by different means to avoid injury to the operator and damage to the machine.

5.4 Memory Module

Micro Memory Card

A micro memory card (MMC) is used as a memory module for the IM 151/CPU. The MMC module can be used as a load memory and portable data carrier. The following data is stored on the MMC:

- User program (all blocks)
- Configuration data (created with *STEP 7*)
- Data for a firmware update (operating system)

Features

The micro memory card has the following features:

- Compact, robust design
- Integrated write protection (temporary and permanent)
- Integrated error correction in the memory field
- Hot swapping is possible
- Fixed individual card identification
- Polarity reversal protection

Micro Memory Cards that Can Be Used

The following memory modules are available:

Table 5-4 Available MMCs

Type	Purpose	Order Numbers
MMC 64 KB	User memory	6ES7 953-8LF00-0AA0
MMC 2 MB	Firmware update, user memory	6ES7 953-8LL00-0AA0

Inserting/Changing the Card

The MMC is designed so that it can also be removed and inserted when the power is on. The chamfered edge of the MMC prevents the card being inserted the wrong way round (reverse polarity protection).

There is an eject button on the memory card slot to enable you to remove the card easily. To eject the card, you require a small screwdriver or ball-point pen.

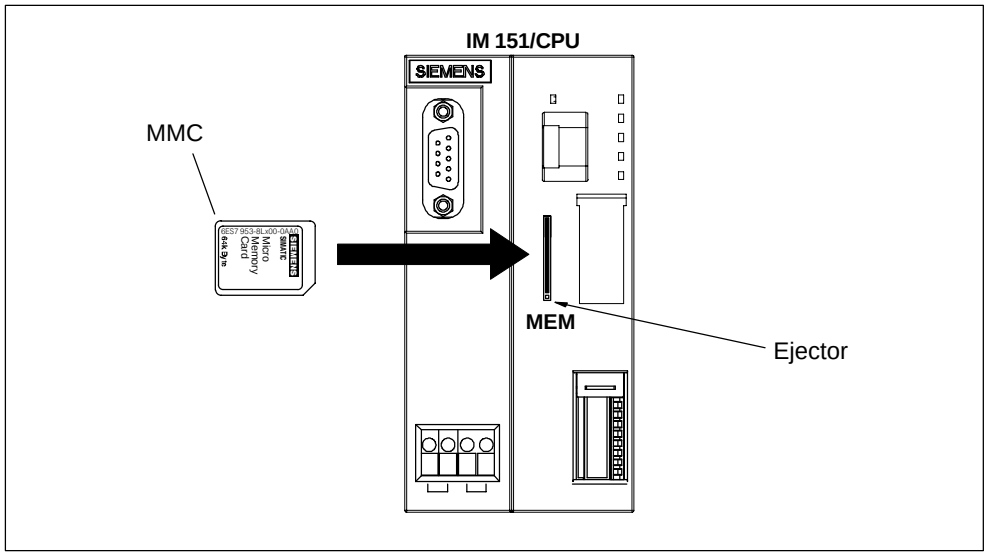


Figure 5-3 Position of the Memory Card Slot for the MMC on the IM 151/CPU

If a new MMC is inserted in the memory card slot, the CPU of the module requests a memory reset.

Note

The user program should be saved explicitly on the MMC. The MMC should always remain inserted because the IM 151/CPU doesn't have a backup battery. This ensures that the user program remains in the CPU even after a power failure (e.g. after power off).

Firmware Update with MMC

To update the firmware, proceed as follows:

Table 5-5 Firmware Update with MMC

Step	Action Required	What Happens in the IM 151/CPU:
1.	Transfer update files to a blank MMC using STEP 7 and your programming device.	-
2.	Deenergize the IM 151/CPU and insert the MMC with the FW update	-
3.	Switch the power on	The IM 151/CPU automatically detects the MMC with the FW update and starts the FW update. All the LEDs light up during the FW update except the FRCE LED. The STOP LED flashes after the FW update has been completed. In this way, the IM 151/CPU requests a memory reset.
4.	Deenergize the IM 151/CPU and remove the MMC with the FW update	-

Backing Up the Operating System on the MMC

To back up the operating system, proceed as follows:

Table 5-6 Backing Up the Operating System

Step	Action Required	What Happens in the IM 151/CPU:
1.	Insert a new micro memory card (2 MB) in the CPU	The CPU requests a memory reset
2.	Hold the mode selector in the MRES position.	–
3.	Switch the power off then on, and keep the mode selector in the MRES position until...	... STOP, RUN and FRCE LEDs start flashing
4.	Move the mode selector to STOP	–
5.	Move the mode selector briefly to MRES, then let it snap back to STOP	The IM 151/CPU starts to back up the operating system on the MMC. All the LEDs light up during backup. The STOP LED flashes after the backup has been completed. In this way, the IM 151/CPU requests a memory reset.
6.	Remove the micro memory card	-

5.5 Clock

The IM 151/CPU has an integrated software clock.

Setting, Reading and Programming the Clock

You set and read the clock using the programming device (see the *STEP 7*) User Manual) or program the clock in the user program using SFCs (see the *System and Standard Functions* Reference Manual and Appendix C).

Features

The table below indicates the features and functions of the clock.

When you parameterize the CPU in *STEP 7*, you can also set functions such as synchronization and correction factors (refer to the *STEP 7* online help system for information on how to do this).

Table 5-7 Features of the Clock

Features	IM 151/CPU
Type	Software clock
Manufacturer setting	DT#1994-01-01-00:00:00
Backup	Not possible
Operating hours counter	1

Behavior of the Clock When Power Is Off

At power on, the CPU clock continues from the time valid at power off.

5.6 Blocks

This section provides an overview of the blocks that run in the CPU component of the IM 151/CPU.

The operating system of the CPU is designed for event-driven processing of the user program. The following tables show which organization blocks (OBs) the operating system automatically invokes in response to which events.

More Information

You will find a detailed description of the blocks and the start events listed below in the *System and Standard Functions* Reference Manual. There is an overview of all the *STEP 7* documentation in the *ET 200S Distributed I/O System* manual.

Overview of All the Blocks

Table 5-8 Overview of the Blocks

Block	Number	Area	Maximum Size	Remarks
OB	13	–	8 KB	All the possible OBs are listed below this table.
FC	128	0 – 127		–
FB	128	0 – 127		–
DB	127	1 – 127		0 is reserved
SFC	48	–	–	You will find a list of all the SFCs of the CPU component in Appendix C.1.
SFB	7	–	–	You will find a list of all the SFBs of the CPU component in Appendix C.2.

OBs for Cycle and Restart

Table 5-9 OBs for Cycle and Restart

Cycle and Restart	Invoked OB	Possible Start Events
Cycle	OB 1	• First OB 1 after restart (power on or STOP-RUN transition of the IM 151/CPU) • Termination of the previous program cycle
Restart (change from STOP to RUN)	OB 100	• Manual restart request (STOP-RUN via mode selector or MPI command) • Automatic restart request (after power off to power on transition)

OBs for Interrupts

Table 5-10 OBs for Interrupts

Interrupts	Invoked OB	Possible Start Events
Time-of-day interrupt	OB 10	• Automatic start after setting and activation of the time-of-day interrupt using *STEP 7* • Activation via SFC 30 after setting with *STEP 7* or SFC 28
Delay Interrupt	OB 20	• Expiration of the time specified in SFC32
Watchdog interrupt (default: 100 ms clock pulse)	OB 35	• Depending on the clock pulse (parameter assignment with *STEP 7*)
Process interrupt	OB 40	• Activation of a process interrupt
Diagnostic interrupt	OB 82	• Outgoing event (cause of event no longer exists) • Incoming event (cause of event still exists)
Removal/insertion interrupt	OB 83	• Removal and insertion of modules in RUN mode

If OBs 10, 20, 40, 82, 83 are missing, or with OB 85 and the corresponding start event, the IM 151/CPU goes into STOP mode.

OBs for Error/Fault Responses

Table 5-11 OBs for Error/Fault Responses

Error/Fault	Invoked OB	Possible Start Events
Time-out	OB 80	• Cycle time exceeded • Acknowledgment error during OB processing • Putting the clock forward (time jump) to start an OB
Program execution error	OB 85	• Start event for an OB not loaded • I/O access error during system updating of the process image
Failure/reintegration of the DP master or a node in direct communication	OB 86	• Outgoing event (cause of event no longer exists) • Incoming event (cause of event still exists)
Programming error	OB 121	• An event occurring due to an error in program scanning, e.g. due to the call of a block not loaded in the CPU
Error during direct access to I/O	OB 122	• Error during read access • Error during write access

If OBs 80, 85, 86, 121 are missing, or with OB 85 and the corresponding start event, the IM 151/CPU goes into STOP.

Points to Note About OB 122

Note

Note the following about OB 122:

The CPU component enters the value ”0” in the following temporary variables of the variable declaration table in the local data of the OB:

- **Byte no. 3:** OB122_BLK_TYPE
  (type of the block in which the error occurred)
- **Byte no. 8 and 9:** OB122_BLK_NUM
  (number of the block in which the error occurred)
- **Byte no. 10 and 11:** OB122_PRG_ADDR
  (address of the block in which the error occurred)

5.7 Parameters

Parameterizable Properties of the CPU

The properties and responses of the CPU component of the IM 151/CPU can be parameterized. You carry out this parameterization on different tabs in *STEP 7*.

Which Parameters Can Be Set for the IM 151/CPU?

The following table contains all the parameter blocks for the IM 151/CPU. The parameters are explained in the *STEP 7* online help system.

Table 5-12 Parameter Blocks, Settable Parameters and Their Ranges for the IM 151/CPU

Parameter Blocks	Settable Parameters	Range
Clock memory	Clock memory	Yes/no
	Memory byte	0 to 255
Startup characteristics	Automatic/manual startup after power on	Warm restart
	Monitoring time for: • ”Finished” message by means of modules (100 ms) • Transfer of the parameters to modules (100 ms)	 • 1 to 65000 • 1 to 65000
	Startup at preset configuration not equal to actual configuration	Yes/no
System diagnostics	Report cause of STOP	Yes/no
Retentivity	Number of memory bytes starting with MB 0	0 to 256
	Number of S7 timers starting with T 0	0 to 128
	Number of S7 timers starting with C 0	0 to 64
	Retentive areas • Data block number • Data byte address • Number of data bytes	8 • 1 to 127 • 0 to 8191 • 0 to 4096
Software clock	Correction factor	–10000 to +10000
Time-of-day interrupts	OB 10 activation	Yes/no
	OB 10 execution	• None • Once • Every minute • Hourly • Daily • Weekly • Monthly • Last day of the month • Annually

Table 5-12 Parameter Blocks, Settable Parameters and Their Ranges for the IM 151/CPU

Parameter Blocks	Settable Parameters	Range
	Start date for the OB 10	Year-month-day
	Start time for the OB 10	Hours:minutes
Cyclic interrupts	Periodicity of the OB 35 (ms)	1 to 60000
Cycle behavior	Cycle load from communication (%)	10 to 50
	Scan cycle monitoring time (ms)	1 to 6000
	OB85 call at I/O access error	• For each access • For incoming and outgoing errors • No call
Protection	Level of protection	• 1: Key switch • 2: Write protection • 3: Write/read protection
	Mode	• Process mode: permissible cycle increase (ms) from 3 to 65535 • Test mode
Module parameters	Number of reference junctions • Activated • Module number • Channel number	8 • Yes/no • From 5 to 66 • 0/1
	Interference frequency suppression	50 Hz / 60 Hz
DP station address	CPU DP address	1 to 125

When Does the CPU ”Accept” the Parameters?

The CPU accepts the parameters (configuration data) you have set:

- After power on or a memory reset of the inserted memory module
- After the configuration data has been transferred without errors to the CPU online in STOP mode

5.8 Parameterization of the Reference Junction for the Connection of Thermocouples

If you want to use the IM 515/CPU in an ET 200S system with thermocouples and reference junctions, set the following parameters in the hardware configuration:

Table 5-13 Parameterization of the Reference Junction

CPU Module Parameter	Range	Explanation
Activation of reference junctions 1 to 8	Activated/not activated Example, see Figure 5-4	You can enable the reference junction with this parameter. Only then can you continue to parameterize the reference junction.
Module number of reference junction 1 to module number of reference junction 8 The module number corresponds to the slot.	None/5 to 66 Example, see Figure 5-4	You can use this parameter to assign the RTD module slot to the corresponding reference junction.
Channel number of reference junction 1 to channel number of reference junction 8	RTD on channel 0 RTD on channel 1 Example, see Figure 5-4	You can use this parameter to define the channel (0/1) for reference temperature measurement (determining the compensation value) for the assigned RTD module slot.
RTD Module Parameter	**Range**	**Explanation**
Measurement type/measurement range	Resistance/temperature measurement, e.g. • RTD-4L Pt 100 standard range	If you use a channel of the RTD module for reference junction parameterization, you must parameterize the measurement type/measurement range for this channel as *RTD-4L Pt 100 climatic range*.
TC Module Parameter	**Range**	**Explanation**
Reference junction number	1 to 8	You can use this parameter to indicate the reference junction (1 to 8) that contains the reference temperature (compensation value).
Reference junction channel 0 and reference junction channel 1	None, RTD	You can enable the use of the reference junction with this parameter.

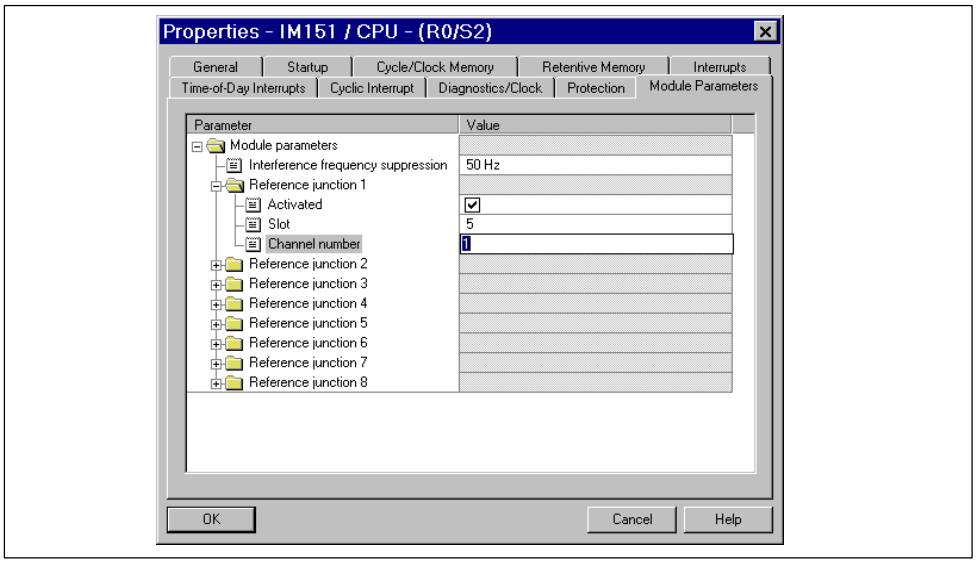


Figure 5-4 Example of a Parameterization Dialog Box for the CPU Module Data in STEP 7 V5.1

Reference

You can find detailed information on the procedure, the connection system and an example of parameterization in the chapter entitled *Analog Electronic Modules* in the *ET 200S Distributed Device* manual.

5.9 Removal and Insertion of Modules During Operation

Removing and inserting one module at a time in the case of the IM 151/CPU with local ET 200S I/Os is possible during operation and in an energized state.

Exceptions

The CPU itself must not be removed during operation and in an energized state.

What Happens when Modules are Removed During Operation

When you remove a module from the ET 200S I/O system during operation, OB 83 is called and a corresponding diagnostic buffer entry is created (event ID 3861_H), irrespective of whether the power module is switched on or off.

If OB 83 is in the memory of the CPU, the IM 151/CPU remains in RUN.

The absence of the module is noted in the system status list.

If the module that has been removed is accessed from the user program, an I/O access error occurs with a corresponding entry in the diagnostic buffer. In addition, OB 122 is called.

If OB 122 is in the memory of the CPU, the IM 151/CPU remains in RUN.

What Happens when Modules are Inserted During Operation

If you reinsert a module that has been removed in the ET 200S I/O system during operation, the CPU carries out a preset/actual comparison for the inserted module. This compares the configured module with the one that is actually inserted. The following activities occur depending on the result of the preset/actual comparison:

Modules that Cannot be Parameterized

The following activities occur irrespective of whether the power module of the inserted module is switched on or off.

Table 5-14 Result of the Preset/Actual Comparison in Modules that **Cannot** be Parameterized

Inserted Module = Configured Module	Inserted Module ≠ Configured Module
OB 83 is called with the corresponding diagnostic buffer entry (event ID 3861_H).	OB 83 is called with the corresponding diagnostic buffer entry (event ID 3863_H).
The module is entered in the system status list as available.	The module remains entered in the system status list as unavailable.
Direct access is again possible.	Direct access is not possible.

Modules that Can be Parameterized

The following activities only occur when the power module of the inserted module is switched **on**.

Table 5-15 Result of the Preset/Actual Comparison in the Case of Parameterizable Modules with the Power Module Switched on

Inserted Module = Configured Module	Inserted Module ≠ Configured Module
OB 83 is called with the corresponding diagnostic buffer entry (event ID 3861_H).	OB 83 is called with the corresponding diagnostic buffer entry (event ID 3863_H).
The CPU reparameterizes the module.	The CPU does not reparameterize the module.
If parameter assignment is successful, the module is entered in the system status list as available.	The module remains entered in the system status list as unavailable. The SF LED on the module remains on.
Direct access is again possible.	Direct access is not possible.

The following activities only occur if the power module of the inserted module is switched **off**.

Table 5-16 Result of the Preset/Actual Comparison in the Case of Parameterizable Modules with the Power Module Switched off

Inserted Module = Configured Module	Inserted Module ≠ Configured Module
OB 83 is called with the corresponding diagnostic buffer entry (event ID 3861_H).	
When the power module is switched on, the CPU reparameterizes the module.	When the power module is switched on, the CPU does not parameterize the module.
If parameter assignment is successful, the module is entered in the system status list as available.	The module remains entered in the system status list as unavailable. The SF LED on the module remains on.
Direct access is again possible.	Direct access is not possible.

5.10 Switching Power Modules Off and On During Operation

What Happens When Power Modules Are Switched Off During Operation

If the load power voltage to a power module is switched off during operation, the following activities take place:

- If you enable diagnostics during parameter assignment for the power module, diagnostic interrupt OB 82 (diagnostic address of the power module) is called with the corresponding diagnostic buffer entry (event 3942_H).
- The power module is entered as present but faulty in the system status list.

Switching off the load power supply has the following effects on the modules supplied by the power module:

- The modules can continue to be accessed without an I/O access error occurring.
- The outputs of the modules are deenergized and inactive for the process.
- The inputs of digital modules and FM modules return 0; the inputs of analog modules return $7FFF_{H.}$

What Happens When Power Modules Are Switched On During Operation

If the load power supply to a power module is switched on during operation, the following activities take place:

- If you enable diagnostics when assigning parameters for the power module, the diagnostic interrupt OB 82 (diagnostic address of the power module) is called with the corresponding diagnostic buffer entry (event 3842_H).
- The power module is entered as present and o.k. in the system status list.

Switching on the load power supply has the following effects on modules supplied by the power module:

- The modules regain their full functionality.

Removal and Insertion of Power Modules During Operation

You can find a list of activities that occur if you remove or insert power modules during operation in Section 5.9.

Removal and insertion has the same effects as switching the load power supply off and on for the modules that are supplied by the power module.

Technical Specifications

6

In This Chapter

In this chapter you will find:

- The technical specifications of the IM 151/CPU interface module

Chapter Overview

In Section	Contents	Page
6.1	Technical Specifications of the IM 151/CPU	6-2

6.1 Technical Specifications of the IM 151/CPU

Order Numbers

IM 151/CPU interface module:	6ES7 151-7AA00-0AB0
IM 151/CPU FO interface module:	6ES7 151-7AB00-0AB0
MMC (Micro Memory Card): (see Section 5.4, Memory Modules)	6ES7 953-8Lx00-0AA0

Features

The IM 151/CPU interface module has the following features:

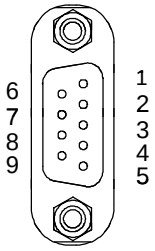
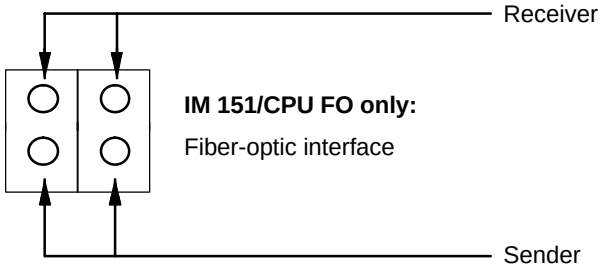
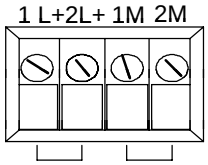
- Intelligent slave with RS485 interface or fiber-optic cable to the PROFIBUS-DP
- Stand-alone operation possible
- 24 KB working memory, not buffered
- 40 KB integrated load memory RAM, not buffered
- Powerfail-proof storage of the user program and configuration via the MMC memory module
- Programmable with *STEP 7*, as of V5.1
- Speed: approx. 0.5 ms per 1000 binary instructions
- Maximum configuration of the local I/Os: 63 ET 200S modules

General Technical Specifications

The IM 151/CPU meets the general technical specifications of the ET 200S distributed I/O system. You will find these standards and test specifications in the chapter entitled ”General Technical Specifications” in the *ET 200S Distributed I/O System* manual.

Terminal Assignment for the IM 151/CPU

Table 6-1 Terminal Assignment of the IM 151/CPU Interface Module

View		Signal Name	Description
IM 151/CPU  RS 485 interface	1	–	–
	2	–	–
	3	RxD/TxD-P	Data line B
	4	RTS	Request To Send
	5	M5V2	Data reference potential (from the station)
	6	P5V2	Supply Plus (from the station)
	7	–	–
	8	RxD/TxD-N	Data line A
	9	–	–
 **IM 151/CPU FO only:** Fiber-optic interface			
 	1 L+		24V DC
	2L+		24 V DC (to loop through)
	1M		Chassis ground
	2M		Chassis ground (to loop through)

Basic Circuit Diagram for the IM 151/CPU

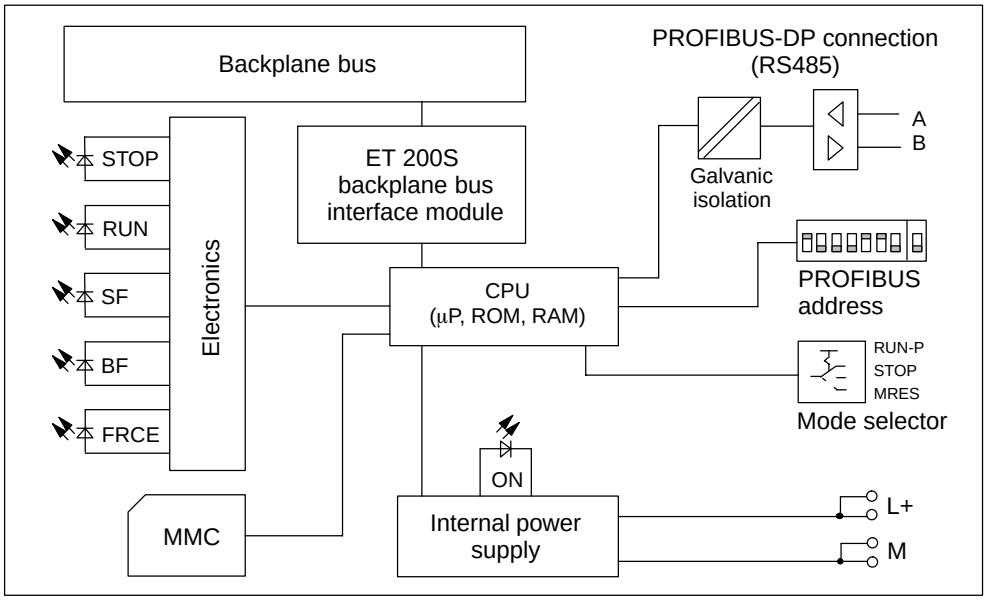


Figure 6-1 Basic Circuit Diagram for the IM 151/CPU

Basic Circuit Diagram for the IM 151/CPU FO

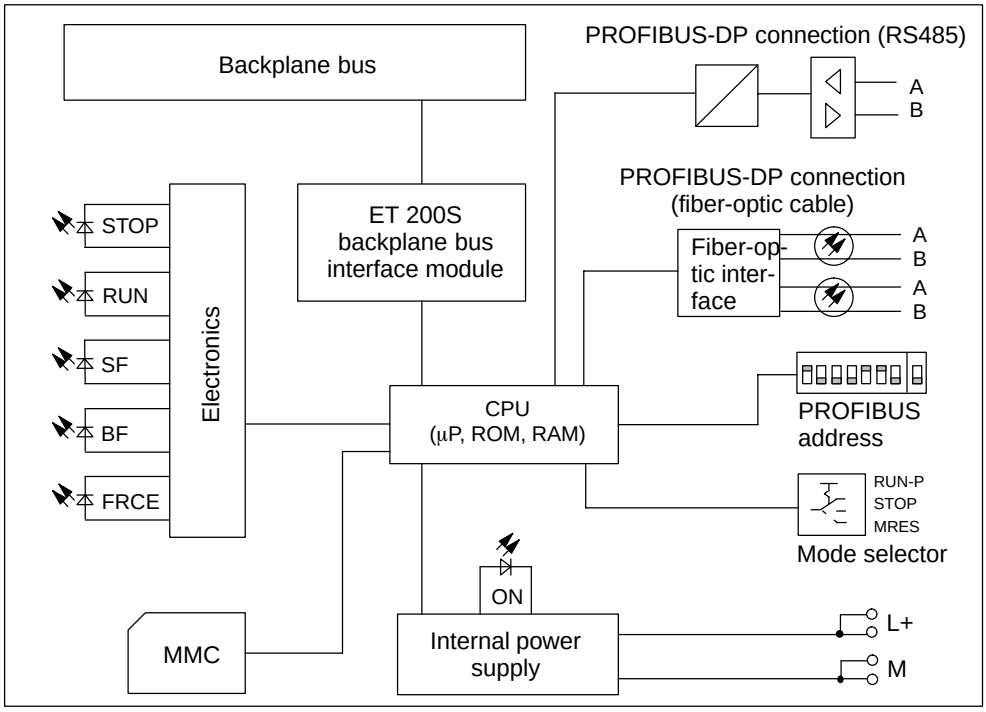


Figure 6-2 Basic Circuit Diagram for the IM 151/CPU FO

Technical Specifications

CPU and Product Version	
MLFB	6ES7 151-7AA00-0AB0 FO: 6ES7 151-7AB00-0AB0
• Hardware version	1
• Firmware version	V1.0.0
• Matching programming package	STEP 7 V5.1 or higher
Memory	
Working memory	
• integral	24 KB
• Expandable	No
Load memory:	
• integral	40 KB RAM
• Expandable	64 KB MMC 2 MB MMC
Backup	No
Processing times	
Processing times for	
• Bit instructions	0.3 µs minimum
• Word instructions	1 µs minimum
• Fixed-point math instructions	2 µs minimum
• Floating-point math instructions	50 µs minimum
Timers, Counters and their Retentive Features	
S7 counters	64
• Adjustable retentivity	from C 0 to C 63
• Preset	from C 0 to C 7
• Counting range	0 to 999
IEC Counters	Yes
• Type	SFB
S7 timers	128
• Adjustable retentivity	from T 0 to T 127
• Preset	No retentive times
• Timing range	10 ms to 9990 s
IEC Timers	Yes
• Type	SFB

Data areas and their retentive characteristics	
Total retentive data area (incl. memory markers, timers, counters)	Max. 4736 bytes
Bit memories	256 bytes
• Adjustable retentivity	MB 0 to MB 255
• Preset retentivity	MB 0 to MB 15
Clock memory	8 (1 memory byte)
Data blocks	max. 127 (DB 0 reserved)
• Size	max. 8 KB
• Adjustable retentivity	max. 8 DB, 4096 data bytes in all
• Preset retentivity	No retentivity
Local data	max. 1536 bytes
• Per priority class	Max. 256 bytes
Blocks	
OBs	See Appendix B
• Size	max. 8 KB
Nesting depth:	
• Per priority class	8
• additional levels within an error OB	4
FBs	max. 128
• Size	max. 8 KB
FCs	max. 128
• Size	max. 8 KB
Address areas (inputs/outputs)	
Total I/O address area	Max. 1536 bytes/1536 bytes
• Distributed	64 bytes/64 bytes
Process image	128 bytes/128 bytes (not adjustable)
Digital channels	Max. 248/248
Analog channels	Max. 124/124
Time	
Clock	Software clock
Operating hours counter	No

S7 message functions	
Process diagnostic messages	ALARM_S, ALARM_SQ
Testing and commissioning functions	
Status/Modify Variables	Yes
• Variable	Inputs, outputs, flags, data, timers, counters
• Number of variables	
– Of which status variables	max. 30
– Of which modify variables	max. 14
Force	Yes
• Variable	Inputs, outputs
• Number	max. 10
Monitor block	Yes
Single sequence	Yes
Breakpoint	2
Diagnostic buffer	Yes
• Number of inputs	Max. 100 (not adjustable)
Communication functions	
PD/OP communication	Yes
Global data communication	No
S7 basic communication	Yes (server)
• User data per job	max. 76 bytes
– Number of which consistent	32 bytes at I_PUT / I_GET
S7 communication	Yes (server)
• User data per job	max. 160 bytes
– Number of which consistent	32 bytes
S7-compatible communication	No
Standard communication	No

Interface		FO
Type of interface	Integrated RS 485 interface	Fiber-optic interface and Integrated RS 485 interface
Physical system	RS 485	FO or RS 485
Galvanically isolated	Yes	No
Power supply to the interface (15 to 30 V DC)	Max. 80 mA	
Number of connection resources	5 for PG/OP/S7 basic/S7 communication One each reserved for the programming device and operator panel	
Functionality		
• MPI	No	
• PROFIBUS DP	DP Slave	
• Point-to-point connection	No	
DP Slave		
• Utilities:		
– PD/OP communication	Yes	
– Routing	No	
– Direct Communication	Yes	
• DDB (Device Database) File	siem806C.gsg FO: siem806D.gsg	
• Transmission rates	Up to 12 Mbps	
• Intermediate memory	64 I bytes/64 O bytes	
– Address areas	32 with a maximum of 32 bytes each *	

* Up to the maximum size of the intermediate memory

Programming	
Programming language	STEP 7
Stored instructions	See Appendix B
Nesting levels	8
System functions (SFCs)	See Appendix C
System function blocks (SFBs)	See Appendix C
User program security	Password protected
Dimensions and Weight	
Installation dimensions W × H × D (mm)	60 x 119.5 x 75
Weight	Approx. 200 g
Voltages, Currents	

Power supply	Rated value 24 V DC
• Permissible range	20,4 to 28,8 V
• Polarity reversal protection	Yes
• Voltage failure buffering	20 ms
Insulation tested at	500 V DC
Current consumption from supply voltage (1L+)	Approx. 250 mA
• Power supply for the ET 200S backplane bus	Max. 700 mA
Power losses	Typically 3.3 W

Cycle and Response Times

7

Introduction

In this chapter you will learn what make up the cycle and response times of the ET 200S with the IM 151/CPU.

You can use the programming device to read out the cycle time of your user program (see the *STEP 7 User Manual*).

The response time is more important for the process. In this chapter we will show you in detail how to calculate the response time.

Chapter Overview

In Section	Contents	Page
7.1	Cycle Time	7-2
7.2	Response Time	7-6
7.3	Interrupt Response Time	7-9

Execution Times

- For the *STEP 7* instructions that can be processed by the CPUs can be found in Appendix B.
- For the SFCs/SFBs integrated in the CPUs can be found in Appendix C.

7.1 Cycle time

Cycle Time – A Definition

The cycle time is the time that the operating system requires to process a program cycle – i.e. an OB 1 cycle – as well as all the program sections and system activities interrupting this cycle.

This time is monitored.

Components of the Cycle Time

Factors	Remarks
Operating system execution time	See Table 7-1
Process image transfer time (PII and PIQ)	See Table 7-2
User program execution time	Is calculated from the execution times of the individual instructions and a CPU-specific factor (see Table 7-3).
S7 timers	See Table 7-4
Loading through interrupts	See Table 7-5

The following figure shows the components that make up the cycle time:

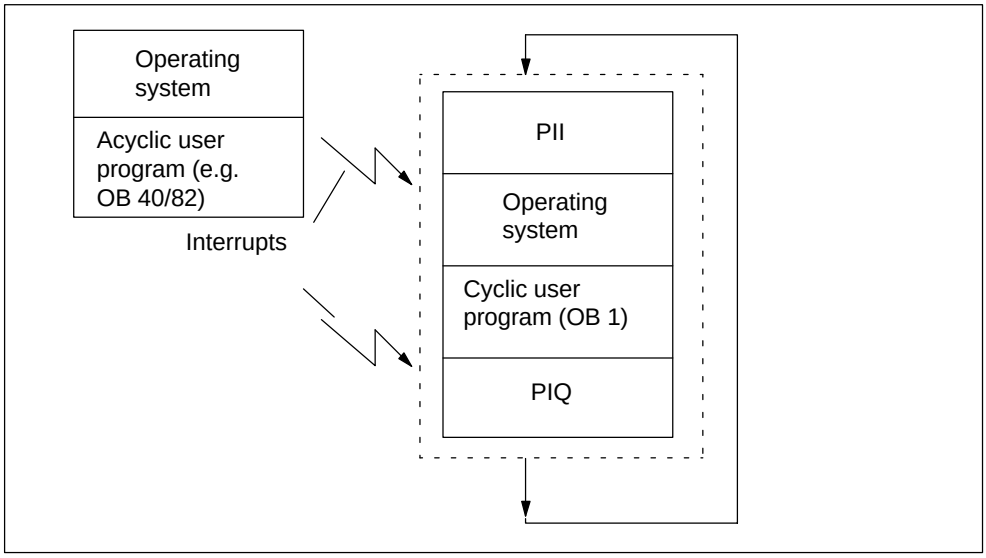


Figure 7-1 Component Parts of the Cycle Time

Extending the Cycle Time

Note that the cycle time of a user program is extended by the following:

- Time-controlled interrupt handling
- Process interrupt handling (see also Section 7.3)
- Diagnostics and error handling (see also Section 7.3)

Operating System Processing Time

The operating system processing time for the IM 151/CPU takes 540 µs to 1040 µs (see Table 7-1).

The specified time does not include the execution of:

- Test functions, e.g. monitor, modify
- Functions: Load block, delete block, compress block
- Communication

Table 7-1 Extending the Cycle by Nesting Interrupts

Sequence	IM 151/CPU
Operating system processing time	540 µs to 1040 µs

Process Image Transfer Time

The table below contains the CPU times for process image updating (process image transfer time). The specified times are ”ideal” values that get increased by interrupts that occur or by communication involving the CPU component of the IM 151/CPU.

(Process image = PI)

The CPU time for process image updating is calculated as follows:

K + A = process image transfer time, in which K and A equal the following:

Table 7-2 Process Image Updating

	Designation	Times in the IM 151/CPU
K	Base load	162 µs
Q	Bytes in the PI for the ET 200S I/O	80 µs per byte

User Program Scanning Time:

The user program scanning time is made up of the sum of the execution times for the instructions and SFCs called. These execution times can be found in the Instruction List. You also have to multiply the user program scanning time by a basic module-specific factor.

In the IM 151/CPU, this factor depends on the following:

Table 7-3 Dependency of the User Program Scanning Time

Dependency	Range
The number of modules inserted	0 to 63
The volume of data transferred via the PROFIBUS-DP between the DP master and DP slave (input data + output data)	0 to 128 bytes
The volume of data obtained through direct communication	0 to 256 bytes

You can calculate the factor for your application approximately using the following rule of thumb for the IM 151/CPU

	1,1
+	0.0055 x number of modules
+	0.0022 x number of bytes transfered (PROFIBUS-DP)
+	0.0011 x number of bytes obtained (direct communication)
=	**Multiplier for your user program**

S7 Timers

The S7 timers are updated every 10 ms.

To include the S7 timers in the calculation of the cycle and response times of the IM 151/CPU, multiply the number of active S7 timers by 8 µs.

Table 7-4 Updating the S7 Timers

Sequence	IM 151/CPU
Updating the S7 timers (every 10 ms)	Number of simultaneously active S7 timers × 8 µs

Delay of the Inputs and Outputs

You have to take into account the following delay times, depending on the expansion module:

- For digital inputs: The input delay time
- For digital outputs: Negligible delay times
- For analog inputs: Cycle time of the analog input
- For analog outputs: Response time of the analog output

Extending the Cycle by Nesting Interrupts

Table 7-5 shows typical extensions of the cycle time through nesting of an interrupt. The program runtime at the interrupt level must be added to these. If several interrupts are nested, the corresponding times need to be added.

Table 7-5 Extending the Cycle by Nesting Interrupts

Interrupts	IM 151/CPU
Process interrupt	Approx. 480 µs
Diagnostic interrupt	Approx. 700 µs
Time-of-day interrupt	Approx. 460 µs
Delay Interrupt	Approx. 370 µs
Watchdog Interrupt	Approx. 280 µs
Programming/access error/runtime system error	Approx. 560 µs

7.2 Response Time

Response Time for the ET 200S with the IM 151/CPU

The response time is the time from the detection of an input signal at the ET 200S with the IM 151/CPU to the modification of an associated output signal via the inputs and outputs of the expansion modules.

Factors

The response time depends on the cycle time and the following factors:

Factors	Remarks
Delay of the inputs and outputs	You can find the delay times in the technical specifications of the signal modules in the *ET 200S Distributed I/O Device* manual.

Range of Fluctuation

The actual response time lies between a shortest and a longest response time. You must always reckon on the longest response time when configuring your system.

The shortest and longest response times are considered below to let you get an idea of the width of fluctuation of the response time.

Shortest Response Time

The following figure shows you the conditions under which the shortest response time is obtained.

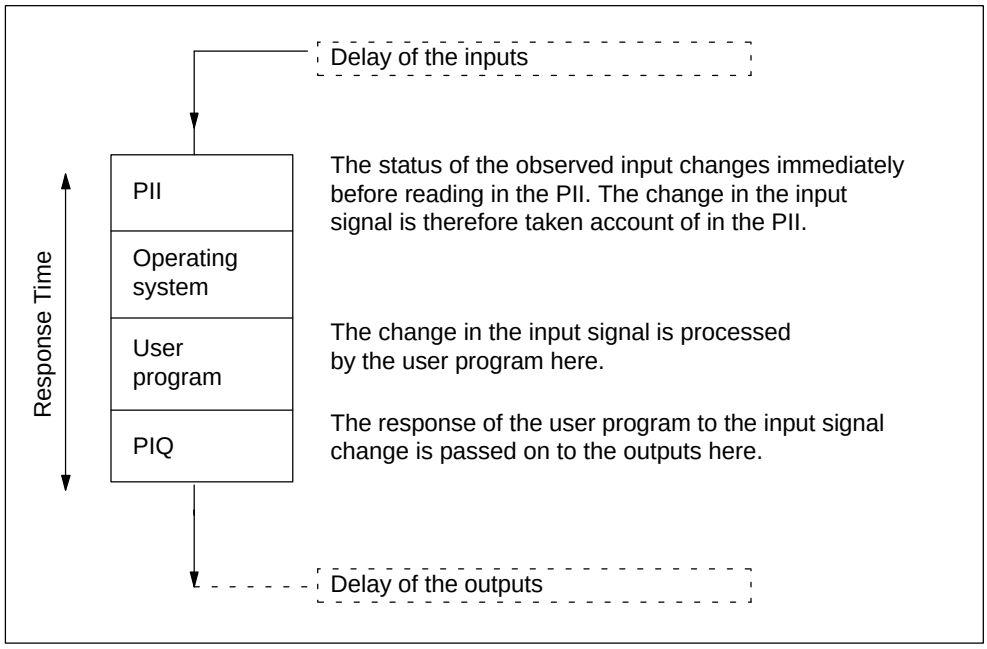


Figure 7-2 Shortest Response Time

Calculation

The (shortest) response time consists of the following:

- 1 × process image transfer time for the inputs +
- 1 × operating system processing time +
- 1 × program scanning time +
- 1 × process image transfer time for outputs +
- Execution time of S7 timer
- Delay of the inputs and outputs

This corresponds to the sum of the cycle time and the delay of the inputs and outputs.

Longest Response Time

The following figure shows what the longest response time consists of.

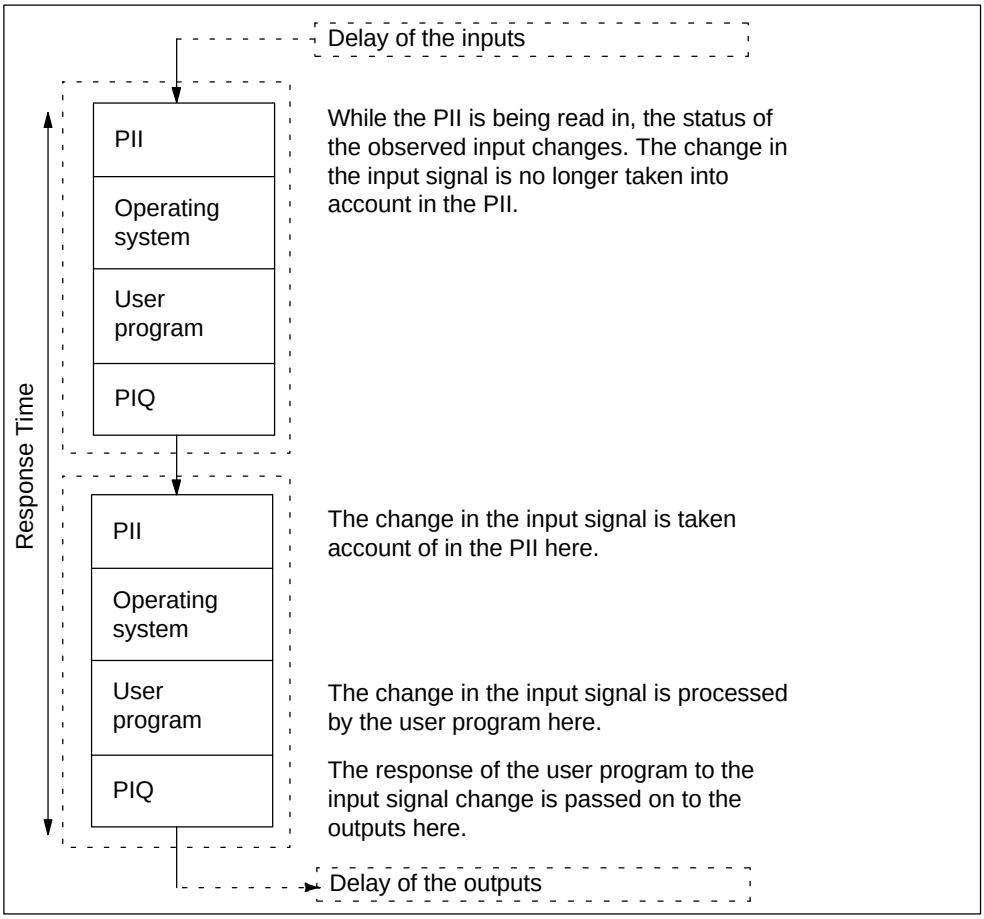


Figure 7-3 Longest Response Time

Calculation

The (longest) response time consists of the following:

- 2 × process image transfer time for the inputs +
- 2 × process image transfer time for the outputs +
- 2 × operating system processing time +
- 2 × program scanning time +
- Processing time of the S7 timers +
- Delay of the inputs and outputs

This corresponds to the sum of the doubled cycle time and the delay of the inputs and outputs.

7.3 Interrupt Response Time

Interrupt Response Time – A Definition

The interrupt response time is the time from the first occurrence of an interrupt signal to the call of the first instruction in the interrupt OB of the IM 151/CPU.

It generally applies that the interrupt response time is increased by the program scanning time of the interrupt OBs that have not yet been processed.

Interrupt Response Times

Table 7-6 Interrupt Response Times of the IM 151/CPU (Without Communication)

Interrupt Response Times (Without Communication) for...	Duration
Process interrupt, diagnostic interrupt	Less than 20 ms

Process Interrupt Handling

Process interrupt handling begins when the process interrupt OB 40 is called. Higher-priority interrupts cause the process interrupt handling routine to be interrupted. Direct accesses to the I/Os are made at the execution time of the instruction. After process interrupt handling has been completed, either cyclic program scanning is continued or additional interrupt OBs of the same priority or a lower priority are called and processed.

Getting Started

8

Introduction

This guide takes you through the 10 commissioning steps required to set up a functioning application by running through a concrete example. In this way, you will get to know the basic functions of your IM 151/CPU for the following:

- Hardware and software
- Stand-alone operation and as an intelligent DP slave

Prerequisites

You must be familiar with the fundamentals of electronic/electrical engineering and have experience of working with computers and Microsoft® Windows™ 95/98/NT.

Danger

The IM 151/CPU-ET 200S and the S7-300 are used in installations and systems that require you to comply with specific rules and regulations that vary depending on the application. Please note the relevant safety and accident-prevention regulations, such as IEC 204 (emergency stop systems).
Non-compliance with these regulations can result in serious injury and damage to both machinery and equipment.

Chapter Overview

In Section	Contents	Page
8.1	Step 1: Installing the ET 200S/IM 151/CPU and S7-300	8-3
8.2	Step 2: Wiring the ET 200S/IM 151/CPU and S7-300	8-4
8.3	Step 3: Commissioning of the ET 200S/IM 151/CPU	8-5
8.4	Step 4: Configuring the IM 151/CPU for Stand-Alone Operation	8-6
8.5	Step 5: Programming the IM 151/CPU	8-8
8.6	Step 6: Test Run	8-9
8.7	Step 7: Upgrading the IM 151/CPU and Commissioning the S7-300	8-10
8.8	Step 8: Configuring the IM 151/CPU as a DP Slave and the S7-300 as a DP Master	8-11
8.9	Step 9: Programming the IM 151/CPU and the S7-300	8-14
8.10	Step 10: Commissioning and Test Run	8-16

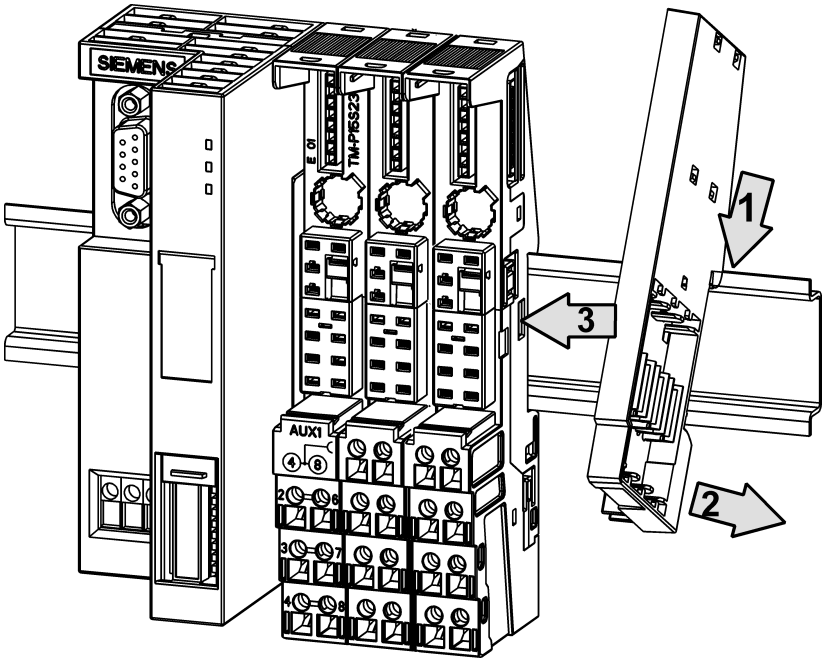
Materials and Tools Required

Quantity	Item	Order Number (SIEMENS)
1	S7-300 system, consisting of power supply (PS), CPU with DP interface(CPU 315 2-DP in this example), digital input module (DI) on slot 4 and digital output module (DO) on slot 5, incl. bus connector and cabling	Various
1	Power supply (PS), e.g. PS 307 with network connecting cable (optional)	e.g. 6ES7 307-1EA00-0AA0
1	IM 151/CPU with terminating module	e.g. 6ES7 151-7AA00-0AB0
1	Power module (PM)	e.g. 6ES7 138-4CA00-0AA0
1	Digital input module (DI)	e.g. 6ES7 131-4BD00-0AA0
1	Digital output module (DO)	e.g. 6ES7 132-4BD00-0AA0
1	Terminal module (TM) for the PM	e.g. 6ES7 193-4CC30-0AA0
2	Terminal modules (TM) for DI and DO	e.g. 6ES7 193-4CB30-0AA0
1	Rail for ET 200S	Various
1	Programming device (PG) with PROFIBUS-DP interface, installed STEP 7 software, version ≥ 5.1 and PG cable (up to 1.5 Mbps)	Various
1	PROFIBUS-DP cable	Various
1	Screwdriver with 3 mm blade	Standard
1	Screwdriver with 4.5 mm blade	Standard
1	Diagonal cutter and tool for insulation stripping	Standard
1	Tool for pressing on wire-end ferrules	Standard
Approx. 2m	Stranded wire with 1 mm^2 cross-section with correct wire end ferrules (type A, length 6 mm and 12 mm)	Standard
4	1-pin on switch (24 V)	Standard

Step 1: Installing the ET 200S/IM 151/CPU and S7-300

Stage	Description
1	Install the S7-300 as described in the *S7-300 Programmable Controller, Hardware and Installation* manual or in *STEP 7 V5.0 Getting Started*.
2	If you want to operate the IM 151/CPU with its own power supply, attach it to the S7-300 rail and tip it down until it snaps into position.
3	Attach the IM 151/CPU to the rail, and tip it down until it snaps into position.
4	Attach the TM for the PM to the rail on the right of the IM 151/CPU, and tip it down until it snaps into position.
5	Move the TM to the left until you can hear it engage with the IM 151/CPU.
6	Repeat steps 3 and 4 with the two TMs for the electronic modules and then with the terminating module (which doesn't snap onto the rail).
7	Push the PM into the corresponding TM until it snaps into position.
8	Push the DI into the left (still unoccupied) TM until it snaps into position.
9	Push the DO into the last (still unoccupied) TM until it snaps into position.

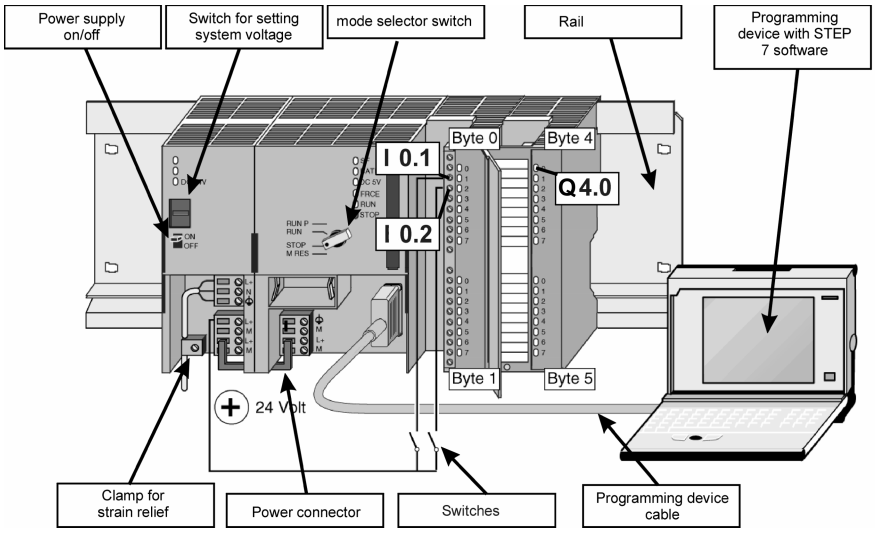
Installing the ET 200S



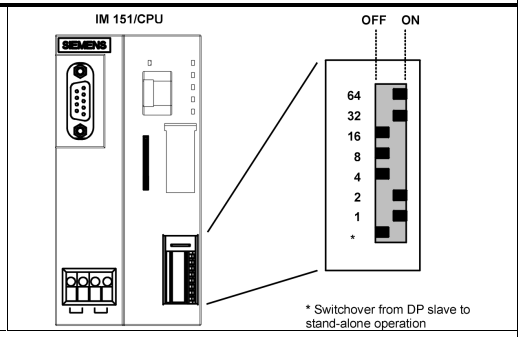
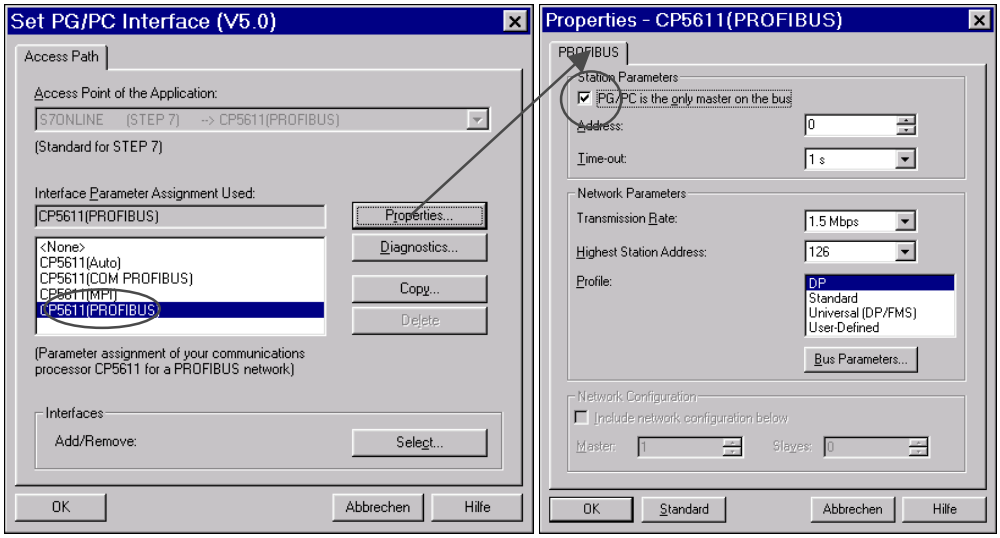
Step 2: Wiring the ET 200S/IM 151/CPU and S7-300

Stage	Description
1	Wire the S7-300 as described in the *S7-300 Programmable Controller, Hardware and Installation* manual or in *STEP 7 Getting Started, V5.0*.
2	Extend the connections of the 4 switches using a cable. Strip 6 mm off the cable ends and attach wire end ferrules.
3	Connect the inputs 1.1 (terminal 13) and 1.2 (terminal 14) on the DI of S7-300 by means of a switch to L+ on the PS of the S7-300.
4	Connect the two remaining 1-pin switches to the DI of the ET 200S as follows: • One switch to terminals 1 and 3 • The other switch to terminals 5 and 7 **A note on spring terminals** To release the spring in a connection, insert a screwdriver with a 3 mm blade into the upper round hole in the terminal until it reaches the stop, moving the screwdriver upwards slightly, if necessary. You can then insert a free cable end into the square hole below. Remove the screwdriver and check that the cable is in position securely.
5	Wire terminal 2 of the TM of the PM to L+ of the PS and terminal 3 of the TM of the PM to M of the PS. Strip 11 mm from the cable ends, and attach wire end ferrules.
6	Wire terminal 1L+ of the IM 151/CPU to L+ of the PS and terminal 1M of the IM 151/CPU to M of the PS. **Note** • Strip 11 mm from the cable ends that have to be connected, and attach wire end ferrules. • You can also use the power supply of the S7-300 for the IM 151/CPU and the PM.
7	Connect the programming device and the IM 151/CPU to the programming device cable, and secure all the connectors.
8	Connect the PS of the ET 200S, the PS of the S7-300 and the programming device to the network.

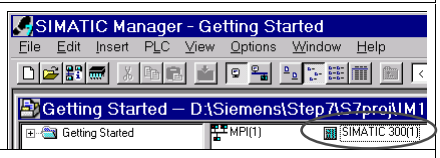
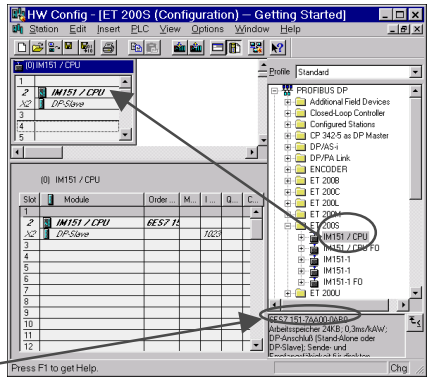
View of the S7-300 (the wiring of the power supply of the DI and the DO is not shown; the programming device is connected to the S7-300)

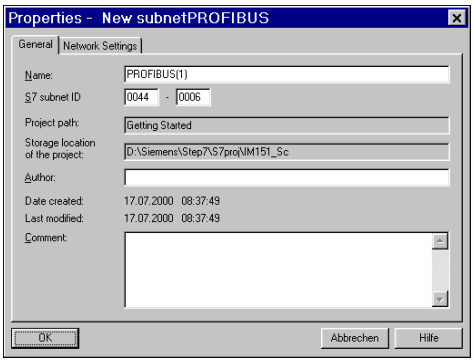
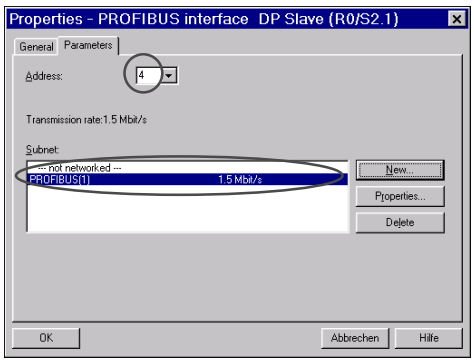
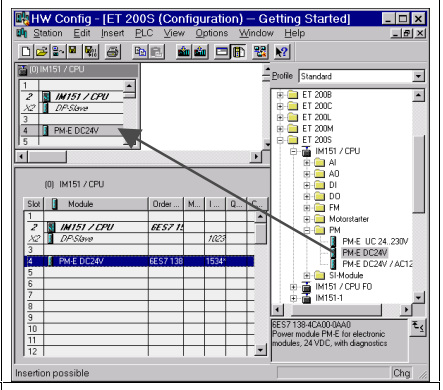
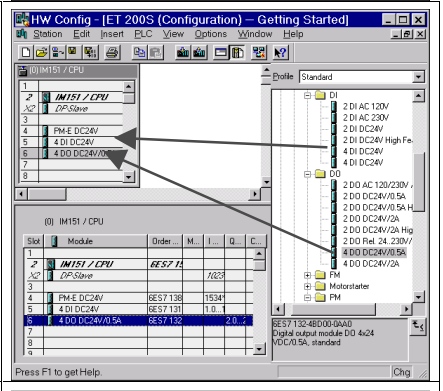
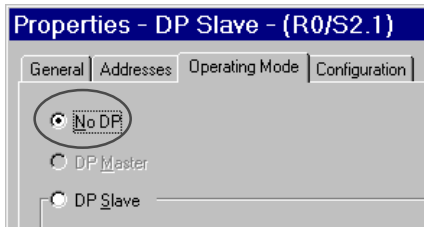


Step 3: Putting the ET 200S/IM 151/CPU into Operation

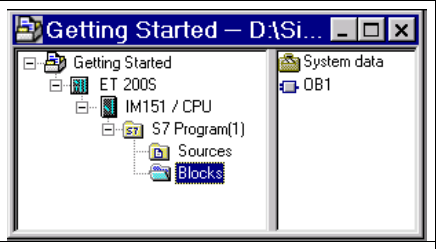
Stage	Description
1	Make sure that the mode selector is at *STOP*. Switch on the lowest DIP switch (for stand-alone operation) and the 4th DIP switch from the bottom (for PROFIBUS-DP address 4) on the IM 151/CPU. Make sure that the other DIP switches are off. 
2	Switch on the PS of the IM 151/CPU. **Result:** • The *DC24V* LED lights up on the PS. • The *PWR* and the *SF* LED light up on the PM. • All the LEDs light up on the IM 151/CPU, the *BF* and the *RUN* LEDs go out, and the *STOP* LED begins to flash slowly. The IM 151/CPU is thus requesting a memory reset.
3	Briefly press down the mode selector of the IM 151/CPU to MRES. The *STOP* LED begins to flash rapidly and then comes on continuously. This indicates that the memory reset has been completed. The *SF* LEDs on the PM and IM 151/CPU go out.
4	Operate the two switches connected to the DI. When you operate the switch on terminals 1 and 3, the *1* LED lights up. When you operate the switch on terminals 5 and 7, the *5* LED lights up.
5	Switch on your programming device, and start SIMATIC Manager on the Windows desktop.
6a	Click Options in SIMATIC Manager, and then choose the **Setting PG/PC Interface** menu command. Configure the programming device/PC interface as follows:
6b	 **Note:** The communication processor may have a different name in your programming device. Make sure that the PROFIBUS version is set.
7	Apply the settings with OK, and close the Setting the PG/PC Interface program.

Step 4: Configuring the IM 151/CPU for Stand-Alone Operation

Stage	Procedure	Result
1	Does the assistant for a new project appear in SIMATIC Manager?	**If so:** Close the assistant because it doesn't support the IM 151/CPU. **If not:** Continue to stage 2
2	In SIMATIC Manager, choose **New** from the **File** menu. Enter the project name "Getting Started", and click **OK**.	A new project is created and opened.
3	Navigate to **Insert** and choose the **Station** menu command. Click the **SIMATIC 300 station** in the list.	
4	Rename this station "ET 200S"	"SIMATIC 300(1)" is renamed "ET 200S".
5	Navigate in SIMATIC Manager to the ET 200S station. Double-click **Hardware** in the right-hand part of the window to open the hardware configuration editor.	
6	If a catalog is not displayed in the right-hand part of the window, activate the catalog by choosing the **Catalog** command from the **View** menu. Navigate to **ET 200S** in the catalog via **PROFIBUS-DP**. Insert the **IM 151/CPU** whose order number corresponds to the order number on your IM 151/CPU by dragging and dropping it in the window in the upper-left corner. A window appears immediately. This is described in the next stage. **Note:** You can find out the order number if you click an IM 151/CPU in the catalog. The order number of this IM 151/CPU then appears in the text box under the catalog.	

7	Click **New** in the dialog box that appears, and check that the settings in the dialog box are as shown here:   Close the dialog box with **OK**	
8	Navigate to the **PM** via the relevant **IM 151/CPU**. Insert the **PM** whose order number corresponds to the order number on your PM by dragging and dropping it onto slot 4.	
9	Do the same with the DI (on slot 5) and the DO (on slot 6)	
10	• Double-click **DP Slave** (line **X2**) in the lower-left part of the window in the hardware configuration program. • Choose the **Operating Mode** tab in the dialog box that appears. • Select the **No DP** check box on this tab. • Confirm with **OK**.	
11	Choose the **Save and Compile** command from the **Station** menu.	The hardware configuration is compiled and saved
12	Close the hardware editor	A icon for the IM 151/CPU appears in the right-hand part of the window in SIMATIC Manager.

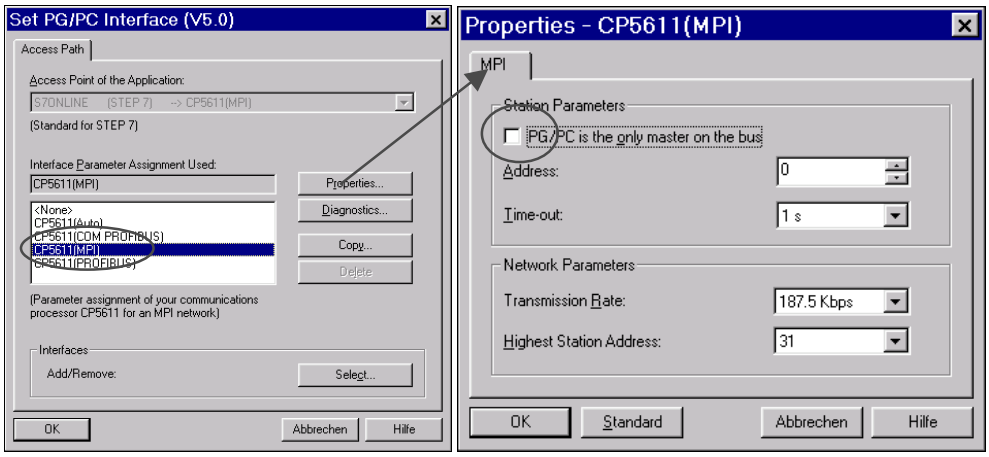
Step 5: Programming the IM 151/CPU

Step	Procedure	Result
1	Navigate in SIMATIC Manager via **IM151 /CPU** and **S7 Program** to the **Blocks** container.	
2	Double-click the **OB1** icon in the right-hand part of the window.	The LAD/FBD/STL editor for editing the OB1 block opens.
3	In the LAD/FBD/STL editor, choose the **LAD** command from the **View** menu to switch to the LAD programming language.	A rung current path is displayed in network 1.
4	Click exactly on the horizontal line of the rung current path.	The line is highlighted.
5	Double-click the -||- button (normally open contact) on the toolbar, and then click the –( ) button (coil).	The icons are inserted into the rung current path.
6	Click the red question mark of the normally open contact on the left in the rung current path.	The normally open contact is highlighted, and a text box with a cursor appears at the point of the question mark.
7	Enter *I1.0* and press *Return*.	The normally open contact on the left is assigned the designation *I1.0*.
8	Enter *I1.1* press *Return*. Enter *Q2.0* and press *Return*.	The normally open contact on the left is assigned the designation *I1.1*. The coil is assigned the designation *Q2.0*
9	Close the editor and confirm that you want to save with **Yes**.	The editor is closed, and the OB1 is saved.

Step 6: Test Run

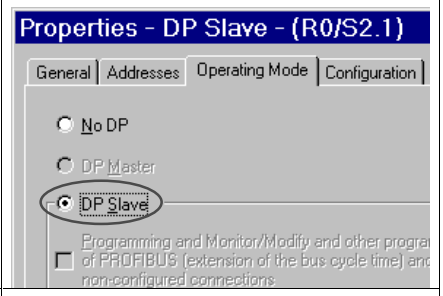
Stage	Procedure	Result
1	In SIMATIC Manager, click **Blocks** in the left-hand part of the dialog box.	**Blocks** is highlighted.
2	Right-click the right-hand part of the window, and insert an empty organization block with the name *OB82* into the block container. This block ensures that the S7-300 CPU will start even if the IM157/CPU is still indicating an I/O error. Generate OB86 in the same way. **Note** OB82 and OB86 are not important until DP slave operation.	The blocks OB82 and OB86 appear next to block OB1.
3	Select the block container in the left-hand part of the window again. From the **PLC** menu, choose the **Download** command to transfer the program and the hardware configuration to the CPU. Click **Yes** in all the dialog boxes that appear.	The program and configuration are downloaded from the programming device to the CPU.
4	Switch the CPU mode selector to *RUN-P*.	The *STOP* LED goes out. The *RUN* LED starts flashing and then comes on continuously.
5	Operate the two switches alternately.	The LEDs of the inputs 1.0 and 1.2 light up alternately. The LED of output 2.0 does not light up.
6	Operate the two switches simultaneously.	The LEDs of inputs 1.0 and 1.2 (*1* and *5* LEDs of the DI) light up together. Because the two switches are ANDed in the program (series connection) and assigned to the output 2.0, the LED of output 2.0 (*1* LED of the DO) lights up. This would switch on a connected actuator or indicator.
7	Switch the CPU mode selector to *STOP*, and switch off the PS of the IM 151/CPU.	All the LEDs go out.

Step 7: Converting the IM 151/CPU to a DP Slave and Putting the S7-300 into Operation

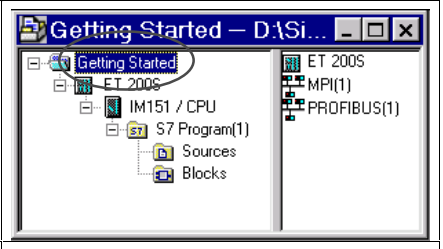
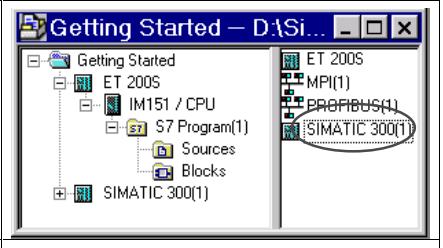
Stage	Description
1	Remove the connector of the programming device cable from the IM 151/CPU.
2a	Start the Setting the PG/PC Interface program as described in stage 6 of step 3. Change the configuration of the programming device/PC interface as follows:
2b	
3	Apply the settings with OK, and close the Setting the PG/PC Interface program.
4	Open the front panel of the S7-300 CPU. Connect the IM 151/CPU to the DP interface of the CPU of the S7-300 using the PROFIBUS-DP cable. **Note:** At least one of the bus connectors has a piggy-back connector. Make sure that this connector is with the S7-300 CPU.
5	Switch on the BUS terminating resistors on the connectors of the PROFIBUS-DP cable.
6	Connect the programming device to the MPI interface of the S7-300 CPU via the programming device cable.
7	Secure all the connectors, and close the front panel of the S7-300 CPU as far as possible.
8	Switch off the lowest DIP switch (for stand-alone operation) on the IM 151/CPU. Make sure that all the DIP switches except the 4th DIP switch from the bottom (for PROFIBUS-DP address 4) are switched off. **Note** Changes in the positions of the DIP switches are only recognized by the IM 151/CPU after the power has been switched off and on again.

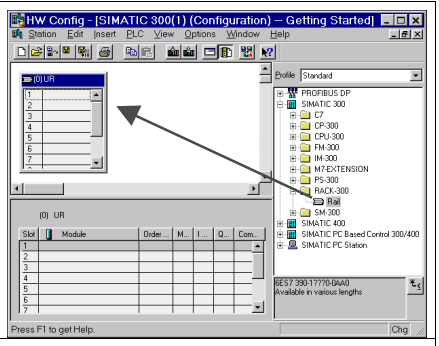
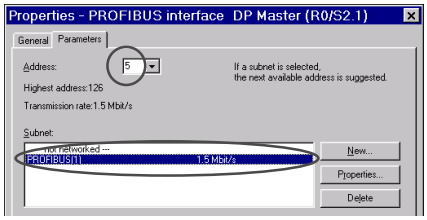
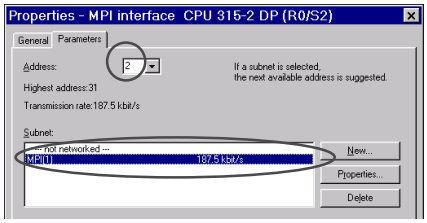
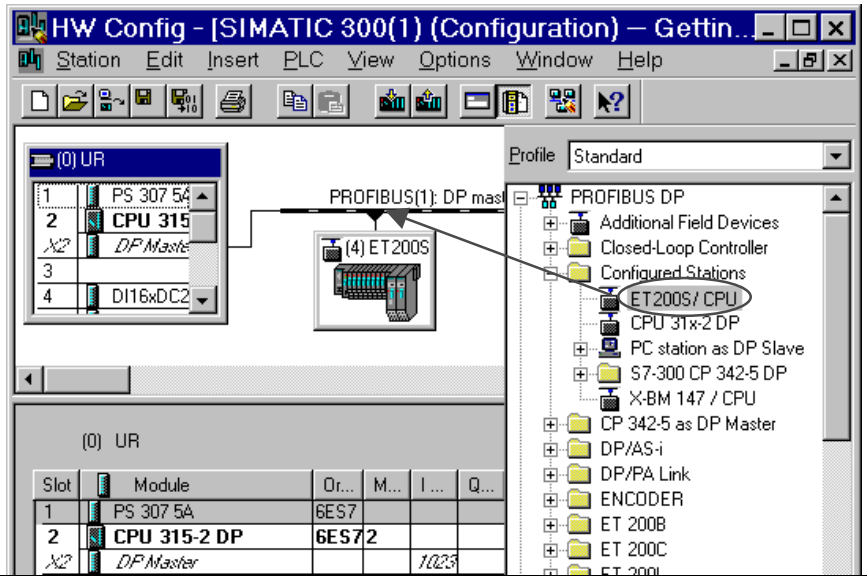
Step 8: Configuring the IM 151/CPU as a DP Slave and the S7-300 as a DP Master

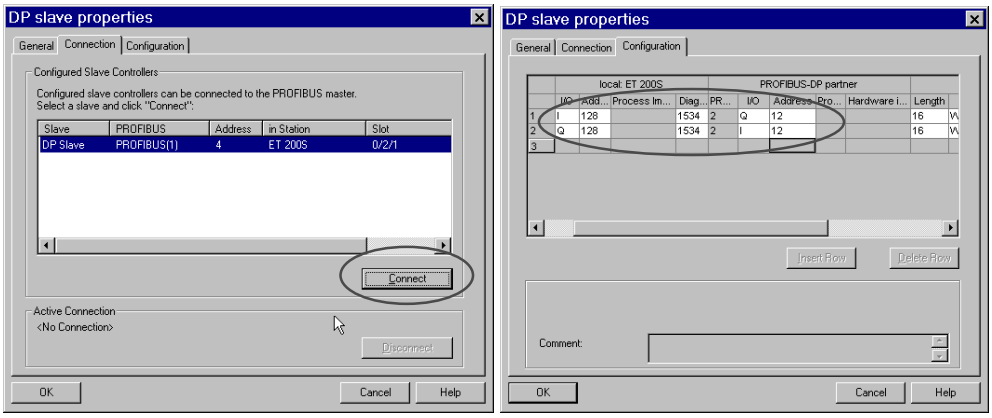
Change the configuration of the IM 151/CPU as shown below:

Stage	Procedure	Result
1	Start the hardware configuration program as described in step 4 for the IM 151/CPU.	The hardware configuration editor opens.
2	• Double-click **DP Slave** (line **X2**) in the lower-left part of the hardware configuration program dialog box. • In the dialog box that appears, select the **Operating Mode** tab. • Select the **DP Slave** option on this tab. • Confirm with **OK**.	
3	From the **Station** menu, choose the **Save and Compile** command. Close the hardware editor	The hardware configuration is compiled and saved, and the editor is closed.

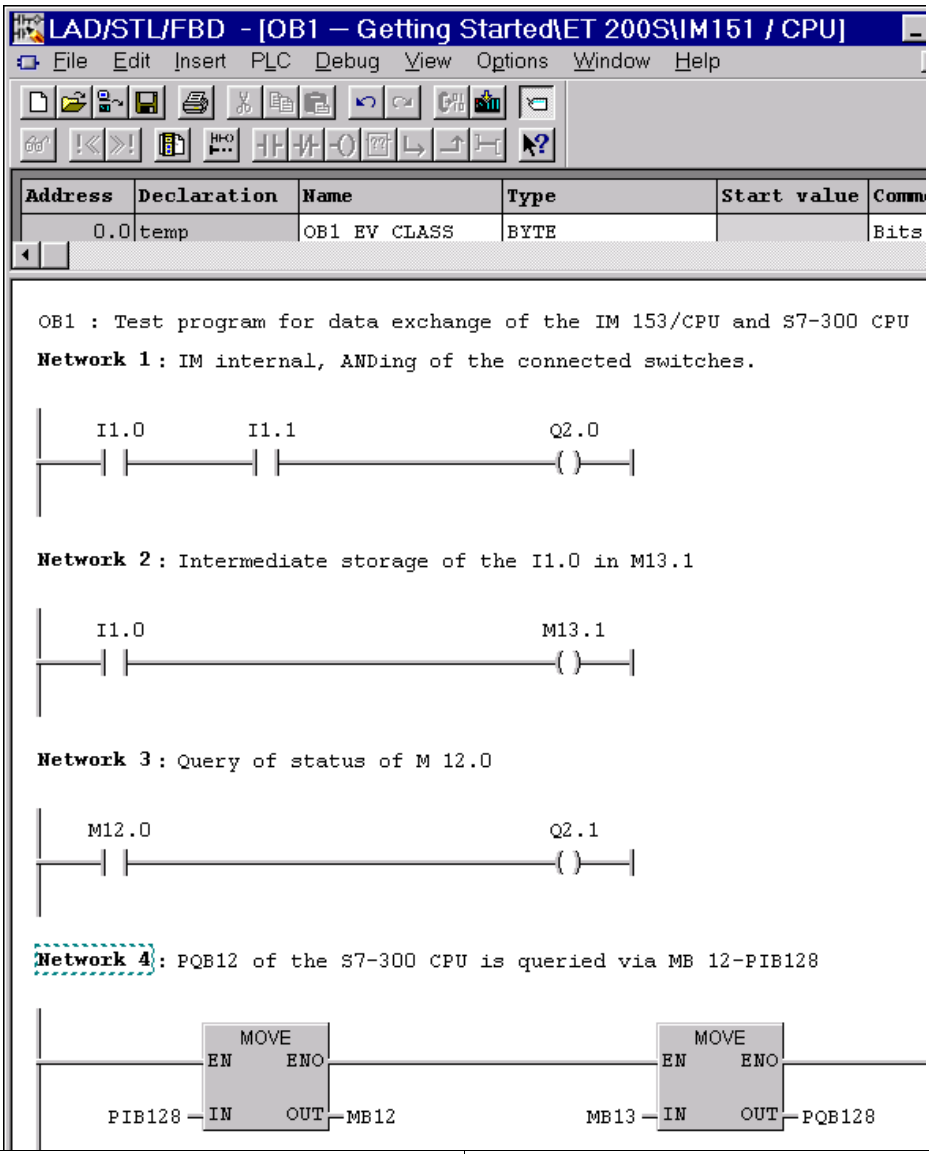
Configure the S7-300 CPU as shown below:

Stage	Procedure	Result
1	In SIMATIC Manager, select the **Getting Started** project in the left-hand part of the window.	
2	Insert a new S7-300 station in the project as described in step 4, stage 4.	
3	In SIMATIC Manager, click the S7-300(1) station in the left-hand part of the window.	The **Hardware** icon appears in the right-hand part of the window.
4	Double-click the **Hardware** icon in the right-hand part of the window.	The hardware editor is opened.

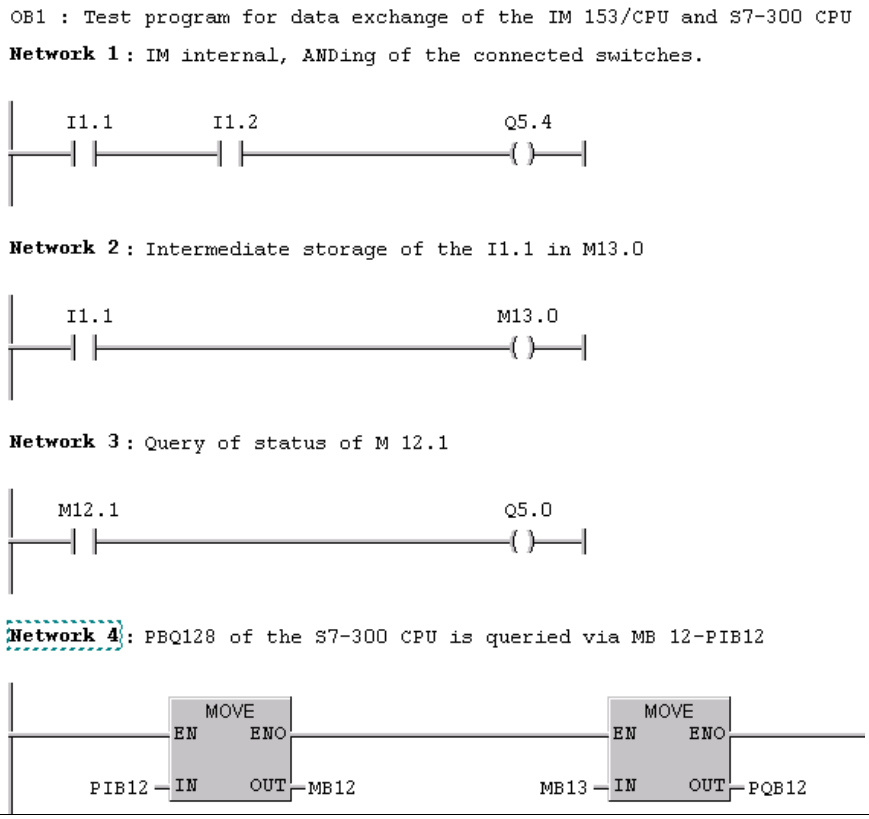
5	If a catalog is not displayed in the right-hand part of the window, activate the catalog by choosing the **Catalog** command from the **View** menu. Navigate in the catalog to **Rack 300** via **SIMATIC 300.** Insert a rail by dragging and dropping it in the window in the upper-left corner.	
6	Insert the PS in slot 1 as described in step 4 (the order number corresponds to your PS order number. Do the same for the S7-300 CPU (slot 2), the S7-300 DI (slot 4) and the S7-300 DO (slot 5). **Note:** When you insert the S7-300 CPU, a dialog box appears. Select the PROFIBUS network from the window, and set address *2*. Confirm with **OK**.	Configuration example (might differ): 
7	• Double-click **CPU 315-2 DP** (line **2**) in the lower-left part of the hardware configuration program window. • In the dialog box that appears, click the **Properties** button on the **General** tab. • In the “MPI Network” dialog box that appears, check whether address *2* is set. If not, set it. Confirm with **OK**.	
8	Navigate in the catalog to **Configured Stations** via **PROFIBUS-DP**. Add the **ET 200S / CPU** station to the **Profibus master system** by dragging and dropping it. 

9	In the dialog box that appears, click the **Interconnect** button.  In the dialog box, select the Configuration tab and fill it in as shown in the screen shot. Confirm with **OK.**	
10	From the **Station** menu, choose the **Save and Compile** command. Close the hardware editor	The hardware configuration is compiled and saved, and the editor is closed.

Step 9: Programming the IM 151/CPU and the S7-300 CPU

Stage	Procedure	Result
1	Navigate in SIMATIC Manager to the block container of the ET 200S. Double-click the **OB1** icon in the right-hand part of the window.	The LAD/FBD/STL editor is opened to edit block OB1.
2	Add to OB1 of the IM 151/CPU as shown below: 	
3	Navigate in SIMATIC Manager to the block container of the S7-300. Double-click the **OB1** icon in the right-hand part of the window.	The LAD/FBD/STL editor is opened to edit block OB1.

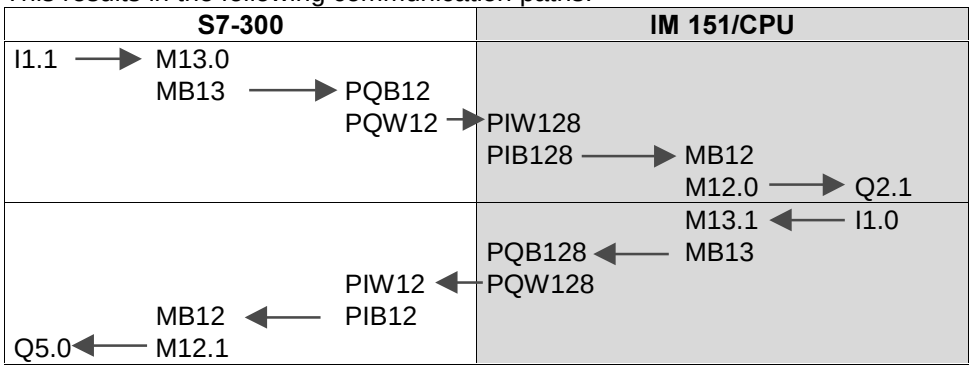
4	Add to the OB1 of the S7-300 CPU as shown below:



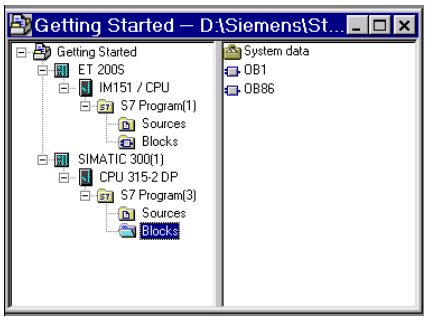
How it works: The status of the switch connected to I1.1 of the S7-300 is queried and temporarily stored in memory marker M13.0. The entire MB13 memory byte is transferred to the PQB12 I/O output byte. You specified in the hardware configuration in step 8 – Configuring the S7-300 (stage 9) – that the area from PQW12 to PQW44 in the S7-300 CPU is to be assigned to the area from PIW128 to PIW160 in the IM 151/CPU.

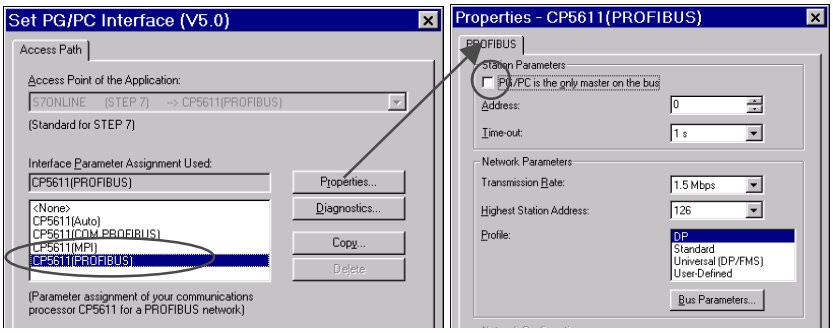
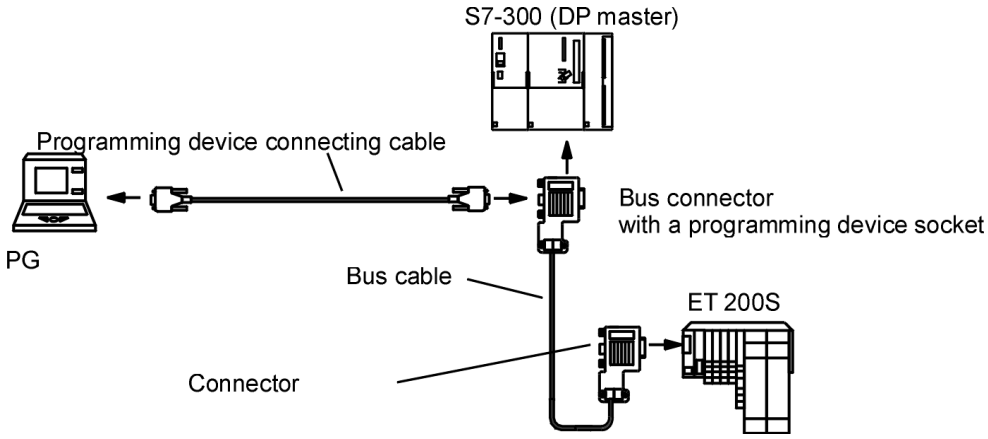
In the IM program, PIB128 is transferred to the MB12 memory byte. The memory marker M12.0 addresses output Q2.1.

This results in the following communication paths:



Step 10: Putting the IM 151/CPU and S7-300 into Operation and Carrying Out a Test Run

Stage	Procedure	Result
1	Navigate in SIMATIC Manager to the block container of the S7-300, and insert an empty organization block with the name *OB86* in the block container. This block ensures that the S7-300 CPU will start even if the IM157/CPU is still indicating an I/O error. Generate OB82 in the same way.	
2	Make sure that the S7 and IM mode selectors are in the *STOP* position. Switch on the PSs of the S7-300 and the ET 200S.	The IM 151/CPU and S7-300 CPU request a memory reset.
3	Reset the memory of the IM157/CPU as described in step 3, stage 3. Execute a memory reset for the S7-300 CPU as follows: • Turn the mode selector to the *MRES* position. Keep the mode selector in this position until the *STOP* LED lights up for the second time and stays on (this takes 3 seconds). • Within the next 3 seconds you must turn the mode selector back to the *MRES* position. The *STOP* LED starts to flash rapidly and the CPU executes a memory reset. When the *STOP* LED comes on continuously again, the CPU has completed the memory reset.	A memory reset is executed for the two CPUs.
4	In SIMATIC Manager, choose the **Download** command from the **PLC** menu to transfer the program and the hardware configuration to the S7-300 CPU. Click **Yes** in all the dialog boxes that appear.	The program and configuration are downloaded from the programming device to the CPU.
5a	Start the Setting the PG/PC Interface program as described in step 3, stage 6. Change the configuration of the programming device/PC interface as shown below:	

5b		
6	Confirm the settings with OK, and close the Setting the PG/PC Interface program.	
7	Open the front panel of the S7-300 CPU. Remove the connector of the programming device cable from the MPI interface of the S7-300 CPU, and put it on the bus connector of the PROFIBUS-DP cable on the S7-300 CPU. Secure the connector. Close the front panel of the S7-300 CPU as far as possible. 	
8	Navigate in SIMATIC Manager to the block container of the ET 200S. Select the block container in the left-hand part of the window. In SIMATIC Manager, choose the **Download** command from the **PLC** menu to transfer the program and the hardware configuration to the IM 151/CPU. Click **Yes** in all the dialog boxes that appear.	The program and configuration are downloaded from the programming device to the CPU.
9	Put the mode selector of the IM 151/CPU into the *RUN-P* position .	The *STOP* LED of the IM goes out. The *RUN* LED starts to flash and then comes on continuously. The *SF* LED comes on.
10	Put the mode selector of the S7-300 CPU into the *RUN-P* position.	The *STOP* LED of the S7 goes out. The *RUN* LED starts to flash and then comes on continuously. The *SF* LED of the IM goes out.

11	Operate the two switches of the S7-300 alternately.	The LEDs of the S7-300 inputs 1.1 and 1.2 light up alternately. The LED of the 5.4 output does not light up.
12	Operate the two switches of the S7-300 simultaneously.	The LEDs of inputs 1.1 and 1.2 light up together. Because the two switches are ANDed in the program (=series connection) and assigned to the output 5.4, the LED of output 5.4 lights up.
13	Operate the switch that is connected to I1.0 of the ET 200S.	The LEDs of the IM input 1.0 and the S7-300 output 5.0 light up.
14	Operate the switch that is connected to I1.1 of the S7-300.	The LEDs of the S7-300 input 1.1 and the IM output 2.1 light up.

Note

Each time the power is switched off and on again, the user program in the memory of the IM 151/CPU is deleted. To download the program again, you have the three following options:

1. Download your program from the programming device to the IM 151/CPU.
2. Store your program on the MMC in the IM 151/CPU (STEP 7: **PLC** menu**, Copy RAM to ROM** menu command).
3. Store your program on the MMC in the programming device (STEP 7: **PLC** menu**, Save Project on PG Memory Card** menu command), and insert the MMC in the IM 151/CPU.

Diagnostics and Debugging

Incorrect operation, incorrect wiring or incorrect hardware configuration can result in faults that the CPU indicates after a memory reset with the group error LED (*SF)*.

You can find out how to evaluate these errors and messages in the following manuals*:*

- *S7-300 Hardware and Installation*; Section 8.3.2
- *Programming with STEP 7 V5.0*; Chapter 21
- *IM 151/CPU Interface Module*, Chapter 5

Additional Help

We recommend that you also read the following Getting Started document: *Getting Started – First Steps with STEP 7 V5.0*.

You can download all the manuals free of charge from the Siemens home page (Automation and Drives Customer Support).

Configuration Frame and Parameter Assignment Frame for the ET 200S

A

By configuring the DP master you can specify which ID format should be used in the configuration frame:

- Configuration in STEP 7 with HWConfig: Integration of the IM 151/CPU slave, during operation on the S7 master: Special ID format **SKF**
- Configuration with another configuration tool: Integration of the IM 151/CPU slave via the DDB: Normal ID format **AKF**

Configuration with *STEP 7* V5.1 or Higher

When you use an S7 master and configure and parameterize it with *STEP 7*, you are supported during input by *STEP 7* and by the online help system.

You do **not** need the information in this appendix.

Configuration with Any Other Configuration Software

If you enter the address areas of the intermediate memory of the CPU using a configuration and parameter assignment frame, you will find the information you require for the IM 151/CPU in the following appendix.

When the configuration is displayed, the ID format in the configuration frame is used.

Appendix	Contents	Page
A.1	Structure of the Configuration Frame (**SKF**)	A-2
A.2	Structure of the Configuration Frame (**AKF**)	A-4
A.3	Structure of the Parameter Assignment Frame	A-6

A.1 Structure of the Configuration Frame (SKF)

The length of the configuration frame depends on the number of address areas configured in the intermediate memory of the CPU component. The first 15 bytes in the configuration frame are already allocated because the first 3 identifiers of the 5-byte IDs are constant. Maximum area that can be displayed: 0-64 bytes/words of inputs, 0-64 bytes/words of outputs, an asymmetrical configuration is possible.

The structure of the configuration frame in the special identifier format for the IM 151/CPU is as follows:

Table A-1 Structure of the Configuration Frame in the special identifier format (SKF)

Configured Address Area	Byte				
	n	n + 1	n + 2	n + 3	n + 4
Fixed range (byte 0 ... 14)	04	00	00	QD	C4
	04	00	00	8B	41
	04	00	00	8F	C0
1st configured address area (byte 15 ... 19)	See Table A-2				
2nd configured address area (byte 20 ... 24)					
...					
32nd configured address area (byte 170 ... 174)					

Identifiers for the Address Areas

The identifiers for configuration depend on the type of the address area. The following table lists all the identifiers for the address areas.

Table A-2 Identifiers for the Address Areas of the Intermediate Memory

Address Area	Identifiers (Hexadecimal)				
	SFF	Length Byte	Manufacturer-Specific Data Comment Length = 3		
	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
Input	See Figure A-1	See Figure A-2	00_H	83_H	40_H
Output			00_H	93_H	40_H

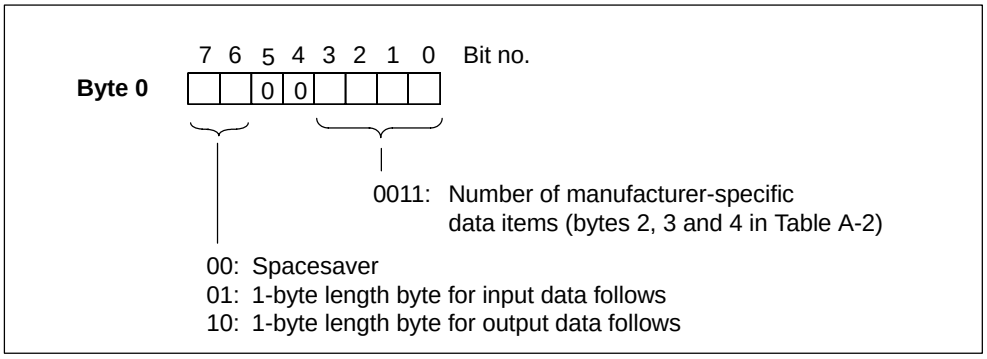


Figure A-1 Description of Byte 0 of the Address Area Identifiers

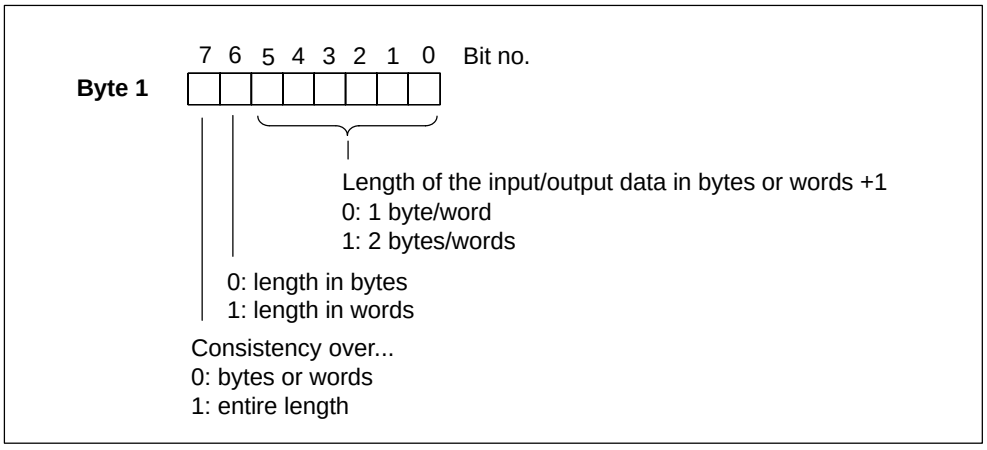


Figure A-2 Description of Byte 1 of the Address Area Identifiers

A.2 Structure of the Configuration Frame (AKF)

DDB (Device Database) File

If your DP master doesn't support the configuration frame in the special ID format (e.g. a non-SIMATIC DP master), you can obtain a DDB file with the normal ID format on the Internet at http://www.ad.siemens.de/csi/gsd (see also Section 5.1).

Structure of the Configuration Frame

The length of the configuration frame depends on the number of configured address areas of the intermediate memory of the CPU component. The first three bytes of the configuration frame are always "0". This represents the 3 fixed 1-byte identifiers. Maximum area that can be displayed: 0-16 bytes/words of inputs, 0-16 bytes/words of outputs, an asymmetrical configuration is possible.

The structure of the configuration frame in the normal identifier format is as follows:

Table A-3 Structure of the Configuration Frame in the Normal Identifier Format (AKF)

Configured Address Areas	Byte
1.	0 0 0 0 0 0 0 0
2.	0 0 0 0 0 0 0 0
3.	0 0 0 0 0 0 0 0
4. : : 32nd	7 6 5 4 3 2 1 0 Bit no. Length of the input/output data in bytes or words +1 01: input data 10: output data 0: length in bytes 1: length in words Consistency over... 0: bytes or words 1: entire length

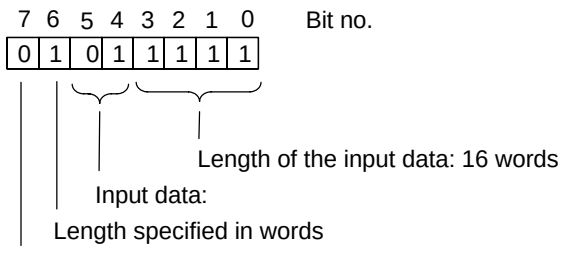
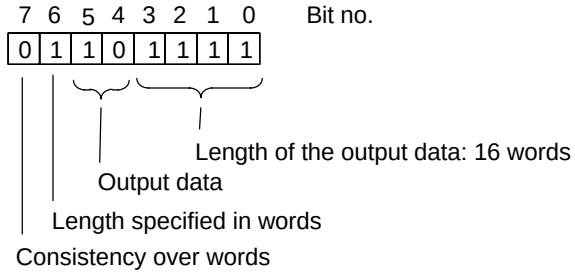
Configuration Frame with the Default Setting for Address Areas

If you don't parameterize the IM 151/CPU and therefore don't specify any address areas for data transfer with the DP master, the IM 151/CPU will start up after commissioning with a default setting on the PROFIBUS-DP.

In the figure below you will find the configuration frame for the default setting:

- 16 words of input data; unit consistency (i.e. word)
- 16 words of output data; unit consistency (i.e. word)

Table A-4 Structure of the Configuration Frame with Default Setting for the Address Areas (Normal Identifier Format in Accordance with AKF)

Configured Address Areas	Byte
1.	0 0 0 0 0 0 0 0
2.	0 0 0 0 0 0 0 0
3.	0 0 0 0 0 0 0 0
4.	
5.	

A.3 Structure of the Parameter Assignment Frame

All the parameterizable values of a DP slave are stored in the parameter assignment frame.

Structure of the Parameter Assignment Frame

The length of the parameter assignment frame for the IM 151/CPU is 16 bytes:

- Standard section; bytes 0 to 6
- Parameters of the IM 151/CPU; bytes 7 to 15

Structure of the Standard Section

The first 7 bytes of the parameter assignment frame are standardized to EN 50170, Volume 2, PROFIBUS and have the following contents:

Byte 0	08_H	Station status
Byte 1	01_H	Watchdog factor 1
Byte 2	06_H	Watchdog factor 2
Byte 3	$0B_H$	Response delay T_{RDY}
Byte 4	80_H	Manufacturer ID, high-byte
Byte 5	$6C_H$ and $6D_H$	Manufacturer ID, low-byte
Byte 6	00_H	Group ID

Figure A-3 Standard Section of the Parameter Assignment Frame

Structure of the General Parameters for the IM 151/CPU

The length of the general parameters for the IM 151/CPU is 3 bytes.

The following parameters can be set:

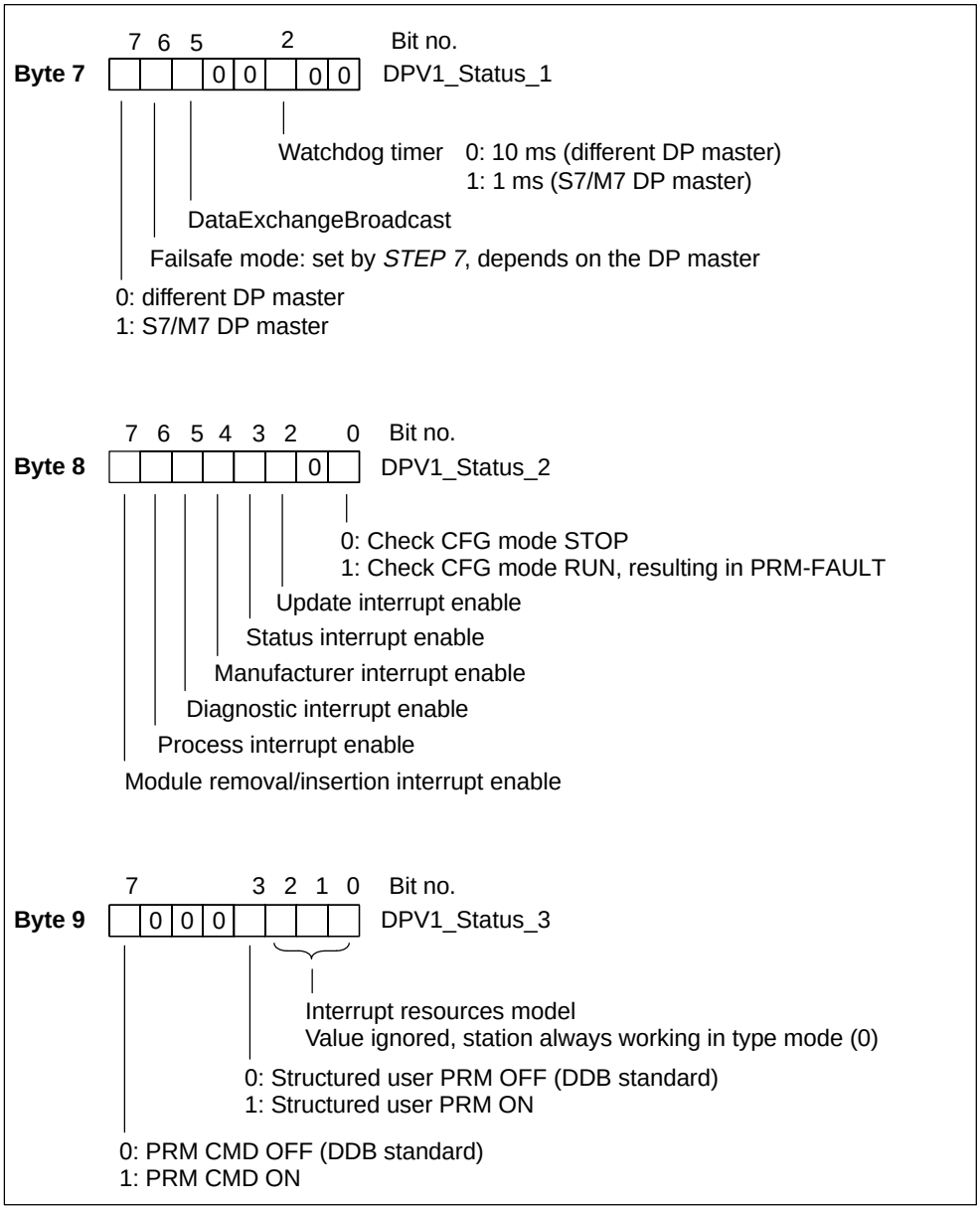


Figure A-4 General Section of the Parameter Assignment Frame for the IM 151/CPU

Instruction List

B

This appendix contains the entire set of instructions available to you for programming the CPU component of the IM 151/CPU using *STEP 7*. The typical execution time is also specified for each instruction.

You will find detailed descriptions of all the instructions, together with examples, in the *STEP 7 programming manuals*.

Note

In the case of indirect addressing (for examples see Appendix B.4), you have to add the time for loading the address of the instruction to the execution times (see Appendix B.5).

Overview

Appendix	Contents	Page
B.1	Address Identifiers and Parameter Ranges	B-2
B.2	Abbreviations	B-3
B.3	Tab	B-3
B.4	Addressing Examples	B-5
B.5	Execution Times in the Case of Indirect Addressing	B-7
B.6	Instruction List	B-13

B.1 Address Identifiers and Parameter Ranges

Address ID	Parameter Range	Description
Q	0.0 to 127.7	Output (in PIO)
QB	0 to 127	Output byte (in PIQ)
QW	0 to 126	Output word (in PIQ)
QD	0 to 124	Output double word (in PIQ)
B	–	Byte with general register-indirect addressing
W	–	Word with general register-indirect addressing
D	–	Double word with general register-indirect addressing
DBX	0.0 to 8191.7	Data bit in data block
DB	1 to 127	Data block
DBB	0 to 8191	Data byte in DB
DBW	0 to 8190	Data word in DB
DBD	0 to 8188	Data double word in DB
DIX	0.0 to 8191.7	Data bit in instance DB
DI	1 to 127	Instance data block
DIB	0 to 8191	Data byte in instance DB
DIW	0 to 8190	Data word in instance DB
DID	0 to 8188	Data double word in instance DB
I	0.0 to 127.7	Input (in PII)
IB	0 to 127	Input byte (in PII)
IW	0 to 126	Input word (in PII)
ID	0 to 124	Input double word (in PII)
L	0.0 to 255.7	Local data
LB	0 to 255	Local data byte
LW	0 to 254	Local data word
LD	0 to 252	Local data double word
M	0.0 to 255.7	Memory markers
MB	0 to 255	Memory byte
MW	0 to 254	Memory word
MD	0 to 252	Memory double word
PQB	0 to 1535	Peripheral output byte
PQW	0 to 1534	Peripheral output word
PQD	0 to 1535	Peripheral output double word
PIB	0 to 1535	Peripheral input byte
PIW	0 to 1534	Peripheral input word
PID	0 to 1532	Peripheral input double word
T	0 to 127	Timer
C	0 to 63	Counter
Parameters	–	Instruction addressed via parameter
B#	–	Constant, 2 or 4 bytes
D#	–	IEC data constant
L#	–	32-bit integer constant
P#	–	Pointer constant
S5T#	–	S5 time constant (16 bits)*
T#	**	Time constant (16/32 bits)
TOD#	–	IEC time constant (32 bits)
C#	–	Counter constant (16/32 bits)
2#	–	Binary constant (16/32 bits)
16#	–	Hexadecimal constant (16/32 bits)

* for loading S5 timers

** T # 1D_5M_3M_1S_2MS

B.2 Abbreviations

The following abbreviations and mnemonics are used in the Instruction List:

Abbreviation	Stands for	Example
k8	8-bit constant	32
k16	16-bit constant	62 531
k32	32-bit constant	127 624
i8	8-bit integer	–155
i16	16-bit integer	+6523
i32	32-bit integer	–2 222 222
m	P#x.y (pointer)	P#240.3
n	Binary constant	1001 1100
p	Hexadecimal constant	EA12
LABEL	Symbolic jump address (max. 4 characters)	DEST

B.3 Tab

ACCU1 and ACCU2 (32 Bits)

The accumulators are registers for processing bytes, words or double words. The address IDs are loaded into the accumulators, where they are logically gated. The result of the logic operation (RLO) is always in ACCU1.

The accumulators are 32 bits long.

Accumulator designations:

ACCU		Bit
ACCU1	ACCU2	Bits 0 to 31
ACCU1-L	ACCU2-L	Bits 0 to 15
ACCU1-H	ACCU2-H	Bits 16 to 31
ACCU1-LL	ACCU2-LL	Bits 0 to 7
ACCU1-LH	ACCU2-LH	Bits 8 to 15
ACCU1-HL	ACCU2-HL	Bits 16 to 23
ACCU1-HH	ACCU2-HH	Bits 24 to 31

Address Registers AR1 and AR2 (32 Bits)

The address registers contain the area-specific or general addresses for instructions that use register-indirect addressing. The address registers are 32 bits long.

The area-internal and/or area-crossing addresses have the following structure:

- Area-internal address:

```
00000000 00000bbb bbbbbbbb bbbbbxxx
```

- Area-crossing address:

```
10000yyy 00000bbb bbbbbbbb bbbbbxxx
```

Legend:

	b	Byte address
	x	Bit number
	y	Area identifier (see Appendix B.4)

Status Word (16 Bits)

The status word bits are evaluated or set by the instructions.

The status word is 16 bits long.

Bit	Assign-ment	Description
0	/FC	First check bit *
1	RLO	Result of (previous) logic operation
2	STA	Status *
3	OR	Or *
4	OS	Stored overflow
5	OV	Overflow
6	A0	Condition code
7	A1	Condition code
8	BR	Binary result
9 ... 15	Unassigned	–

* Bit cannot be evaluated in the user program with the L STW instruction, since it is not updated at program runtime.

B.4 Examples of Addressing

Addressing Examples	Description
Direct Addressing	
L +27	Load 16-bit integer constant ”27” into ACCU1
L L#–1	Load 32-bit integer constant ”–1” into ACCU1
L 2#1010101010101010	Load binary constant into ACCU1
L DW#16#A0F0 BCFD	Load hexadecimal constant into ACCU1
L ’END’	Load ASCII character into ACCU1
L T#500 ms	Load time value into ACCU1
L P#10.0	Load area-internal pointer into ACCU1
L P#E20.6	Load area-crossing pointer into ACCU1
L –2.5	Load real number into ACCU1
L D#1997–01–20	Load date
L TOD#13:20:33.125	Load time of day
Direct Addressing	
A I 0.0	ANDing of input bit 0.0
L IB 1	Load input byte 1 into ACCU1
L IW 0	Load input word 0 into ACCU1
L ID 0	Load input double word 0 into ACCU1
Indirect Addressing of Timers/Counters	
SP T [LW 8]	Start timer; the timer number is in local data word 8
CU C [LW 10]	Start counter; the counter number is in local data word 10
Area-Internal, Memory-Indirect Addressing	
A I [LD 12] Example: L P#22.2 T LD 12 A I [LD 12]	AND operation: The address of the input is in local data double word 12 as a pointer
A I [DBD 1]	AND operation: The address of the input is in data double word 1 of the DB as a pointer
A Q [DID 12]	AND operation: The address of the output is in data double word 12 of the instance DB as a pointer
A Q [MD 12]	AND operation: The address of the output is in memory marker double word 12 as a pointer
Area-Internal, Memory-Indirect Addressing	
A I [AR1,P#12.2]	AND operation: The address of the input is calculated from the pointer value in AR1 + P#12.2

Addressing Examples	Description
Area-Crossing, Memory-Indirect Addressing	
For area-crossing, register-indirect addressing, bits 24 to 26 of the address must also contain an area identifier. The address is in the address register. **Area identifier** / **Coding Binary** / **Hex.** / **Area** P / 1000 0**000** / 80 / I/O area I / 1000 0**001** / 81 / Input area Q / 1000 0**010** / 82 / Output area M / 1000 0**011** / 83 / Memory marker area DB / 1000 0**100** / 84 / Data area DI / 1000 0**101** / 85 / Instance data area L / 1000 0**110** / 86 / Local data area VL / 1000 0**111** / 87 / Predecessor local data area (access to local data of the calling block; see page B-5)	
L B [AR1,P#8.0]	Load byte into ACCU1: The address is calculated from the pointer value in AR 1 + P#8.0
A [AR1,P#32.3]	AND operation: The address of the address ID is calculated from the ”pointer value in AR1 + P#32.3”
Addressing Via Parameters	
A Parameter	Addressing via parameters

Calculating Pointers

Here are 2 examples of how to calculate pointers:

Example for the sum of the bit addresses ≦ 7:

```
LAR1 P#8.2
A I [AR1,P#10.2]
```

Result: Input 18.4 is addressed (by adding the byte and bit addresses)

Example for the sum of the bit addresses > 7:

```
L MD 0      Random pointer, e.g. P#10.5
LAR1
A I [AR1,P#10.7]
```

Result: Input 21.4 is addressed (by adding the byte and bit addresses with carryover)

B.5 Execution Times with Indirect Addressing

Two-Part Statement

A statement with indirectly addressed instructions consists of 2 parts:

Part 1: Loading the Address of the Address ID

Part 2: Executing the Instruction

In other words, you must calculate the execution time of a statement with indirectly addressed instructions from these two parts.

Calculating the Execution Time

The total execution time is calculated as follows:

	Time required for loading the address
+	execution time of the instruction
=	Total execution time of the instruction

The execution times given in Appendix B.6 and the following chapters are the execution times for the second part of a statement (i.e. for the actual execution of an instruction).

You then have to add the time required for loading the address of the instruction to this execution time.

Loading an Address

The execution time for loading the address of the instruction from the various areas is shown in the following table.

Address Is In ...	Execution Time in µs
Memory marker area M	
Word	0.8
Double word	2.1
Data block DB/DX	
Word	3.0
Double word	4.1
Local data area L	
Word	0.9
Double word	2.2
AR1/AR2 (area-internal)	1.7
AR1/AR2 (area-crossing)	3.2

Address Is In ...	Execution Time in µs
Parameter (word) ... for: • Timers • Counters • Block calls	2.1
Parameter (double word) ... for: Bits, bytes, words and double words	4.3

B.5.1 Example of Memory-Indirect, Area-Internal Addressing

Example

Example: A I [DBD 12]

Step 1

Load the contents of DBD 12

Address Is In ...	Execution Time in µs
Memory marker area M	
Word	0.8
Double word	2.1
Data block DB/DX	
Word	3.0
Double word	4.1

Step 2

ANDing of the addressed input (the execution time is in Appendix B.6 and the following chapters).

Typical Execution Time in µs	
Direct Addressing	**Indirect Addressing**
0.3 :	**1.6+** Time for A I **:**

Total Execution Time

This results in a total execution time of:

	4.1 µs
+	1.6 µs
=	5.7 µs

B.5.2 Example of Register-Indirect, Area-Internal Addressing

Example

A I [AR1, P#34.3]

Step 1

Load the contents of AR1 and increment it by the offset 34.3

Address Is In ...	Execution Time in µs
:	:
AR1/AR2 (area-internal)	1.7
:	:

Step 2

ANDing of the addressed input (the execution time is in Appendix B.6 and the following chapters).

Typical Execution Time in µs	
Direct Addressing	**Indirect Addressing**
0.3 :	Time for A I **1.6+ :**

Total Execution Time

This results in a total execution time of:

Total execution time:

	1.7 µs
+	1.6 µs
=	3.3 µs

B.5.3 Example of Memory-Indirect, Area-Crossing Addressing

Example

U [AR1, P#23.1] ... with P#E 1.0 in the AR1

Step 1

Load the contents of AR1 and increment it by the offset 23.1

Address Is In ...	Execution Time in µs
:	:
AR1/AR2 (area-crossing)	3.2
:	:

Step 2

ANDing of the addressed input (the execution time is in Appendix B.6 and the following chapters).

Typical Execution Time in µs	
Direct Addressing	**Indirect Addressing**
0.3 :	**1.6+** Time for A I **:**

Total Execution Time

This results in a total execution time of:

Total execution time:

	3.2 µs
+	1.6 µs
=	4.8 µs

B.5.4 Example of Addressing Via Parameters

Example

A parameter ... with I 0.5 in the block parameter list

Step 1

Loading the I 0.5 addressed via the parameter.

Address Is In ...	Execution Time in µs
:	:
:	:
Parameter (double word)	4.3

Step 2

ANDing of the addressed input (the execution time is in Appendix B.6 and the following chapters).

Typical Execution Time in µs	
Direct Addressing	**Indirect Addressing**
0.3 :	**1.6+** Time for A I **:**

Total Execution Time

This results in a total execution time of:

Total execution time:

	4.3 µs
+	1.6 µs
=	5.9 µs

B.6 Bit Logic Instructions

Examination of the signal state of the addressed instruction and gating of the result with the RLO in accordance with the appropriate logic function.

Instruction	Address ID	Description	Length in Words	Typical Execution Time in µs	
				Direct Addressing	Indirect Addressing*
A		AND			
	I/O	Input/output	1**/2	0.3	1.6+
	M	Memory markers	1**/2	0.6	1.7+
	L	Local data bit	2	0.9	1.8+
	DBX/DIX	Data bit	2	2.8	2.5+
	[AR1,m]	I/Q/M/L/DBX/DIX (addressed	2	–	+
	[AR2,m]	(area-crossing) via AR1/AR2 or		–	+
	Parameters	via parameter)		–	+
AN		AND NOT			
	I/O	Input/output	2	0.5	1.9+
	M	Memory markers		0.8	2.1+
	L	Local data bit		1.0	2.2+
	DBX/DIX	Data bit		3.1	2.8+
	[AR1,m]	I/Q/M/L/DBX/DIX (addressed	2	–	–
	[AR2,m]	(area-crossing) via AR1/AR2 or		–	–
	Parameters	via parameter)		–	–

Status word for: **A, AN**	BR	A1	A0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	–	–	–	–	–	Yes	–	Yes	Yes
Instruction controls:	–	–	–	–	–	Yes	Yes	Yes	1

O		OR			
	I/O	Input/output	1**/2	0.3	1.6+
	M	Memory markers	1**/2	0.7	1.7+
	L	Local data bit	2	0.9	1.8+
	DBX/DIX	Data bit	2	2.9	2.5+
	[AR1,m]	I/Q/M/L/DBX/DIX (addressed	2	–	+
	[AR2,m]	(area-crossing) via AR1/AR2 or via		–	+
	Parameters	parameter)		–	+
ON		OR NOT			
	I/O	Input/output	1**/2	0.5	1.6+
	M	Memory markers	1**/2	0.8	2.0+
	L	Local data bit	2	2.0	2.2+
	DBX/DIX	Data bit	2	3.1	2.8+
	[AR1,m]	I/Q/M/L/DBX/DIX (addressed	2	–	+
	[AR2,m]	(area-crossing) via AR1/AR2 or via		–	+
	Parameters	parameter)		–	+

Status word for: **O, ON**	BR	A1	A0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	–	–	–	–	–	–	–	Yes	Yes
Instruction controls:	–	–	–	–	–	0	Yes	Yes	1

* + time for loading the address of the instruction
** With direct instruction addressing

Instruction	Address ID	Description	Length in Words	Typical Execution Time in µs	
				Direct Addressing	Indirect Addressing*
X		EXCLUSIVE OR			
	I/O	Input/output	2	0.3	1.6+
	M	Memory markers		0.7	1.7+
	L	Local data bit		0.9	1.9+
	DBX/DIX	Data bit		2.9	2.5+
	[AR1,m]	I/Q/M/L/DBX/DIX (addressed	2	–	+
	[AR2,m]	(area-crossing) via AR1/AR2 or via		–	+
	Parameters	parameter)		–	+
XN		EXCLUSIVE OR NOT			
	I/O	Input/output	2	0.5	1.9+
	M	Memory markers		0.8	2.0+
	L	Local data bit		1.0	2.2+
	DBX/DIX	Data bit		3.1	2.8+
	[AR1,m]	I/Q/M/L/DBX/DIX (addressed	2	–	+
	[AR2,m]	(area-crossing) via AR1/AR2 or via		–	+
	Parameters	parameter)		–	+

Status word for: **X, XN**	BR	A1	A0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	–	–	–	–	–	–	–	Yes	Yes
Instruction controls:	–	–	–	–	–	0	Yes	Yes	1

* + time for loading the address of the instruction

B.7 Logic Instructions with Parenthetical Expressions

Saving of the BR, RLO and OR bits and a function identifier (A, AN, ...) to the nesting stack. 7 nesting levels are possible per block.

Instruction	Address ID	Description	Length in Words	Typical Execution Time in µs
A(		AND left parenthesis	1	1.7
AN(		AND NOT left parenthesis	1	1.7
O(		OR left parenthesis	1	1.7
ON(		OR NOT left parenthesis	1	1.7
X(		EXCLUSIVE OR left parenthesis	1	1.7
XN(		EXCLUSIVE OR NOT left parenthesis	1	1.7

Status word for: **A(, AN(, O(, ON(, X(, XN(**	BR	A1	A0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	Yes	–	–	–	–	Yes	–	Yes	Yes
Instruction controls:	–	–	–	–	–	0	1	–	0

)		Right parenthesis, removing an entry from the nesting stack, gating the RLO with the current RLO in the processor	1	1.9

Status word for: **)**	BR	A1	A0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	–	–	–	–	–	–	–	Yes	–
Instruction controls:	Yes	–	–	–	–	Yes	1	Yes	1

B.8 ORing of AND Operations

The ORing of AND functions is carried out according to the principle: AND before OR.

In-struc-tion	Address ID	Description	Length in Words	Typical Execution Time in µs
O		The ORing of AND functions is carried out according to the principle: AND before OR.	1	0.5

Status word for: **O**	BR	A1	A0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	–	–	–	–	–	Yes	–	Yes	Yes
Instruction controls:	–	–	–	–	–	Yes	1	–	Yes

B.9 Logic Instructions with Timers and Counters

Querying of the signal state of the addressed timer/counter and gating of the result with the RLO in accordance with the corresponding function.

Instruction	Address ID	Description	Length in Words	Typical Execution Time in µs: Direct Addressing	Typical Execution Time in µs: Indirect Addressing*
A	T C	AND timer AND counter	1**/2	0.9 0.6	2.1+ 1.8+
	Timer para. Counter p.	AND timer/counter (addressed via parameter)	2	– –	+ +
AN	T C	AND NOT timer AND NOT counter	1**/2	1.1 0.9	2.3+ 2.1+
	Timer para. Counter p.	AND NOT timer/counter (addressed via parameter)	2	– –	+ +

Status word for: **A, AN**	BR	A1	A0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	–	–	–	–	–	Yes	–	Yes	Yes
Instruction controls:	–	–	–	–	–	Yes	Yes	Yes	1

O	T C	OR timer OR counter	1**/2	0.9 0.6	2.1+ 1.8+
	Timer para. Counter p.	OR timer/counter (addressed via parameter)	2	– –	+ +
ON	T C	OR NOT timer OR NOT counter	1**/2	1.1 0.9	2.3+ 2.1+
	Timer para. Counter p.	OR NOT timer/counter (addressed via parameter)	2	– –	+ +
X	T C	EXCLUSIVE OR timer EXCLUSIVE OR counter	2	0.9 0.6	2.1+ 1.8+
	Timer para. Counter p.	EXCLUSIVE OR timer/counter (addressed via parameter)	2	– –	+ +
XN	T C	EXCLUSIVE OR NOT timer EXCLUSIVE OR NOT counter	2	1.1 0.9	2.3+ 2.1+
	Timer para. Counter p.	EXCLUSIVE OR NOT timer/counter (addressed via parameter)	2	– –	+ +

Status word for: **O, ON, X, XN**	BR	A1	A0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	–	–	–	–	–	–	–	Yes	Yes
Instruction controls:	–	–	–	–	–	0	Yes	Yes	1

* + time for loading the address of the instruction
** With direct instruction addressing

B.10 Logic Instructions with the Contents of Accumulator 1

Gating of the contents of ACCU1 or ACCU1-L with a word or double word in accordance with the appropriate function. The word or double word is either a constant in the instruction or in ACCU2. The result is in ACCU1 or ACCU1-L.

In-struc-tion	Address ID	Description	Length in Words	Typical Execution Time in µs
AW		AND ACCU2-L	1	0.6
	k16	AND 16-bit constant	2	0.9
OW		OR ACCU2-L	1	0.6
	k16	OR 16-bit constant	2	0.9
XOW		EXCLUSIVE OR ACCU2–L	1	0.6
	k16	EXCLUSIVE OR 16-bit constant	2	0.9
AD		AND ACCU2	1	2.0
	k16	AND 16-bit constant	3	2.3
OD		OR ACCU2	1	2.0
	k16	OR 16-bit constant	3	2.3
XOD		EXCLUSIVE OR ACCU2	1	2.0
	k16	EXCLUSIVE OR 16-bit constant	3	2.3

Status word for: **AW, OW, XOW, AD, OD, XOD**	BR	A1	A0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	–	–	–	–	–	–	–	–	–
Instruction controls:	–	Yes	0	0	–	–	–	–	–

B.11 Logic Instructions with Condition Code Bits

Examination of the signal state of the specified conditions and gating of the result with the RLO in accordance with the appropriate logic function.

In-struc-tion	Address ID	Description	Length in Words	Typical Execution Time in µs
A	==0	AND result=0 (CC 1=0) and (CC 0=0)	1	0.6
	>0	AND result>0 (CC 1=1) and (CC 0=0)	1	0.9
	<0	AND result<0 (CC 1=0) and (CC 0=1)	1	0.9
	<>0	AND result ≠ 0 ((CC 1=0) and (CC 0=1) or (CC 1=1) and (CC 0=0))	1	0.6
	<=0	AND result<=0 ((CC 1=0) and (CC 0=1) or (CC 1=0) and (CC 0=0))	1	0.6
	>=0	AND result>=0 ((CC 1=1) and (CC 0=0) or (CC 1=0) and (CC 0=0))	1	0.6
	AO	AND unordered (CC 1=1) and (CC 0=1)	1	0.6
	OS	AND OS=1	1	0.3
	BR	AND BR=1	1	0.3
	OV	AND OV=1	1	0.3

Status word for: **A condition**	BR	A1	A0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	Yes	Yes	Yes	Yes	Yes	Yes	–	Yes	Yes
Instruction controls:	–	–	–	–-	–	Yes	Yes	Yes	1

In-struc-tion	Address ID	Description	Length in Words	Typical Execution Time in µs
AN	==0	AND NOT result=0 (CC 1=0) and (CC 0=0)	1	0.6
	>0	AND NOT result>0 (CC 1=1) and (CC 0=0)	1	0.9
	<0	AND NOT result<0 (CC 1=0) and (CC 0=1)	1	0.9
	<>0	AND NOT result ≠ 0 ((CC 1=0) and (CC 0=1) or (CC 1=1) and (CC 0=0))	1	0.9
	<=0	AND NOT result<=0 ((CC 1=0) and (CC 0=1) or (CC 1=0) and (CC 0=0))	1	0.3
	>=0	AND NOT result>=0 ((CC 1=1) and (CC 0=0) or (CC 1=0) and (CC 0=0))	1	0.3
	AO	AND NOT unordered (CC 1=1) and (CC 0=1)	1	0.9
	OS	AND NOT OS=1	1	0.6
	OV	AND NOT OV=1	1	0.6
	BR	AND NOT BR=1	1	0.6

Status word for: **AN condition**	BR	A1	A0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	Yes	Yes	Yes	Yes	Yes	Yes	–	Yes	Yes
Instruction controls:	–	–	–	–	–	Yes	Yes	Yes	1

In-struc-tion	Address ID	Description	Length in Words	Typical Execution Time in µs
O	==0	OR result=0 (CC 1=0) and (CC 0=0)	1	0.6
	>0	OR result>0 (CC 1=1) and (CC 0=0)	1	0.9
	<0	OR result<0 (CC 1=0) and (CC 0=1)	1	0.9
	<>0	OR result ≠ 0 ((CC 1=0) and (CC 0=1) or (CC 1=1) and (CC 0=0))	1	0.6
	<=0	OR result<=0 ((CC 1=0) and (CC 0=1) or (CC 1=0) and (CC 0=0))	1	0.6
	>=0	OR result>=0 ((CC 1=1) and (CC 0=0) or (CC 1=0) and (CC 0=0))	1	0.6
	AO	OR unordered (CC 1=1) and (CC 0=1)	1	0.6
	OS	OR OS=1	1	0.3
	OV	OR OV=1	1	0.3
	BR	OR BR=1	1	0.3

Status word for: **O condition**	BR	A1	A0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	Yes	Yes	Yes	Yes	Yes	–	–	Yes	Yes
Instruction controls:	–	–	–	–	–	0	Yes	Yes	1

In-struc-tion	Address ID	Description	Length in Words	Typical Execution Time in µs
ON	==0	OR NOT result=0 (CC 1=0) and (CC 0=0)	1	0.6
	>0	OR NOT result>0 (CC 1=1) and (CC 0=0)	1	0.9
	<0	OR NOT result<0 (CC 1=0) and (CC 0=1)	1	0.9
	<>0	OR NOT result ≠ 0 ((CC 1=0) and (CC 0=1) or (CC 1=1) and (CC 0=0))	1	0.9
	<=0	OR NOT result<=0 ((CC 1=0) and (CC 0=1) or (CC 1=0) and (CC 0=0))	1	0.3
	>=0	OR NOT result>=0 ((CC 1=1) and (CC 0=0) or (CC 1=0) and (CC 0=0))	1	0.3
	AO	OR NOT unordered (CC 1=1) and (CC 0=1)	1	0.9
	OS	OR NOT OS=1	1	0.6
	OV	OR NOT OV=1	1	0.6
	BR	OR NOT BR=1	1	0.6

Status word for: **ON condition**	BR	A1	A0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	Yes	Yes	Yes	Yes	Yes	–	–	Yes	Yes
Instruction controls:	–	–	–	–	–	0	Yes	Yes	1

In-struc-tion	Address ID	Description	Length in Words	Typical Execution Time in µs
X	==0	EXCLUSIVE OR result=0 (CC 1=0) and (CC 0=0)	1	0.6
	>0	EXCLUSIVE OR result>0 (CC 1=1) and (CC 0=0)	1	0.9
	<0	EXCLUSIVE OR result<0 (CC 1=0) and (CC 0=1)	1	0.9
	<>0	EXCLUSIVE OR result ≠ 0 ((CC 1=0) and (CC 0=1) or (CC 1=1) and (CC 0=0))	1	0.6
	<=0	EXCLUSIVE OR result<=0 ((CC 1=0) and (CC 0=1) or (CC 1=0) and (CC 0=0))	1	0.6
	>=0	EXCLUSIVE OR result>=0 ((CC 1=1) and (CC 0=0) or (CC 1=0) and (CC 0=0))	1	0.6
	AO	EXCLUSIVE OR unordered (CC 1=1) and (CC 0=1)	1	0.6
	OS	EXCLUSIVE OR OS=1	1	0.3
	OV	EXCLUSIVE OR OV=1	1	0.3
	BR	EXCLUSIVE OR BR=1	1	0.3

Status word for: **X condition**	BR	A1	A0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	Yes	Yes	Yes	Yes	Yes	–	–	Yes	Yes
Instruction controls:	–	–	–	–	–	0	Yes	Yes	1

In-struc-tion	Address ID	Description	Length in Words	Typical Execution Time in µs
XN	==0	EXCLUSIVE OR NOT result=0 (CC 1=0) and (CC 0=0)	1	0.6
	>0	EXCLUSIVE OR NOT result>0 (CC 1=1) and (CC 0=0)	1	0.9
	<0	EXCLUSIVE OR NOT result<0 (CC 1=0) and (CC 0=1)	1	0.9
	<>0	EXCLUSIVE OR NOT result≠0 ((CC 1=0) and (CC 0=1) or (CC 1=1) and (CC 0=0))	1	0.9
	<=0	EXCLUSIVE OR NOT result<=0 ((CC 1=0) and (CC 0=1) or (CC 1=0) and (CC 0=0))	1	0.3
	>=0	EXCLUSIVE OR NOT result>=0 ((CC 1=1) and (CC 0=0) or (CC 1=0) and (CC 0=0))	1	0.3
	AO	EXCLUSIVE OR NOT unordered (CC 1=1) and (CC 0=1)	1	0.9
	OS	EXCLUSIVE OR NOT OS=1	1	0.6
	OV	EXCLUSIVE OR NOT OV=1	1	0.6
	BR	EXCLUSIVE OR NOT BR=1	1	0.6

Status word for: **XN condition**	BR	A1	A0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	Yes	Yes	Yes	Yes	Yes	–	–	Yes	Yes
Instruction controls:	–	–	–	–	–	0	Yes	Yes	1

B.12 Edge-Triggered Instructions

Detection of an edge change. The current signal state of the RLO is compared with the signal state of the instruction or ”edge memory marker”.
FP detects an edge change from ”0” to ”1”.
FN detects an edge change from ”1” to ”0”.

Instruction	Address ID	Description	Length in Words	Typical Execution Time in µs	
				Direct Addressing	Indirect Addressing*
FP	I/O M L DBX/DIX [AR1,m] [AR2,m] Parameters	Detection of the rising edge in the RLO. The bit addressed in the instruction is the auxiliary edge memory marker.	2	0.8 1.5 1.6 4.0 – – –	2.4+ 2.7+ 2.7 3.6+ + + +
FN	I/O M L DBX/DIX [AR1,m] [AR2,m] Parameters	Detection of the falling edge in the RLO. The bit addressed in the instruction is the auxiliary edge memory marker.	2	1.0 1.6 1.7 4.1 – – –	2.6+ 2.8+ 2.8+ 3.7+ + + +

Status word for: **FP, FN**	BR	A1	A0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	–	–	–	–	–	–	–	Yes	–
Instruction controls:	–	–	–	–	–	0	Yes	Yes	1

* + time for loading the address of the instruction

B.13 Setting/Resetting Bit Addresses

Assignment of the value ”1” or ”0” or the RLO to the addressed instruction. The instructions can be dependent on the MCR.

Instruction	Address ID	Description	Length in Words	Typical Execution Time in µs	
				Direct Addressing	Indirect Addressing*
S	I/O	Set input/output to ”1”	1**/2	0.3	2.2+
		(MCR-dependent)		0.5	2.9+
	M	Set memory marker to ”1”	1**/2	0.8	2.5+
		(MCR-dependent)		2.3	3.0+
	L	Set local data bit to ”1”	2	1.3	2.5+
		(MCR-dependent)		2.9	2.5+
	DBX/DIX	Set data bit to ”1”	2	3.7	3.5+
		(MCR-dependent)		4.3	4.1+
	[AR1,m] [AR2,m] Parameters	Set I/Q/M/L/DBX/DIX (MCR-dependent) to ” 1”. (addressed (area-crossing) via AR1/AR2 or via parameter)	2	– – –	+ + +
R	I/O	Reset input/output to ”0”	1**/2	0.4	2.3+
		(MCR-dependent)		0.5	3.0+
	M	Reset memory marker to ”0”	1**/2	0.9	2.6+
		(MCR-dependent)		2.4	3.2+
	L	Reset local data bit to ”0”	2	1.3	2.6+
		(MCR-dependent)		3.0	2.7+
	DBX/DIX	Reset data bit to ”0”	2	3.8	3.6+
		(MCR-dependent)		4.3	4.3+
	[AR1,m] [AR2,m] Parameters	Set I/Q/M/L/DBX/DIX (MCR-dependent) to ”0”. (addressed (area-crossing) via AR1/AR2 or via parameter)	2	– – –	+ + +
=	I/O	Assign RLO to input/output	1**/2	0.3	2.2+
		(MCR-dependent)		0.5	2.9+
	M	Assign RLO to memory marker	1**/2	0.9	2.5+
		(MCR-dependent)		2.3	3.0+
	L	Assign RLO to local data bit	2	1.1	2.5+
		(MCR-dependent)		2.6	2.3+
	DBX/DIX	Assign RLO to data bit	2	3.8	3.6+
		(MCR-dependent)		4.4	4.3+
	[AR1,m] [AR2,m] Parameters	Assign RLO to I/Q/M/L/DBX/DIX (via AR1, AR2 (area-crossing) or via parameter)	2	–	+

Status word for:	BR	A1	A0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	–	–	–	–	–	–	–	Yes	–
Instruction controls:	–	–	–	–	–	0	Yes	–	0

* + time for loading the address of the instruction
** With direct instruction addressing

B.14 Instructions Directly Affecting the RLO

The following instructions have a direct effect on the RLO.

Instruction	Address ID	Description	Length in Words	Typical Execution Time in µs
CLR		Set RLO to ”0”	1	0.3

Status word for: **CLR**	BR	A1	A0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	–	–	–	–	–	–	–	–	–
Instruction controls:	–	–	–	–	–	0	0	0	0

SET		Set RLO to ”1”	1	0.3

Status word for: **SET**	BR	A1	A0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	–	–	–	–	–	–	–	–	–
Instruction controls:	–	–	–	–	–	0	1	1	0

NOT		Negate RLO	1	0.3

Status word for: **NOT**	BR	A1	A0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	–	–	–	–	–	Yes	–	Yes	–
Instruction controls:	–	–	–	–	–	–	1	Yes	–

SAVE		Save the RLO in the BR bit	1	0.3

Status word for: **SAVE**	BR	A1	A0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	–	–	–	–	–	–	–	Yes	–
Instruction controls:	Yes	–	–	–	–	–	–	–	–

B.15 Timer Instructions

Starting or resetting of a timer (addressed directly or via a parameter). The duration must be in ACCU1-L.

Instruction	Address ID	Description	Length in Words	Typical Execution Time in µs	
				Direct Addressing	Indirect Addressing*
SP	T	Start timer as pulse on edge change from "0" to "1"	1**/2	9.2	9.7+
	Timer para.		2	–	–
SE	T	Start timer as extended pulse on edge change from "0" to "1"	1**/2	9.2	9.7+
	Timer para.		2	–	–
SD	T	Start timer as ON delay on edge change from "0" to "1"	1**/2	9.7	10.2+
	Timer para.		2	–	–
SS	T	Start timer as retentive ON delay on edge change from "0" to "1"	1**/2	9.7	10.2+
	Timer para.		2	–	–
SF	T	Start timer as OFF delay on edge change from "1" to "0"	1**/2	10.0	10.5+
	Timer para.		2	–	–
FR	T	Enable timer for restarting on edge change from "0" to "1" (reset edge memory marker for starting timer)	1**/2	2.1	2.7+
	Timer para.		2	–	–
R	T	Reset timer	1**/2	1.8	2.4+
	Timer para.		2	–	–

Status word for: **SP, SE, SD, SS, SF, FR, R**	BR	A1	A0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	–	–	–	–	–	–	–	Yes	–
Instruction controls:	–	–	–	–	–	0	–	–	0

* + time for loading the address of the instruction
** With direct instruction addressing

B.16 Counter Instructions

The count is in ACCU1-L and in the address passed as a parameter.

In-stru-ction	Address ID	Description	Length in Words	Typical Execution Time in µs	
				Direct Addres-sing	Indirect Ad-dressing*
S	C	Preset counter at edge change from ”0” to ”1”	1**/2	6.6	7.1+
	Counter p.		2	–	–
R	C	Reset counter to ”0” at edge change from ”0” to ”1”	1**/2	1.8	2.3+
	Counter p.		2	–	–
CU	C	Increment counter by 1 on edge change from ”0” to ”1”	1**/2	2.8	3.4+
	Counter p.		2	–	–
CD	C	Decrement counter by 1 on edge change from ”0” to ”1”	1**/2	3.0	3.5+
	Counter p.		2	–	–
FR	C	Enable counter at edge change from ”0” to ”1” (reset edge memory marker for up and down counting of a counter)	1**/2	2.2	2.7+
	Counter p.		2	–	–

Status word for: **S, R, CU, CD, FR**	BR	A1	A0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	–	–	–	–	–	–	–	Yes	–
Instruction controls:	–	–	–	–	–	0	–	–	0

* + time for loading the address of the instruction
** With direct instruction addressing

B.17 Load Instructions

Loading the instructions in ACCU1. The old contents of ACCU1 are saved to ACCU2 first.
The status word is not affected.

Instruction	Address ID	Description	Length in Words	Typical Execution Time in µs Direct Addressing	Typical Execution Time in µs Indirect Addressing*
L		Load ...			
	IB	Input byte	1**/2	0.6	1.7+
	QB	Output byte	1**/2	0.6	1.7+
	PIB	Peripheral input byte	1**/2	< 125	< 127
	MB	Memory byte	1**/2	0.8	1.8+
	LB	Local data byte	2	1.1	2.0+
	DBB	Data byte	2	2.8	2.6+
	DIB	Instance data byte	2	2.8	2.6+
		... in ACCU1			
	Parameters	Load IB/QB/PIB/MB/LB/DBB/DIB into ACCU1 (addressed via parameter)	2	–	+
		Load ...			
	IW	Input word	1**/2	0.9	1.9+
	QW	Output word	1**/2	0.9	1.9+
	PIW	Peripheral input byte	1**/2	< 135	< 137
	MW	Memory word	1**/2	1.1	2.1+
	LW	Local data word	2	1.3	2.3+
	DBW	Data word	2	3.3	3.2+
	DIW	Instance data word	2	3.3	3.2+
		... in ACCU1-L			
	Parameters	Load IW/QW/PIW/MW/LW/DBW/DIW into ACCU1 (addressed via parameter)	2	–	+
		Load ...			
	ID	Input double word	1**/2	1.1	2.1+
	QD	Output double word	1**/2	1.1	2.1+
	PID	Peripheral input double word	1**/2	< 145	< 147
	MD	Memory double word	1**/2	1.5	2.5+
	LD	Local data double word	2	1.6	2.7+
		Data double word	2	4.3	4.2+
	DBD	Instance data double word	2	4.3	4.2+
	DID	... in ACCU1			
	Parameters	Load ID/QD/PID/MD/LD/DBD/DID into ACCU1 (addressed via parameter)	2	–	+

* + time for loading the address of the instruction
** With direct instruction addressing

Instruction	Address ID	Description	Length in Words	Typical Execution Time in µs	
				Direct Addressing	Indirect Addressing*
L		Load (area-crossing addressing)			
	B[AR1,m]	Byte			40.1+
	B[AR2,m]		2	–	40.1+
	W[AR1,m]	Word			45.6+
	W[AR2,m]		2	–	45.6+
	D[AR1,m]	Double word			57.4+
	D[AR2,m]		2	–	57.4+
		... in ACCU1			
		Load ...			
	k8	8-bit constant in ACCU1-LL	1	0.6	–
	k16	16-bit constant in ACCU1-L	2	0.6	–
	L#k32	32-bit constant in ACCU1	3	0.8	–
	Parameters	Load constant in ACCU1 (addressed via parameter)	2	–	+
	2#n	Load 16-bit binary constant in ACCU1-L	2	0.6	–
		Load 32-bit binary constant in ACCU1	3	0.7	–
	16#p	Load 16-bit hexadecimal constant in ACCU1-L	2	0.6	–
		Load 32-bit hexadecimal constant in ACCU1	3	0.7	–
	'xx'	Load 2 characters	2	0.7	–
	'xxxx'	Load 4 characters	3	0.88	–
	D# date	Load IEC date (binary coded decimal)	2	0.8	–
	S5T# time value	Load S5 time constant (16 bits)	2	0.8	–
	TOD# time value	Load 32-bit time constant (IEC time of day)	3	0.88	–
	T# time value	Load 16-bit time constant	2	0.88	–
	C# count value	Load 16-bit counter constant	2	0.88	–
		Load 32-bit counter constant	3	0.88	–
	P# bit pointer	Load bit pointer	3	0.88	–
	L# integer	Load 32-bit integer constant	3	0.88	–
	q	Load real number	3	0.88	–

* + time for loading the address of the instruction

B.18 Load Instructions for Timers and Counters

Loading of a time value or count value into ACCU1. The contents of ACCU1 are saved to ACCU2 first.
The indicators are not affected.

Instruction	Address ID	Description	Length in Words	Typical Execution Time in µs	
				Direct Addressing	Indirect Addressing*
L	T	Load time value	1**/2	1.7	2.1+
	Timer para.	Load time value (addressed via parameter)	2	–	+
	C	Load count value	1**/2	1.5	2.1+
	Counter p.	Load time value (addressed via parameter)	2	–	+
LD	T	Load value of a timer (binary coded decimal)	1**/2	5.4	5.9+
	Timer para.	Load value of a timer (binary coded decimal, addressed via the parameter)	2	–	+
	C	Load count value (binary coded decimal)	1**/2	4.9	5.4+
	Counter p.	Load count value (addressed via parameter)	2	–	+

* + time for loading the address of the instruction
** With direct instruction addressing

B.19 Transfer Instructions

Transfer of the contents of ACCU1 to the addressed instruction. The status word is not affected. Remember that some transfer instructions depend on the MCR.

Instruction	Address ID	Description	Length in Words	Typical Execution Time in µs	
				Direct Addressing	Indirect Addressing*
T		Transfer contents of ACCU1–LL to ...			
	IB	Input byte	1**/2	0.3	1.2+
		(MCR-dependent)		1.4	1.6+
	QB	Output byte	1**/2	0.3	1.2+
		(MCR-dependent)		1.4	1.6+
	PQB	Peripheral output byte	1***/2	< 125	< 127
		(MCR-dependent)		<126	< 128
	MB	Memory byte	1**/2	0.4	1.3+
		(MCR-dependent)		1.5	1.7+
	LB	Local data byte	2	0.6	1.5+
		(MCR-dependent)	2	1.8	2.0+
	DBB	Data byte		2.5	2.3+
		(MCR-dependent)	2	3.0	2.8+
	DIB	Instance data byte		2.5	2.3+
		(MCR-dependent)		3.0	2.8+
	B[AR1,µ] B[AR2,m] Parameters	Transfer contents of ACCU1–LL to IB/QB/PQB/MB/LB/DBB/DIB (addressed via AR1, AR2 (area-crossing) or via parameter)	2	–	+

* + time for loading the address of the instruction
** With direct instruction addressing
*** Direct addressing of PQB 0 to 255

Instruction	Address ID	Description	Length in Words	Direct Addressing	Indirect Addressing*
T		Transfer contents of ACCU1–L to ...			
	IW	Input word	1**/2	0.5	1.5+
		(MCR-dependent)		1.5	1.8+
	QW	Output word	1**/2	0.5	1.5+
		(MCR-dependent)		1.5	1.8+
	PQW	Peripheral output byte	1***/2	< 135	< 137
		(MCR-dependent)		< 136	< 138
	MW	Memory word	1**/2	0.8	1.7+
		(MCR-dependent)		1.8	2.1+
	LW	Local data word	2	0.9	1.8+
		(MCR-dependent)		2.0	2.3+
	DBW	Data word	2	3.0	2.9+
		(MCR-dependent)		3.5	3.4+
	DIW	Instance data word	2	3.0	2.9+
		(MCR-dependent)		3.5	3.4+
	W[AR1,m] W[AR2,m] Parameters	Transfer contents of ACCU1-L to IW/QW/PQW/MW/LW/DBW/DIW (addressed via AR1, AR2 area-crossing or via parameter)	2	–	+
		Transfer contents of ACCU1 to ...			
	ID	Input double word	1**/2	0.8	1.7+
		(MCR-dependent)		1.8	2.1+
	QD	Output double word	1**/2	0.8	1.7+
		(MCR-dependent)		1.8	2.1+
	PQD	Peripheral output double word	2	< 150	< 152
		(MCR-dependent)		< 151	< 153
	MD	Memory double word	1**/2	1.3	2.2+
		(MCR-dependent)		2.3	2.6+
	LD	Local data double word	2	1.4	3.0+
		(MCR-dependent)		2.6	3.5+
	DBD	Data double word	2	4.1	4.0+
		(MCR-dependent)		4.6	4.5+
	DID	Instance data double word	2	4.1	4.0+
		(MCR-dependent)		4.6	4.5+
	D[AR1,m] D[AR2,m] Parameters	Transfer contents of ACCU1 to ID/QD/PQD/MD/LD/DBD/DID (addressed via AR1, AR2 (area-crossing) or via parameter)	2	–	+

* + time for loading the address of the instruction
** With direct instruction addressing
*** Direct addressing of PQW 0 to 254

B.20 Load and Transfer Instructions for Address Registers

Loading of a double word from a memory area or register into AR1 or AR2 or transfer of a double word from AR1 or AR2 to a memory area or register.
The status word is not affected.

In-struc-tion	Address ID	Description	Length in Words	Typical Execution Time in µs
LAR1		Load contents from ...		
	–	ACCU1	1	0.3
	AR2	Address register 2	1	0.3
	DBD	Data double word	2	3.8
	DID	Instance data double word	2	3.8
	m	32-bit constant as pointer	3	0.5
	LD	Local data double word	2	1.5
	MD	Memory double word	2	1.4
		... into AR1		
LAR2		Load contents from ...		
	–	ACCU1	1	0.3
	DBD	Data double word	2	3.8
	DID	Instance data double word	2	3.8
	m	32-bit constant as pointer	3	0.5
	LD	Local data double word	2	1.5
	MD	Memory double word	2	1.4
		... into AR2		
TAR1		Transfer contents of AR1 to ...		
	–	ACCU1	1	0.7
	AR2	Address register 2	1	0.3
	DBD	Data double word	2	4.3
	DID	Instance data double word	2	4.3
	LD	Local data double word	2	1.6
	MD	Memory double word	2	1.5
TAR2		Transfer contents of AR2 to ...		
	–	ACCU1	1	0.7
	DBD	Data double word	2	4.3
	DID	Instance data double word	2	4.3
	LD	Local data double word	2	1.6
	MD	Memory double word	2	1.5
CAR		Exchange the contents of AR1 and AR2	1	0.5

B.21 Load and Transfer Instructions for the Status Word

Instruction	Address ID	Description	Length in Words	Typical Execution Time in µs
L	STW	Load status word* into ACCU1		1.5

Status word for: **L STW**	BR	A1	A0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	Yes	Yes	Yes	Yes	Yes	0	0	Yes	0
Instruction controls:	–	–	–	–	–	–	–	–	–

T	STW	Transfer ACCU1 (bits 0 to 8) to the status word*		1.4

Status word for: **T STW**	BR	A1	A0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	–	–	–	–	–	–	–	–	–
Instruction controls:	Yes	Yes	Yes	Yes	Yes	0	0	Yes	0

* Structure of the Status Word

B.22 Load Instructions for DB Numbers and DB Length

Load the number/length of a data block into ACCU1. The contents of ACCU1 are saved to ACCU2 first.
The indicators are not affected.

Instruction	Address ID	Description	Length in Words	Typical Execution Time in µs
L	DBNO	Load number of data block	1	3.3
L	DINO	Load number of the instance data block	1	3.3
L	DBLG	Load length of data block into byte	1	0.6
L	DILG	Load length of instance data block into byte	1	0.6

B.23 Fixed-Point Math Instructions (16 Bits)

Math instructions of two 16-bit numbers. The result is in ACCU1–L.

In-struc-tion	Address ID	Description	Length in Words	Typical Execution Time in µs
+I	–	Add 2 integers (16 bits) (ACCU1-L)=(ACCU1-L)+ (ACCU2-L)	1	1.5
–I	–	Subtract 2 integers (16 bits) (ACCU1-L)=(ACCU2-L)– (ACCU1-L)	1	1.6
*I	–	Multiply 2 integers (16 bits) (ACCU1-L)=(ACCU2-L)* (ACCU1-L)	1	2.4
/I	–	Divide 2 integers (16 bits) (ACCU1-L)=(ACCU2-L): (ACCU1-L) The remainder of the division is in ACCU1-H.	1	3.4

Status word for: **+I, –I,*I, /I**	BR	A1	A0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	–	–	–	–	–	–	–	–	–
Instruction controls:	–	Yes	Yes	Yes	Yes	–	–	–	–

B.24 Fixed-Point Math Instructions (32 Bits)

Math instructions of two 32-bit numbers. The result is in ACCU1.

Instruction	Address ID	Description	Length in Words	Typical Execution Time in µs
+D	–	Add 2 integers (32 bits) (ACCU1)=(ACCU2)+(ACCU1)	1	2.0
–D	–	Subtract 2 integers (32 bits) (ACCU1)=(ACCU2)–(ACCU1)	1	2.7
*D	–	Multiply 2 integers (32 bits) (ACCU1)=(ACCU2)*(ACCU1)	1	9.9
/D	–	Divide 2 integers (32 bits) (ACCU1)=(ACCU2):(ACCU1)	1	10.8
MOD	–	Divide 2 integers (32 bits) and load the remainder of the division in ACCU1: (ACCU1)=Remainder from [(ACCU2):(ACCU1)]	1	11.3

Status word for: **+D, –D,*D, /D, MOD**	BR	A1	A0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	–	–	–	–	–	–	–	–	–
Instruction controls:	–	Yes	Yes	Yes	Yes	–	–	–	–

B.25 Floating-Point Math Instructions (32 Bits)

The result of the math instructions is in ACCU1. The execution time of the instruction depends on the value to be calculated.

Instruction	Address ID	Description	Length in Words	Typical Execution Time in µs
+R	–	Add 2 real numbers (32 bits) (ACCU1)=(ACCU2)+(ACCU1)	1	< 35
–R	–	Subtract 2 real numbers (32 bits) (ACCU1)=(ACCU2)–(ACCU1)	1	< 35
*R	–	Multiply 2 real numbers (32 bits) (ACCU1)=(ACCU2)*(ACCU1)	1	< 35
/R	–	Divide 2 real numbers (32 bits) (ACCU1)=(ACCU2):(ACCU1)	1	< 40

Status word for: **+R, –R,*R, /R**	BR	A1	A0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	–	–	–	–	–	–	–	–	–
Instruction controls:	–	Yes	Yes	Yes	Yes	–	–	–	–

NEGR	–	Negate real number in ACCU1	1	0.3
ABS	–	Make absolute value of the real number in ACCU1	1	0.3

Status word for: **NEGR, ABS**	BR	A1	A0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	–	–	–	–	–	–	–	–	–
Instruction controls:	–	–	–	–	–	–	–	–	–

B.26 Addition of Constants

Addition of integer constants to ACCU1. The indicators are not affected.

Instruction	Address ID	Description	Length in Words	Typical Execution Time in µs
+	i8	Add an 8-bit integer constant	1	0.3
+	i16	Add a 16-bit integer constant	2	0.3
+	i32	Add an 32-bit integer constant	3	0.6

B.27 Addition via the Address Register

Addition of an integer (16 bits) to the contents of the address register. The value is in the instruction or in ACCU 1-L. The indicators are not affected.

Instruction	Address ID	Description	Length in Words	Typical Execution Time in µs
+AR1		Add contents of ACCU1-L to AR1	1	0.3
+AR1	m	Add pointer constant to AR1	2	0.3
+AR2		Add contents of ACCU1-L to AR2	1	0.3
+AR2	m	Add pointer constant to AR2	2	0.3

B.28 Comparison Instructions with Integers (16 Bits)

Comparison of the integers (16 bits) in ACCU1-L and ACCU2-L. RLO=1 if the condition is satisfied.

Instruction	Address ID	Description	Length in Words	Typical Execution Time in µs
==I		ACCU2-L=ACCU1-L	1	1.4
<>I		ACCU2-L≠ACCU1-L	1	1.5
<I		ACCU2-L<ACCU1-L	1	1.5
<=I		ACCU2-L<=ACCU1-L	1	1.4
>I		ACCU2-L>ACCU1-L	1	1.5
>=I		ACCU2-L>=ACCU1-L	1	1.4

Status word for: **==I, <>I, <I, <=I, >I, >=I**	BR	A1	A0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	–	–	–	–	–	–	–	–	–
Instruction controls:	–	Yes	Yes	0	–	0	Yes	Yes	1

B.29 Comparison Instructions with Integers (32 Bits)

Comparison of the 32-bit integers in ACCU1 and ACCU2. RLO=1 if the condition is satisfied.

In-struc-tion	Address ID	Description	Length in Words	Typical Execution Time in µs
==D		ACCU2=ACCU1	1	2.0
<>D		ACCU2≠ACCU1	1	2.0
<D		ACCU2<ACCU1	1	2.0
<=D		ACCU2<=ACCU1	1	2.0
>D		ACCU2>ACCU1	1	2.0
>=D		ACCU2>=ACCU1	1	2.0

Status word for: **==D,< >D, <D, <=D, >D, >=D**	BR	A1	A0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	–	–	–	–	–	–	–	–	–
Instruction controls:	–	Yes	Yes	0	–	0	Yes	Yes	1

B.30 Comparison Instructions (32-Bit Real Numbers)

Comparison of the 32-bit real numbers in ACCU1 and ACCU2. RLO=1 if the condition is satisfied. The execution time of the instruction depends on the value to be compared.

In-struc-tion	Address ID	Description	Length in Words	Typical Execution Time in µs
==R		ACCU2=ACCU1	1	< 45
<>R		ACCU2 ≠ ACCU1	1	< 45
<R		ACCU2<ACCU1	1	< 45
<=R		ACCU2<=ACCU1	1	< 45
>R		ACCU2>ACCU1	1	< 45
>=R		ACCU2>=ACCU1	1	< 45

Status word for: **==R,< >R, <R, <=R, >R, >=R**	BR	A1	A0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	–	–	–	–	–	–	–	–	–
Instruction controls:	–	Yes	Yes	Yes	Yes	0	Yes	Yes	1

B.31 Shift Instructions

Shift the contents of ACCU1 or ACCU1-L by the specified number of places to the left/right. If an address ID is not specified, shift by the number specified in ACCU2-LL. Positions that become free are padded with zeros or a sign. The last bit shifted is in condition code bit CC 1.

In-struc-tion	Address ID	Description	Length in Words	Typical Execution Time in µs
SLW		Shift the contents of ACCU1–L to the left	1	2.0
SLW	0 to 15			0.7
SLD		Shift the contents of ACCU1 to the left	1	3.1
SLD	0 to 32			3.1
SRW		Shift the contents of ACCU1–L to the right	1	2.0
SRW	0 to 15			0.7
SRD		Shift the contents of ACCU1 to the right	1	3.1
SRD	0 to 32			3.1
SSI		Shift the contents of ACCU1–L with sign to the right	1	1.8
SSI	0 to 15			0.7
SSD		Shift the contents of ACCU1 with sign to the right	1	3.1
SSD	0 to 32			3.2

Status word for: **SLW, SLD, SRW, SRD, SSI, SSD**	BR	A1	A0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	–	–	–	–	–	–	–	–	–
Instruction controls:	–	Yes	Yes	Yes	–	–	–	–	–

B.32 Rotate Instructions

Rotation of the contents of ACCU1 to the left or right by the specified number of places. If an address ID is not specified, rotate by the number specified in ACCU2-LL.

In-struc-tion	Address ID	Description	Length in Words	Typical Execution Time in µs
RLD		Rotate the contents of ACCU1 to the left	1	3.3
RLD	0 to 32			3.4
RRD		Rotate the contents of ACCU1 to the right	1	3.5
RRD	0 to 32			3.5

Status word for: **RLD, RRD**	BR	A1	A0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	–	–	–	–	–	–	–	–	–
Instruction controls:	–	Yes	Yes	Yes	–	–	–	–	–

RLDA		Rotate the contents of ACCU1 one bit position to the left through condition code bit CC 1		1.9
RRDA		Rotate the contents of ACCU1 one bit position to the right through condition code bit CC 1		1.9

Status word for: **RLDA, RRDA**	BR	A1	A0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	–	–	–	–	–	–	–	–	–
Instruction controls:	–	Yes	0	0	–	–	–	–	–

B.33 ACCU Transfer Instructions, Incrementing, Decrementing

Status Word

The status word is not affected.

Instruction	Address ID	Description	Length in Words	Typical Execution Time in µs
CAW		Reverse the order of the bytes in ACCU1–L LL, LH becomes LH, LL.	1	0.3
CAD		Reverse the order of the bytes in ACCU1 LL, LH, HL, HH becomes HH, HL, LH, LL.	1	0.6
TAK		Swap the contents of ACCU1 and ACCU2	1	0.8
PUSH		The contents of ACCU1 are transferred to ACCU2	1	0.3
POP		The contents of ACCU2 are transferred to ACCU1	1	0.3
INC	0 to 255	Increment ACCU1-LL	1	0.3
DEC	0 to 255	Decrement ACCU1-LL	1	0.3

B.34 Program Display and Null Operation Instructions

Status Word

The status word is not affected.

Instruction	Address ID	Description	Length in Words	Typical Execution Time in µs
BLD	0 to 255	Program display instruction; Is treated by the CPU like a null operation instruction. The indicators are not affected.	1	0.3
NOP	0 1	Null operation instruction: The indicators are not affected.	1	0.3 0.3

B.35 Data Type Conversion Instructions

Result

The results of the conversion are in ACCU1.

The execution time for the conversion of real numbers depends on the value.

In-struc-tion	Address ID	Description	Length in Words	Typical Execution Time in µs
BTI	–	Convert contents of ACCU1 from BCD to integer (16 bits) (**B**CD **T**o **I**nt)	1	4.7
BTD	–	Convert contents of ACCU1 from BCD to double integer (32 bits) (**B**CD **T**o **D**oubleint)	1	11.5
DTR	–	Convert contents of ACCU1 from double integer (32 bits) to real (32 bits) (**D**oubleint **T**o **R**eal)	1	< 15
ITD	–	Convert contents of ACCU1 from integer (16 bits) to double integer (32 bits) (**I**nt **T**o **D**oubleint)	1	0.1

Status word for: **BTI, BTD, DTR, ITD**	BR	A1	A0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	–	–	–	–	–	–	–	–	–
Instruction controls:	–	–	–	–	–	–	–	–	–

ITB	–	Convert contents of ACCU1 from integer (16 bits) to BCD (**I**nt **T**o **B**CD)	1	5.1
DTB	–	Convert contents of ACCU1 from double integer (32 bits) to BCD (**D**oubleint **T**o **B**CD)	1	11.8

Status word for: **ITB, DTB**	BR	A1	A0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	–	–	–	–	–	–	–	–	–
Instruction controls:	–	–	–	Yes	Yes	–	–	–	–

RND	–	Convert real number to 32-bit double integer.	1	< 20
RND–	–	Convert real number to 32-bit double integer. Rounded up to next integer.	1	< 20
RND+	–	Convert real number to 32-bit double integer. Rounded up to next integer.	1	< 20
TRUNC	–	Convert real number to 32-bit double integer. Digits after the decimal point are truncated.	1	< 20

Status word for: **RND, RND–, RND+, TRUNC**	BR	A1	A0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	–	–	–	–	–	–	–	–	–
Instruction controls:	–	–	–	Yes	Yes	–	–	–	–

B.36 Complement Formation

In-struc-tion	Address ID	Description	Length in Words	Typical Execution Time in σs
INVI		Create ones complement of ACCU1-L	1	0.3
INVD		Create ones complement of ACCU1	1	0.3

Status word for: **INVI, INVD**	BR	A1	A0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	–	–	–	–	–	–	–	–	–
Instruction controls:	–	–	–	–	–	–	–	–	–

NEGI		Create twos complement of ACCU1-L (16-bit integer)	1	1.5
NEGD		Create twos complement of ACCU1 (32-bit integer)	1	2.0

Status word for: **NEGI, NEGD, NEGR**	BR	A1	A0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	–	–	–	–	–	–	–	–	–
Instruction controls:	–	Yes	Yes	Yes	Yes	–	–	–	–

B.37 Block Call Instructions

Instruction	Address ID	Description	Length in Words	Typical Execution Time in µs	
				Direct Addressing	Indirect Addressing*
CALL	FC	Unconditional call of a function	1**/2	7.7	-
	SFC	Unconditional call of a function of the operating system	2	See Appendix C for execution times	
UC	FC	Unconditional call of blocks without parameters	1**/2	5.3	6.1+

Status word for: **CALL, UC**	BR	A1	A0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	–	–	–	–	–	–	–	–	–
Instruction controls:	–	–	–	–	0	0	1	–	0

CC	FC	Conditional call of blocks without parameters	1**/2	5.3	6.1+

Status word for: **CC**	BR	A1	A0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	–	–	–	–	–	–	–	Yes	–
Instruction controls:	–	–	–	–	0	0	1	–	0

OPN	DB	Open a data block	2	1.5	2.6+
	DB para.	Open a data block (addressed via parameter)	2	–	–
	DI	Open an instance data block	2	1.5	2.6+

Status word for: **OPN**	BR	A1	A0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	–	–	–	–	–	–	–	–	–
Instruction controls:	–	–	–	–	–	–	–	–	–

* + time for loading the address of the instruction
** With direct instruction addressing

B.38 Block End Instructions

In-struc-tion	Address ID	Description	Length in Words	Typical Execution Time in µs
BE		End block	1	2.8
BEU		End block unconditionally	1	–

Status word for: **BE, BEU**	BR	A1	A0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	–	–	–	–	–	–	–	–	–
Instruction controls:	–	–	–	–	0	0	1	–	0

BEC		End block conditionally if RLO=”1”		3.2

Status word for: **BEC**	BR	A1	A0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	–	–	–	–	–	–	–	Yes	–
Instruction controls:	–	–	–	–	Yes	0	1	1	0

B.39 Exchanging Data Blocks

Exchange of the two current data blocks. The current data block becomes the current instance data block and vice versa. The indicators are not affected.

In-struc-tion	Address ID	Description	Length in Words	Typical Execution Time in µs
CDB		Exchanging Data Blocks	1	0.4

B.40 Jump Instructions

Jumping as a function of conditions. The jump width in an 8-bit address ID is between (–128 to +127). The jump width in 16-bit address IDs is between (–32768 to –129) or (+128 to +32767).

Note on JC

Please note in the case of CPU 614 programs that the jump destination always forms the **beginning** of a Boolean logic string. The jump destination must not be included in the logic string.

In-struc-tion	Address ID	Description	Length in Words	Typical Execution Time in µs
JU	LABEL	Jump unconditionally	1*/2	1.8

Status word for: **JU**	BR	A1	A0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	–	–	–	–	–	–	–	–	–
Instruction controls:	–	–	–	–	–	–	–	–	–

JC	LABEL	Jump if RLO=”1”	1/2	1.5
JCN	LABEL	Jump if RLO=”0”	2	1.6

Status word for: **JC, JCN**	BR	A1	A0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	–	–	–	–	–	–	–	Yes	–
Instruction controls:	–	–	–	–	–	0	1	1	0

JCB	LABEL	Jump if RLO=”1” Save the RLO in the BR bit	2	1.8
JNB	LABEL	Jump if RLO=”0” Save the RLO in the BR bit	2	1.8

Status word for: **JCB, JNB**	BR	A1	A0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	–	–	–	–	–	–	–	Yes	–
Instruction controls:	Yes	–	–	–	–	0	1	1	0

JBI	LABEL	Jump if BR=”1”	2	1.5
JNBI	LABEL	Jump if BR=”0”	2	1.5

Status word for: **JBI, JNBI**	BR	A1	A0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	Yes	–	–	–	–	–	–	–	–
Instruction controls:	–	–	–	–	–	0	1	–	0

* 1 word long for jump widths of between –128 and +127

Instruction	Address ID	Description	Length in Words	Typical Execution Time in µs
JO	LABEL	Jump on stored overflow (OV="1")	1*/2	1.5

Status word for: **JO**	BR	A1	A0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	–	–	–	Yes	–	–	–	–	–
Instruction controls:	–	–	–	–	–	–	–	–	–

JOS	LABEL	Jump on stored overflow (OS="1")	2	1.6

Status word for: **JOS**	BR	A1	A0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	–	–	–	–	Yes	–	–	–	–
Instruction controls:	–	–	–	–	0	–	–	–	–

JUO	LABEL	Jump if "instruction unordered" (CC 1=1 and CC 0=1)	2	1.8
JZ	LABEL	Jump if result=0 (CC 1=0 and CC 0=0)	1*/2	1.7
JP	LABEL	Jump if result>0 (CC 1=1 and CC 0=0)	1*/2	1.8
JM	LABEL	Jump if result<0 (CC 1=0 and CC 0=1)	1*/2	1.8
JN	LABEL	Jump if result ≠ 0 (CC 1=1 and CC 0=0) or (CC 1=0) and (CC 0=1)	1*/2	1.8
JMZ	LABEL	Jump if result ≤ 0 (CC 1=0 and CC 0=1) or (CC 1=0 and CC 0=0)	2	1.5
JPZ	LABEL	Jump if result ≥ 0 (CC 1=1 and CC 0=0) or (CC 1=0) and (CC 0=0)	2	1.6

Status word for: **JUO, JZ, JP, JM, JN, JMZ, JPZ**	BR	A1	A0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	–	Yes	Yes	–	–	–	–	–	–
Instruction controls:	–	–	–	–	–	–	–	–	–

JL	LABEL	Jump distributor This instruction is followed by a list of jump instructions. The address ID is a jump label to the instruction following the list. ACCU1-L contains the number of the jump instruction to be executed	2	2.7
LOOP	LABEL	Decrement ACCU1-L and jump to ACCU1-L ≠ 0 (loop programming)	2	1.6

Status word for: **JL, LOOP**	BR	A1	A0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	–	–	–	–	–	–	–	–	–
Instruction controls:	–	–	–	–	–	–	–	–	–

* 1 word long for jump widths of between –128 and +127

B.41 Instructions for the Master Control Relay (MCR)

MCR writes the value ”0” or leaves the memory contents unchanged.

MCR=0→MCR is deactivated
MCR=1→MCR is activated; ”T” instruction writes zeros to the corresponding address IDs; ”S”/”R” instructions do not change the memory contents.

In-struc-tion	Address ID	Description	Length in Words	Typical Execution Time in µs
MCR(		Open a MCR zone. Save the RLO to the MCR stack.	1	1.7

Status word for: **MCR(**	BR	A1	A0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	–	–	–	–	–	–	–	Yes	–
Instruction controls:	–	–	–	–	–	0	1	–	0

)MCR		Close an MCR zone. Remove an entry from the MCR stack.	1	1.6

Status word for: **)MCR**	BR	A1	A0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	–	–	–	–	–	–	–	–	–
Instruction controls:	–	–	–	–	–	0	1	–	0

MCRA		Activate the MCR	1	0.3
MCRD		Deactivate the MCR	1	0.3

Status word for: **MCRA, MCRD**	BR	A1	A0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	–	–	–	–	–	–	–	–	–
Instruction controls:	–	–	–	–	–	–	–	–	–

Execution Times of the SFCs and SFBs

C

The CPU component of the IM 151/CPU provides you with different system functions and system function blocks for program scanning and diagnostics, for example. You can call these system functions/system function blocks in your user program using the number of the SFC.

For a detailed description of all system functions/system function blocks, refer to the reference manual *STEP 7 System and Standard Functions*. This includes how to call the system functions and which parameters you have to enter.

C.1 System Functions (SFCs)

SFC No.	Name	Description	Execution Time in µs
0	SET_CLK	Sets the clock time	295
1	READ_CLK	Reads the clock time	193
2	SET_RTM	Sets the operating hours counter	62
3	CTRL_RTM	Starts/stops the operating hours counter	57
4	READ_RTM	Reads the operating hours counter	86
5	GADR_LGC	Reads the free address of the channel *x* of the signal module on module slot *y*.	148
6	RD_SINFO	Read the start information of the current OB.	116 +3 per byte
7	DP_PRAL	Triggers a process interrupt from the user program of the CPU as DP slave through to DP master.	228
14	DPRD_DAT	Reads consistent user data	141 +3 per byte
15	DPWR_DAT	Writes consistent user data	143
17	ALARM_SQ	Creates acknowledgeable, block-specific messages	252
18	ALARM_S	Creates non-acknowledgeable, block-specific messages	248
19	ALARM_SC	Determines the acknowledgment status of the last ALARM_SQ arrived message	115
20	BLKMOV	Copies variables of any type **Note:** You can only use SFC20 for process-related DBs, i.e. the ”Unlinked” check box in the object properties is not selected. Otherwise, you will receive the W#16#8092 error code (=block is not process-related) as the RET_VAL.	84 +1.6 per byte

SFC No.	Name	Description	Execution Time in µs
21	FILL	Presets a field	84 +2.5 per byte
22	CREATE_DB	Creates a data block	126 +5 per byte
23	DEL_DB	Deletes a data block	102
24	TEST_DB	Tests a data block	110
28	SET_TINT	Sets a time-of-day interrupt	212
29	CAN_TINT	Cancels a time-of-day interrupt	64
30	ACT_TINT	Activates a time-of-day interrupt	68
31	QRY_TINT	Queries a time-of-day interrupt	79
32	SRT_DINT	Starts a delay interrupt	89
33	CAN_DINT	Cancels a delay interrupt	78
34	QRY_DINT	Queries a delay interrupt	85
36	MSK_FLT	Masks sync faults	118
37	DMSK_FLT	Enables sync faults	131
38	READ_ERR	Reads and erases programming and access errors that have occurred or have been disabled	128
39	DIS_IRT	Disables the handling of new interrupts	284
40	EN_IRT	Enables the handling of new interrupts	702
41	DIS_AIRT	Delays the handling of interrupts	48
42	EN_AIRT	Enables the handling of interrupts	76
43	RE_TRIGR	Re-triggers cycle time monitoring	95
44	REPL_VAL	Copies a replacement value into accumulator 1 of the level causing the error	51
46	STP	Forces the CPU into STOP mode	No value
47	WAIT	Implements waiting times	861
49	LGC_GADR	Converts a free address to the slot and rack for a module	281
50	RD_LGADR	Reads all the declared free addresses for a module	252
51	RDSYSST	Reads out the information from the system state list The SFC 51 cannot be interrupted by interrupts.	280 +8 per byte
52	WR_ USMSG	Writes specific diagnostic information in the diagnostic buffer	964
54	RD_DPARM	Reads predefined dynamic parameters of a module	2390
55	WR_PARM	Writes dynamic parameters for a module (currently not used because all the module parameters are static)	2500
56	WR_DPARM	Writes predefined dynamic parameters for a module (currently not used because all the module parameters are static)	2520
57	PARM_MOD	Assigns a module's parameters	2520

SFC No.	Name	Description	Execution Time in µs
58	WR_REC	Writes a module-specific data record (currently not used because there isn't a module to which user records can be written)	2200
59	RD_REC	Reads a module-specific data record (currently only reads diagnostic records 0 and 1)	352
64	TIME_TICK	Displays the system time with an accuracy of 10 ms	50
81	UBLKMOV	Copies the variable without interruption	140

C.2 System Function Blocks (SFBs)

SFB No.	Name	Description	Execution Time in µs
IEC Counters			
0	CTU	Counts up. The counter is incremented by 1 for each rising edge.	70
1	CTD	Counts down. The counter is decremented by 1 for each rising edge.	69
2	CTUD	Counts up and count down.	85
IEC Timers			
3	TP	Generates a pulse of duration PT.	90
4	TON	Delays a rising edge of duration PT.	91
5	TOF	Delays of falling edge of duration PT.	91
Implementation of a Sequence Processor			
32	DRUM	Implements a sequence processor with a maximum of 16 sequences.	302

Migration of the IM 151/CPU

D

In this chapter, you will find out the most important differences of two selected CPUs in the S7-300 SIMATIC family.

We will also show you how to rewrite programs you have written for the S7-300-CPUs for the IM 151/CPU.

Chapter Overview

In Section	Contents	Page
D.1	Differences to Selected S7-300 CPUs	D-2
D.2	Porting the User Program	D-3

More Information

You can find further information on how to create and structure programs in the *STEP 7* manuals and online help system.

D.1 Differences to Selected S7-300 CPUs

The following table lists the most important programming differences between two selected CPUs of the S7-300 SIMATIC family and the IM 151/CPU.

Table D-1 Differences to Selected S7-300 CPUs

Features	CPU 314	CPU 315-2 DP	IM 151/CPU
Real-time clock	Hardware	Hardware	Software
Memory backup	Yes, with battery	Yes, with battery	Not possible
Memory card	Memory card	Memory card	MMC
Number of connections to programming device and operator panel	4 (as of 10/99: 12)	4 (as of 10/99: 12)	5
Setting the PROFIBUS address	Hardware configuration	Hardware configuration	Hardware configuration must match address setter
Transmission rate to programming device and operator panel	187.5 kbps (MPI)	187.5 kbps (MPI) 12 Mbps (DP)	12 Mbps (DP)
Interface	Active	Active	Passive
Communication:			
Programming device/operator panel	Yes	Yes	Yes
Global data communication	Yes	Yes	No
S7 basic communication	Yes	Yes	Yes (server)
S7 communication	Yes (server)	Yes (server)	Yes (server)
Direct communication	No	Yes	Yes
Range of uses with DP	Stand-alone	As a DP master As a DP slave Stand-alone	As DP slave Stand-alone
Addressing	Slot-oriented	Free	Free
Interrupt Response Time	1-1.5 ms	1-1.5 ms	Less than 20 ms
Removal/insertion of modules during operation	No	No	Yes

D.2 Porting the User Program

Introduction

By porting we mean making available on a distributed basis a program that was previously used centrally on a master. Certain adjustments may be necessary to relocate an existing program partially or completely from a master to an intelligent slave. The resources required for porting sections of a user program to an intelligent slave depend on how the address assignment of inputs and outputs is stored in the FBs in the source program.

The inputs and outputs can be used in the FCs in the source program in different ways. Addresses can be packed in the current ET 200S, which is not possible in the IM 151/CPU.

See the description of IM 151/CPU addressing in Section 2.1.

Porting with Unpacked Addresses

If you use FBs with unpacked I/O addresses, the required program sections can be transferred easily to the IM 151/CPU without the need for porting.

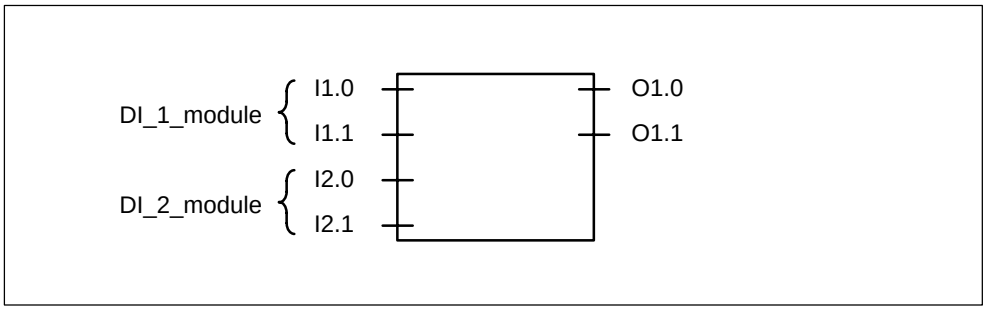


Figure D-1 Example: FB with Unpacked Addresses

Porting with Packed Addresses

If FBs with packed I/O addresses are copied to the IM 151/CPU, the packed addresses there can no longer be assigned to the I/Os of the I/O modules locally because the CPU of the IM 151/CPU cannot work with packed addresses. This requires rewiring of the corresponding FBs. Rewiring corresponds to ”unpacking” the addresses.

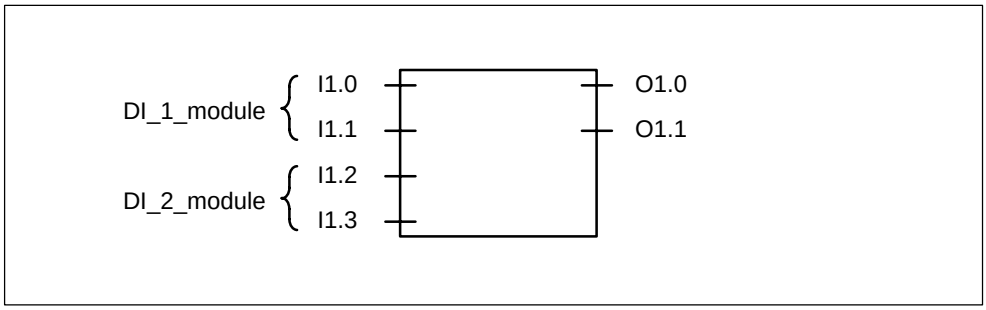


Figure D-2 Example: FB with Packed Addresses

Rewiring

The following blocks and address IDs can be reassigned:

- Inputs, outputs
- Memory markers, timers, counters
- Functions, function blocks

Proceed as follows to rewire the signals:

1. In SIMATIC Manager, select the “Blocks” folder, which contains the blocks with the packed addresses that you want to port to the IM 151/CPU.
2. Choose **Options→ Rewire**.
3. Enter the desired replacements in the displayed “Rewiring” dialog box in the table (old address ID/new address ID).

Table D-2 Example: Replacements under Options→ Rewire

	Old Address ID	New Address ID
1	I 1.2	I 2.0
2	I 1.3	I 2.1

4. Click OK.

This starts rewiring. After rewiring, you can decide in a dialog box whether you want to look at the rewiring information file. The file contains the list of old and new address IDs. The various blocks are also listed together with the number of rewirings carried out in the block.

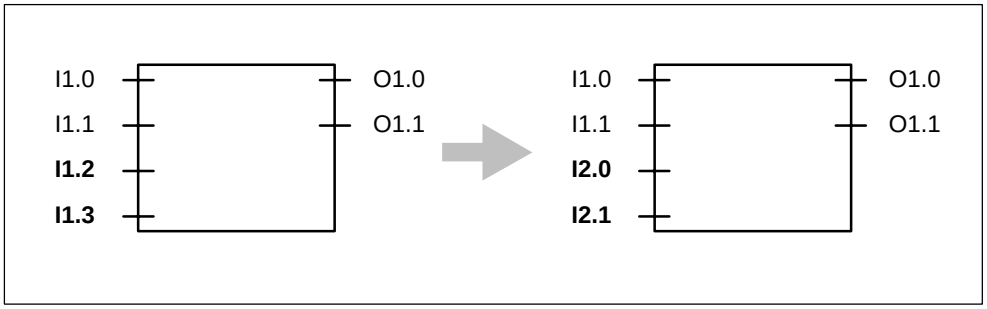


Figure D-3 Example: Rewiring of the Signals

If you assign symbols to the inputs and outputs using the symbol table in *STEP 7*, you must change the symbol table to adjust the subprogram for use in the IM 151/CPU.

See also the *STEP 7* online help system.

Porting FBs with I/Os in an I/O Word

If you map the addresses of the inputs and outputs via an I/O word to a function block you have programmed, porting is considerably more involved.

It is possible to program a shell around the FB that makes an adjustment so that the FB can be used with the IM 151/CPU. The other option is to reprogram the FB. We recommend you reprogram the FB because this is easier than programming a shell.

See also the *STEP 7* online help system.

Glossary

Accumulator

The accumulators are registers in the → CPU and are used as intermediate memory for loading, transfer, comparison, calculation and conversion operations.

Address

An address is the designation for a certain address ID or address area (e.g. input I 12.1; memory word MW 25; data block DB 3).

Aggregate Current

The sum of the currents of all the output channels of a digital output module.

Automation System

An automation system is a programmable logic controller consisting of at least one CPU, various input and output modules and operator interfaces.

Backup Memory

The backup memory is for memory areas of the→ CPU without a backup battery. A configurable number of timers, counters, memory markers and data bytes (retentive timers, counters, memories and data bytes) is backed up.

Bus

A common transmission path connecting all nodes and having two defined ends. In the case of the ET 200S, the bus is a two-wire or fiber-optic cable.

Chassis Ground

Chassis ground refers to all the interconnected inactive parts of a piece of equipment that, even in the event of a fault, cannot carry voltage that is dangerous to the touch.

Compress

The programming device online function “Compress” is used to align all valid blocks contiguously in the RAM of the CPU at the start of the user memory. This eliminates all gaps which arose when blocks were deleted or modified.

Consistent Data

Data that belongs together in terms of its content and should not separated is known as consistent data.

For example, the values of analog modules must always be handled consistently. In other words, the value of an analog module must not be corrupted by reading it out at two different times.

Counter

Counters are part of the → system memory of the CPU. The content of the “counter cells” can by modified by *STEP 7* instructions (e.g. count up/down).

CPU

Central processing unit of the S7 programmable controller with a control unit and arithmetic logic unit, memory, operating system and interface for a programming device.

Cycle Time

The cycle time is the time taken by the → CPU to scan the → user program once.

Data Block

Data blocks (DB) are data areas in the user program which contain user data. Global data blocks can be accessed by all code blocks, whereas instance data blocks are assigned to a specific FB call.

Device Database File

All the DP slave-specific properties are stored in a device database file (DDB file). The format of the device database file is defined in EN 50170, Volume 2, PROFIBUS.

Diagnostic Buffer

The diagnostic buffer is a buffered memory area in the CPU in which diagnostic events are stored in the order of their occurrence.

Diagnostic Interrupt

Diagnostics-capable modules use diagnostic interrupts to report system errors which they have detected to the central CPU.

In SIMATIC S7/M7: When an error is detected or disappears (e. g. wire break), the ET 200S triggers a diagnostic interrupt, provided the interrupt is enabled. The CPU of the DP master interrupts the processing of the user program and lower-priority classes and processes the diagnostic interrupt block (OB 82).

In SIMATIC S5: The diagnostic interrupt appears in the station diagnosis. Using cyclical querying of the diagnostic bits in the station diagnosis you can detect errors such as a wire break.

Diagnostics

Diagnostics is the detection, localization, classification, display and further evaluation of errors, faults and messages.

Diagnostics includes monitoring functions that run automatically while the system is in operation. This increases the availability of systems by reducing setup times and downtimes.

Direct Communication

Direct communication is a special communication relationship between PROFIBUS-DP nodes. Direct communication is characterized by the fact that the PROFIBUS-DP nodes ”listen in” to find out which data a DP slave is sending back to its DP master.

Direct Data Communication

See Direct Communication

Distributed I/O Systems

These are input/output units that are not located in the base unit; instead, they are distributed at some distance from the CPU. The following are examples of such units:

- ET 200S, ET 200M, ET 200B, ET 200C, ET 200U, ET 200X, ET 200L
- DP/AS-I Link
- S5-95U with a PROFIBUS-DP slave interface
- Other DP slaves from either Siemens or other vendors

The distributed I/O systems are connected to the DP master via PROFIBUS-DP.

DP master

A→ Master that complies with EN 50170, Volume 2, PROFIBUS, is referred to as a DP master.

DP Slave

A→ slave on the PROFIBUS bus system with the PROFIBUS-DP protocol that complies with EN 50170, Volume 2, PROFIBUS standard is referred to as a DP slave.

DP Standard

The DP standard is the bus protocol of the ET 200 distributed I/O system based on EN 50170, Volume 2, PROFIBUS.

Error Display

The error display is one of the possible responses of the operating system to a runtime error. The other possible responses are: → error response in the user program, STOP mode of the CPU.

Error Handling via OB

If the operating system detects a specific error (e.g. an access error in the case of *STEP 7*), it calls the organization block (error OB) provided for this event that specifies the subsequent behavior of the CPU.

Error Response

Response to a → runtime error. The operating system can respond in the following ways: conversion of the programmable controller to STOP mode, call of an organization block in which the user can program a response, or display of the error.

ET 200

The ET 200 distributed I/O system with the PROFIBUS-DP protocol is a bus for connecting distributed I/O devices to a CPU or an appropriate DP master. The ET 200 is notable for its fast response times, since only a small amount of data (bytes) is transferred.

The ET 200 is based on IEC 61158/EN 50170, Volume 2, PROFIBUS.

The ET 200 works on the master/slave principle. Examples of DP masters are the IM 308-C master interface module and the CPU 315-2 DP.

DP slaves can be the distributed I/O devices ET 200S, ET 200B, ET 200C, ET 200M, ET 200X, ET 200U, ET 200L or DP slaves from Siemens or other vendors.

FC → Function

Force

During commissioning, for example, the “Force” function allows certain outputs to be set to “ON” for any length of time, even if the logic operations of the user program are not fulfilled (e.g. because inputs are not wired).

FREEZE

This is a control command of the DP master to a group of DP slaves.

When a DP slave receives the FREEZE control command, it freezes the current status of the **inputs** and transfers them cyclically to the DP master.

After each FREEZE control command, the DP slave freezes the status of the **inputs** again.

The input data is not transferred from the DP slave to the DP master again cyclically until the DP master sends the UNFREEZE control command.

Function

A function (FC) in accordance with IEC 1131-3 is a code block without static data. A function allows parameters to be passed in the user program. Functions are therefore suitable for programming frequently repeated complex functions (e.g. calculations).

Intelligent DP Slave

The defining feature of an intelligent DP slave is that input/output data is not made available to the DP master by a real input/output of the DP slave directly, but by a preprocessing CPU (in this case the IM 151/CPU interface module).

Interrupt

The operating system of the CPU recognizes 10 different priority classes that control the processing of the user program. These runtime levels include interrupts, e.g. diagnostic interrupts. When an interrupt is triggered, the operating system automatically calls an assigned organization block in which the user can program the desired response (for example in an FB).

Interrupt, Diagnostic → Diagnostic Interrupt

Interrupt, Process → Process Interrupt

Load Memory

The load memory is part of the CPU. It contains objects generated by the programming device. It is implemented either as a plug-in memory card or a permanently integrated memory.

Master

When they are in possession of the token, masters can send data to and request data from other nodes (= active node). An example of a DP master is the CPU 315-2 DP.

Master System

All the DP slaves that are assigned to a DP master for reading and writing make up a master system with the DP master.

Memory Marker

Memory markers are part of the→ system memory of the CPU for storing interim results. They can be accessed in units of a bit, byte, word or doubleword.

MMC

Micro Memory Card Memory module for SIMATIC systems. Can be used as a load memory and portable data carrier.

MPI

The multipoint interface (MPI) is the SIMATIC S7 interface for programming devices.

Nesting depth

One block can be called from another by means of a block call. The nesting depth is the number of code blocks called at the same time.

Node

A device that can send, receive or amplify data via the bus, such as a DP master, DP slave, RS 485 repeater or active star coupler.

OB → Organization Block

OB Priority

The operating system of the CPU distinguishes between various priority classes, such as cyclic program scanning and process interrupt-driven program scanning. Each priority class is assigned → organization blocks (OB) in which the S7 user can program a response. The OBs have different standard priorities which determine the order in which they are executed or interrupted in the event that they are activated simultaneously.

Operating Mode

The SIMATIC S7 programmable controllers have the following operating modes: STOP, →START-UP, RUN.

Operating system of the CPU

The operating system of the CPU organizes all the functions and processes of the CPU which are not associated with a special control task.

Organization block

Organization blocks (OBs) represent the interface between the operating system of the CPU and the user program. The processing sequence of the user program is defined in the organization blocks.

Parameter

1st variable of a *STEP 7* code block

2nd variable to set the behavior of a module (one or more per module). Each module is delivered with a suitable default setting, which can be changed by configuring the parameters in *STEP 7*.

PG → Programming Device

PLC → Programmable Controller

Priority Class

The operating system of an S7-CPU provides up to 26 priority classes (or “runtime levels”) to which various organization blocks are assigned. The priority classes determine which OBs interrupt other OBs. If a priority class includes several OBs, they do not interrupt each other, but are executed sequentially.

Process Image

The process image is part of the → system memory of the CPU. The signal states of the inputs are written into the process input image at the start of the cyclic program. At the end of the cyclic program, the signal states in the process output image are transferred to the outputs.

PROFIBUS

Process **Fi**eld **Bus** is a German process and field bus standard, defined in the IEC 61158/EN 50170, Volume 2, PROFIBUS standard. It defines the functional, electrical and mechanical features of a bit-serial field bus system.

PROFIBUS is available with the protocols DP (the German abbreviation for distributed I/O), FMS (= field bus message specification), PA (= process automation) or TF (= technology (process-related) functions).

PROFIBUS Address

Each bus node must receive a PROFIBUS address to identify it uniquely on the PROFIBUS bus system.

The PC/programming device has the PROFIBUS address ”0”.

The PROFIBUS addresses 1 to 125 are permissible for the ET 200S distributed I/O system.

Programmable Controller

Programmable controllers (PLCs) are electronic controllers whose function is saved as a program in the control unit. The configuration and wiring of the unit are therefore independent of the function of the control system. The programmable logic controller has the structure of a computer; it consists of the → CPU (central processing unit) with memory, input and output modules and an internal bus system. The I/Os and the programming language are designed for the purpose of control engineering.

Programming Device

Programming devices are essentially personal computers which are compact, portable and suitable for industrial applications. They are equipped with special hardware and software for SIMATIC programmable controllers.

Process Interrupt

A process interrupt is triggered by interrupt-triggering modules on the occurrence of a specific event in the process. The process interrupt is reported to the CPU. The assigned → organization block is then processed in accordance with the priority of this interrupt.

Publisher

A sender in direct data communication *See* Direct Communication

Restart

When a CPU is started up (e.g. by switching the mode selector from STOP to RUN or by switching the power on), the organization block OB 100 (complete restart) is executed before cyclic program scanning (OB 1) commences. On a complete restart, the process image input table is read in and the *STEP 7* user program is processed starting with the first command in OB 1.

Runtime Error

Error which occurs during processing of the user program on the programmable controller (i.e. not in the process).

Scan Cycle Checkpoint

The point during CPU program scanning at which the process image is updated, for example.

SFC → System Function

Slave

A slave can only exchange data with a → master when requested by it to do so. By slaves we mean, for example, all DP slaves such as ET 200S, ET 200B, ET 200X, ET 200M, etc.

Stand-Alone Operation

The device is operated on a stand-alone basis without data exchange to a superordinate master and without direct communication with other DP slaves. All the modules power up using default parameters and with the maximum configuration (32 slots, 64 bytes consistently).

Start Event

Start events are defined events such as faults, time stamps or interrupts. They cause the operating system to start a corresponding organization block (if programmed to do so by the user). The start events are displayed in the header information of the corresponding OB. The user can respond to start events in the user program.

Startup

RESTART mode is activated during the transition from STOP mode to RUN mode.

Can be triggered by the mode selector or after power on or an operator action on the programming device. In the case of the ET 200S a restart is carried out.

STEP 7

Programming language for developing user programs for SIMATIC S7 PLCs.

Subscriber

A recipient in direct communication *See* Direct Communication

SYNC

This is a control command of the DP master to a group of DP slaves.

By means of the SYNC control command, the DP master causes the DP slave to freeze the current statuses of the **outputs**. In the subsequent frames, the DP slave stores the output data, but the statuses of the outputs remain unchanged.

After each new SYNC control command, the DP slave sets the outputs that it has saved as output data. The outputs are not cyclically updated again until the DP master sends the UNSYNC control command.

System Diagnostics

System diagnostics is the term used to describe the detection, evaluation and indication of errors which occur within the programmable controller. Examples of such errors are program errors or module failures. System errors can be indicated by means of LEDs or in *STEP 7*.

System Function

A system function (SFC) is a function integrated in the operating system of the CPU that can be called up in the *STEP 7* user program, if necessary.

System Memory

The system memory is integrated on the central processing unit and implemented as RAM. The system memory includes the address areas (for example timers, counters, memory markers, etc.) as well as the data areas (e.g. communication buffers) required internally by the operating system.

Timers (timer cells)

Timers are part of the → system memory of the CPU. The contents of the “timer cells” are updated automatically by the operating system asynchronously to the user program. *STEP 7* instructions are used to define the exact function of the timer cells (e.g. on-delay) and initiate their execution (e.g. start).

Token

Access right on bus

Transmission Rate

The transmission rate of a data transfer is measured in bits transmitted per second (transmission rate = bit rate).

In the case of the ET 200S, transmission rates of 9.6 kbps to 12 Mbps are possible.

User Memory

The user memory contains the code and data blocks of the user program. The user memory can be integrated in the CPU (IM 151/CPU) or can be provided on plug-in memory cards or memory modules. The user program is always executed in the → working memory of the CPU, however.

User Program

The SIMATIC system distinguishes between the → operating system of the CPU and user programs. The latter are created with the programming software → *STEP 7* in the possible programming languages (ladder logic and statement list) and saved in code blocks. Data is stored in data blocks.

Working Memory

The working memory is random-access memory in the → CPU that is accessed by the processor during program execution.

Index

Symbols

), B-15
)MCR, B-52
+, B-39
+AR1, B-40
+AR2, B-40
+D, B-38
+I, B-37
+R, B-39
–D, B-38
–I, B-37
–R, B-39
*D, B-38
*I, B-37
*R, B-39
/D, B-38
/I, B-37
/R, B-39
=, B-26
==D, B-41
==I, B-40
==R, B-42
<=D, B-41
<=I, B-40
<=R, B-42
<> R, B-42
<>D, B-41
<>I, B-40
<D, B-41
<I, B-40
<R, B-42
>=D, B-41
>=R, B-42
>>=I, B-40
>>D, B-41
>>I, B-40
>>R, B-42

A

A, B-13, B-17, B-19
A(, B-15
Access, to the ET 200S from a PG/PC, 3-3
Accumulator, B-3, Glossary-1
AD, B-18
Address, Glossary-1
Address area
    data consistency, 2-7
    default setting, 2-10
    for user data transfer, 2-7
    of the expansion modules, 2-3
Address assignment, for analog and digital
    modules, 2-3
Address identifiers, B-2
Address registers, B-4
Addresses
    base address, 2-5
    for diagnostics, 2-7
    for user data transfer, 2-7
Addressing, 2-1
    address allocation, 2-5
    indirect, B-7
    interface in STEP 7, 2-9
    rules, 2-9
    slot-oriented, 2-2
    user-defined, 2-5
Addressing examples, STEP 7
    instructions, B-5
Aggregate current, Glossary-1
AN, B-13, B-15, B-17, B-20
Area of application, 1-2
Automation system, Glossary-1
AW, B-18

B

Backup, memory, Glossary-1
BE, B-49
BEC, B-49
BEU, B-49
BLD, B-45
Blocks, of the BM 147/CPU, 5-13
BM 147/CPU
    mode selector, 5-4
    parameters, 5-16
BTD, B-46
BTI, B-46
Bus, Glossary-1

Bus connector, 3-7
Bus faults, remedy, 3-9

C

Cables, 3-7
CAD, B-45
CALL, B-48
CAR, B-35
CAW, B-45
CC, B-48
CD, B-29
CDB, B-49
Changing the PROFIBUS address, 3-6
Chassis ground, Glossary-1
Clock, 5-12, 5-16
Clock memory, 5-16
CLR, B-27
Commissioning, 4-1, 4-7, 8-1, 8-2, 8-6, 8-11, 8-17
Communication, direct, Glossary-3
Components, ET 200S, 1-5
Compression, Glossary-2
Configuration, ET 200S stand-alone, 3-4
Configuration data, accepting, 5-17
Configuration frame, A-1
  configuration, normal ID format, A-4
  default setting, A-5
  structure, A-4
    special identifier format, A-2
Configuration software, 1-4
Configuring, 8-1, 8-7, 8-12
  IM 151/CPU, 4-2
Consistency, **2-7**, 2-8
Consistent data, Glossary-2
Counter, Glossary-2
CPU, Glossary-2
  operating system, Glossary-8
CU, B-29
Cycle, OB 1, 5-14
Cycle behavior, 5-17
Cycle extension, through interrupts, 7-5
Cycle time, 7-2, Glossary-2
  extending, 7-3
  structure, 7-2
Cyclic interrupts, 5-17

D

Data, consistent, Glossary-2
Data block, Glossary-2
Data consistency, 2-7, 2-8
Data exchange, direct, 3-10
Data interchange, sample program, 2-10
Data transfer
  principle behind, 2-1
  with the DP master, 2-6
DBs, 5-13
DDB (device database) file, **5-2**
DEC, B-45
Default addressing, 2-2
Default setting, A-5
  for the address area, 2-10
Delay, of the inputs/outputs, 7-5
Device database file, Glossary-2
Diagnosis
  module, 4-20
  station, 4-14, 4-22
Diagnostic address, 2-7, 4-11, 4-13
Diagnostic buffer, Glossary-3
  entry, 4-11
  reading out, 4-5
Diagnostic interrupt, 4-23, Glossary-3
  OB 82, 5-14
Diagnostic interrupt response time, 7-9
Diagnostics, 4-1, Glossary-3
  system, Glossary-11
  using LEDs, 4-9
Direct communication, 3-10, Glossary-3
Direct data communication, Glossary-3
Display, LED, 4-9
Distributed I/O device, Glossary-4
DP diagnostic address, 2-7
DP master, Glossary-4
DP slave, Glossary-4
  intelligent, 4-2, Glossary-6
DP slave diagnosis, structure, 4-15
DP standard, Glossary-4
DTB, B-46
DTR, B-46

E

Error display, Glossary-4
Error during direct access to I/O, OB 122, 5-15
Error handling via OB, Glossary-4
Error response, Glossary-5
Error/fault, direct access to I/O, OB 122, 5-15
Error/fault responses, 5-15
ET 200, Glossary-5

ET 200S
components, 1-5
manuals, 1-5
ET 200S manuals, guide to, 1-5
Event identification, in the DP master/DP slave, 4-12
Execution time, user program, 7-2

F

FBs, 5-13
FCs, 5-13
Features, 6-2
of the IM 151/CPU, 1-4, 5-2
FN, B-25
Force, 5-5, Glossary-5
FP, B-25
FR, B-28, B-29
FRCE, LED, 5-5
FREEZE, Glossary-5
Function, FC, Glossary-6
Functions, via PG, 3-9

G

General technical specifications, 6-2

I

Identifier, configuration frame, special identifier format, A-3
IM 147/CPU, blocks, 5-13
IM 151/CPU
configuring, 4-2
features, 1-4
important features, 5-2
memory reset, 4-4
IM 151/CPU FO interface module, 6-3, 6-4
basic circuit diagram, 6-4
terminal assignment, 6-3
IM 151/CPU interface module, 6-4
basic circuit diagram, 6-4
technical specifications, 6-5
INC, B-45
Inputs, delay time, 7-5
Installation, 8-1, 8-4
Integrated clock, 5-12
Intelligent DP slave, 4-2, Glossary-6
Intermediate memory
accessing in the user program, 2-8
in the IM 151/CPU, 2-6
Internode communication, see direct communication, 3-10
Interrupt, 4-14, Glossary-6
process, Glossary-9
Interrupts, cycle extension, 7-5
INVD, B-47
INVI, B-47
ITB, B-46
ITD, B-46

J

JBI, B-50
JC, B-50
JCB, B-50
JCN, B-50
JL, B-51
JM, B-51
JMZ, B-51
JN, B-51
JNB, B-50
JNBI, B-50
JO, B-51
JOS, B-51
JP, B-51
JPZ, B-51
JU, B-50
JUO, B-51
JZ, B-51

L

L, B-30, B-31, B-32, B-36
LAR1, B-35
LAR2, B-35
LD, B-32
LED, 4-5
display, 4-9
FRCE, 5-5
indicator, 1-4
ON, 5-5
RUN, 5-5
SF, 5-5
STOP, 5-5
Load instruction, 2-8
Load memory, Glossary-6
LOOP, B-51

M

Manual contents, brief overview, 1-5

Manuals, guide to, 1-5
Manufacturer ID, CPU 31x-2 as DP slave, 4-19
Master, Glossary-6
Master PROFIBUS address, 4-18
Master system, Glossary-6
MCR(, B-52
MCRA, B-52
MCRD, B-52
Memory
  backup, Glossary-1
  load, Glossary-6
  system, Glossary-11
  user, Glossary-12
  working, Glossary-12
Memory marker, Glossary-7
Memory module, 5-8
Memory reset
  internal CPU events, 4-5
  resetting the memory of the IM 151/CPU, 4-4
  with mode selector, 5-4
Micro Memory Card, Glossary-7
Micro memory card, 5-8
MMC, Glossary-7
MMC module, 5-8
MOD, B-38
Mode
  RUN, 5-5
  STOP, 5-5
Mode selector, 5-4
  memory reset, 4-4
  MRES, 5-4
  RUN-P, 5-4
  STOP, 5-4
Module diagnosis, 4-20
Module status, 4-21
MPI, 3-2, Glossary-7
MPI node address, 5-17
MRES, mode selector, 5-4

N

NEGD, B-47
NEGI, B-47
Nesting depth, Glossary-7
Network, structure, 3-1
Network components, 3-7
Node, Glossary-7
NOP, B-45
NOT, B-27

O

O, B-13, B-16, B-21
O(, B-15
OB, Glossary-8
  start event, Glossary-10
OB 1, 5-14
OB 100, 5-14
OB 122, 4-12, 5-15
OB 40, 4-14, 5-14
OB 82, 4-8, 4-12, 5-14
OB 86, 4-8, 4-12, 5-15
OB priority, Glossary-7
OB10, 5-14
OB121, 5-15
OB20, 5-14
OB35, 5-14
OB80, 5-15
OB83, 5-14
OB85, 5-15
OBs
  of the CPU, 5-13
  start events, 5-14
OD, B-18
ON, B-13, B-17, B-22
  LED, 5-5
ON(, B-15
Online functions, for the ET 200S, 3-9
Operating mode, Glossary-7
Operating mode changes, 4-12
Operating system
  of the CPU, Glossary-8
  processing time, 7-3
OPN, B-48
Order number
  IM 151/CPU, 6-2
  memory module, 5-8
  network components, 3-7
Organization block, Glossary-8
Outputs, delay time, 7-5
OW, B-18

P

Parameter, Glossary-8
Parameter assignment, with the DP master, A-1
Parameter assignment frame, A-1
  configuration, 5-18
  structure, A-6

Parameter ranges, STEP 7 instructions, B-2
Parameters, BM 147/CPU, 5-16
PC
  connection to ET 200S, 3-4
  prerequisites, 3-2
PG, Glossary-9
  connection to ET 200X, 3-4
PLC, Glossary-9
Pointers, calculating, B-6
POP, B-45
Priority, OB, Glossary-7
Priority class, Glossary-8
Process image, Glossary-8
Process image updating, processing time, 7-3
Process interrupt, 4-14, 4-23, Glossary-9
  OB 40, 5-14
Process interrupt handling, 7-9
Process interrupt response time, 7-9
Processing time
  operating system, 7-3
  process image updating, 7-3
  user program, 7-4
Product overview, 1-1
PROFIBUS, Glossary-9
PROFIBUS address, 3-3, Glossary-9
PROFIBUS network
  network components, 3-7
  structural principles, 3-2
PROFIBUS-DP, specifications, 5-2
Programming, 8-1, 8-2, 8-9, 8-15
Programming device
  connecting cable, 3-3
  prerequisites, 3-2
Programming device connecting cable, 3-7
Programming device functions, 3-9
Programming software, 1-4
Publisher, Glossary-9
PUSH, B-45

R

R, B-26, B-28, B-29
Response time, 7-6
  diagnostic interrupt, 7-9
  longest, 7-8
  process interrupt, 7-9
  shortest, 7-7
Restart, Glossary-10
  OB 100, 5-14
Retentivity, 5-16
Rewiring, D-4
RLD, B-44
RLDA, B-44
RND, B-46
RND+, B-46
RND–, B-46
RRD, B-44
RRDA, B-44
Rules, for addressing, 2-9
RUN
  LED, 5-5
  mode, 5-5
RUN-P, mode selector, 5-4
Runtime error, Glossary-10

S

S, B-26, B-29
S7 timers, updating, 7-4
SAVE, B-27
Scan cycle checkpoint, Glossary-10
SD, B-28
SE, B-28
SET, B-27
SF, B-28
  LED, 5-5
SFBs, 5-13
SFC, C-1
SFC DP_PRAL, 4-14
SFC DPRD_DAT, 2-8
SFC DPWR_DAT, 2-8
SFCs, 5-13
  execution times, C-1
Slave, Glossary-10
Slave diagnosis, 4-14
SLD, B-43
Slot assignment, ET-200S, 2-2
Slot-oriented addressing, 2-2
SLW, B-43
Software clock, 5-12, 5-16
SP, B-28
SRD, B-43
SRW, B-43
SS, B-28
SSD, B-43
SSI, B-43
Stand-alone operation, Glossary-10
  of ET 200S, 3-4
  of the ET 200S, 3-9
Start event, OB, Glossary-10
Start events, for OBs, 5-14
Startup, Glossary-11
Startup characteristics, 5-16
Station diagnosis, 4-14, 4-22

Station failure, OB 86, 5-15
Station status 1 to 3, 4-16
Status word, B-4
STEP 7, Glossary-11
  addressing interface, 2-9
  configuring the IM 151/CPU, 4-2
  settings, 3-9
STOP
  LED, 5-5
  mode, 5-5
  mode selector, 5-4
Structure, of the PROFIBUS network, 3-2
Subscriber, Glossary-11
SYNC, Glossary-11
System diagnostics, 5-16, Glossary-11
System function, SFC, Glossary-11
System functions, C-1
System memory, Glossary-11

T

T, B-33, B-34, B-36
TAK, B-45
TAR1, B-35
TAR2, B-35
Technical specifications
  general, 6-2
  of the IM 151/CPU, 6-2
  PROFIBUS-DP, 5-2
Test functions, 3-9
Test run, 8-1, 8-2, 8-10, 8-17
Time-of-day interrupts, 5-16
Timers (timer cells), Glossary-12
Token, Glossary-12
Transfer instruction, 2-8
Transmission rate, Glossary-12
Troubleshooting, 4-1
TRUNC, B-46

U

UC, B-48
Updating, of the S7 timers, 7-4
Upgrading, 8-1, 8-11
User data transfer, to the DP master, 2-6
User memory, Glossary-12
User program, Glossary-12
  processing time, 7-4
User program execution time, 7-2
User-defined addressing, 2-5

W

Wiring, 8-1, 8-5
Working memory, Glossary-12

X

X, B-14, B-17, B-23
X(, B-15
XN, B-14, B-17, B-24
XN(, B-15
XOD, B-18
XOW, B-18

Siemens AG
A&D AS E82
Postfach 1963
D–92209 Amberg
Federal Republic of Germany

From:

Your Name: . . . . . .

Your Title: . . . . . .

Company Name: . . . . . .

Street: . . . . . .

City, Zip Code: . . . . . .

Country: . . . . . .

Phone: . . . . . .

Please check any industry that applies to you:

- ❒ Automotive
- ❒ Chemical
- ❒ Electrical Machinery
- ❒ Food
- ❒ Instrument and Control
- ❒ Nonelectrical Machinery
- ❒ Petrochemical
- ❒ Pharmaceutical
- ❒ Plastic
- ❒ Pulp and Paper
- ❒ Textiles
- ❒ Transportation
- ❒ Other . . . . . .

Remarks Form

Your comments and recommendations will help us to improve the quality and usefulness of our publications. Please take the first available opportunity to fill out this questionnaire and return it to Siemens.

Please give each of the following questions your own personal mark within the range from 1 (very good) to 5 (poor).

1. Do the contents meet your requirements? ☐
2. Is the information you need easy to find? ☐
3. Is the text easy to understand? ☐
4. Does the level of technical detail meet your requirements? ☐
5. Please rate the quality of the graphics/tables: ☐

Additional comments:

. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .

Siemens AG
A&D AS E82
Postfach 1963
D–92209 Amberg
Federal Republic of Germany

From:

Your Name: . . . . . .

Your Title: . . . . . .

Company Name: . . . . . .

Street: . . . . . .

City, Zip Code: . . . . . .

Country: . . . . . .

Phone: . . . . . .

Please check any industry that applies to you:

- ❒ Automotive
- ❒ Chemical
- ❒ Electrical Machinery
- ❒ Food
- ❒ Instrument and Control
- ❒ Nonelectrical Machinery
- ❒ Petrochemical
- ❒ Pharmaceutical
- ❒ Plastic
- ❒ Pulp and Paper
- ❒ Textiles
- ❒ Transportation
- ❒ Other . . . . . .

Remarks Form

Your comments and recommendations will help us to improve the quality and usefulness of our publications. Please take the first available opportunity to fill out this questionnaire and return it to Siemens.

Please give each of the following questions your own personal mark within the range from 1 (very good) to 5 (poor).

1. Do the contents meet your requirements? ☐
2. Is the information you need easy to find? ☐
3. Is the text easy to understand? ☐
4. Does the level of technical detail meet your requirements? ☐
5. Please rate the quality of the graphics/tables: ☐

Additional comments:

. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .