

NT4S/NT15S/NT18S

Programmable Terminal

OPERATION MANUAL



NT4S/NT15S/NT18S

Programmable Terminal

Operation Manual

Produced March 1999

OMRON Product References

All OMRON products are capitalized in this manual. The word 'Unit' is also capitalized when it refers to an OMRON product, regardless of whether or not it appears in the proper name of the product.

The abbreviation 'Ch,' which appears in some displays and on some OMRON products, often means 'word' and is abbreviated 'Wd' in documentation in this sense.

The abbreviation 'PC' means Programmable Controller and is not used as an abbreviation for anything else.

The abbreviation 'Host' means a controller such as an FA computer which controls a PT (programmable terminal).

Visual Aids

The following headings appear in the left column of the manual to help you locate different types of information.

Note Indicates information of particular interest for efficient and convenient operation of the product.

1, 2, 3... 1. Indicates lists of one sort or another, such as procedures, checklists, etc.

© OMRON, 1999

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, mechanical, electronic, photocopying, recording, or otherwise, without the prior written permission of OMRON.

No patent liability is assumed with respect to the use of the information contained herein. Moreover, because OMRON is constantly striving to improve its high-quality products, the information contained in this manual is subject to change without notice. Every precaution has been taken in the preparation of this manual. Nevertheless, OMRON assumes no responsibility for errors or omissions. Neither is any liability assumed for damages resulting from the use of the information contained in this publication.

TABLE OF CONTENTS

PRECAUTIONS	xi
1 Intended Audience	xii
2 General Precautions	xii
3 Safety Precautions	xii
SECTION 1	
The Terminals of the NT-Series	1
1.1 Front View	4
1.2 Keyboard.....	8
1.3 Rear View.....	16
1.4 Mounting the Terminals.....	21
1.5 Pin Assignments	31
1.6 Shield	36
1.7 Display.....	37
1.8 User-Mode Switch	41
1.9 Battery.....	42
1.10 Fuse.....	43
1.11 Application Memory.....	43
SECTION 2	
Technical Data	45
2.1 NT4S-SF121B-E	47
2.2 NT4S-SF122B-E	48
2.3 NT4S-SF123B-E	50
2.4 NT15S-SF121B-E.....	51
2.5 NT18S-SF121B-E.....	53
SECTION 3	
Operating Modes	55
3.1 Setting the Operating Mode	57
SECTION 4	
Standard Mode	59
4.1 Setting the Operating Mode	63
4.2 Startup Process	64
4.3 Communication in the Standard Mode	65
4.4 Operating Concept.....	65
4.5 Masks	70
4.6 Variables	72
4.7 Graphics	124
4.8 Recipes.....	126
4.9 Message System	136
4.10 Help System.....	155
4.11 Function Keys	156
4.12 System Parameters.....	159
4.13 Version Number	161
4.14 Running Time Meter.....	161

4.15	Parallel Outputs	163
4.16	Screen Saver.....	163
4.17	Image of the Mask Number	164
4.18	Image of the Mode Selector Switch	164
4.19	Terminal Clock	164
4.20	Read Coordination Byte	166
4.21	Write Coordination Byte.....	168
4.22	Cyclic Poll Area	169
4.23	Control Codes	173
4.24	Cyclic Variables	175
4.25	Interface Parameters.....	175
4.26	Variable Definition	175
4.27	Application Programming.....	176
4.28	Downloading the User Description	180
4.29	Simulation without the Controller	184

SECTION 5

Transparent Mode 185

5.1	Setting the Operating Mode	187
5.2	Start-up Processes	187
5.3	Communication in the Transparent Mode.....	188
5.4	Function Setup Menu	189
5.5	Display.....	191
5.6	Keys	192
5.7	Interface Control Characters	193
5.8	Error Messages	197

SECTION 6

Linking to OMRON Controllers..... 199

6.1	OMRON Host-Link	201
6.2	OMRON NT-Link.....	205

INDEX..... 213

Revision History..... 217

About this Manual:

This manual describes the basic functions and operation procedures of the NT-series programmable terminal NT4S/NT15S/NT18S, its operations when connected to a PC or a Host, and includes the sections described below.

Please read this manual carefully and be sure you understand the information provided before attempting to install and operate the NT-series programmable terminal NT4S/NT15S/NT18S.

Section 1 describes the physical size of the terminals and the electrical connections.

Section 2 describes the electrical and mechanical specifications of the terminals.

Section 3 describes the two operation modes, the standard mode and transparent mode and how to set these.

Section 4 describes the standard mode in detail.

Section 5 describes the transparent mode in detail.

Section 6 describes how to connect to OMRON PLC using hostlink or NT-Link.

Related Manuals and Their Contents:

The related manuals are listed below.

The □ symbol at the end of the manual number is the revision history number.

[Operating the programmable terminal and communicating with the host]

- NT4S/NT15S/NT18S Programmable Terminal Operation Manual
(V056-E1-□) This manual

This operation manual is the manual for the NT4S/NT15S/NT18S itself.

This operation manual describes the functions and handling of both the program-
mable terminal body and the host interface function.

[Connecting the NT4S/NT15S/NT18S to PLC's other than Omron.]

- NT4S/NT15S/NT18S Programmable terminal, multi-vendor connections
(V058-E1-□)

The NT4S/NT15S/NT18S can also be connected to other PLC's then Omron only.
This manual describes how to connect to other PLC's.

PRECAUTIONS

This section provides general precautions for using the Programmable Terminal.

The information contained in this section is important for the safe and reliable application of the Programmable Terminal. You must read this section and understand the information contained before attempting to set up or operate a Programmable Terminal.

1	Intended Audience	xii
2	General Precautions	xii
3	Safety Precautions	xii

1 Intended Audience

This manual is intended for the following personnel, who must also have knowledge of electrical systems (an electrical engineer or the equivalent).

- Personnel in charge of introducing FA systems into production facilities.
- Personnel in charge of designing FA systems.
- Personnel in charge of installing and connecting FA systems.
- Personnel in charge of managing FA systems and facilities.

2 General Precautions

The user must operate the product according to the performance specifications described in the operation manuals.

Before using the product under conditions which are not described in the manual or applying the product to nuclear control systems, railroad systems, aviation systems, vehicles, combustion systems, medical equipment, amusement machines, safety equipment, and other systems, machines and equipment that may have a serious influence on lives and property if used improperly, consult your OMRON representative.

Make sure that the ratings and performance characteristics of the product are sufficient for the systems, machines, and equipment, and be sure to provide the systems, machines, and equipment with double safety mechanisms.

This manual provides information for using the Programmable Terminal. Be sure to read this manual before attempting to use the software and keep this manual close at hand for reference during operation.



WARNING

It is extremely important that Programmable Terminals and related devices be used for the specified purpose and under the specified conditions, especially in applications that can directly or indirectly affect human life. You must consult with your OMRON representative before applying Programmable Terminals to the above-mentioned applications.



WARNING

Do not use input functions such as PT keys for applications where danger to human life or serious damage is possible, or for emergency switch applications.

3 Safety Precautions

Safety Conventions and their Meanings

Read these safety precautions carefully and make sure you understand them before using the Programmable Terminal so that you can use it safely and correctly.

This operation manual uses the following conventions and symbols to indicate cautions, warnings, and dangers in order to ensure safe use of the NT4S/NT15S/NT18S.

The cautions, warnings, and dangers shown here contain important information related to safety. This instructions in these cautions, warnings, and dangers must be observed.

Precautions

The conventions used and their meanings are presented below.



Indicates information that, if not heeded, could possibly result in loss of life or serious injury.



Indicates information that, if not heeded, could result in relatively serious or minor injury, damage to the product, or faulty operation.

Explanation of Symbols

This manual uses the following symbols to indicate notes and hazardous situations.



Notes for the User



General Danger



Specific Danger

SECTION 1

The Terminals of the NT-Series

This section describes the terminal hardware. This includes all the drawings, the description of the function of the keys, the electrical connections and the dip-switch settings. It also describes how to mount the terminal. Furthermore it describes how to handle the battery.

1.1	Front View	4
1.1.1	NT4S-SF121B-E	4
1.1.2	NT4S-SF122B-E	5
1.1.3	NT4S-SF123B-E	6
1.1.4	NT15S-SF121B-E	7
1.1.5	NT18S-SF121B-E	8
1.2	Keyboard	8
1.2.1	Editing Keys	9
1.2.2	Control Keys	10
1.2.3	Special Keys	10
1.2.4	Function Keys	11
1.2.4.1	Function Key Arrangement	11
1.2.4.1.1	NT4S-SF121B-E	11
1.2.4.1.2	NT4S-SF122B-E	12
1.2.4.1.3	NT4S-SF123B-E	12
1.2.4.1.4	NT15S-SF121B-E	13
1.2.4.1.5	NT18S-SF121B-E	13
1.2.5	Slide-in Identification Strips for the Function Keys	14
1.2.5.1	NT4S-SF121B-E	14
1.2.5.2	NT4S-SF122B-E	14
1.2.5.3	NT4S-SF123B-E	15
1.2.5.4	NT15S-SF121B-E	15
1.2.5.5	NT18S-SF121B-E	15
1.3	Rear View	16
1.3.1	NT4S-SF121B-E	16
1.3.2	NT4S-SF122B-E	17
1.3.3	NT4S-SF123B-E	18
1.3.4	NT15S-SF121B-E	19
1.3.5	NT18S-SF121B-E	20
1.4	Mounting the Terminals	21
1.4.1	Front Panel Dimensions	22
1.4.1.1	NT4S-SF121B-E	22
1.4.1.2	NT4S-SF122B-E	22
1.4.1.3	NT4S-SF123B-E	23
1.4.1.4	NT15S-SF121B-E	23
1.4.1.5	NT18S-SF121B-E	24
1.4.2	Side View, Mounting Depth	24
1.4.2.1	NT4S-SF121B-E	24
1.4.2.2	NT4S-SF122B-E	25
1.4.2.3	NT4S-SF123B-E	26
1.4.2.4	NT15S-SF121B-E	27
1.4.2.5	NT18S-SF121B-E	28

1.4.3	Panel Cutout	29
1.4.3.1	NT4S-SF121B-E	29
1.4.3.2	NT4S-SF122B-E	29
1.4.3.3	NT4S-SF123B-E	30
1.4.3.4	NT15S-SF121B-E	30
1.4.3.5	NT18S-SF121B-E	31
1.5	Pin Assignments	31
1.5.1	Pin Assignment X1 Supply Voltage	32
1.5.2	Pin Assignment X3 SER1 TTY / 20 mA Current Loop	33
1.5.3	Pin Assignment X3 SER1 RS485	34
1.5.4	Pin Assignment X3 SER1 RS232c	35
1.5.5	Pin Assignment X3 SER2 RS232c	35
1.5.6	Pin Assignment X4 Parallel Outputs	36
1.6	Shield	36
1.7	Display	37
1.7.1	NT4S-SF121B-E, NT4S-SF122B-E, NT4S-SF123B-E	37
1.7.2	NT15S-SF121B-E	37
1.7.3	NT18S-SF121B-E	38
1.7.4	Contrast Setting	38
1.7.5	Default Contrast Setting	38
1.7.6	Character Attributes	39
1.7.6.1	Font Normal	39
1.7.6.2	Font Zoom	39
1.7.6.3	ASCII Character Set Table	40
1.8	User-Mode Switch	41
1.9	Battery	42
1.10	Fuse	43
1.11	Application Memory	43

1 The Terminals of the NT-Series

This manual is only valid for the following operating terminals:

- NT4S-SF121B-E
- NT4S-SF122B-E
- NT4S-SF123B-E
- NT15S-SF121B-E
- NT18S-SF121B-E

The terminals of the NT-series are the favourite Man-Machine-Interfaces for all applications, where an optimum relation of size to function is needed.

All terminals of the NT-series permit sealed installation in accordance with the IP65 degree of protection.

Each terminal is equipped with function keys, that can be individually labelled with slide-in strips.

A built-in lithium battery buffers the data in the RAM and also supplies the real time clock with power.

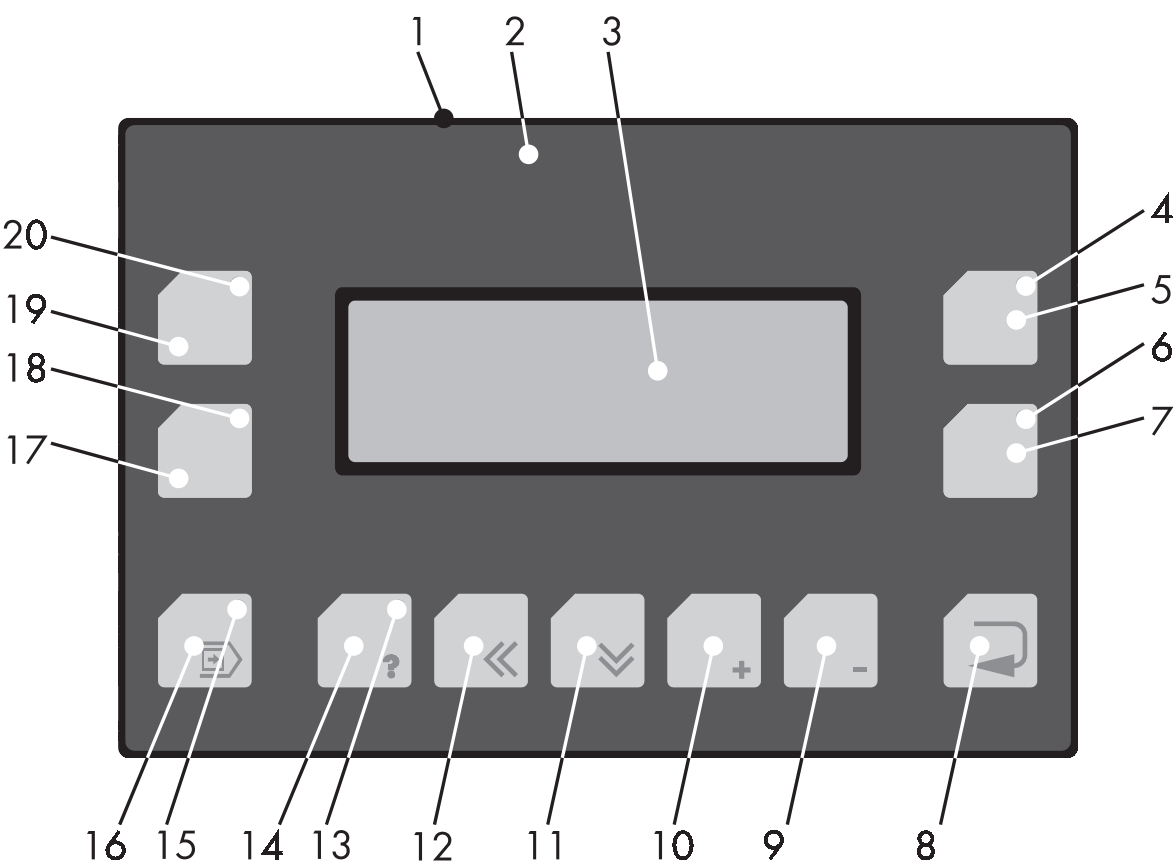
The discharge state of the battery is monitored constantly by the system.

The communication with a controller and logging printer is supported by a universal interface, unique for all terminals of the NT-series.

The operator guidance is done by means of a comfortable programming software.

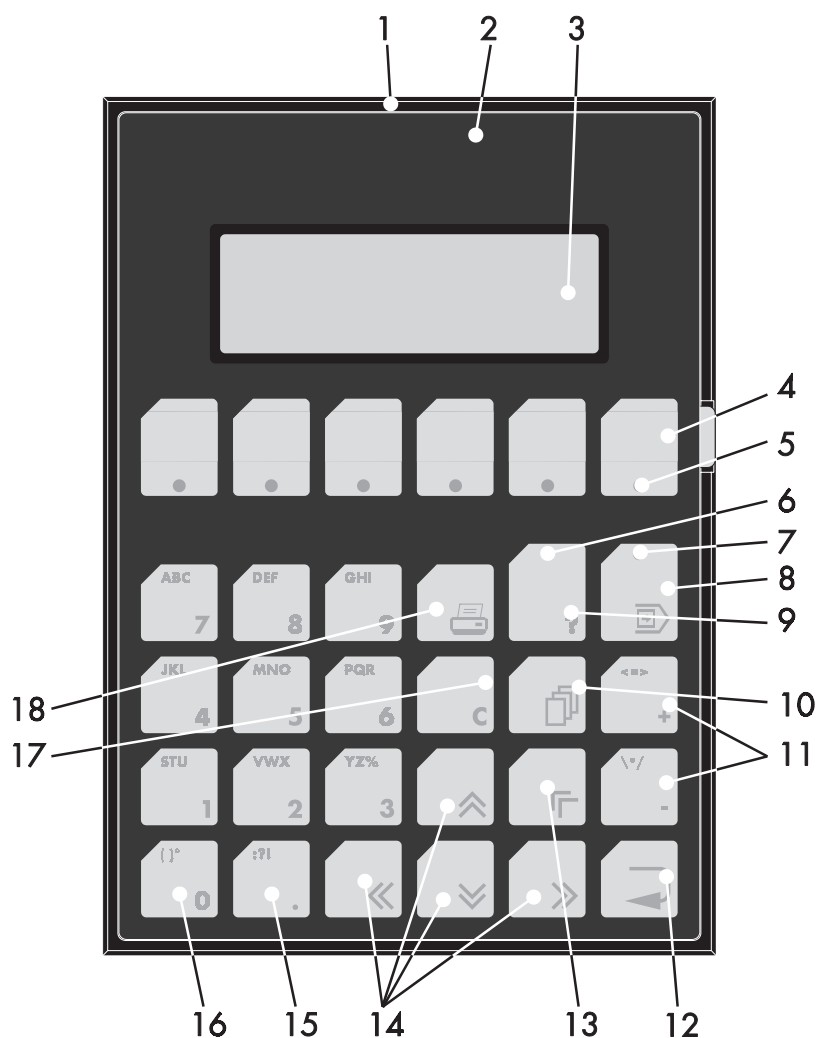
1.1 Front View

1.1.1 NT4S-SF121B-E



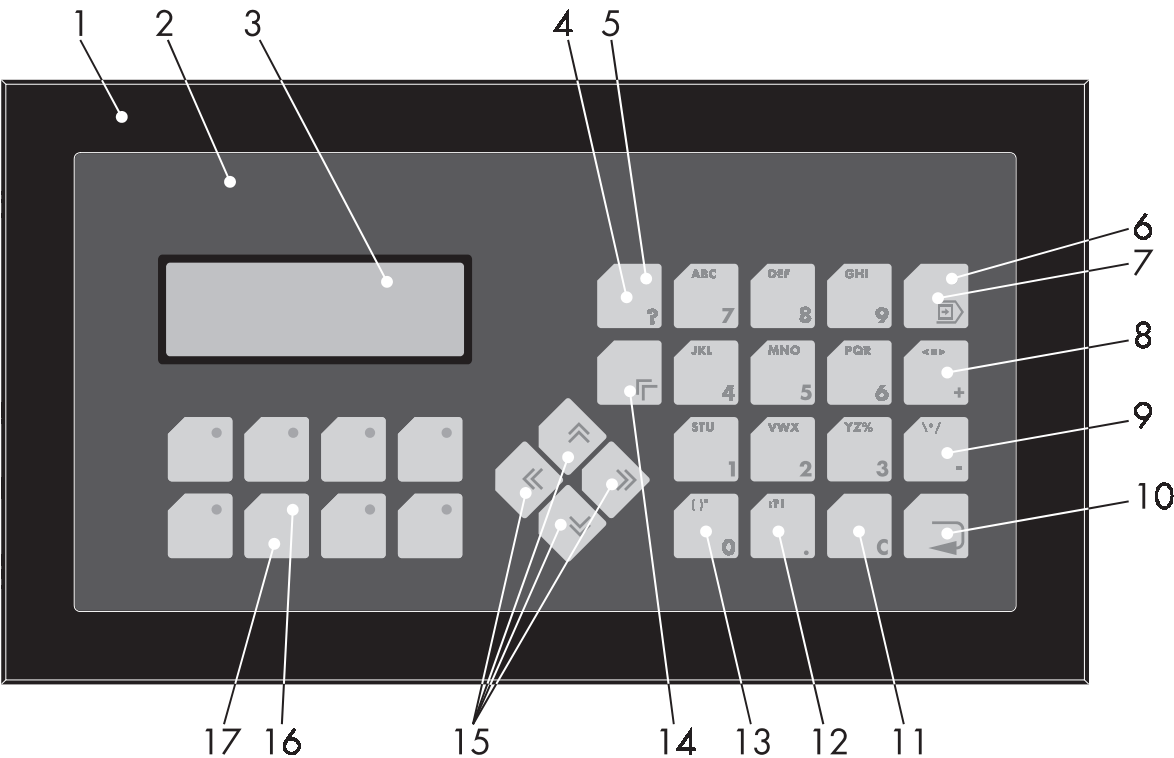
- | | | | |
|----|------------------------------|----|------------------------------|
| 1 | Front Panel | 11 | Key Cursor Down |
| 2 | Front Cover | 12 | Key Cursor Left |
| 3 | Display | 13 | Status-LED Key Help |
| 4 | Status-LED Function Key (F3) | 14 | Key Help |
| 5 | Function Key (F3) | 15 | Status-LED Key Data Release |
| 6 | Status-LED Function Key (F4) | 16 | Key Data Release |
| 7 | Function Key (F4) | 17 | Function Key (F2) |
| 8 | Key Enter | 18 | Status-LED Function Key (F2) |
| 9 | Key Minus | 19 | Function Key (F1) |
| 10 | Key Plus | 20 | Status-LED Function Key (F1) |

1.1.2 NT4S-SF122B-E



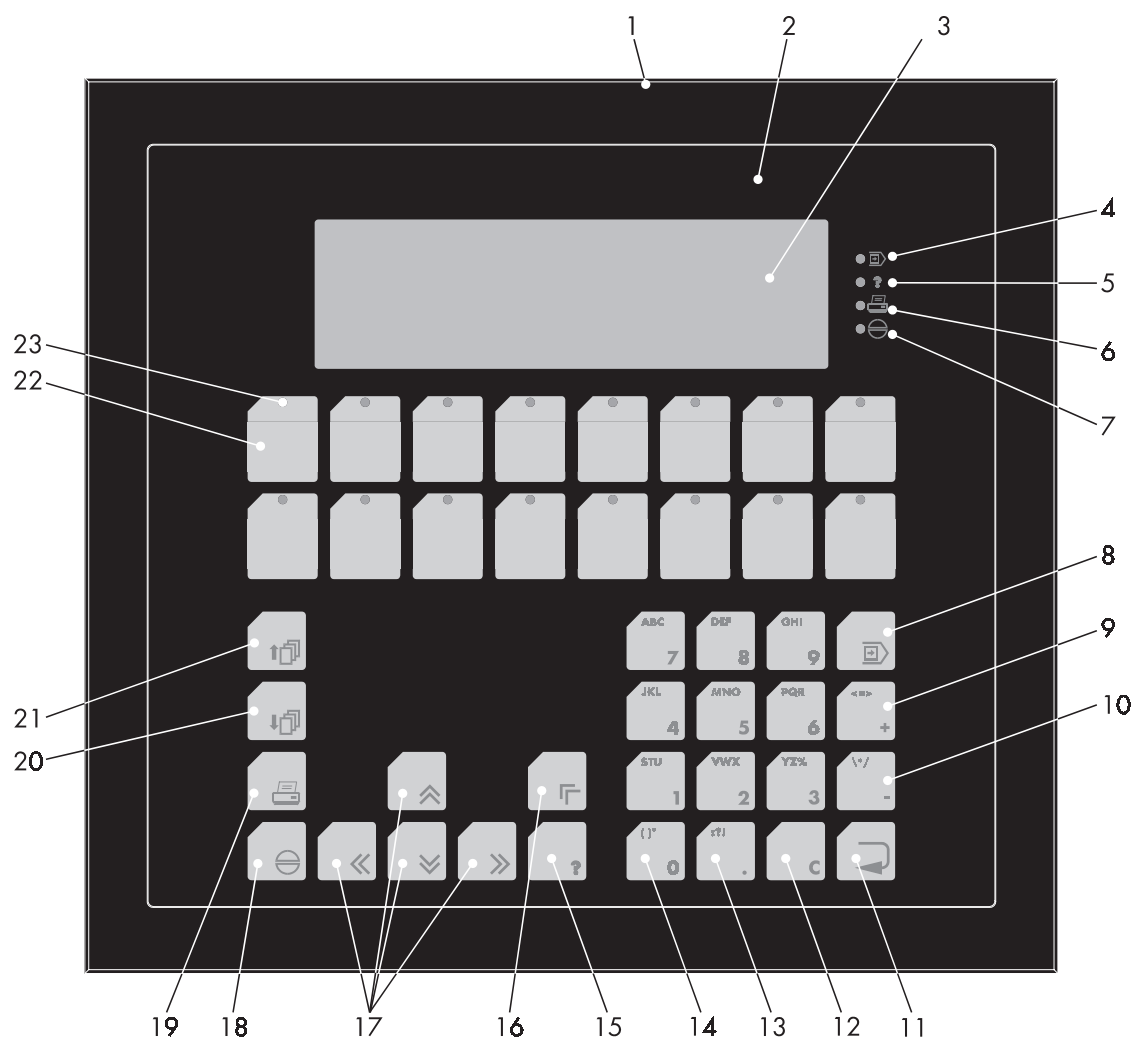
- | | |
|--------------------------------|-------------------------------------|
| 1 Front Panel | 10 Key Scroll |
| 2 Front Cover | 11 Key Plus, Minus |
| 3 Display | 12 Key Enter |
| 4 Function Key (F6) | 13 Key Cursor Home |
| 5 Status-LED Function Key (F6) | 14 Key Cursor Right, Left, Up, Down |
| 6 Status-LED Key Help | 15 Key Dot |
| 7 Status-LED Key Data Release | 16 Keys 0 to 9 |
| 8 Key Data Release | 17 Key Clear |
| 9 Key Help | 18 Key Print |

1.1.3 NT4S-SF123B-E



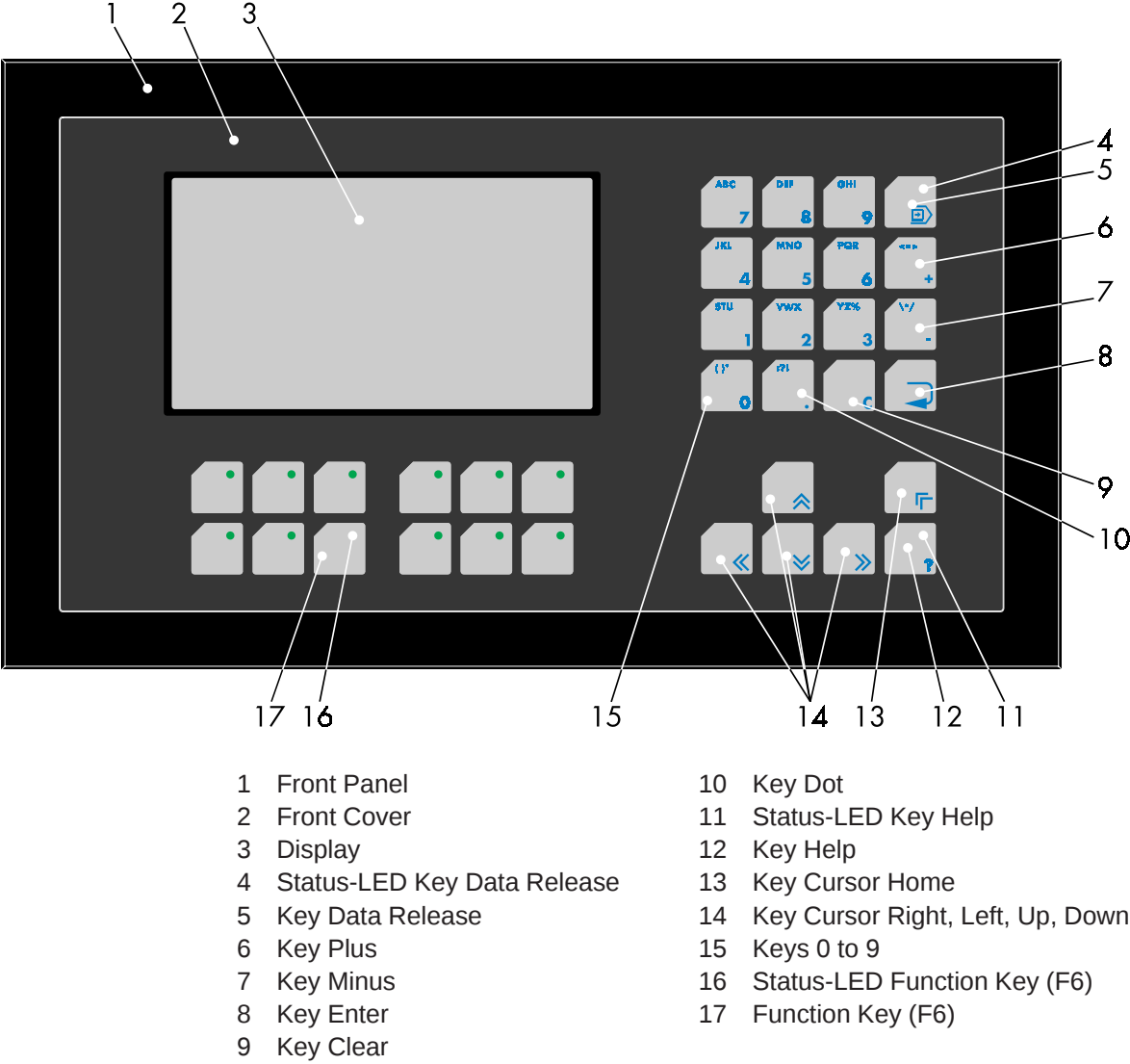
- | | | | |
|---|-----------------------------|----|----------------------------------|
| 1 | Front Panel | 10 | Key Enter |
| 2 | Front Cover | 11 | Key Clear |
| 3 | Display | 12 | Key Dot |
| 4 | Key Help | 13 | Keys 0 to 9 |
| 5 | Status-LED Key Help | 14 | Key Cursor Home |
| 6 | Status-LED Key Data Release | 15 | Key Cursor Right, Left, Up, Down |
| 7 | Key Data Release | 16 | Status-LED Function Key (F6) |
| 8 | Key Plus | 17 | Function Key (F6) |
| 9 | Key Minus | | |

1.1.4 NT15S-SF121B-E



- | | |
|---------------------------|-------------------------------------|
| 1 Front Panel | 13 Key Dot |
| 2 Front Cover | 14 Keys 0 to 9 |
| 3 Display | 15 Key Help |
| 4 Status-LED Data Release | 16 Key Cursor Home |
| 5 Status-LED Help | 17 Key Cursor Right, Left, Up, Down |
| 6 Status-LED Print | 18 Key Acknowledge |
| 7 Status-LED Acknowledge | 19 Key Print |
| 8 Key Data Release | 20 Key Page Down |
| 9 Key Plus | 21 Key Page Up |
| 10 Key Minus | 22 Function Key (F1) |
| 11 Key Enter | 23 Status-LED Function Key (F1) |
| 12 Key Clear | |

1.1.5 NT18S-SF121B-E



1.2 Keyboard

The terminals NT4S-SF121B-E, NT4S-SF123B-E and NT18S-SF121B-E are equipped with a keyboard made of short-stroke keys. The stroke length is approximately 0.5 mm.

The terminals NT4S-SF122B-E and NT15S-SF121B-E are equipped with a keyboard made of membrane keys. The stroke length is approximately 0.3 mm.

The key area of each key is 16 × 16 mm. The keys are covered by an embossed polyester sheet which is resistant to environmental effects. These combinations allows a sensitive use of the keys.

The status-LEDs of the function keys illuminate green. In transparent mode the keys supply a fixed start and stop code. In standard mode, the function of the keys is as defined in the application.

1.2.1 Editing Keys



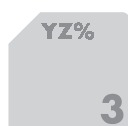
Key: **0 and () °** is used to edit data within the editor. If the system variable **Shift** or **ShiftCase** is programmed, the characters (and) and ° can be entered.



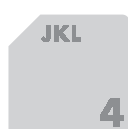
Key: **1 and STU** is used to edit data within the editor. If the system variable **Shift** or **ShiftCase** is programmed, the characters S and T and U can be entered.



Key: **2 and VWX** is used to edit data within the editor. If the system variable **Shift** or **ShiftCase** is programmed, the characters V and W and X can be entered.



Key: **3 and YZ%** is used to edit data within the editor. If the system variable **Shift** or **ShiftCase** is programmed, the characters Y and Z and % can be entered.



Key: **4 and JKL** is used to edit data within the editor. If the system variable **Shift** or **ShiftCase** is programmed, the characters J and K and L can be entered.



Key: **5 and MNO** is used to edit data within the editor. If the system variable **Shift** or **ShiftCase** is programmed, the characters M and N and O can be entered.



Key: **6 and PQR** is used to edit data within the editor. If the system variable **Shift** or **ShiftCase** is programmed, the characters P and Q and R can be entered.



Key: **7 and ABC** is used to edit data within the editor. If the system variable **Shift** or **ShiftCase** is programmed, the characters A and B and C can be entered.



Key: **8 and DEF** is used to edit data within the editor. If the system variable **Shift** or **ShiftCase** is programmed, the characters D and E and F can be entered.



Key: **9 and GHI** is used to edit data within the editor. If the system variable **Shift** or **ShiftCase** is programmed, the characters G and H and I can be entered.



Key: **Decimal Point and :?!** is used to edit data within the editor. If the system variable **Shift** or **ShiftCase** is programmed, the characters : and ? and ! can be entered.



Key: **Minus and */** can be used to enter negative values within the editor. In the increment editor, the variable value is decremented by 1. When the key is held down, the function is repeated at a rate of repetition that is automatically increased. If the system variable **Shift** or **ShiftCase** is programmed, the characters \ and * and / can be entered.



Key: **Plus and <=>** can be used to enter positive values within the editor. In the increment editor, the variable value is incremented by 1. When the key is held down, the function is repeated at a rate of repetition that is automatically increased. If the system variable **Shift** or **ShiftCase** is programmed, the characters < and = and > can be entered.

1.2.2 Control Keys



Key: **Cursor left** can be programmed to directly select I/O masks. In the editor, it moves the cursor to the left.



Key: **Cursor right** can be programmed to directly select I/O masks. In the editor, it moves the cursor to the right.



Key: **Cursor up** can be programmed to directly select I/O masks. In the editor, it moves the cursor upwards.



Key: **Cursor down** can be programmed to directly select I/O masks. In the editor, it moves the cursor downwards.



Key: **Cursor home** can be programmed to directly select I/O masks. In the editor, it moves the cursor to the position of the first input variable.



Key: **Page down** is used to page through tables, recipes and messages. The functionality corresponds to the system variable 'TabPgDn'. The key allows data contents towards the bottom of the table to be viewed.



Key: **Page up** is used to page through tables, recipes and messages. The functionality corresponds to the system variable 'TabPgUp'. The key allows data contents towards the top of the table to be viewed.

1.2.3 Special Keys



Key: **Help key** always displays the current help text (online help). When the status-LED help flashes, it signals that an error message is pending. The error or system message is always displayed in plain-text.



Key: **Data Release** key is used to switch from a menu into the editor. The status-LED data release lights up when the editing mode is active. When the Data Release key is pressed within the editor, the editing mode is exited.



Key: **Enter** is used to conclude data entry. When pressed while in the startup mask, the key switches into the setup mask.



Key: **Clear** deletes the character beneath the cursor when it is used in an editor. Deletes the selected messages from the data memory.



Key: **Acknowledge** is used as an acknowledge key for the message system.



Key: **Print** can be used as a soft key to activate various print processes.

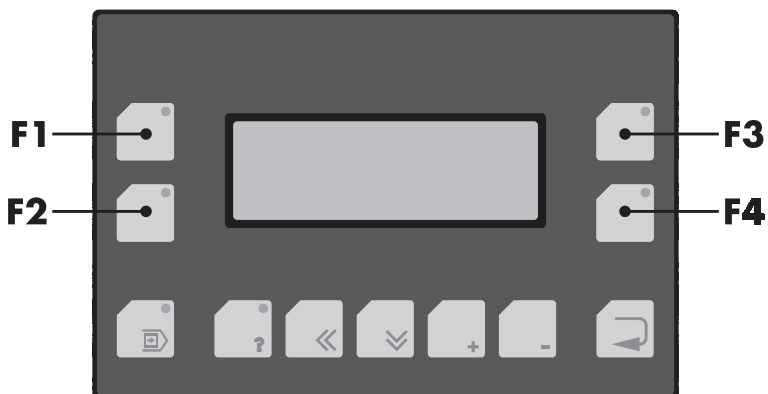
1.2.4 Function Keys



Keys: **Function Keys F1 to F16** with status-LEDs for functional feedback. In standard mode, the key functions can be freely assigned to a soft key functionality; either as direct keys for menu control or to activate a function in the controller.

1.2.4.1 Function Key Arrangement

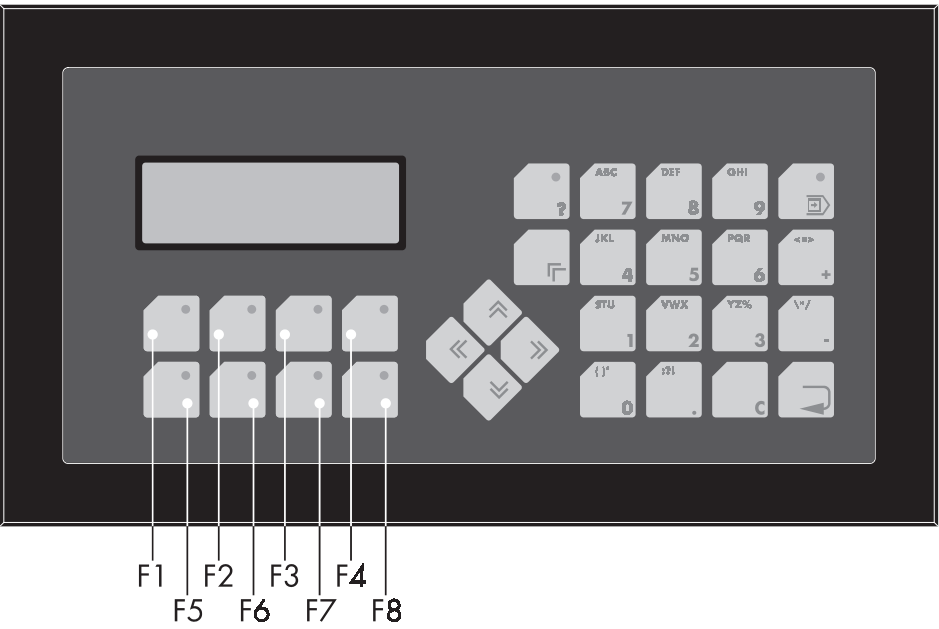
1.2.4.1.1 NT4S-SF121B-E



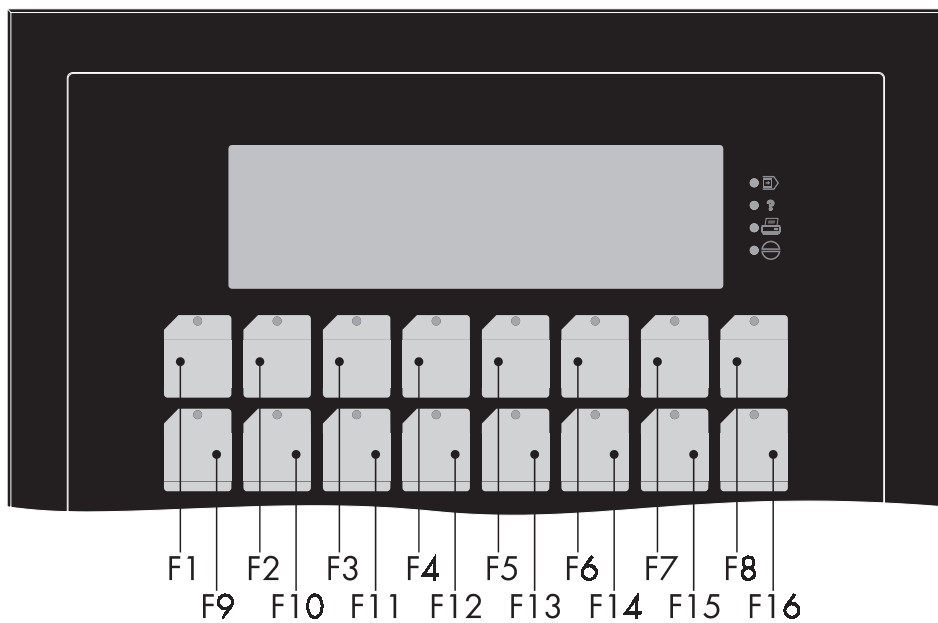
1.2.4.1.2 NT4S-SF122B-E



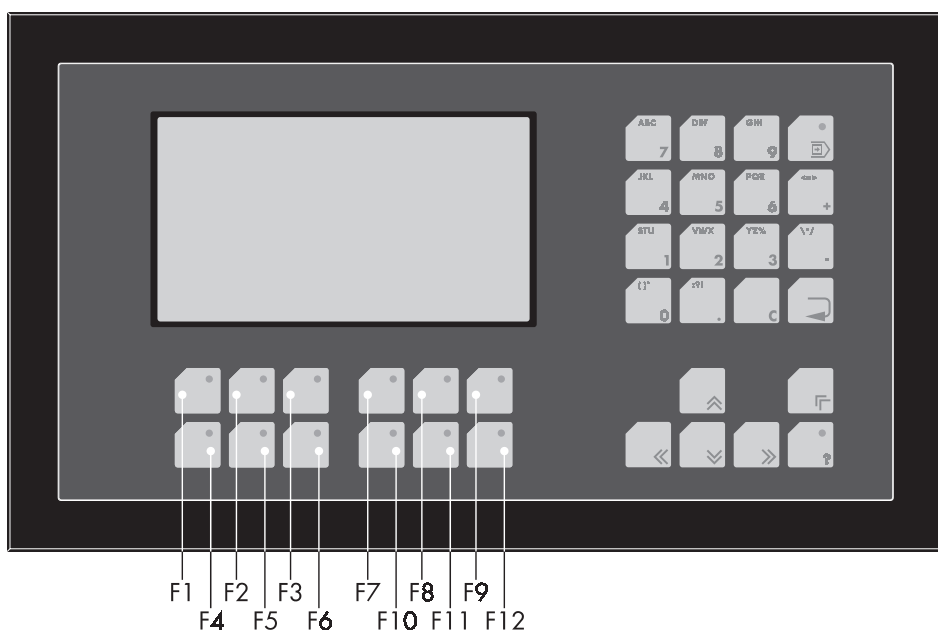
1.2.4.1.3 NT4S-SF123B-E



1.2.4.1.4 NT15S-SF121B-E



1.2.4.1.5 NT18S-SF121B-E



1.2.5 Slide-in Identification Strips for the Function Keys

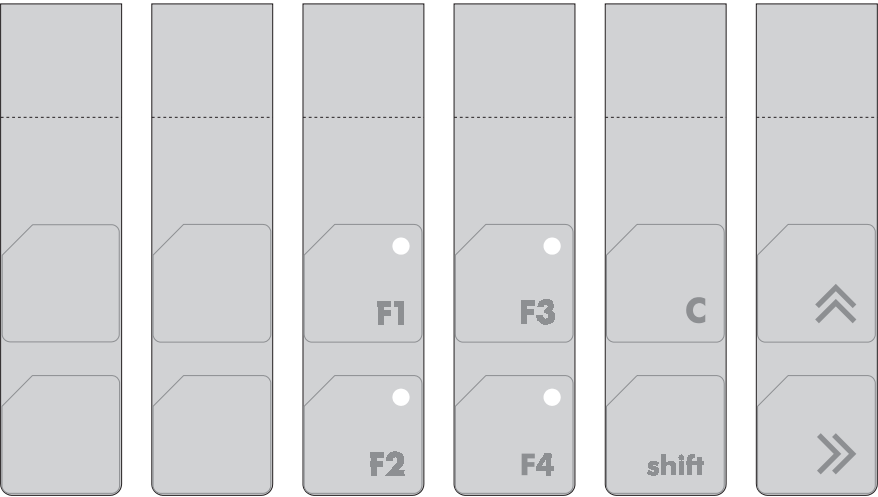
The terminals are equipped with a set of slide-in identification strips. Each set consists of ready-to-use labelled and blank slide-in identification strips.

Various labeling methods are recommended, depending on the number of terminals involved.

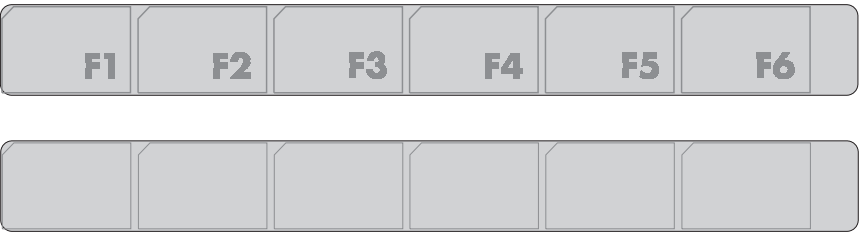
Suitable labeling methods for:

- single terminal:
- small number of terminals:
- large number of terminals:
- labeling with an indelible pen
- transparency with laser printing
- identification strips printed according to customer's needs

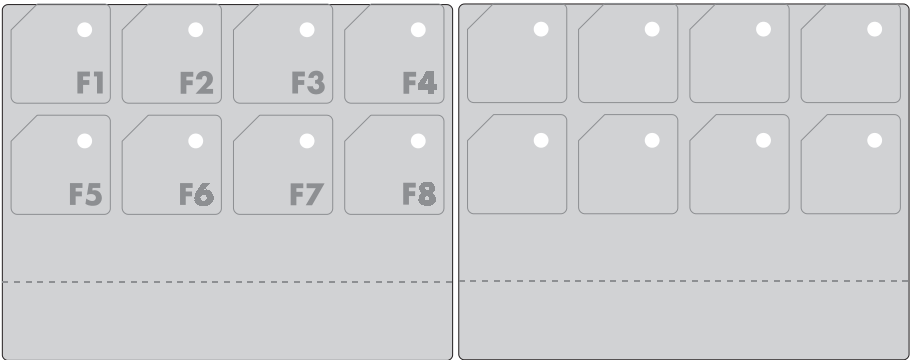
1.2.5.1 NT4S-SF121B-E



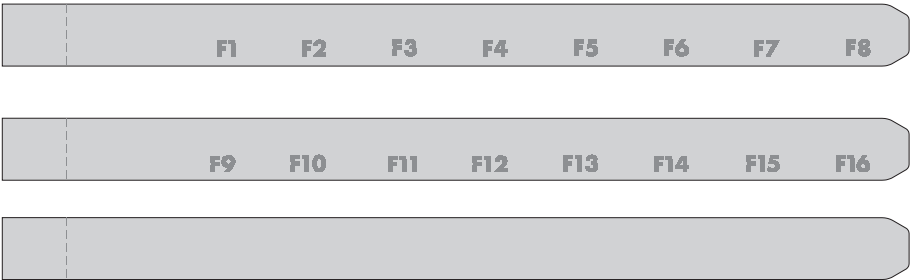
1.2.5.2 NT4S-SF122B-E



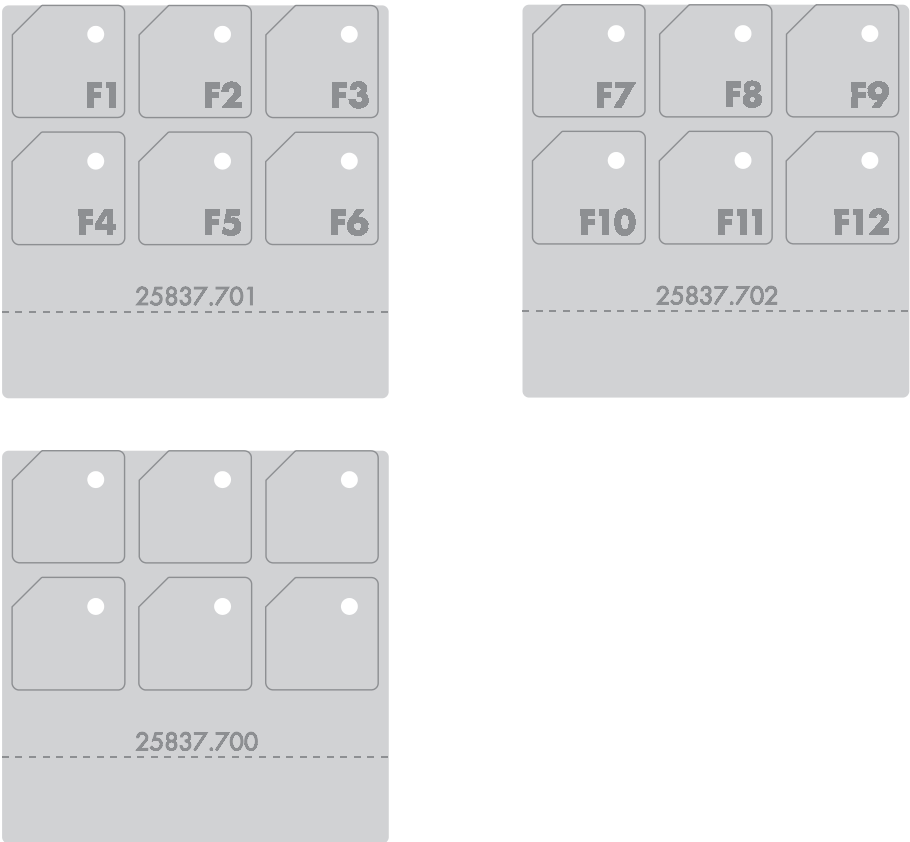
1.2.5.3 NT4S-SF123B-E



1.2.5.4 NT15S-SF121B-E

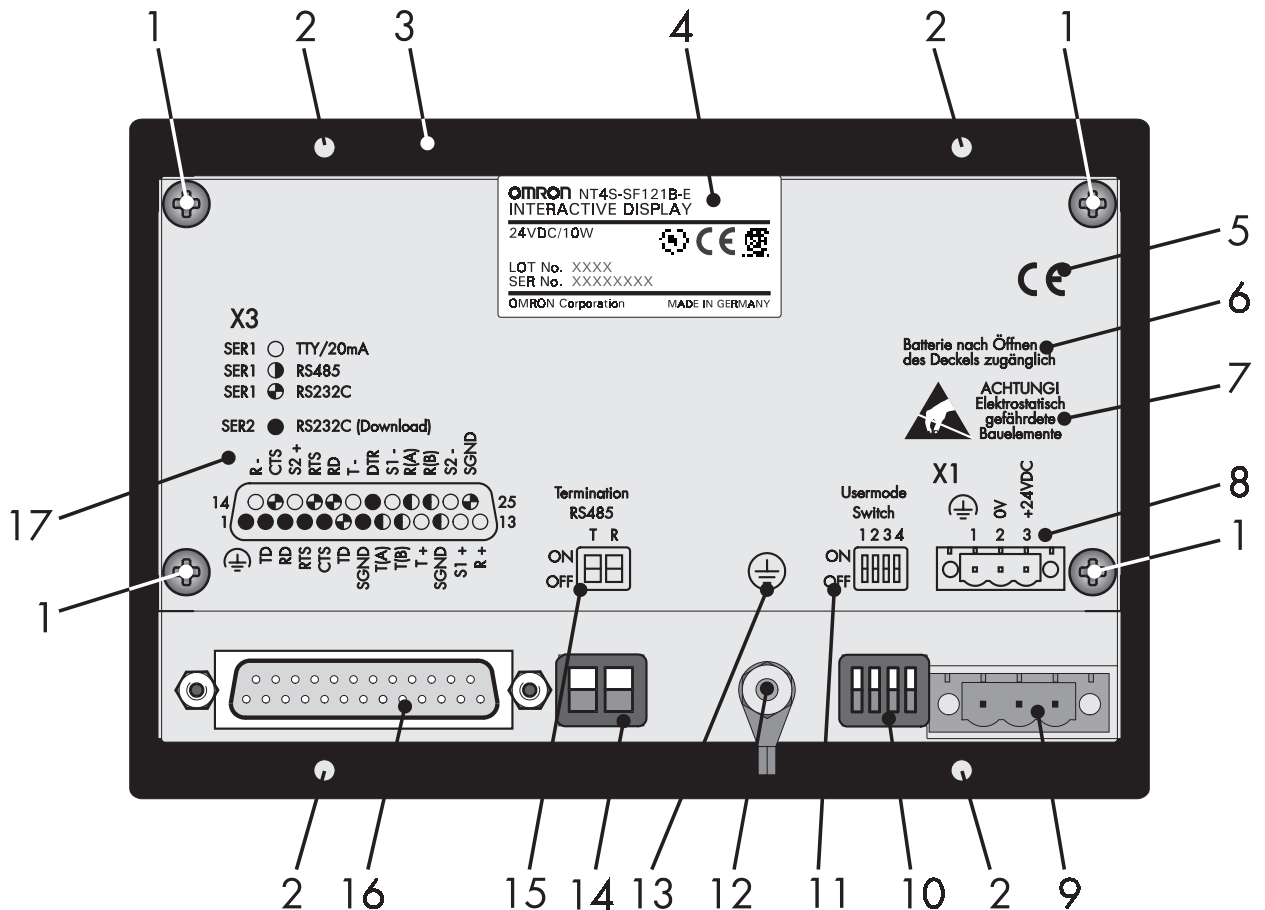


1.2.5.5 NT18S-SF121B-E



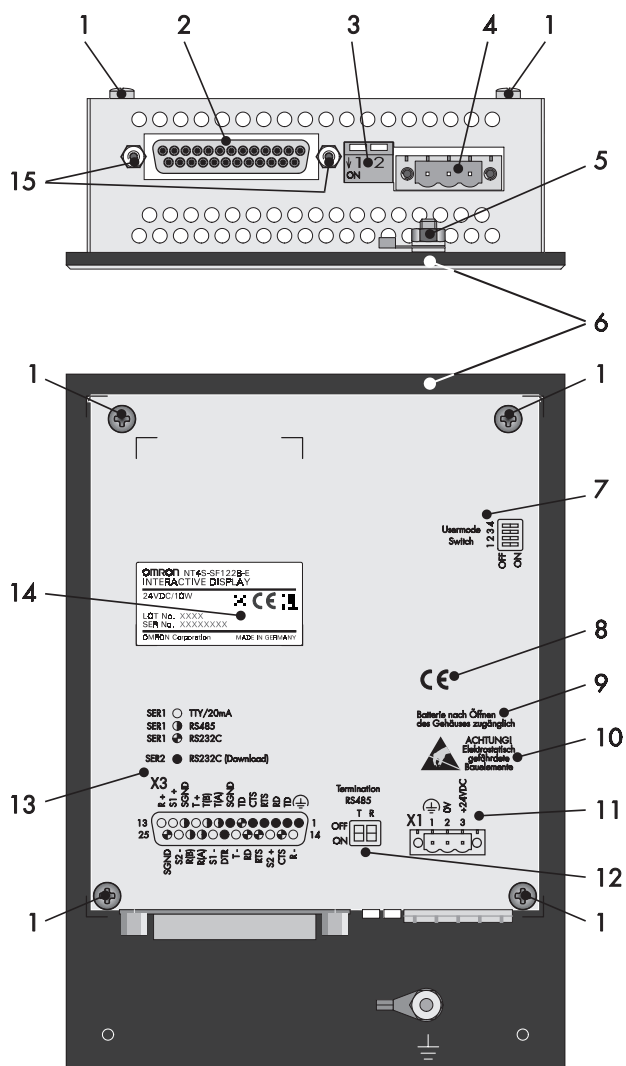
1.3 Rear View

1.3.1 NT4S-SF121B-E



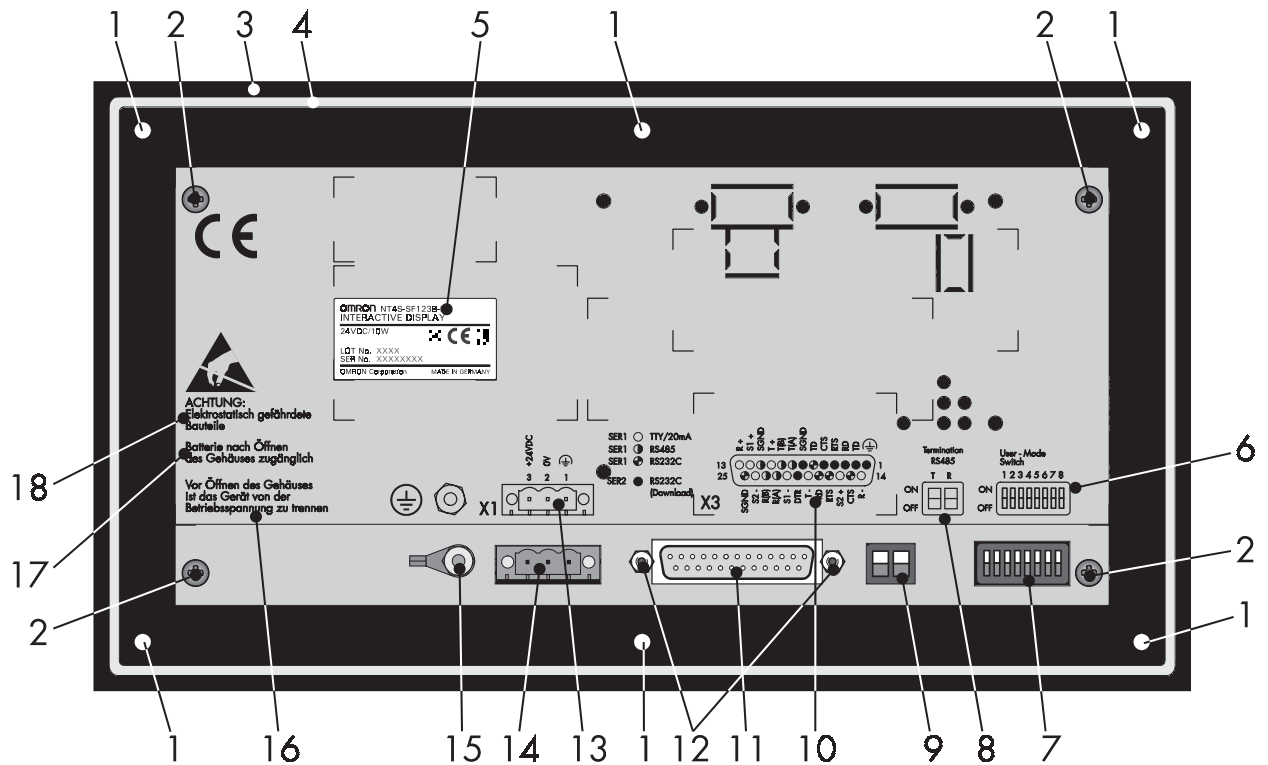
- | | |
|---------------------------------|--|
| 1 Fastening Screw for Enclosure | 10 User-Mode Switch |
| 2 Mounting Bolts | 11 Switch Assignment User-Mode Switch |
| 3 Front Panel | 12 Ground Screw |
| 4 Name Plate | 13 Sign for Ground Screw |
| 5 CE-Sign | 14 Terminator Switch (RS485) |
| 6 Note for Battery | 15 Switch Assignment Terminator Switch |
| 7 Warning Note | 16 Female Connector X2 |
| 8 Pin Assignment Connector X1 | 17 Pin Assignment Female Connector X2 |
| 9 Connector X1 (Power Supply) | |

1.3.2 NT4S-SF122B-E



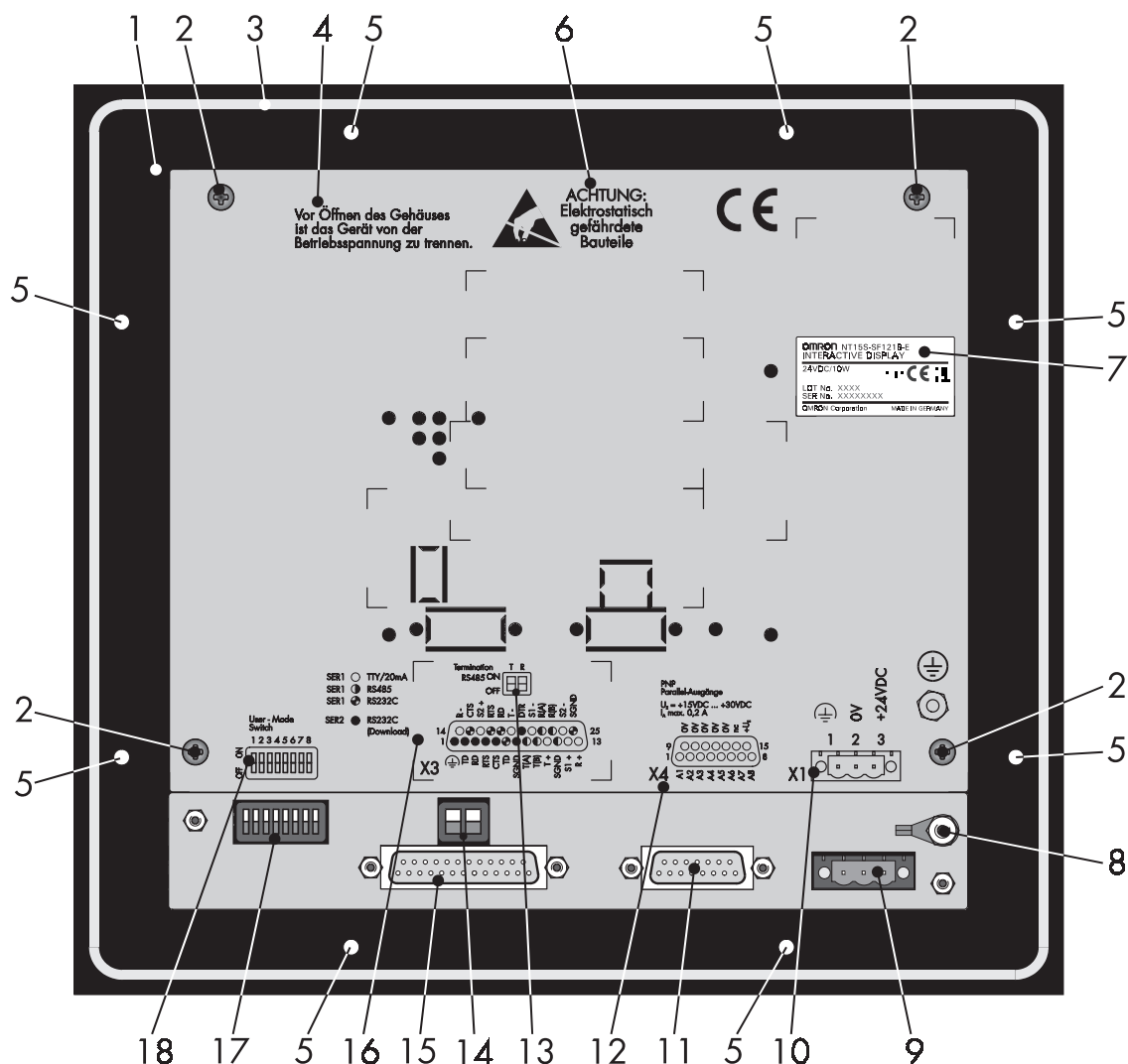
- | | |
|--------------------------------------|--|
| 1 Fastening Screw for Enclosure | 10 Warning Note |
| 2 Female Connector X3 | 11 Pin Assignment Connector X1 |
| 3 Terminator Switch (RS485) | 12 Switch Assignment Terminator Switch |
| 4 Connector X1 (Power Supply) | |
| 5 Ground Screw | 13 Pin Assignment Female Connector X3 |
| 6 Front Panel | 14 Name Plate |
| 7 Switch Assignment User-Mode Switch | 15 Fastening Screw for Female Connector X3 |
| 8 CE-Sign | |
| 9 Note for Battery | |

1.3.3 NT4S-SF123B-E



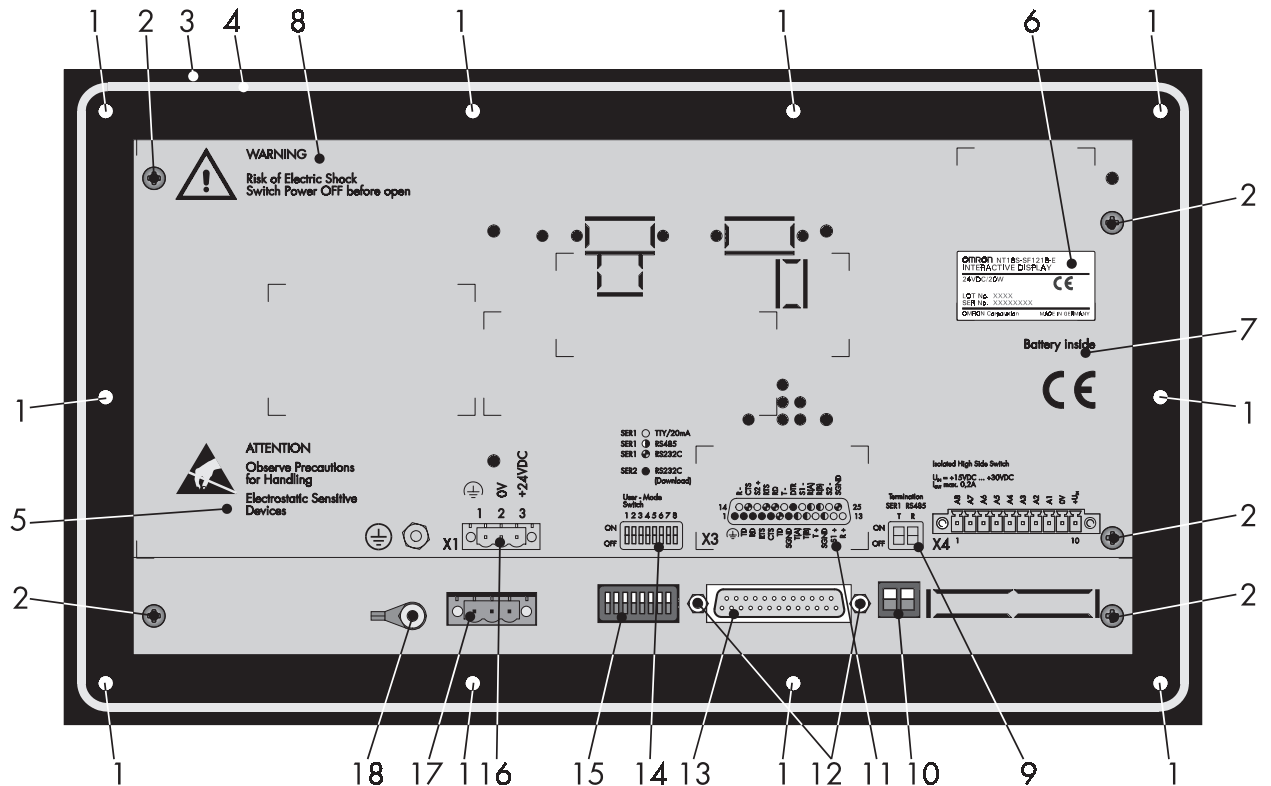
- | | |
|---------------------------------------|--|
| 1 Mounting Bolts | 10 Pin Assignment Female Connector X3 |
| 2 Fastening Screw for Enclosure | 11 Female Connector X3 |
| 3 Front Panel | 12 Fastening Screw for Female Connector X3 |
| 4 Rubber Sealing | 13 Pin Assignment Connector X1 |
| 5 Name Plate | 14 Connector X1 (Power Supply) |
| 6 Switch Assignment User-Mode Switch | 15 Ground Screw |
| 7 User-Mode Switch | 16 Warning Note |
| 8 Switch Assignment Terminator Switch | 17 CE-Sign |
| 9 Terminator Switch (RS485) | 18 Note for Battery |

1.3.4 NT15S-SF121B-E



- | | |
|------------------------------------|---|
| 1 Front Panel | 12 Pin Assignment Connector X4 (Parallel Outputs) |
| 2 Fastening Screw for Enclosure | 13 Switch Assignment Terminator Switch |
| 3 Rubber Sealing | 14 Terminator Switch (RS485) |
| 4 Warning Note | 15 Female Connector X3 |
| 5 Mounting Bolts | 16 Pin Assignment Female Connector X3 |
| 6 Note for Battery | 17 User-Mode Switch |
| 7 Name Plate | 18 Switch Assignment User-Mode Switch |
| 8 Ground Screw | |
| 9 Connector X1 (Power Supply) | |
| 10 Pin Assignment Connector X1 | |
| 11 Connector X4 (Parallel Outputs) | |

1.3.5 NT18S-SF121B-E



- | | |
|---------------------------------------|--|
| 1 Mounting Bolts | 10 Terminator Switch (RS485) |
| 2 Fastening Screw for Enclosure | 11 Pin Assignment Female Connector X3 |
| 3 Front Panel | 12 Fastening Screw for Female Connector X3 |
| 4 Rubber Sealing | 13 Female Connector X3 |
| 5 Warning Note | 14 Switch Assignment User-Mode Switch |
| 6 Name Plate | 15 User-Mode Switch |
| 7 Note for Battery | 16 Pin Assignment Connector X1 |
| 8 Warning Note | 17 Connector X1 (Power Supply) |
| 9 Switch Assignment Terminator Switch | 18 Ground Screw |

1.4 Mounting the Terminals

The terminals with mounting bolts or lateral mounting slits are mounted from the rear side. The rear panel mounting is suitable for easy and sealed installation places where the rear side of the unit is accessible.

The terminal NT15S-SF121B-E is mounted from the front side. This type of installation is suitable for easy and sealed installation where the rear side of the unit is not accessible.

The terminals are particularly suitable for mounting in control cabinets with a mounting plate thickness of approximately 1 to 10 mm.

The front panels permit sealed installation of the units in accordance with the IP65 degree of protection.

The terminals

- NT4S-SF123B-E
- NT18S-SF121B-E
- NT15S-SF121B-E

have a circumferential groove milled into the rear side of the front plate, containing a rubber sealing.

The terminals

- NT4S-SF121B-E
- NT4S-SF122B-E

are equipped with a foam sealing on the rear side of the front plate.

All parts for mounting the units are supplied with the spare parts set.

Special care needs to be taken during installation to maintain this high degree of protection. The appliance is inserted from the front through the mounting panel cutout and fastened with the fasteners of the spare parts set. The sealing must be positioned evenly and the fastening elements tightened uniformly.

When installing the terminal, keep a minimum space of 30 mm around the terminal for adequate air circulation.

The tightness between the front panel and the mounting surface depends on the care during installation.



Mounting and connection of the operating terminal to the power supply must be carried out by authorized and qualified personnel with EMC training.

Operating terminals must be grounded according to regulations.

The application-specific safety- and accident prevention regulations must be observed during the mounting and connection to the power supply, e.g.:

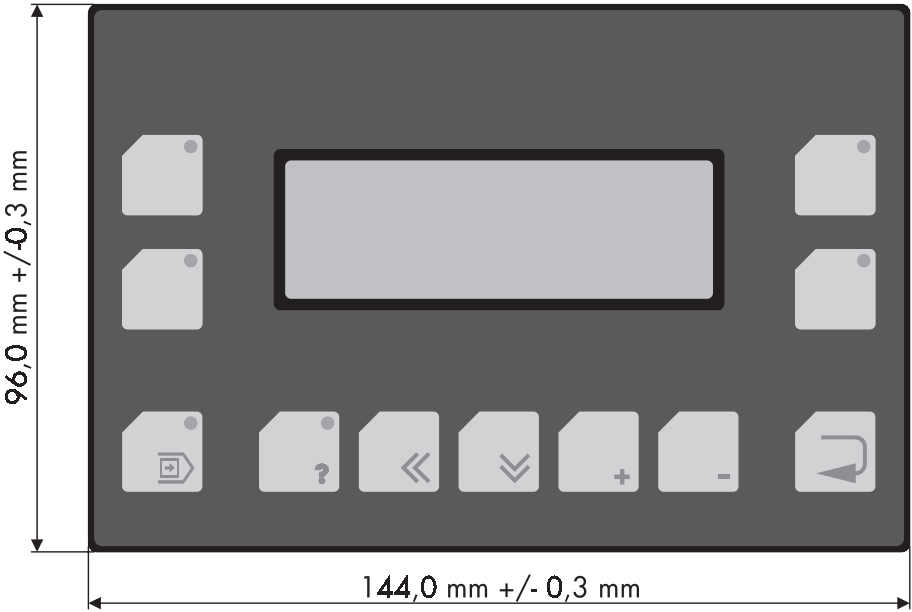
- EN 60204, Electrical Equipment of Machines
- EN 292, Safety of Machinery, General Principles for Design
- DIN 57 100 Part 410, Protection against Electric Shock



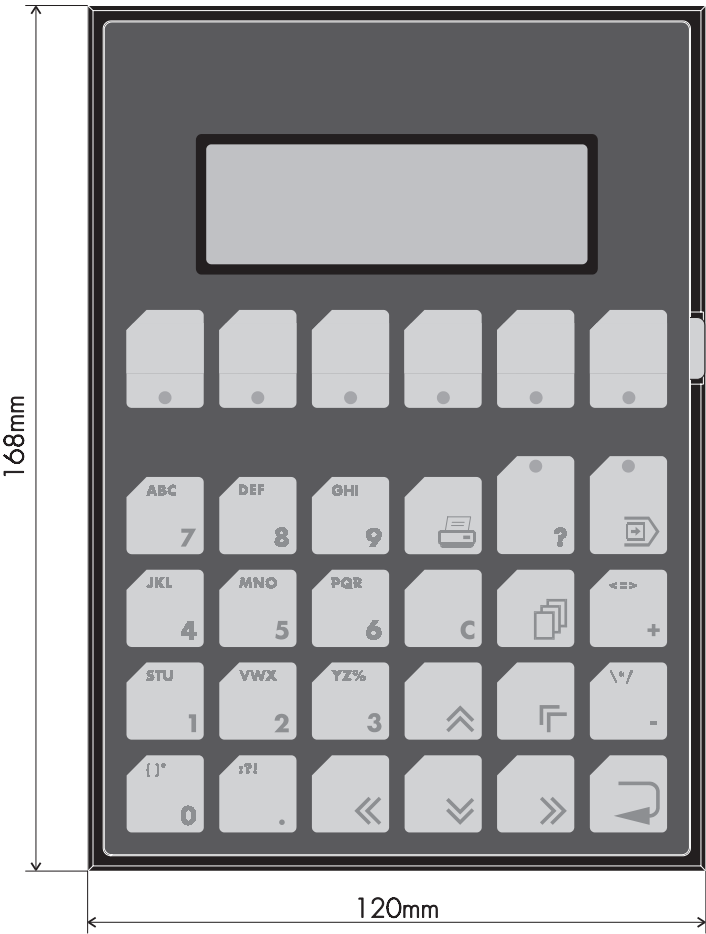
Hazardous voltages can exist inside electrical installations that can pose a danger to humans. Coming in contact with live parts may result in **electric shock!**

1.4.1 Front Panel Dimensions

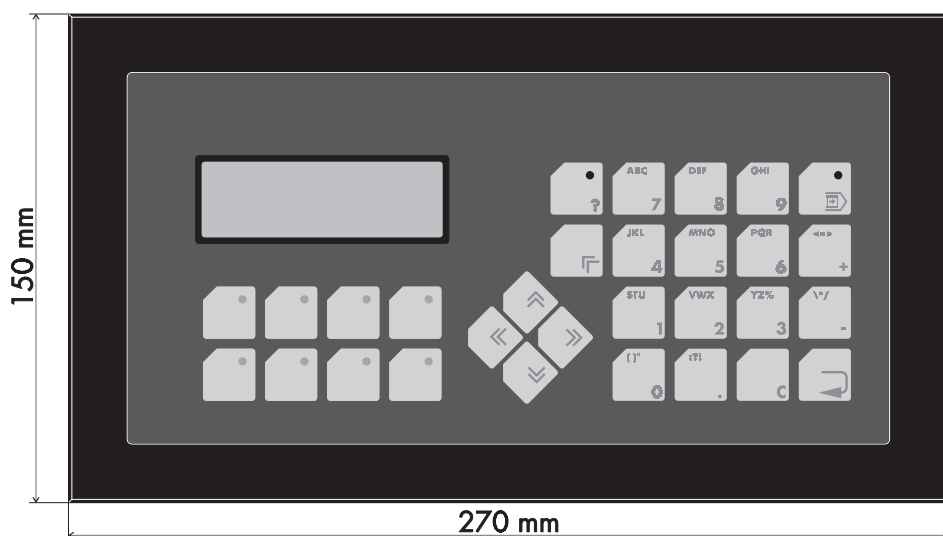
1.4.1.1 NT4S-SF121B-E



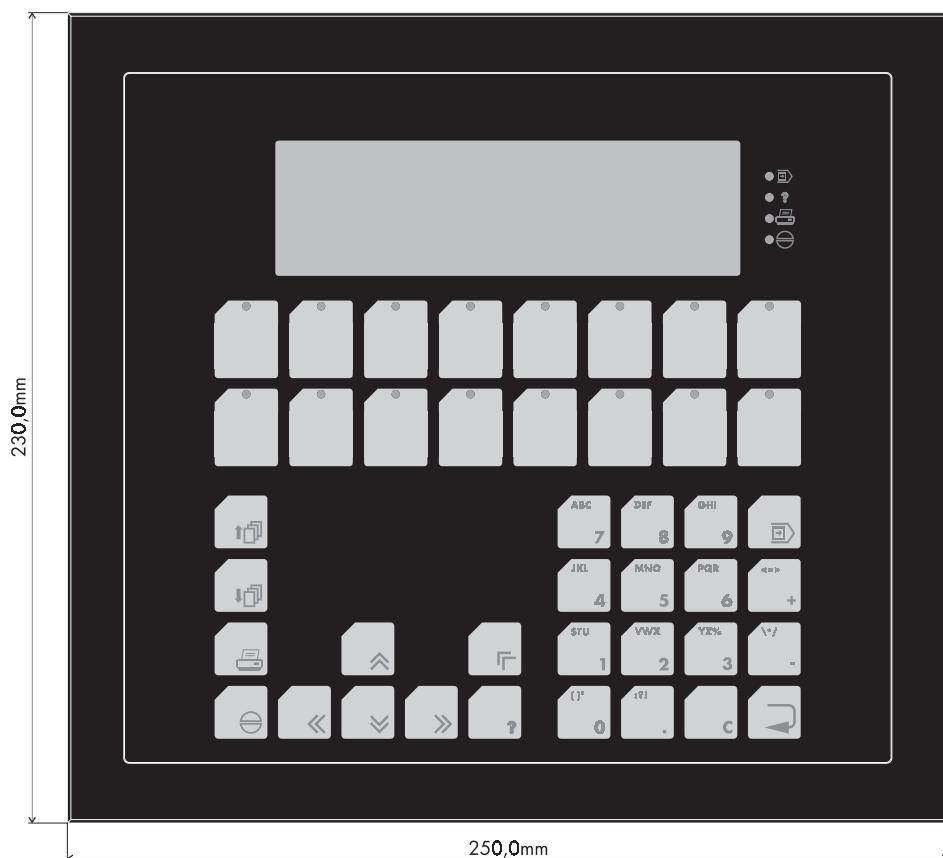
1.4.1.2 NT4S-SF122B-E



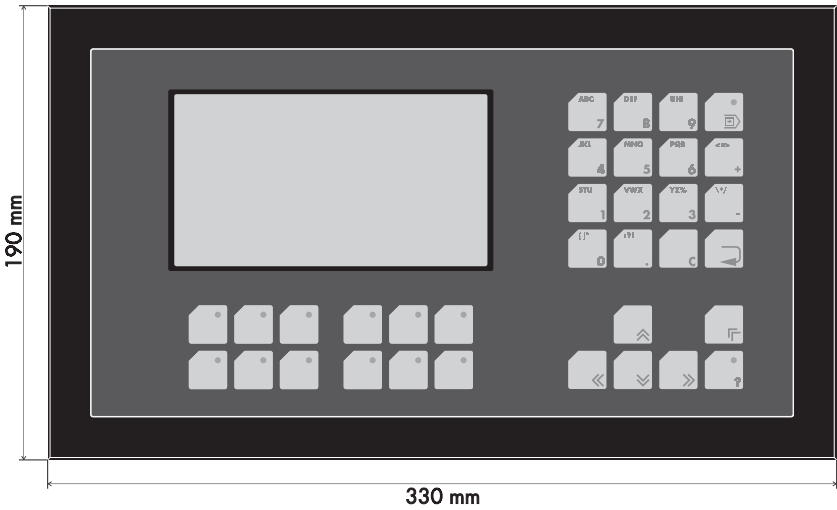
1.4.1.3 NT4S-SF123B-E



1.4.1.4 NT15S-SF121B-E

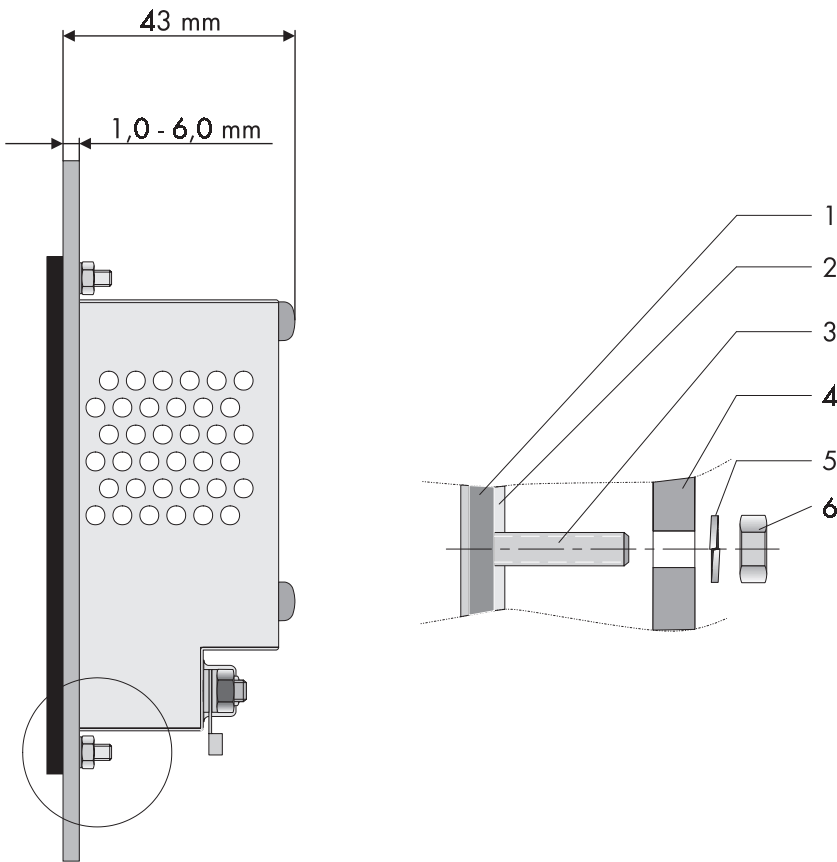


1.4.1.5 NT18S-SF121B-E



1.4.2 Side View, Mounting Depth

1.4.2.1 NT4S-SF121B-E



- 1 Front Panel

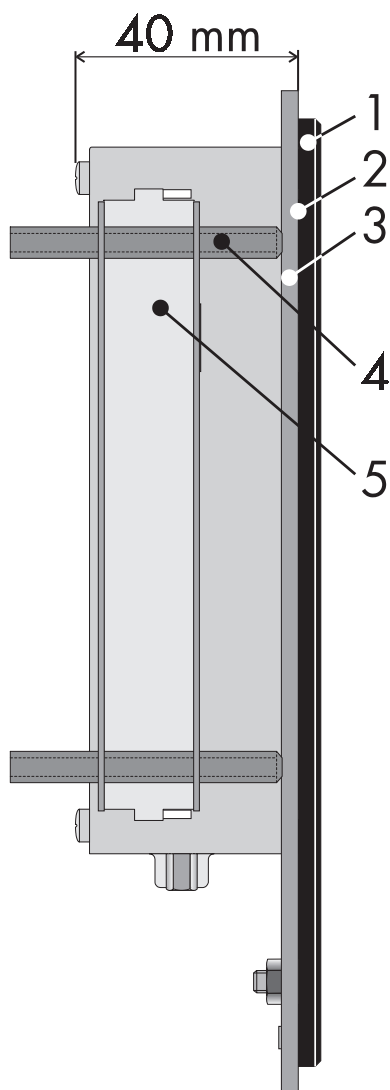
2 Foam Sealing

3 Press-In Threaded Bolt M3 x 9
- 4 Mounting Surface Thickness 1 to 6 mm

5 Spring Lock Washer B3 DIN127 Form B

6 Nut M3 DIN934

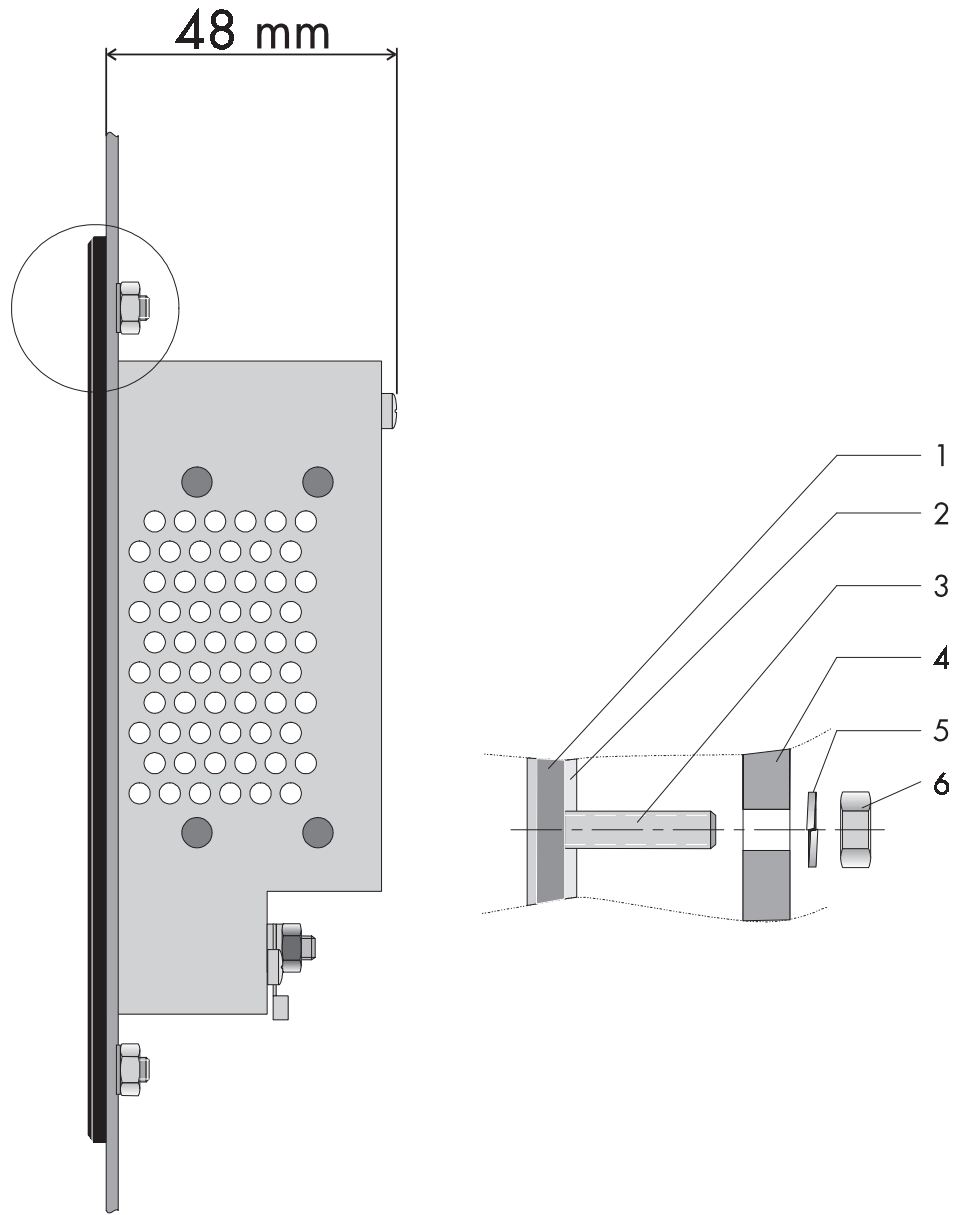
1.4.2.2 NT4S-SF122B-E



- 1 Front Panel
- 2 Foam Sealing
- 3 Mounting Surface Thickness 1 to 14 mm

- 4 Threaded Pin DIN 914 M4 x 35
- 5 Mounting Bracket

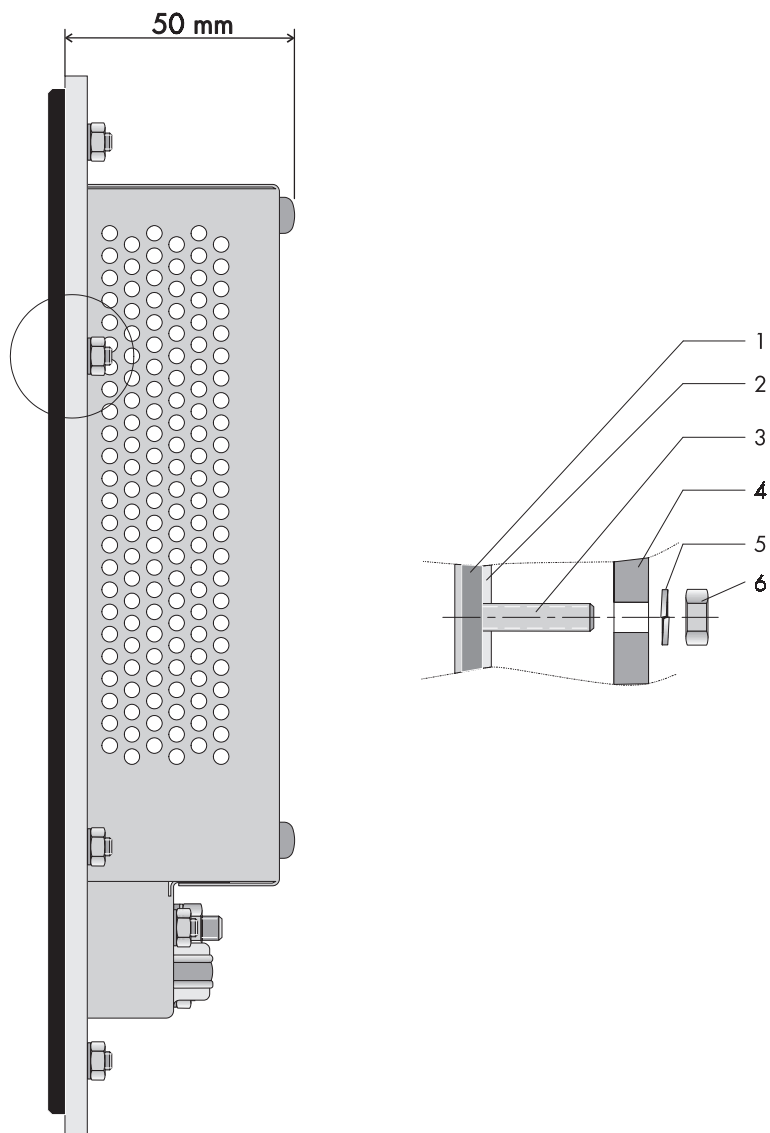
1.4.2.3 NT4S-SF123B-E



- | | |
|---|---------------------------------------|
| 1 | Front Panel |
| 2 | Circumferential Rubber Sealing |
| 3 | Press-In Threaded Bolt M4 x 16 |
| 4 | Mounting Surface Thickness 1 to 10 mm |

- | | |
|---|--|
| 5 | Spring Lock Washer B4
DIN127 Form B |
| 6 | Nut M4 DIN934 |

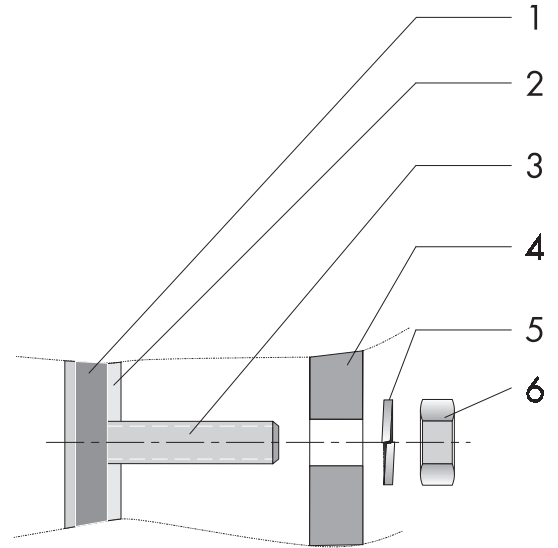
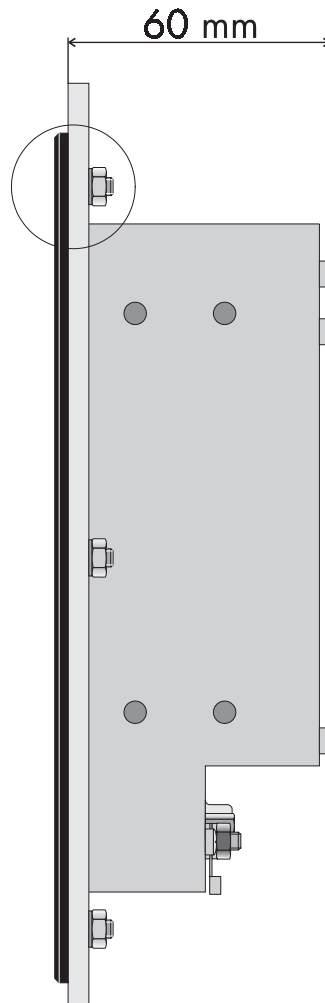
1.4.2.4 NT15S-SF121B-E



- 1 Front Panel
- 2 Circumferential Rubber Sealing
- 3 Press-In Threaded Bolt M4 x 16
- 4 Mounting Surface Thickness 1 to 10 mm

- 5 Spring Lock Washer B4 DIN127 Form B
- 6 Nut M4 DIN934

1.4.2.5 NT18S-SF121B-E

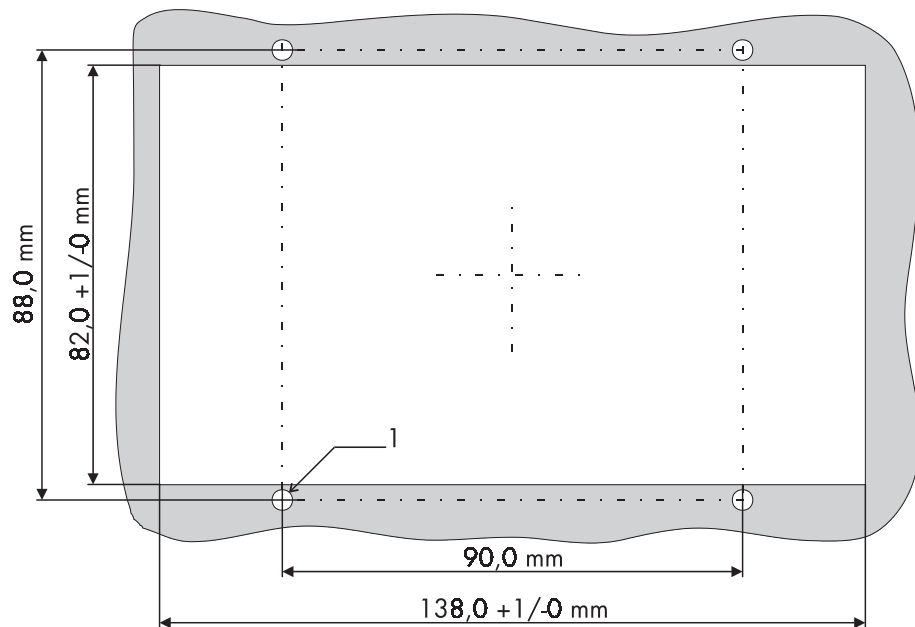


- 1 Front Panel
- 2 Circumferential Rubber Sealing
- 3 Press-In Threaded Bolt M4 x 16
- 4 Mounting Surface Thickness 1 to 10 mm

- 5 Spring Lock Washer B4 DIN127 Form B
- 6 Nut M4 DIN934

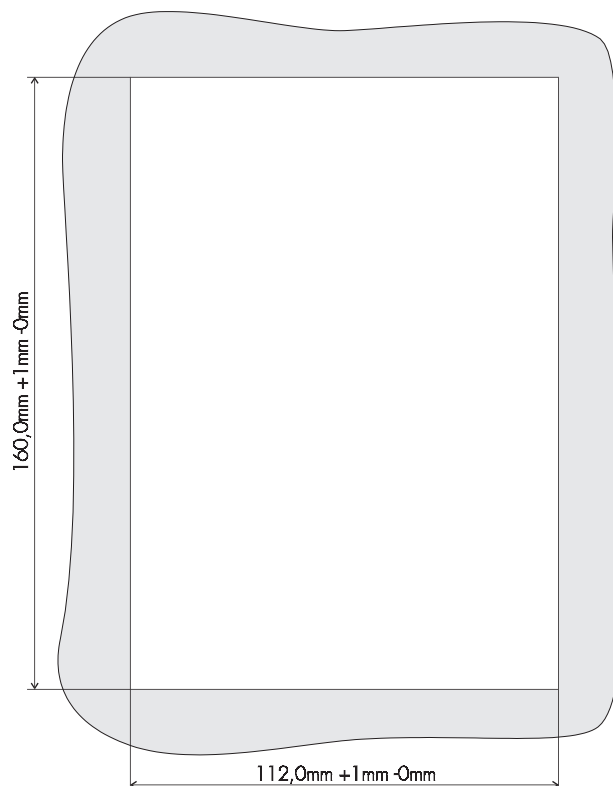
1.4.3 Panel Cutout

1.4.3.1 NT4S-SF121B-E

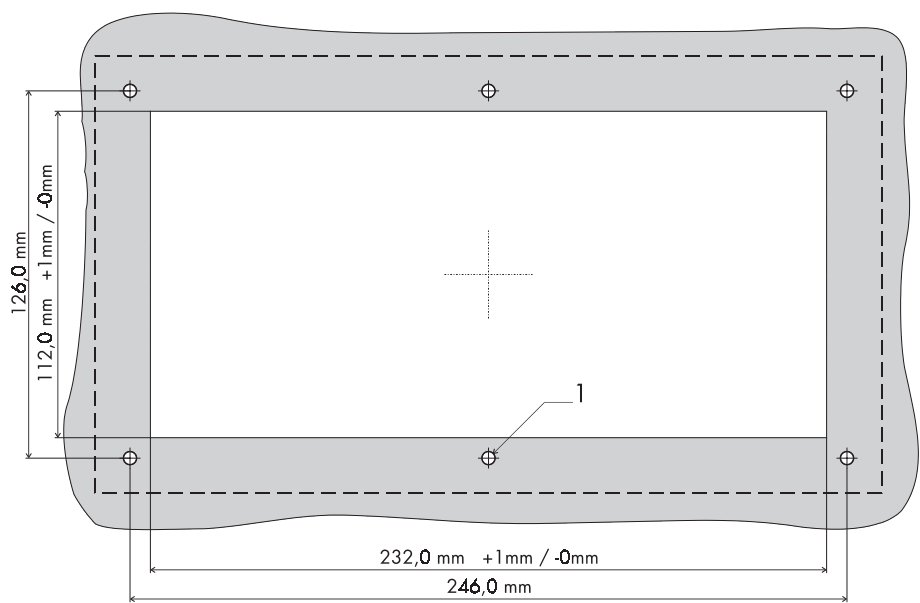


1 4 Holes with a Diameter of 3.5 mm

1.4.3.2 NT4S-SF122B-E

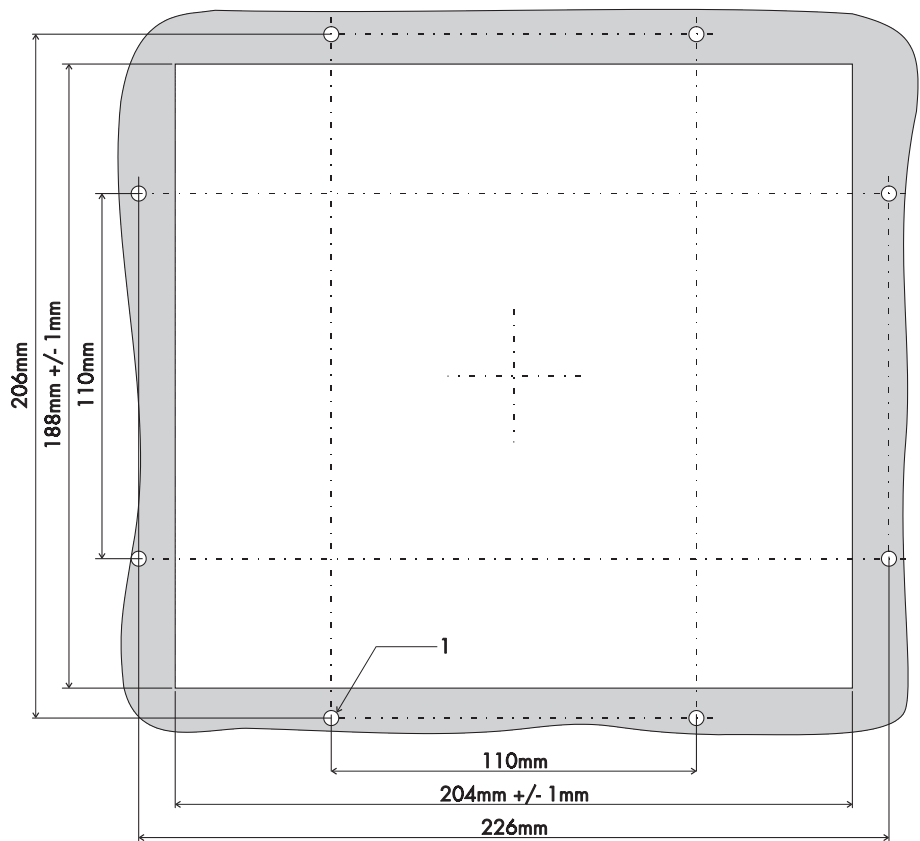


1.4.3.3 NT4S-SF123B-E



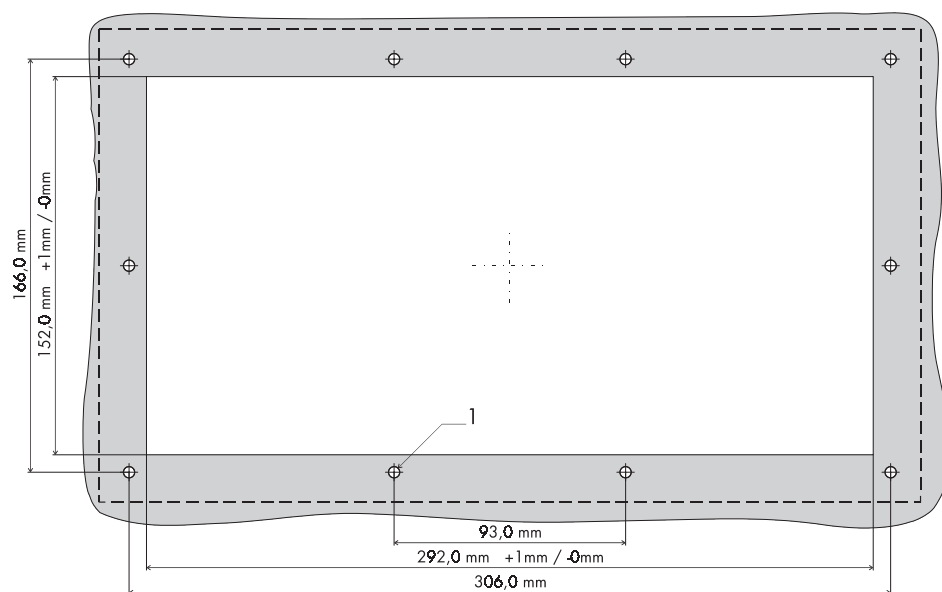
1 6 Holes with a Diameter of 4.5 mm

1.4.3.4 NT15S-SF121B-E



1 8 Holes with a Diameter of 4.5 mm

1.4.3.5 NT18S-SF121B-E



1 10 Holes with a Diameter of 4.5 mm

1.5 Pin Assignments

The operating terminals are equipped with a universal interface by default. This universal interface combines several interface standards within one physical connector. The connector is divided into two channels. The channel for communication (SER1) is divided from the channel for download/upload/printer (SER2). The channels can operate independent of each other.

The channel for communication (SER1) can handle - specified by protocol - only one of the three interface standards.

Connector X124 VDCSupply Voltage

Connector X3 SER1 ...TTY / 20 mACommunication

Connector X3 SER1 ...RS485Communication

Connector X3 SER1 ...RS232cCommunication

Connector X3 SER2 ...RS232cDownload/Upload/Logging Printer

The operating terminal NT15S-SF121B-E has additional parallel outputs:

Connector X4Parallel Outputs

The parallel outputs are suitable for access of PLC-inputs.

1.5.1 Pin Assignment X1 Supply Voltage

The supply voltage is connected via the connector X1.

Connector in the terminal: 3-pin male connector strip Phoenix COMBICON MSTBV 2,5/3-GF

Pin	Designation	Function
1		Signal Ground
2	0 V	Supply Voltage 0 V
3	24 VDC	Supply Voltage 24 VDC

The supply voltage is connected via a plug-in 3-pin female connector strip. The cable is secured in the female connector strip by means of screw terminals. Cables with fine wires with a cross-section of up to 2.5mm² can be used. The female connector strip is secured in position by means of a screw-type locking. The female connector strip of the type **Phoenix COMBICON MSTB 2.5/3-STF** is supplied.



Hazardous voltages can exist inside electrical installations that can pose a danger to humans. Coming in contact with live parts may result in **electric shock!**



Please note with respect to pin assignment:
If shielded connecting cables are used for the supply voltage, the shield should be connected to the ground screw.
Any protective conductors in the cable must be connected with pin 1.



Separate ground screw for protective grounding
A separate ground conductor must be provided for the ground screw in each case. The minimum cross-section of the ground conductor must be 1.5 mm². Compliance with this information increases the operational safety.

1.5.2 Pin Assignment X3 SER1 TTY / 20 mA Current Loop

TTY / 20 mA current loop, passive

Pin	Designation	Channel	Function
10	T+	SER1	Transmit Data, Positive Polarity
13	R+	SER1	Receive Data, Positive Polarity
14	R-	SER1	Receive Data, Negative Polarity
19	T-	SER1	Transmit Data, Negative Polarity

TTY / 20 mA current loop, active

Pin	Designation	Channel	Function
10	T+	SER1	Transmit Data, Positive Polarity
12	S1+	SER1	Power Source 2, Positive Polarity
13	R+	SER1	Receive Data, Positive Polarity
14	R-	SER1	Receive Data, Negative Polarity
16	S2+	SER1	Power Source 1, Positive Polarity
19	T-	SER1	Transmit Data, Negative Polarity
21	S1-	SER1	Power Sink 1, Negative Polarity
24	S2-	SER1	Power Sink 2, Negative Polarity

Termination:

When using the channel SER1 as current loop the terminator switches for RS485 must be switched OFF!

The interface can be connected as either an active or passive current loop depending on the wiring. The transmit line and the receive line are provided with separate 20mA power sources. The compliance voltage is approximately 24 VDC.

The maximum baud rate is 19200Bd. The maximum cable length depends on the baud rate and rate of transmission errors.

For longer cable lengths, the 20mA power supply should be fed by the transmitting unit. This can decrease crosstalk on the signal lines considerably.

In idle state (signal logical 1) a current loop of 20 mA can be measured on the cable.

Signal logical 1	-	Current flow 20mA
Signal logical 0	-	Current flow is interrupted

A shielded cable with twisted pair wires (cable type LiYCY-TP) and a minimum cross section of 0.08 mm² must be used. The maximum cable length is 100 m.



Connect the cable shield to the metal hoods of the connectors over as large a surface as possible!

1.5.3 Pin Assignment X3 SER1 RS485

The interface RS485 is suitable for point-to-point connections and multipoint connections.

Termination for point-to-point connection:

For operation with point-to-point connection the **termination** must always be activated.

Termination for multipoint connection:

For operation with multipoint connections only the **termination** at the cable end must be activated.

The signals of the interface are electrically isolated.
The configuration of the hardware can be adapted to different systems. The associated wires are marked with 'A' and 'B'. Some descriptions refer to the pins with '+' and '-', where the following applies: A = + and B = -. The voltage levels comply with the standards and are defined as follows:

- Signal logical 1 - $U_A - U_B \leq -0.3 \text{ V}$ i.e. ($U_A < U_B$)
- Signal logical 0 - $U_A - U_B \geq +0.3 \text{ V}$ i.e. ($U_A > U_B$)

Pin	Designation	Channel	Function
8	T(A)	SER1	Transmit Data Channel A
9	T(B)	SER1	Transmit Data Channel B
11	SGND	SER1	Signal Ground
22	RD(A)	SER1	Receive Data Channel A
23	RD(B)	SER1	Receive Data Channel B

A shielded cable with twisted pair wires (cable type LiYCY-TP) and a minimum cross section of 0.34 mm² (for 400 m) must be used. The maximum cable length is 400 m.



Connect the cable shield to the metal hoods of the connectors over as large a surface as possible!

1.5.4 Pin Assignment X3 SER1 RS232c

Interface for communication with controller.

Pin	Designation	Channel	Function
1	Shield	SER1	Shield
6	TD	SER1	Transmit Data
15	CTS	SER1	Clear To Send
17	RTS	SER1	Request To Send
18	RD	SER1	Receive Data
25	SGND	SER1	Signal Ground

A shielded cable with stranding in layers (cable type LiYCY) and with a minimum cross-section of 0.25 mm² must be used. The maximum cable length is 15 m.



Connect the cable shield to the metal hoods of the connectors over as large a surface as possible!

1.5.5 Pin Assignment X3 SER2 RS232c

Interface for download, upload and logging printer.

Pin	Designation	Channel	Function
2	TD	SER2	Transmit Data
3	RD	SER2	Receive Data
4	RTS	SER2	Request to Send
5	CTS	SER2	Clear To Send
7	SGND	SER2	Signal Ground
20	DTR	SER2	Data Terminal Ready

A shielded cable with stranding in layers (cable type LiYCY) and with a minimum cross-section of 0.25 mm² must be used. The maximum cable length is 15 m.



Connect the cable shield to the metal hoods of the connectors over as large a surface as possible!

1.5.6 Pin Assignment X4 Parallel Outputs

Open-collector-outputs, which switch the positive potential, are used as parallel outputs. These outputs are suitable for direct control of PLC inputs. The outputs can be activated by means of function keys or the controller. The assignment of these functions is carried out in the programming software. The parallel outputs are designed for use in standard mode only.

Technical Data:	
Input voltage	24VDC in accordance with EN61131-2
Output Current	max. 0.2 A

The outputs are short-circuit-proof!
The voltage supply must be provided externally. Pins 9 through 13 for the negative potential are connected internally.

Connector in the operating terminal: 15-pin SubminD male connector strip

Pin	Designation	Function
1	A1	Output 1
2	A2	Output 2
3	A3	Output 3
4	A4	Output 4
5	A5	Output 5
6	A6	Output 6
7	A7	Output 7
8	A8	Output 8
9	0V	Negative Potential
10	0V	Negative Potential
11	0V	Negative Potential
12	0V	Negative Potential
13	0V	Negative Potential
14	nc	Not Connected
15	+U _E	Positive Potential

1.6 Shield

The shield must be connected to the metal hoods of the connector housings at both ends and over as large a surface as possible. It should be noted that a potential equalization line with a minimum cross-section equal to 10 times that of the shield may be necessary as a result of the grounding on both sides.

1.7 Display

The displays of the operating terminals either consist of backlight LCD modules of different sizes and different amounts of functions.

All displays have an optimum viewing angle of approximately 90°.

Also all displays are capable of displaying the enhanced ASCII character set (semi graphics).

The contrast of the display is stabilized over the full temperature range.

The default contrast can be adjusted at operating time by means of a system variable.



If the display is damaged, do not swallow or breathe in the liquids or gases being emitted and avoid direct contact with skin.

Danger of Poisoning! Could Result in Burns!

1.7.1 NT4S-SF121B-E, NT4S-SF122B-E, NT4S-SF123B-E

Overview of the technical data:

Type :	LCD-Module
Graphics Capability :	Semi Graphics Capability
Backlight :	LED-Backlight
Background Colour :	Yellow-green
Lines (Font Normal) :	4
Characters/Line (Font Normal) :	20
Character Matrix :	5 x 7 Dots + Cursor
Character Height (Font Normal) :	4.3 mm
Visible Display Size :	74.0 mm x 23.0 mm

1.7.2 NT15S-SF121B-E

Overview of the technical data:

Type :	LCD-Module
Resolution :	240 x 64 Dots
Graphics Capability :	Full Graphics Capability
Backlight :	LED-Backlight
Background Colour :	Yellow-green
Lines (Font Normal) :	8
Characters/Line (Font Normal) :	40
Character Matrix (Font Normal) :	6 x 8 Dots
Character Matrix (Font Zoom) :	12 x 16 Dots
Dot Colour :	Black
Dot Size :	0.49 mm x 0.49 mm
Dot Gap Size :	0.04 mm
Visible Display Size :	134.0 mm x 40.4 mm

1.7.3 NT18S-SF121B-E

Overview of the technical data:

Type :	LCD-Module
Resolution:	240 x 128 Dots
Graphics Capability:	Full Graphics Capability
Backlight :	LED-Backlight
Background Colour :	White
Lines (Font Normal) :	16
Characters/Line (Font Normal) :	40
Character Matrix (Font Normal) :	6 x 8 Dots
Character Matrix (Font Zoom) :	12 x 16 Dots
Dot Colour :	Black
Dot Size :	0.49 mm x 0.49 mm
Dot Gap Size :	0.04 mm
Visible Display Size :	131.0 mm x 72.0 mm

1.7.4 Contrast Setting

The contrast for the display can be adjusted by means of the software. This requires the system variable **LCDCONTRAST** to be set up in an I/O mask of the application. The value can then be modified using any editor that can handle integer numbers.

The limit values for the contrast must be set to

Lower level: -25
Upper level: +70

If this variable is not defined in the menus or the value is out of the range of values, the default setting (value 25) will be loaded when the system is initialized.

The system variable can be stated in any I/O-mask of the application!

1.7.5 Default Contrast Setting

If the contrast of the display should be such that the masks are no longer legible, the default contrast setting can be restored using the user mode switch.

Position of the switch to restore the contrast:

S1 ON
S2 OFF
S3 OFF
S4 ON

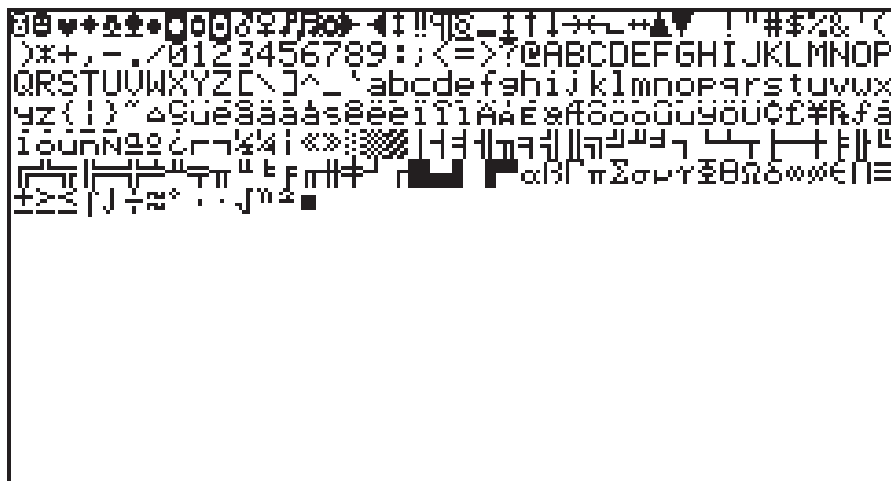
This switch position coincides with 'activating download by hardware'. The contrast will be reset before the warning is displayed. The warning will be displayed in a legible manner. Upon display of this warning, switch off the terminal, set the switch 4 to the OFF-position and switch the terminal on again. The application description is not lost.

1.7.6 Character Attributes

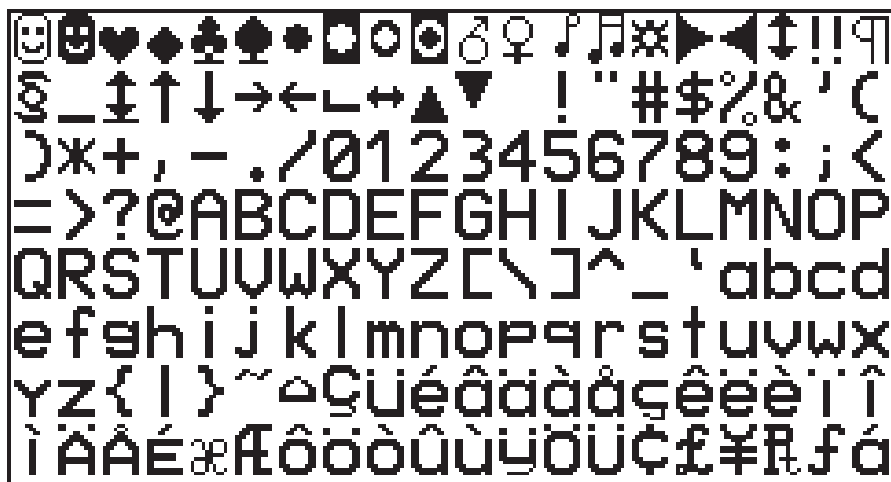
The characters of the fonts **Normal** and **Zoom** can be displayed with different attributes.

Operating Terminal	Normal	Flashing	Underlined	Inverse
NT4S-SF121B-E	Yes	Yes	No	No
NT4S-SF122B-E	Yes	Yes	No	No
NT4S-SF123B-E	Yes	Yes	No	No
NT15S-SF121B-E	Yes	Yes	Yes	Yes
NT18S-SF121B-E	Yes	Yes	Yes	Yes

1.7.6.1 Font Normal



1.7.6.2 Font Zoom



1.7.6.3 ASCII Character Set Table

000		032	!	064	@	096	`	128	Ç	160	Á	192	Ł	224	α
001	☺	033	"	065	A	097	a	129	Ü	161	Í	193	ł	225	β
002	☹	034	"	066	B	098	b	130	É	162	Ó	194	Ł	226	Γ
003	♥	035	#	067	C	099	c	131	Â	163	Ú	195	ł	227	Π
004	♦	036	\$	068	D	100	d	132	Ä	164	Ń	196	Ł	228	Σ
005	♣	037	%	069	E	101	e	133	À	165	Ñ	197	ł	229	σ
006	♠	038	&	070	F	102	f	134	Å	166	Œ	198	Ł	230	μ
007	•	039	'	071	G	103	g	135	Ç	167	Ž	199	Ł	231	τ
008	■	040	<	072	H	104	h	136	Ê	168	ı	200	Ł	232	ø
009	○	041	>	073	I	105	i	137	Ë	169	ı	201	Ł	233	θ
010	□	042	*	074	J	106	j	138	È	170	ı	202	Ł	234	Ω
011	♂	043	+	075	K	107	k	139	İ	171	½	203	Ł	235	δ
012	♀	044	,	076	L	108	l	140	Î	172	¾	204	Ł	236	ω
013	♪	045	-	077	M	109	m	141	Ï	173	ı	205	Ł	237	ø
014	♫	046	.	078	N	110	n	142	Ä	174	«	206	Ł	238	€
015	✳	047	/	079	O	111	o	143	Å	175	»	207	Ł	239	π
016	►	048	0	080	P	112	p	144	É	176	///	208	Ł	240	≡
017	◄	049	1	081	Q	113	q	145	Æ	177	///	209	Ł	241	±
018	†	050	2	082	R	114	r	146	Œ	178	///	210	Ł	242	≥
019	‡	051	3	083	S	115	s	147	Ô	179		211	Ł	243	≤
020	¶	052	4	084	T	116	t	148	Ö	180		212	Ł	244	ƒ
021	§	053	5	085	U	117	u	149	Ø	181		213	Ł	245	J
022	■	054	6	086	V	118	v	150	Ù	182		214	Ł	246	÷
023	±	055	7	087	W	119	w	151	Ù	183	π	215	Ł	247	≈
024	↑	056	8	088	X	120	x	152	ÿ	184	ı	216	Ł	248	°
025	↓	057	9	089	Y	121	y	153	Û	185	ı	217	Ł	249	·
026	→	058	:	090	Z	122	z	154	Ü	186	ı	218	Ł	250	√
027	←	059	;	091	[	123	[	155	É	187	ı	219	Ł	251	√
028	└	060	<	092	\	124	]	156	Ê	188	ı	220	Ł	252	n
029	↔	061	=	093	]	125	]	157	ÿ	189	ı	221	Ł	253	z
030	▲	062	>	094	^	126	~	158	Ŕ	190	ı	222	Ł	254	
031	▼	063	?	095	_	127	Δ	159	Ŕ	191	ı	223	Ł	255	

1.8 User-Mode Switch

The user-Mode switch is placed at the side of the terminal NT4S-SF122B-E and at the rear side of all other terminals.

User-Mode Switch

4 Switches

NT4S-SF121B-E

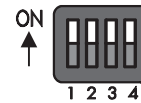
NT4S-SF122B-E

8 Switches

NT4S-SF123B-E

NT15S-SF121B-E

NT18S-SF121B-E



The switches S5 to S8 can be used by the operator as needed. The switch positions are stored at initialization time and afterwards they can be overtaken to the controller.

S1	S2	S3	S4	S5	S6	S7	S8	Function
I	X	-	-	X	X	X	X	Standard-Mode with PLC (delivery state)
I	X	I	-	X	X	X	X	Standard-Mode without PLC
-	I	-	-	X	X	X	X	Transparent-Mode with start and stop code of the keys
-	-	-	I	X	X	X	X	Transparent-Mode without stop code of the keys
I	-	-	I	X	X	X	X	Activate download (deletes application memory) and default contrast setting

Table Legend:

I = Switch position ON

- = Switch position OFF

X = Switch position irrelevant

1.9 Battery

A built-in lithium battery buffers the data in the RAM memory and also supplies the real-time clock with power. The discharge degree of the battery is monitored constantly to prevent any loss of data.

The battery provides a minimum life of 5 years, even under unfavourable operating conditions.

If the battery is drained the system message 'change battery' is generated.

A new battery is supported by OMRON EUROPE.

Replacing the battery:

The battery can be replaced while the operating voltage is connected to ensure that the message data and time setting are not lost. Mind the safety instructions!

- Remove the mounting bolts of the 25-pin interface connector
- Remove the fastening screws of the enclosure and remove the enclosure
- Replace the cable fastener, which is used to fasten the battery
- Plug off the cable of the battery and replace the discharged battery
- Plug on the cable of the new battery
- Fasten the new battery on the plastic holder on the printed circuit board

OBSERVE THE CORRECT POLARITY OF (-) AND (+)

- Place the enclosure on the rear side of the terminal
- At first fasten the bolts of the interface connector and at last fasten the screws of the enclosure properly

Changing the battery may only be performed by qualified and authorized personnel!

Sewage and refuse disposal:

Dispose only drained batteries into the collection box of the community or of the local dealer. The battery is stated as drained when the message 'change battery' appears on the display of the terminal.

To prevent short circuitry in the collection boxes insulate the poles of each battery with insulation tape or put each single battery into a plastic bag.



Do not put lithium batteries in fire or heat them above 100 °C and do not recharge them. **Danger of Explosion!**



Do not open lithium batteries. **Danger of Poisoning!**



Hazardous voltages can exist inside electrical installations that can pose a danger to humans. Coming in contact with live parts may result in **electric shock!**



Electrostatic discharges can damage electronic components! ESD protective measures must be observed!

1.10 Fuse

A semiconductor fuse is used to prevent damage to the operating terminal. Once the fuse has been activated, the device must be disconnected from the supply voltage to allow the semiconductor fuse to regenerate. With an ambient temperature of 20 °C, the regeneration takes about 20 seconds. The higher the ambient temperature, the longer the regeneration period.

The semiconductor fuse is not designed to be replaced.



Hazardous voltages can exist inside electrical installations that can pose a danger to humans. Coming in contact with live parts may result in **electric shock!**



Electrostatic discharges can damage electronic components! ESD protective measures must be observed!

1.11 Application Memory

A 256 kByte flash memory is used as an application memory for each operating terminal. This memory area is available to store the user application, the loadable protocol driver, the fonts and the recipe data. The advantage of the flash memory is that programming and deleting processes can be carried out directly in the terminal.

Changing the application memory may only be performed by qualified and authorized personnel!



Hazardous voltages can exist inside electrical installations that can pose a danger to humans. Coming in contact with live parts may result in **electric shock!**



Electrostatic discharges can damage electronic components! ESD protective measures must be observed!

SECTION 2
Technical Data

This section lists all the electrical and mechanical specifications of the terminals.

2.1	NT4S-SF121B-E	47
2.2	NT4S-SF122B-E	48
2.3	NT4S-SF123B-E	50
2.4	NT15S-SF121B-E	51
2.5	NT18S-SF121B-E	53

2 Technical Data

2.1 NT4S-SF121B-E

Keyboard	a Total of 11 Keys, Mechanical with Tactile Feedback	
	Divided into	
	2	Control Keys
	4	Function Keys with LEDs and Slide-in Identification Strips
	1	Special Key without LED
	2	Special Keys with LEDs
Display	2	Editing Keys
	Backlit LCD Module, 4 Lines with 20 Characters Each,	
	Display Area 23 × 74 mm (H × W)	
	with Glare Suppression for Increased Contrast	
Interface X3	Variable Baud Rates and Data Formats	
	600 to 19200 Bd	
	SER1 RS485, Electrically Isolated	Communication
	SER1 TTY / 20 mA, Electrically Isolated	Communication
	SER1 RS232c, Electrically Isolated	Communication
	SER2 RS232c, Not Electrically Isolated	Download/Logging Printer
Central Unit	Z80-CPU, 10 MHz, Watchdog Timer, Real-Time Clock, Programmable Interface	
	Parameters, Temperature Compensation of the Display, Adjustment of Contrast,	
	Battery Monitoring, User Mode Switch	
Memory	256 Kbyte Flash Memory, Application Memory	
	256 kByte Flash Memory, Firmware	
	128 Kbyte stat. CMOS-RAM, Battery-Backed	
Fuse	Semiconductor Fuse	
Connection System	Plug-in Type, via SubminD Female Connector Strip	
Supply Voltage	24 V Direct Voltage, Residual Ripple Max. 10%	
	Minimum Voltage	19.2 V
	Maximum Voltage	30.2 V
	Typ. Power Consumption	<0.3 A
	Peak Current (10 ms)	<0.5 A
Connected Load	~ 10 W	

Noise Immunity	EC Electromagnetic Compatibility Directive 89/336/EEC EN 55011 Limit Class B EN 50081-1 Table A1 EN 50082-2 EN 61000-4-2 EN 61000-4-3 EN 61000-4-4 EN 61000-4-5 EN 61000-4-6		
Environmental Test	DIN40040 Code Letter - Temperature Code Letter - Humidity	Operation KW F	Storage HU F
Degrees of Protection	DIN 40050 Mechanical Degrees of Protection Front: IP65 Rear: IP20		
Front Panel	Aluminum, Black Anodized with Affixed Polyester Cover, Circumferential Polyurethane Foam Seal at Rear Side of Front Panel 96.0 × 144.0 × 3.5 mm (H × W × D)		
Panel Cutout	82 × 138 mm (H × W)		
Mounting Depth	43 mm without Connector		
Enclosure	Zinc-Coated Steel Plate		
Total Weight	Approx. 400 g		

2.2 NT4S-SF122B-E

Keyboard	a Total of 30 Keys, Membrane with Tactile Feedback Divided into 6 Control Keys 6 Function Keys with LEDs and Slide-in Identification Strips 2 Special Keys with LEDs 3 Special Keys without LEDs 13 Editing Keys		
Display	Backlit LCD Module, 4 Lines with 20 Characters Each, Display Area 23 × 74 mm (H × W) with Glare Suppression for Increased Contrast		
Interface X3	Variable Baud Rates and Data Formats 600 to 19200 Bd SER1 RS485, Electrically Isolated Communication SER1 TTY / 20 mA, Electrically Isolated Communication SER1 RS232c, Electrically Isolated Communication SER2 RS232c, Not Electrically Isolated Download/Logging Printer		

Central Unit	Z80-CPU, 10 MHz, Watchdog Timer, Real-Time Clock, Programmable Interface Parameters, Temperature Compensation of the Display, Adjustment of Contrast, Battery Monitoring, User Mode Switch		
Memory	256 Kbyte Flash Memory, Application Memory 256 kByte Flash Memory, Firmware 128 Kbyte stat. CMOS-RAM, Battery-Backed		
Fuse	Semiconductor Fuse		
Connection System	Plug-in Type, via SubminD Female Connector Strip		
Supply Voltage	24 V Direct Voltage, Residual Ripple Max. 10% Minimum Voltage 19.2 V Maximum Voltage 30.2 V Typ. Power Consumption <0.3 A Peak Current (10 ms) <0.5 A		
Connected Load	~ 10 W		
Noise Immunity	EC Electromagnetic Compatibility Directive 89/336/EEC EN 55011 Limit Class B EN 50081-1 Table A1 EN 50082-2 EN 61000-4-2 EN 61000-4-3 EN 61000-4-4 EN 61000-4-5 EN 61000-4-6		
Environmental Test	DIN40040 Code Letter - Temperature Code Letter - Humidity	Operation KW F	Storage HU F
Degrees of Protection	DIN 40050 Mechanical Degrees of Protection Front: IP65 Rear: IP20		
Front Panel	Aluminum, Black Anodized with Affixed Polyester Cover, Circumferential Rubber Sealing around rear of front panel 168.0 × 120.0 × 4.0 mm (H × W × D)		
Panel Cutout	160 × 112 mm (H × W)		
Mounting Depth	40 mm without Connector		
Enclosure	Zinc-Coated Steel Plate		
Total Weight	500 g		

2.3 NT4S-SF123B-E

Keyboard	a Total of 30 Keys, Mechanical with Tactile Feedback Divided into 5 Control Keys 8 Function Keys with LEDs and Slide-in Identification Strips 2 Special Key without LED 2 Special Keys with LEDs 13 Editing Keys	
Display	Backlit LCD Module, 4 Lines with 20 Characters Each, Display Area 23 × 74 mm (H × W) with Glare Suppression for Increased Contrast	
Interface X3	Variable Baud Rates and Data Formats 600 to 19200 Bd - SER1 RS485, Electrically Isolated Communication - SER1 TTY / 20 mA, Electrically Isolated Communication - SER1 RS232c, Electrically Isolated Communication - SER2 RS232c, Not Electrically Isolated Download/Logging Printer	
Central Unit	Z80-CPU, 10 MHz, Watchdog Timer, Real-Time Clock, Programmable Interface Parameters, Temperature Compensation of the Display, Adjustment of Contrast, Battery Monitoring, User Mode Switch	
Memory	256 Kbyte Flash Memory, Application Memory 256 kByte Flash Memory, Firmware 128 Kbyte stat. CMOS-RAM, Battery-Backed	
Fuse	Semiconductor Fuse	
Connection System	Plug-in Type, via SubminD Female Connector Strip	
Supply Voltage	24 V Direct Voltage, Residual Ripple Max. 10% - Minimum Voltage 19.2 V - Maximum Voltage 30.2 V - Typ. Power Consumption <0.35 A - Peak Current (10 ms) <0.5 A	
Connected Load	~ 10 W	
Noise Immunity	EC Electromagnetic Compatibility Directive 89/336/EEC EN 55011 Limit Class B EN 50081-1 Table A1 EN 50082-2 EN 61000-4-2 EN 61000-4-3 EN 61000-4-4 EN 61000-4-5 EN 61000-4-6	

Environmental Test	DIN40040	Operation	Storage
	Code Letter - Temperature	KW	HU
	Code Letter - Humidity	F	F
Degrees of Protection	DIN 40050 Mechanical Degrees of Protection Front: IP65 Rear: IP20		
Front Panel	Aluminum, Black Anodized with Affixed Polyester Cover, Circumferential Sealing Around Rear of Front Panel 150.0 × 270.0 × 3.0 mm (H × W × D)		
Panel Cutout	112 × 232 mm (H × W)		
Mounting Depth	48 mm without Connector		
Enclosure	Zinc-Coated Steel Plate		
Total Weight	approx. 800 g		

2.4 NT15S-SF121B-E

Keyboard	a Total of 42 Keys, Mechanical with Tactile Feedback Divided into		
	7 Control Keys		
	16 Function Keys with LEDs and Slide-in Identification Strips		
	2 Special Key without LED		
	4 Special Keys with LEDs		
	13 Editing Keys		
Display	Backlit LCD Module, 8 Lines with 40 Characters Each, Display Area 40.4 × 134.0 mm (H × W) with Glare Suppression for Increased Contrast		
Interface X3	Variable Baud Rates and Data Formats 600 to 19200 Bd		
	SER1 RS485, Electrically Isolated	Communication	
	SER1 TTY / 20 mA, Electrically Isolated	Communication	
	SER1 RS232c, Electrically Isolated	Communication	
	SER2 RS232c, Not Electrically Isolated	Download/Logging Printer	
Central Unit	Z80-CPU, 10 MHz, Watchdog Timer, Real-Time Clock, Programmable Interface Parameters, Temperature Compensation of the Display, Adjustment of Contrast, Battery Monitoring, User Mode Switch		
Memory	256 Kbyte Flash Memory, Application Memory		
	256 kByte Flash Memory, Firmware		
	128 Kbyte stat. CMOS-RAM, Battery-Backed		

Fuse	Semiconductor Fuse		
Connection System	Plug-in Type, via SubminD Female Connector Strip		
Supply Voltage	24 V Direct Voltage, Residual Ripple Max. 10%		
	Minimum Voltage	19.2 V	
	Maximum Voltage	30.2 V	
	Typ. Power Consumption	<0.4 A	
	Peak Current (10 ms)	<0.6 A	
Connected Load	~ 10 W		
Noise Immunity	EC Electromagnetic Compatibility Directive 89/336/EEC EN 55011 Limit Class B EN 50081-1 Table A1 EN 50082-2 EN 61000-4-2 EN 61000-4-3 EN 61000-4-4 EN 61000-4-5 EN 61000-4-6		
Environmental Test	DIN40040	Operation	Storage
	Code Letter - Temperature	KW	HU
	Code Letter - Humidity	F	F
Degrees of Protection	DIN 40050 Mechanical Degrees of Protection Front: IP65 Rear: IP20		
Front Panel	Aluminum, Black Anodized with Affixed Polyester Cover, Circumferetial Rubber Sealing at Rear Side of Front Panel 230.0 × 250.0 × 4.0 mm (H × W × D)		
Panel Cutout	188 × 204 mm (H × W)		
Mounting Depth	50 mm without Connector		
Enclosure	Zinc-Coated Steel Plate		
Total Weight	Approx. 1200 g		

2.5 NT18S-SF121B-E

Keyboard	a Total of 34 Keys, Mechanical with Tactile Feedback Divided into 5 Control Keys 12 Function Keys with LEDs and Slide-in Identification Strips 2 Special Key without LED 2 Special Keys with LEDs 13 Editing Keys	
Display	Backlit LCD Module, 16 Lines with 40 Characters Each, Display Area 72 × 131 mm (H × W) with Glare Suppression for Increased Contrast	
Interface X3	Variable Baud Rates and Data Formats 600 to 19200 Bd - SER1 RS485, Electrically Isolated Communication - SER1 TTY / 20 mA, Electrically Isolated Communication - SER1 RS232c, Electrically Isolated Communication - SER2 RS232c, Not Electrically Isolated Download/Logging Printer	
Central Unit	Z80-CPU, 10 MHz, Watchdog Timer, Real-Time Clock, Programmable Interface Parameters, Temperature Compensation of the Display, Adjustment of Contrast, Battery Monitoring, User Mode Switch	
Memory	256 Kbyte Flash Memory, Application Memory 256 kByte Flash Memory, Firmware 128 Kbyte stat. CMOS-RAM, Battery-Backed	
Fuse	Semiconductor Fuse	
Connection System	Plug-in Type, via SubminD Female Connector Strip	
Supply Voltage	24 V Direct Voltage, Residual Ripple Max. 10% - Minimum Voltage 19.2 V - Maximum Voltage 30.2 V - Typ. Power Consumption <0.7 A - Peak Current (10 ms) <3.0 A	
Connected Load	~ 20 W	
Noise Immunity	EC Electromagnetic Compatibility Directive 89/336/EEC EN 55011 Limit Class B EN 50081-1 Table A1 EN 50082-2 EN 61000-4-2 EN 61000-4-3 EN 61000-4-4 EN 61000-4-5 EN 61000-4-6	

Environmental Test	DIN40040	Operation	Storage
	Code Letter - Temperature	KW	HU
	Code Letter - Humidity	F	F
Degrees of Protection	DIN 40050 Mechanical Degrees of Protection		
	Front: IP65		
	Rear: IP20		
Front Panel	Aluminum, Black Anodized with Affixed Polyester Cover, Circumferential Sealing Around Rear of Front Panel		
	190.0 × 330.0 × 3.0 mm (H × W × D)		
Panel Cutout	152 x 292 mm (H x W)		
Mounting Depth	60 mm without Connector		
Enclosure	Zinc-Coated Steel Plate		
Total Weight	approx. 1500 g		

SECTION 3

Operating Modes

This section describes the two available operation modes, the standard mode and the transparant mode and indicates how to select the operation mode.

3.1	Setting the Operating Mode	57
-----	----------------------------------	----

3 Operating Modes

There are two types of operating modes available in the operating terminals, the standard mode of operation and the transparent mode of operation.

Terminals operating in the transparent mode represent full-size ANSI-terminals. Each key generates a press and release code, which will be transmitted via the interface SER1 in the form of an ASCII character. The displays and the key LEDs are controlled via ESC sequences. The number of character sets and character attributes varies with the type of display. For a detailed description please refer to the chapter 'Transparent Mode'.

In the standard mode of operation, the entire operating system is integrated in the terminal. The standardized operating concept completely frees the connected controller from any operator guidance tasks as well as data display. In standard mode, a decoding of the keys or selection of masks from within the controller is not required.

3.1 Setting the Operating Mode

The operating mode can be set by means of the User-Mode-Switch. The terminals are factory-set to the standard mode of operation.

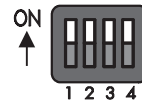
The user-mode switch is accessible at the side of the terminal NT4S-SF122B-E and at the rear side of all other operating terminals.

User-Mode Switch

4 Switches

NT4S-SF121B-E

NT4S-SF122B-E



8 Switches

NT4S-SF123B-E

NT15S-SF121B-E

NT18S-SF121B-E



The switches S5 to S8 can be used by the operator as needed. The switch positions are stored at initialization time and afterwards they can be overtaken to the controller.

S1	S2	S3	S4	S5	S6	S7	S8	Function
I	X	-	-	X	X	X	X	Standard-Mode with PLC (delivery state)
I	X	I	-	X	X	X	X	Standard-Mode without PLC
-	I	-	-	X	X	X	X	Transparent-Mode with start and stop code of the keys
-	-	-	I	X	X	X	X	Transparent-Mode without stop code of the keys
I	-	-	I	X	X	X	X	Activate download (deletes application memory) and default contrast setting

Table Legend:

I = Switch position ON

- = Switch position OFF

X = Switch position irrelevant

SECTION 4

Standard Mode

This section describes the standard operation mode in detail. This section contains all detailed information to understand all features of the terminal in order to make a terminal application.

4.1	Setting the Operating Mode	63
4.2	Startup Process.....	64
4.2.1	Startup Process without a Valid User Description	65
4.3	Communication in the Standard Mode	65
4.4	Operating Concept	65
4.4.1	Mask Structure	65
4.4.2	External Mask Selection	66
4.4.3	Password Protection, Access Authorization	66
4.4.3.1	Reactivating the Password Protection.....	69
4.4.3.2	Password Management.....	69
4.4.3.3	Password Mask and Password Functionality.....	69
4.5	Masks	70
4.5.1	Mask Parameters	70
4.5.2	System Masks	70
4.5.2.1	Setup Mask.....	71
4.5.2.1.1	Password Protection - Setup Mask	71
4.5.2.1.2	Function Without the Setup Mask	71
4.5.2.2	Start Mask.....	71
4.5.2.3	Password Mask	72
4.5.2.4	I/O Mask	72
4.6	Variables	72
4.6.1	Output Variables	73
4.6.1.1	'Decimal Number' Representation.....	76
4.6.1.1.1	'Standard' Variable Type	76
4.6.1.1.2	'Timer' Variable Type	76
4.6.1.1.3	'Counter' Variable Type	77
4.6.1.1.4	'BCD-Number' Variable Type	78
4.6.1.2	'Alphanumeric' Representation	78
4.6.1.3	'Selection Text' Representation	78
4.6.1.4	'Selection Image' Representation	79
4.6.1.5	'Floating Point Number' Representation	80
4.6.1.6	'Hexadecimal Number' Representation	80
4.6.1.7	'Binary Number' Representation	81
4.6.1.8	Bar Representation	81
4.6.1.9	Curve Representation (Trendline).....	82
4.6.2	Input Variables	83
4.6.3	System Variables	85
4.6.3.1	Basic Functions.....	85
4.6.3.2	Communication SER1	87
4.6.3.3	Error Statistics SER1	89
4.6.3.4	Communication SER2	90
4.6.3.5	Real-Time Clock	92
4.6.3.6	Serial Message System	93

4.6.3.7	Parallel Message System	96
4.6.3.8	Printer Control	98
4.6.3.9	Menu Control / Keys.....	99
4.6.3.10	Password	106
4.6.3.11	Recipes	107
4.6.3.12	Running Time Meter.....	112
4.6.3.13	Loop-Through Operation	112
4.6.3.14	Loadable Font	113
4.6.3.15	Maintenance.....	113
4.6.3.16	Editors	114
4.6.3.17	Help	114
4.6.4	Editors	115
4.6.4.1	Decimal Number Editor	116
4.6.4.2	Floating Point Number Editor.....	117
4.6.4.3	Hexadecimal Editor.....	118
4.6.4.4	Alphanumerical Editor.....	118
4.6.4.5	Selection Text Editor	119
4.6.4.6	Selection Image Editor	119
4.6.4.7	Table Editor.....	119
4.6.5	External Data Release	121
4.6.6	PLC-Handshake	123
4.6.7	Refreshing One-Time Output Data.....	123
4.6.8	Modified Data	124
4.6.8.1	Input Plausibility Check.....	124
4.7	Graphics.....	124
4.7.1	Images	124
4.7.2	Graphics on Operating Terminals.....	125
4.7.2.1	Background Images.....	125
4.8	Recipes.....	126
4.8.1	Structure of a Recipe	127
4.8.2	Processing Recipes and Data Sets.....	127
4.8.2.1	Selecting a Recipe	128
4.8.2.2	Selecting a Data Set.....	128
4.8.2.3	Copying a Data Set	128
4.8.2.4	Deleting a Data Set	129
4.8.2.5	Modifying a Data Set	129
4.8.3	Data Set Transfer to / from a Controller	129
4.8.3.1	Transfer to a Controller.....	129
4.8.3.2	Transfer from a Controller	131
4.8.4	Transferring Data Sets to / from a PC.....	132
4.8.4.1	Transfer to a PC	132
4.8.4.2	Transfer from a PC	133
4.8.4.3	Structure of a Data Set File	133
4.8.5	Printing Data Sets	135
4.8.6	Memory Requirements for Storing Data Sets.....	136
4.9	Message System.....	136
4.9.1	Internal Messages.....	136
4.9.1.1	System Messages.....	136
4.9.1.1.1	Suppressing the Display of System Messages	140
4.9.1.2	Error Messages	140

4.9.2	External Messages	146
4.9.2.1	Structure of an External Message	146
4.9.2.1.1	Assigning Message Numbers	146
4.9.2.1.2	Message Buffer Size	147
4.9.2.1.3	Message Texts, Variables	147
4.9.2.1.4	Sorting Messages	148
4.9.2.1.5	Message Priority for Direct Display	148
4.9.2.1.6	Printing the Message Memory	148
4.9.2.2	Message Mask, Status Message Mask	149
4.9.2.2.1	Direct Selection of the Message Mask	150
4.9.2.2.2	Output Formats for Messages	150
4.9.2.2.3	Zooming Messages	151
4.9.2.2.4	Acknowledging Messages	151
4.9.2.3	Serial Message System	151
4.9.2.3.1	Full-Page Message Output	152
4.9.2.3.2	Messages Directly to a Logging Printer	152
4.9.2.3.3	Erasing the Message Memory Externally	152
4.9.2.3.4	Information about the Serial Message System	153
4.9.2.4	Parallel Message System (Status Messages)	154
4.9.2.4.1	Number of Bytes for Status Messages	154
4.9.2.4.2	Image of the Status Messages	154
4.9.2.4.3	Time-Controlled Transfer of the Status Message	155
4.9.2.4.4	Event-Controlled Transfer of the Status Message	155
4.10	Help System	155
4.10.1	Default Help Text	155
4.10.2	Help Text For Masks	156
4.10.2.1	Help Text for the Message Mask	156
4.10.3	Help Text For Variables	156
4.11	Function Keys	156
4.11.1	Direct Selector Keys	156
4.11.2	Function Keys of the Controller	157
4.11.3	Soft Keys	157
4.11.3.1	Reaction Time of Function and Soft Keys	158
4.11.3.2	Control Keys as Function Keys	158
4.11.4	Function Keys Controlling Parallel Outputs	158
4.11.5	Status LEDs in the Function Keys	158
4.12	System Parameters	159
4.12.1	System Parameters: General Parameters	159
4.12.2	System Parameters: Poll Area	159
4.12.3	System Parameters: Terminal Clock	160
4.12.4	System Parameters: Running Time Meter	160
4.12.5	System Parameters: Message System	160
4.12.6	System Parameters: Variant Buffer	160
4.12.7	System Parameters: Password Management	160
4.12.8	System Parameters: Printer Interface	160
4.12.9	System Parameters: Gateway	161
4.12.10	System Parameters: Data Set Transfer	161
4.12.11	System Parameters: Parallel Outputs	161
4.13	Version Number	161
4.14	Running Time Meter	161
4.15	Parallel Outputs	163
4.16	Screen Saver	163

4.17	Image of the Mask Number	164
4.18	Image of the Mode Selector Switch	164
4.19	Terminal Clock	164
4.19.1	Image of Date and Time	165
4.20	Read Coordination Byte	166
4.20.1	Editing Request Bit (Bit 'EA')	166
4.20.2	Editing Status Bit (Bit 'EZ')	166
4.20.3	Refresh Request Bit (Bit 'RA')	167
4.20.4	Liveness Flag (Bit 'LM')	167
4.20.5	Data Set Download Active (Bit 'DDA')	167
4.21	Write Coordination Byte	168
4.21.1	External Data Release (Bit 'ED')	168
4.21.2	Refresh Acknowledgement (Bit 'RQ')	168
4.21.3	Resetting the Password.	168
4.21.4	Liveness Flag (Bit 'LM')	168
4.21.5	Data Set Download Release (Bit 'DDF')	169
4.22	Cyclic Poll Area	169
4.22.1	Byte-Oriented	170
4.22.2	Word-Oriented	171
4.22.3	Image of the LEDs	172
4.22.4	Serial Message Channel	172
4.22.5	Polling Time	173
4.22.6	Size of the Poll Area	173
4.23	Control Codes	173
4.23.1	Triggering Data Set Printouts	173
4.23.2	Setting the Clock in the Operating Terminal	173
4.23.3	Transferring Data Sets from the Controller to the Terminal	174
4.23.4	Transferring Data Sets from the Terminal to the Controller	174
4.23.5	Transferring Data Sets from the Controller to the Terminal (Individually)	174
4.23.6	Refreshing the Message System	175
4.24	Cyclic Variables	175
4.25	Interface Parameters	175
4.26	Variable Definition	175
4.26.1	Variable List	176
4.27	Application Programming	176
4.27.1	Configuring the System	177
4.27.2	Programming Systems	178
4.27.3	Project Description	178
4.27.3.1	Multilingual Projects	178
4.27.3.2	Variants of a Project	179
4.27.4	Project Documentation	179
4.27.4.1	Creating Documentation	179
4.27.5	Project Back-up	180
4.27.6	Optimizing the Transmission Rate	180
4.28	Downloading the User Description	180
4.28.1	Downloading with Windows	182
4.28.2	Application Memory	182
4.28.3	Loading an Application	182
4.28.4	Activating the Download Function using the Software	183
4.28.5	Activating the Download Function using the Hardware	183
4.28.6	Automatic Download Function	184
4.28.7	Download Cable	184
4.29	Simulation without the Controller	184

4 Standard Mode

The standard mode is the mode of operation in which the terminals are most commonly used and that provides the highest performance. In this mode of operation, the terminals act as intelligent peripheral devices for the controller by performing controlled pre-processing of the visualization data, thus completely relieving the connected controller of all operating tasks. This reduces the programming effort required by the user, the run time and memory required in the PLC.

Communication processors have been dispensed with in the PLC and programming unit interfaces have been used throughout, making the PLC connection very economical. The operating concept is a universal one, thanks to the large number of supported protocols. Operation of the machine has been made independent of the type and manufacturer of the controller.

The standard mode is supported by every operating terminal. The system's flexibility means you can customize all menus and dialogs according to your particular application. Extremely sophisticated operator guidances can be configured with the aid of masks, system variables and control and function keys. In standard mode, all functions which can be executed with the terminal, mask contents, texts, messages and variables are stored in the user description (mask definition). After the programming phase is complete, the user description is stored in the operating terminal's Flash memory.

In the description which follows, the term '*user*' refers to individuals who configure or program the operator interface, while '*operator*' refers to individuals who monitor and operate the data of the installation on site.

4.1 Setting the Operating Mode

The standard mode of operation can be set with the user-mode switch. Turn off the terminal before selecting the mode of operation.

All terminals are factory-set to the standard mode.
The ON/OFF positions on the mode selector switch are marked.
The selected mode of operation becomes active as soon as power is supplied.
Position of the switch for the standard mode:

S1	ON	S5	not used
S2	not used	S6	not used
S3	OFF	S7	not used
S4	OFF	S8	not used



Fig. 1: User-mode switch

4.2 Startup Process

All LEDs in the terminal are initially lit when the power is applied. A system-test is then performed that includes testing and initialization of the modules in the operating terminal. Various system and error messages may be output during this system-test. If the application memory contains a valid user description, the first mask, the Start-Mask, or the mask specified in the language parameters as the mask to be used as the Start-Mask will appear on the display.

This mask is displayed for 5 seconds (fixed time setting). This time can be used by the operator to visually inspect the LEDs and the screen for proper functioning. After this time period, the Main Mask or the mask specified in the language parameters as the Main Mask is displayed. This mask is the first mask of the operator guidance.

If the **Enter key** is pressed while the Startup Mask is displayed, the Setup Mask will appear on the display. This Setup Mask can be used to set the parameters for the interface and the operating terminal.

If the Enter key is pressed before the Startup Mask is displayed, an error message will be generated during the keyboard test.



.....The self-test performed on the keyboard during the terminal start-up has detected that a key is pressed. Comply with the request and release the key. If this message is generated when no key is pressed, this indicates a defective keyboard. Generally, the message will disappear after releasing the key/keys.

A longer delay may occur before the Startup Mask is displayed if the statistics memory contains a large number of messages from the serial message system. This time period (initialization period) is required to set up the structures for message management, which will then enable faster sorting of messages later. During the initialization phase, the following message is displayed:



.....When the terminal is switched on, the messages that are present in the terminal are sorted. This process requires a certain length of time (initialization time).

4.2.1 Startup Process without a Valid User Description

The system-test performed on the application memory will detect whether a module is missing, defective or of the wrong type. This is then indicated to the operator by the following message:

```
NO FLASH EPROM
```

If the memory test establishes that the right type of Flash memory is used but that it does not contain a valid user description, the Flash memory will be erased and the terminal will automatically switch to the download operating mode. The following messages appear on the display indicating the various phases:

```
ERASE FLASH EPROM
```

```
FLASH IS ERASED  
FLASH xxx kByte  
HFxxxxxx
```

```
DOWNLOAD 1  
FLASH xxx kByte
```

The message 'DOWNLOAD 1' remains on the display to indicate that the terminal is now ready to receive a valid user description via the download interface.

Until the memory contains a valid user description, communication with a controller will not take place, nor will the keyboard be operational. Once a valid user description has been downloaded, the terminal becomes active.

4.3 Communication in the Standard Mode

In standard mode, any interface except the interfaces for the logging printer and the parallel outputs can be used for communication between the PLC (host computer, etc.) and the operating terminal. The interface labeling, however, always depends on the connected counter part or the network.

See relevant chapters in the manual for a more detailed description of the various interfaces.

4.4 Operating Concept

The operating concept of the terminal has been designed to allow the operator to access all masks - and thus all the data - quickly and easily.

A number of means are available to the user to help guide the operator through the mask system.

4.4.1 Mask Structure

By means of the programming system, a network of I/O masks without a real hierarchical structure is formed. I/O masks are located at the nodes in the network and contain a selection field which is used to call up other masks by their names. It is possible to access any other I/O mask from an I/O mask by using the control and function keys.

4.4.2 External Mask Selection

Mask calls from the controller are handled in the same way as messages. This requires simply that 8000H is added to the desired mask number prior to transfer.



When the same number is assigned to a mask and message, both the mask and the message are called up during external mask selections.

This effect can be used to provide help in a mask. Otherwise, ensure that different numbers are used for masks and messages.

A mask called up externally is considered selected once the desired mask number appears in the variable <Image of Mask Number>. The acknowledgement from the serial message channel is not a reliable confirmation of the mask output.

Example 1: Mask number and message number are not identical

A project comprises masks with numbers ranging from 100 to 200; the first message has the number 10.

Mask 118 is to be called up by the controller.

The following adding operation must be performed in the controller:

$$118 + 8000H = 76H + 8000H = 8076H.$$

The value 8076H must be written to the address of the serial message channel.

Only mask 118 is displayed.

Example 2: Mask number and message number are identical

A project comprises masks with numbers ranging from 1 to 100; the first message has the number 10.

Mask 50 and serial message 50 are to be called up by the controller.

The following adding operation must be performed in the controller:

$$50 + 8000H = 32H + 8000H = 8032H.$$

The value 8032H must be written to the address of the serial message channel.

Both the mask 50 is displayed and message 50 are written to the message buffer.

4.4.3 Password Protection, Access Authorization

The operating concept incorporates a password protection function. Password protection prevents masks from being accessed and the data they contain from being altered without proper authorization. The protective function is available in every operating terminal. It is obtained by assigning access levels to masks and by using passwords.

Unless otherwise specified by the programmer, the access levels of all masks automatically default to the lowest level (=0), i.e. no password is required to access all masks with this access level.

Two authorization levels, referred to as the edit level and the view level, are assigned to every password.

View level means that the next mask can be viewed after entering the password but the values in it can also not be edited.

Edit level means that the mask can be viewed after entering the password and the values in it can be edited.

Up to eight different passwords with a length of up to 11 characters can be defined. When defining the passwords, the access authorizations should be entered in a hierarchical structure.

Example:

- password for the manufacturer of the system, machine, etc.
- password for servicing on site
- password for the machine setter, master craftsman, foreman etc.
- password for the system operator

The following figure illustrates how the access levels, edit and view levels work:

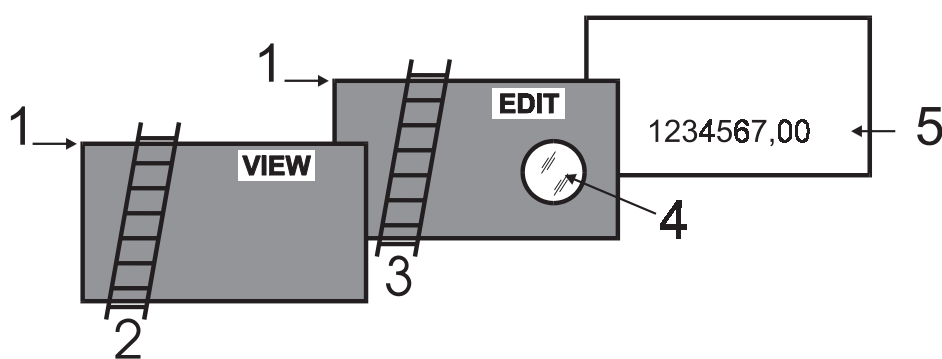


Fig. 2: Access levels

An access level (represented here by the height (1) of the walls) can be assigned to every mask and a view level (length of ladder 2) and edit level (length of ladder 3) can be assigned to every password.

A password must be entered for **viewing only** of a mask (5) that has been assigned an access level (access level is greater than 0). The password must have been assigned a view level (ladder 2) high enough to be able to get over the access level (view wall). The mask (5) can then be viewed (through window 4) but the variable values can not be edited.

For **viewing and editing** the variable values in a mask (5), a password must be supplied which has a view level (ladder 2) and an edit level (ladder 3) that are both high enough.

The following rules apply to passwords:

- Access is permitted if the view and edit level values are greater than or equal to the values specified for the access level.
- The edit level must be equal to or less than the view level.
- The higher the values for the view level and edit level, the higher the degree of authorization.
- The valid range of values for the view level and edit level is 0 to 255.
- The default setting for both is 0.

The Data Release key will have no function if pressed with an insufficient edit level. The password can be entered in any I/O mask. Note that the Setup Mask is an exception. The system variable **MskchgPasswd** is designed to prompt for password entry. If the Password Editor is selected in the programming system, passwords will not appear as they are entered on the terminal. Instead the character 'X' appears for every character that is entered.

Example:
Enter PASSWORD: XXXX

This example illustrates hidden entry of a 4-digit number. If an invalid password is entered, the authorization levels will automatically reset to 0. It is recommended that at least one password - the master password - is programmed with the highest authorization level. The first password entered in the password list via the programming system is the master password and as such has a special function. The master password is unique in that it can not be changed on the operating terminal. Furthermore, it can be used to reset all modified passwords to the default values entered in the programming system.

Example:

Mask 5	access level = 10	
Mask 6	access level = 20	
Mask 7	access level = 30	
Password 4712	Edit level = 15	View level = 25

Once the password '4712' has been entered, the following accesses are permitted:

- Mask 5 will be displayed, editing of values is authorized
- Mask 6 will be displayed, editing of values is **not** authorized
- Mask 7 will **not** be displayed, editing of values is **not** authorized

The access level for the Startup Mask is always 0. The Setup Mask is an exception with regards to the password and external data release functions. Since no communication is taking place when the Setup Mask is displayed, the external data release function is not applicable. To restrict access, passwords must be used. By defining the first editable variable in the Setup Mask as a Password Editor, all further variables can be protected against unauthorized access. The view level does not apply when accessing the Setup Mask. Viewing is always permitted if a value less than or equal to 254 is selected for the access level of the Setup Mask. The edit level for all variables of the Setup Mask, with the exception of the Password Editor, is the same as that defined as the access level. Access to the mask is always denied if an access level of 255 is defined for the setup mask. This means that it will no longer be displayed during initialization of the terminal and can therefore not be selected. However, all terminal-specific parameters can also be edited in any I/O mask. The new parameters become effective by restarting the terminal or with the system variable **Boot**.

4.4.3.1 Reactivating the Password Protection

The access authorization for a mask or variable is reset whenever

- the operating terminal is switched off and turned back on
- an incorrect password is entered
- bit 2 in the <Write Coordination Byte> is set
- the system variable **MskchgResPasswd** is activated
- the option **Reset Password** in the mask parameters of the password-protected mask is selected.

4.4.3.2 Password Management

Passwords are stored in the operating terminal's Flash memory. These are the default passwords that are used when the terminal is started up initially after a download. At the same time, the passwords are stored in the operating terminal's RAM.

The passwords stored in the flash memory can be reactivated by writing to the system variable **FlashPasswd**.

Every password (except for the master password = first password in the list) can be edited from the terminal. For this process, the password to be edited is written to the system variable **MskchgPasswd** first. The new password is then written twice to the system variable **ChangePasswd**. If both entries for the new password are identical, the password will become effective immediately; otherwise, a system message will be displayed and the password reset.

Passwords are stored and compared as 11-character strings. The passwords are entered with the Alphanumeric Editor.

Passwords are only globally programmable (not language-specifically).

4.4.3.3 Password Mask and Password Functionality

- It is possible to create a special mask which requests entry of a password. This password mask is then displayed whenever an attempt is made to call up a password-protected mask without first entering a password with sufficient authorization. When a password with sufficient authorization is entered into this mask, the previously selected password-protected mask is called up once the Data Release key is pressed. There are no restrictions with respect to the remaining mask contents (texts, further variables, soft keys, etc.).
- You can specify for every mask separately whether the password protection is to be reactivated after the mask is exited.
- For those cases where no valid password has been entered, an option to exit the mask must be provided. The Cursor home key can be programmed to perform this function, for example.
- If no such mask has been created to prompt for password entry, the operator will be required to enter the password in masks specifically designed for this purpose.
- The entire password protection can be deactivated by setting the system variable **PasswdInactive** to the value 1. The operating terminal will then act as if all masks were created with an edit and view level of 0. The system variable is battery-backed, so deactivation will therefore still be in effect after the operating terminal is restarted.

4.5 Masks

In conjunction with the operating system, the term 'mask' always refers to the contents of one screen. The size of masks therefore varies from operating terminal to operating terminal.

Masks with a specific functionality form the basic elements of the operator guidance. The first step when programming a mask is to specify its functionality. The process of designing the operator guidance is greatly simplified by the restricted number of different mask types.

The following mask types are available in the operating concept:

- Setup Mask (system mask)
- Start Mask (system mask)
- Password Mask (system mask)
- Main Mask (system mask)
- I/O Mask (user mask)

4.5.1 Mask Parameters

The number of mask parameters that is available for a specific mask depends on the mask type. The mask parameters determine the functionality of the control keys in the operator guidance. Any mask parameters which are not required may remain unassigned. If no mask parameters at all are programmed, only the function keys or external mask selection can be used to exit the mask in question.

The functionality of the control keys specified in the mask parameters is subject to access control of the password system. This prevents unauthorized access to masks via the control keys.

In addition to the control key functionality, the mask parameters also contain the access level definition. The access level indicates the minimum value that a password (view level, edit level) must have to be authorized to access a mask and edit its values. The default value used for the access level is '0' (meaning free access). A selected Automatic Data Release option in the mask parameters will allow the operator to supply input into the mask without having to press the Data Release key first.

An option for automatic reactivation of the password protection is also available in the mask parameters to ensure that the password has to be reentered when a password-protected mask is called up again.

4.5.2 System Masks

System masks facilitate initial programming. They also get your system up and running directly. This makes the initialization phase an integral part of the user description. Any mask can be selected as a system mask. System masks are basically I/O type masks with a few restrictions. These restrictions result from the operator-prompted initialization phase and the fact that communication to the controller is not yet established.



The system masks Setup Mask and Start Mask can not be accessed via external mask selection!

4.5.2.1 Setup Mask

Only terminal-specific parameters can be defined in the Setup Mask. This is because no communication takes place with the connected controller while the Start-up Mask and Setup Mask are displayed. The external data release therefore has no function. To protect data, a password must be assigned.

From the operator guidance, the Setup Mask and Startup Mask can be reached through the function keys, provided they have not been assigned an access level. After the masks are exited, however, the terminal is not reinitialized.

Examples of terminal-specific parameters:

- Printer interface settings
- Default contrast/default intensity setting of the display
- Date and time settings
- Activation of the download function.

4.5.2.1.1 Password Protection - Setup Mask

The Setup Mask is an exception with regard to password protection.

To password-protect the Setup Mask, the system variable **MskchgPasswd** must be set up as the first variable which can be edited in the Setup Mask. A password can be input regardless of the access level (with the exception of the access level 255).

For the setup mask, the access level applies to the edit level only, meaning that its contents are always visible to the operator.

4.5.2.1.2 Function Without the Setup Mask

If the Setup Mask is not needed, its access level can be set to the value 255, thus preventing access to the Setup Mask via the Start Mask (using the Data Release key).

4.5.2.2 Start Mask

The Start Mask is displayed for around 5 seconds after the terminal is switched on. This is a fixed time setting; the startup process can not be changed.

With regards to the mask design, the user can only design the text to be displayed in the mask. Any combination of characters, character sizes and text attributes can be chosen in the programming system for this purpose.

Only system variables can be output in the Start Mask. Variable input is not possible due to the time restriction. While the Start Mask is displayed, the Setup Mask can be called by pressing the Data Release key. However, this is not possible if the access level for the setup mask is set to 255.

Some examples of information that may be chosen to be displayed in the Start Mask are listed below:

- address for servicing
- machine type
- version number of the program

4.5.2.3 Password Mask

The password mask is an I/O type of mask. The functionality is as illustrated in the password description.

4.5.2.4 I/O Mask

As a basic type of mask, the I/O mask offers a broad range of functions - enough in fact for configuring complete operator guidances based entirely on it.

I/O masks provide the following options:

- Selection of menus
- Definition of data formats
- Scaling of values
- Output of values once or cyclically
- Text display
- Message output
- Direct selection of up to five adjacent masks with the control keys
- Display of large mask contents over several screen pages
- Display of values in tables

Key Functions in the I/O Mask

Key: Cursor up	Can be programmed freely to select adjacent masks
Key: Cursor down	Can be programmed freely to select adjacent masks
Key: Cursor left	Can be programmed freely to select adjacent masks
Key: Cursor right	Can be programmed freely to select adjacent masks
Key: Cursor home	Can be programmed freely to select adjacent masks. Exits the I/O mask for the higher-level menu.
Key: Data Release	Switches into the Editor and exits the Editor
Key: Enter	Only has a function in conjunction with the Editor
Key: ?, Help	Displays the help text specified for the mask for as long as the key is held down

4.6 Variables

The number of valid variable types is determined by the connected controller. All standard variable types which are commonly used are supported by the operating terminals. The data type that is used also determines the valid range of values and number of significant digits.

Variable Type	Size	Range of Values
Bit	1 bit	[0, 1]
Byte	1 byte	[-128 to +127]
Byte	1 byte	[0 to 255]
Word	2 bytes	[-32768 to +32767]
Word	2 bytes	[0 to 65535]
Lword	4 bytes	[-2147483648 to +2147483647]
Lword	4 bytes	[0 to 4294967295]
Lword	4 bytes	[$\pm 1.2 \cdot 10^{-38}$ to $\pm 3.4 \cdot 10^{+38}$]
ASCII	42 bytes	[0 to 255]

4.6.1 Output Variables

Output variables are numerical or alphanumeric data stored in the memory of the connected controller. The variables are retrieved from the controller whenever required, and output on the display at the programmed location according to the defined method of representation, formatting and scaling. Generally, a distinction is made between displaying one-time variables (variables are transferred only once and are then displayed) or at cyclic intervals (variables are continually updated while displayed). The character attributes that are available depend on the display type of the operating terminal.

One-Time Output Variables or Cyclic Output Variables:

Pure output variables are transmitted by the controller only once and are displayed in the mask. This method offloads the communication process and is suitable for all variables such as setpoint values, constant values and parameters that change either rarely or never. All output variables can be displayed on a scaled and formatted basis.

Cyclic output variables are used to display actual values, in other words values which vary continually while a mask is displayed. The cycle time (and thus the intervals at which the displayed actual values are refreshed) corresponds to the polling time predefined by the user.

Data transfer between the operating terminal and the PLC can be offloaded by using variables of the same data type only (data word, flag word, input word, flag word) and by using continuous addresses within one mask.

Cyclic output variables can also be scaled and formatted. Note that the display of floating point numbers requires more computing time when the values are scaled. These will therefore not be displayed in 'real time'. For this reason, the cycle times selected here should be > 500 ms.

Also note that the larger the amount of cyclic data for transfer, the longer the response time to new values from the PLC. As a result, longer polling times are required.

Formatted Output:

Formatting a numerical variable allows the display format of the variable to be adapted to the output range of values. Formatting includes the number of digits to be displayed, the number of post-decimal places (fractional digits) and whether signs are to be used or not. The number of post-decimal places gives the operator the impression that a division has been performed. A division is not actually carried out, however. The variable in the controller must be available in the appropriate (higher) resolution.

Example:

Actual value of a length in the controller:

Word variable	Range of values 0 to 65535
	Resolution 1/100 mm

Display of actual values on the operating terminal:

Display range	0.00 to 655.35 mm
----------------------	-------------------

The output is in millimeters, so a conversion in the controller is not necessary. The following format has been selected in the variable definition:

PLC address: Data type Word
 Variable type: Decimal number (without sign)
 Length: 6 digits
 Post-decimal places: 2 digits

Selecting 'with sign' would cause the output to change as follows:

Display range -327.68 to 327.67 mm

Because of the sign, the output length has increased to 7 digits. The format specified in the variable definition must be adapted accordingly.

<u>Format</u>	<u>Data type</u>	<u>formatting</u>	<u>scaling</u>
Binary	Bit, Byte, Word, Lword	x	-
Hexadecimal	Byte, Word, Lword	-	x
Decimal	Bit, Byte, Word, Lword	x	x
BCD	Byte, Word, LWord	-	x
Floating point	Lword	x	x
Selection text	Bit, Byte, Word	x	-
Text	ext. ASCII font	-	-

The format corresponds to scaling without the need for calculation.

Scaled Output

When a variable is scaled, its range of values can be adapted to the operator guidance. Scaling applies to data input and output. The operands can be entered in the programming software.

Scaling for integers:

<u>Operand</u>	<u>Range of values</u>
Factor	-32768 to +32767 (excluding the value 0!)
Divisor	1 to +32767
Summand	-32768 to +32767

The operands factor and divisor must be >0!

Scaling for floating point numbers:

<u>Operand</u>	<u>Range of values</u>
Factor	+/-999999999,99999999 (excluding the value 0!)
Divisor	+/-999999999,99999999 (excluding the value 0!)
Summand	+/-999999999,99999999

The operands are stored in IEEE-format

The variable also retains its entire value range for formatting.

Formula for Scaling the Output:

Scaling is always performed on the variable in the operating terminal. Through scaling, the measured values collected in the controller are adapted to the operator guidance. The output representation is determined by *formula 1* only. Unlike the factor and divisor, the summand (representing an offset) can be set to the value 0. Before a variable or constant is output on the display, it is converted as follows:

$$\boxed{\text{Terminal value}} = \frac{\boxed{\text{PLC value}} \times \text{Factor}}{\text{Divisor}} + \text{Summand}$$

Fig. 3: Formula 1

Output Representation Types:

Certain types of variables can be recognized more easily if they are displayed in their own specific format. A type-specific representation makes interpretation of the variable content easier. For this reason, a wide range of representation formats is provided.

Examples:

- | | |
|-------------------------------------|-------------------------|
| - Input statuses of an input module | Binary representation |
| - Filling level of a container | Bar (bar chart) |
| - Temperature | Curve (line graph) |
| - Valve states | Graphics showing valves |

Example of a coded text used for outputting an end position condition:

<u>Binary</u>	<u>Hex</u>	<u>Decimal</u>	<u>Selection Text (Coded Text)</u>
0	00	0	Not in end position
1	20	32	In end position

Representation With Leading Zeros:

The option of displaying leading zeros can be used in conjunction with every integer. Leading zeros are required in particular for displaying dates and times and hexadecimal or binary formats. The 'Display Leading Zeros' option can be selected in the variable definition.

	<u>Representation without Leading Zeros</u>	<u>Representation with Leading Zeros</u>
Binary Number	10 0101	0010 0101
Time	8: 2:33	08:02:33
Date	5. 3.1997	05.03.1997

4.6.1.1 ‘Decimal Number’ Representation

The decimal representation of numbers is the most frequently used representation method.

The operating concept differentiates between the following decimal number types of representation: Standard, Timer, Counter and BCD-Format.

Decimal representation includes integers and floating point numbers.

4.6.1.1.1 ‘Standard’ Variable Type

The positional significance of the displayed positions increases from right to left. Leading zeros or the decimal point can be displayed as an option. Decimal representation is suitable for the data types Bit, Byte, Word and Lword. The maximum length depends on the data type. There are no blanks between the characters. In the controller, this variable is in the binary format or special timer or counter formats.

10 ³	10 ²	10 ¹	10 ⁰	10 ⁻¹	10 ⁻²	Positional significance
0	1	2	3	4	5	Display (123,45 _D)

4.6.1.1.2 ‘Timer’ Variable Type

The variable type Timer has a special function only when used in combination with Simatic S5 controllers.

The kind of formatting of the variable type Timer depends on the memory area of the PLC where the value was read. If the value is read directly from a timer word, the 10 bit binary time value contained in it and the 2-bit time base are converted to the time value to the base 10 ms. If read from a different memory area, i.e. data word (DW), flag word (MW), input word (EW) or output word (AW), it is then assumed that the value is BCD-coded (3 digits BCD-code and 2 bit time base). This value will also be converted to a time value to the base 10 ms.

The resulting time value to the base 10 ms can now be formatted in the same way as a fixed point number, i.e. use of post-decimal places is possible and scaling can be applied.

Example:

A setpoint value is entered on the operating terminal: Address MW100

The current actual value is displayed on the operating terminal: Address MW200

Representation Input Variable:	Decimal Number / Timer / 7 Digits / 2 Post-Decimal Places / Factor 1 / Divisor 1 / Summand 0
Representation Output Variable:	Decimal Number/ Timer / 6 Digits / 1 Post-Decimal Place/ Factor 1 / Divisor 10 / Summand 0

The command sequence

```
L   MW 100
SI  T   1
```

has caused the BCD-coded time value to be loaded into flag word 100.

The command sequence

```
LC T 1 (Load current time value as a BCD-number)
T MW 200
```

now causes the current time value to be loaded into flag word 200. Before being output, the operating terminal reads this value as a BCD-coded time value and interprets it.

For this process, the time value is converted to the time base 10 ms first and is then scaled.

The example lists the output values with one post-decimal place (fractional digit). This produces the following to be displayed on the operating terminal:

Input Value	Output Value	Resolution	KT Values (S5)
0000,01 to 0000,09	0000,0 to 0000,0	0,01 s	001.0 to 009.0
0000,10 to 0000,99	0000,1 to 0000,9	0,01 s	010.0 to 099.0
0001,00 to 0009,99	0001,0 to 0009,9	0,01 s	100.0 to 999.0
0010,00 to 0099,90	0010,0 to 0099,9	0,1 s	100.1 to 999.1
0100,00 to 0999,00	0100,0 to 0999,0	1 s	100.2 to 999.2
1000,00 to 9990,00	1000,0 to 9990,0	10 s	100.3 to 999.3

The resolution and as a result, the precision of the input and displayed values can be influenced by modifying the post-decimal places and scaling.

4.6.1.1.3 'Counter' Variable Type

The variable type Counter can only be used in conjunction with controllers that support this variable type. This will be explained in more detail using the variable type Counter in combination with the Siemens S5 as an example.

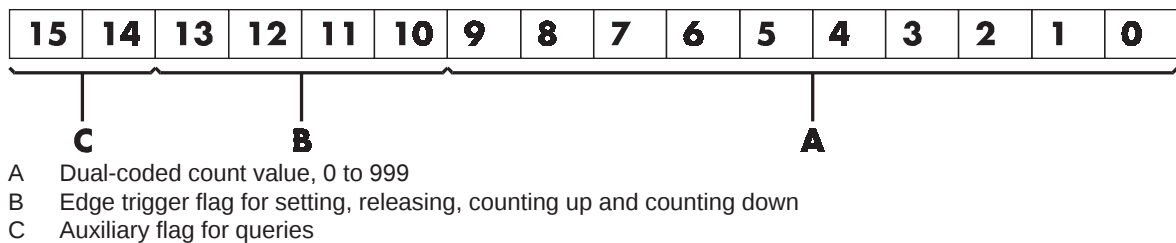


Fig. 4: Structure of the counter word with the Siemens SIMATIC S5-115U

The kind of formatting of the variable type Counter depends on the memory area of the PLC where the value was read. If read directly from a counter word, the 10 bit binary count value contained in it will be displayed directly, so conversion is not necessary.

If read from another memory area, i.e. data word (DW), flag word (MW), input word (EW) or output word (AW), it is assumed that the value is BCD-coded (3 digits BCD-code). The value will then be converted to a binary count value first.

The resulting binary count value can then be formatted in the same way as an integer, i.e. scaling is possible.

4.6.1.1.4 'BCD-Number' Variable Type

The BCD format is partially used to represent numbers in the PLC. This output is required in special cases. The positional significance of the displayed digits increases from right to left. The numbers 0 to 9 are used for the representation; leading zeros can be included optionally. The BCD mode of representation is suitable for the data types Byte, Word and Lword. The length is limited to a maximum of 8 digits. There are no blanks between the characters. The variable value is stored in the controller in BCD format. The range of numbers for a byte is 00 to 99.

10 ⁴	10 ³	10 ²	10 ¹	10 ⁰	Positional significance
0	1	2	3	4	Display (1234 ₀)

4.6.1.2 'Alphanumerical' Representation

In the alphanumerical mode of representation, ASCII strings are read from the controller in byte format and are represented on the display. The number of characters which can be displayed depends on the capabilities of the type of terminal involved. One display line is the maximum permissible length for a variable; longer texts are truncated. The address in the variable list indicates the beginning of the character string. A definition of the variable size is not included and is not necessary. The alphanumerical representation provides another means of editing mask texts during runtime.

4.6.1.3 'Selection Text' Representation

In the Selection Text mode of representation, a text string is assigned to a numerical value.
This allows two different tasks to be performed:

1. visualization or selection of statuses by means of text strings
2. change of mask.

An example of using a one-line selection text is the option of selecting the parity for a serial interface.
In this case, the text list will contain three entries:

Value	Text
0	No Parity
1	Odd
2	Even

A variable (**ComParityA** / **ComParityB**) is created in the mask.
Only these three options are selectable on the operating terminal by means of the Standard or Mix-Mode Editor; invalid inputs are therefore prevented.

Multiple-line selection texts are primarily used for menu control. Unlike one-line selection texts, multiple-line selection texts display the text list either in full or in part. If the value for the selection text field height is less than the number of text elements to be displayed, the cursor keys can be used to scroll through the text elements. After the final text list entry is reached, the selection proceeds with the first entry.

To indicate that a text is selected, the entire selection text line is represented in the inverse format.

With the representation type Selection Text, text lists are also used to assign mask names to the mask numbers:

<u>Value</u>	<u>Text</u>
10	Machine parameters
20	Serial message mask
30	Status messages
40	Message configuration
50	Interface parameters

A variable (**NewMask**) of the type Selection Text is created in the Main Mask with the maximum height of 5 lines and a length of 25 characters. The values in the text list and the mask numbers must match.

One of these entries can then be selected from the selection field displayed on the operating terminal: the desired mask is then displayed.

4.6.1.4 'Selection Image' Representation

In the Selection Image mode of representation, an image (pixel graphic) is assigned to a numerical value. This image is assigned in an image list. The image list is linked to the variable whose values are to be represented as images. In this way, language-independent visualization of operating states, inputs and outputs, etc. is possible.

Numbers may be freely assigned to the images. They need not necessarily be contiguous or sorted in a consecutive order. In addition, a default image exists for every image list which is displayed, whenever the variable assumes a value that does not exist in the image list. The images used are cut to the size that applies to the output format. This representation is limited to bits, bytes or words. Selection image variables can be both displayed and edited using the +/- key (as with the Selection Text Editor). Any modifications that are made are directly transferred to the controller.

The example below illustrates how images are assigned to numerical values:

<u>Value</u>	<u>Image Name</u>
120	Symbol1
34	Symbol2
7	Symbol3
1201	Symbol4

The names for the images represent the images in the image list. The actual list will look something like this:

120	
34	
7	
1201	

This image list has been defined with four entries. The graphic selected by the controller will be displayed.

All of the images in an image list should have the same output size to ensure that they overwrite (overlap) each other completely. An I/O mask can contain multiple selection image variables. These selection image variables can be of the types: one-time output variable, cyclical output variable, or input variable. When the cyclical output of selection images ('animation') is chosen, it must be remembered that the rate of change will be slow due to the limitations of the hardware performance. The more images that are output, the smaller the output performance. The selection image variable should therefore mainly be used for switching states or nearly static processes.

4.6.1.5 'Floating Point Number' Representation

The rules that apply to the Floating Point Number mode of representation are basically the same as those applying to the representation type Decimal Number - variable type Standard. The only difference is that with this representation type, the factor that is used to achieve scaling can be a floating point number; a divisor is therefore not required. With floating point numbers, the reciprocal value can also be generated, before the value is displayed.

Not every controller type supports the floating point format. There are different floating point formats available for processing in the controller (for example, the IEEE format).

4.6.1.6 'Hexadecimal Number' Representation

The hexadecimal representation of numbers is frequently used to display addresses when the PLC is programmed. This format is aimed for use by more experienced operators! The positional significance of the displayed digits increases from right to left. Numbers are expressed by the characters 0 to 9 and A to F. Only capital letters and leading zeros are used for representation. The hexadecimal representation is suitable for the data types Byte, Word and Lword. The length is limited to a maximum of 8 digits. There are no blanks between the characters.

16 ⁴	16 ³	16 ²	16 ¹	16 ⁰	Positional significance
0	E	4	5	A	Display (0E45A _H)

4.6.1.7 'Binary Number' Representation

The binary format permits the representation of single bits, bytes, words or Lwords. The value for the 'length' entered in the variable definition corresponds to the number of bits to be represented. Counting always begins with bit 0. The number of blanks indicates how many gaps (spaces) are inserted between each bit. Output is always in the horizontal position. The direction of the positional significance can be set in the variable definition.

Bit 3	Bit 2	Bit 1	Bit 0	
0	1	0	0	Direction = 76543210
Bit 0	Bit 1	Bit 2	Bit 3	
0	0	1	0	Direction = 01234567
0100				Blanks = 0
0 1 0 0				Blanks = 1
0 1 0 0				Blanks = 2

4.6.1.8 Bar Representation

Variables in I/O masks can be represented as bars. They can be programmed to be either horizontal or vertical. The bars can start at a reference point (e.g. in the center of the display) and extend in both the positive and the negative direction. The range of values of the bars can be defined by specifying the upper and lower limit (corner values). The bars are represented on the terminal with the aid of four different graphical objects, the simplest of which is a fill pattern. Separate fill patterns can be specified for:

- the empty part of the bar
- the filled-in bar
- the 'lower limit exceeded' bar
- the 'upper limit exceeded' bar

The programming software includes seven default fill patterns for representing bars on the operating terminal. The other, above-mentioned graphical objects can also be used to design your own fill patterns and shapes.

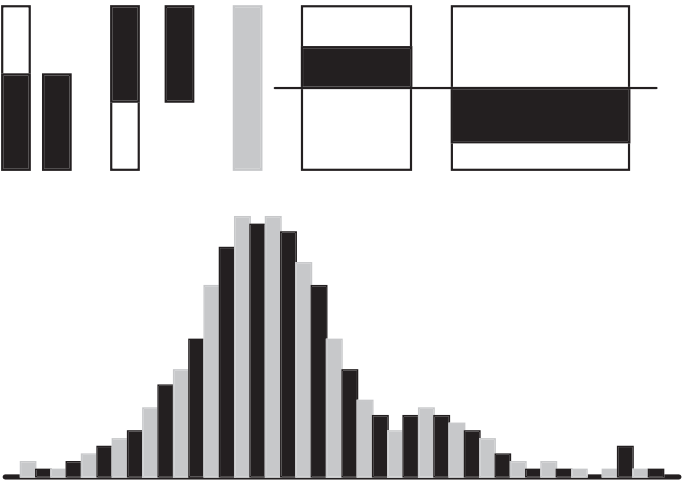


Fig.5: Different forms of bar representation

Bar charts can be output once only, cyclically or after defined events (event-controlled). They are used purely to indicate actual values.

The range of values for outputs extends from -32768 to +32767.

If your mask contains several bars, you should address the variables such that they can be transferred contiguously.

Each bar varies between a lower limit and an upper limit. Outside these limits, it changes to the predefined fill pattern for Upper or Lower Limit Exceeded.

Possible applications:

- Vertical and horizontal trend displays
- Visualized monitoring of limit values
- Histograms
- Filling-level indications

Example of a filling-level indication:

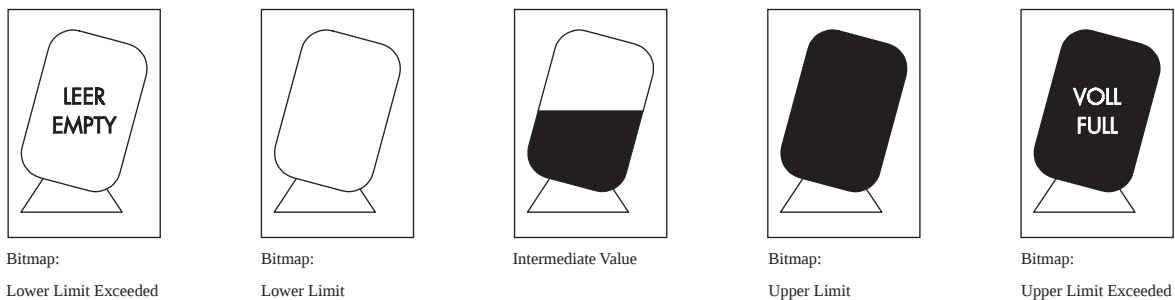


Fig. 6: Example of a filling-level indication

The bar mode of representation always means a slower output rate. Each bar therefore always corresponds to an integer multiple of a character. The smallest possible bar is equivalent in size to a single character, while the largest bar covers the entire display. The bars 'grow' one pixel at a time.

4.6.1.9 Curve Representation (Trendline)

The curve mode of representation permits value tables to be output as dotted lines. A 'curve' variable must be defined in a mask in order to represent a curve. The size of the curve is determined by its length and height. If a coordinate system is required, it can be displayed with the aid of background images.

The address of the curve variable represents the start of a value table in the PLC. Each value in the table describes one pixel on the curve. The graphical representation of the value table resembles that for cyclic output variables.

Examples of a variable time history:

- Output of one-time processes
- Memory function of a point recorder
- Filling-level curves

A curve is limited by the following parameters:

- Maximum height: Height of the display
- Maximum width: 54 pixels per curve variable

If you need a width of more than 54 pixels, you can output several curves directly adjacent to one another.

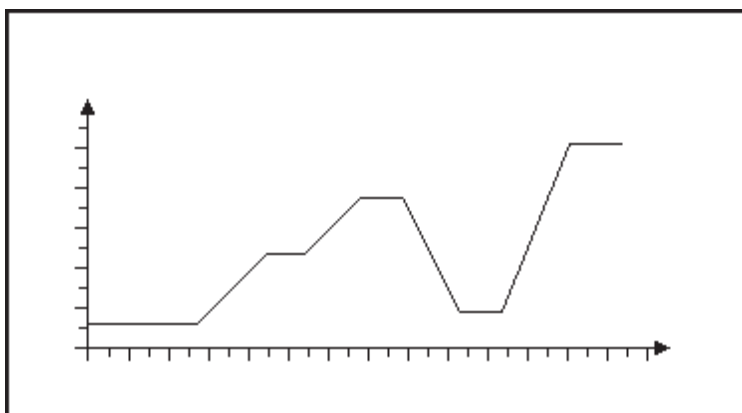


Fig. 7: Example of a curve representation

The height information, in other words the variable values, is read from the controller in a contiguous field by means of one read job. The height information in the field element with the starting address ($address + 0$) is displayed on the far left. All subsequent height information ($address + n$) is shifted one pixel position to the right. The curve height information is refreshed cyclically.

The output is thereby deleted and written again one pixel at a time.

Only two parameters are required to represent a curve:

- Curve width as a multiple of the character width (normal font)
- Curve height as a multiple of the character height (normal font)

4.6.2 Input Variables

Input variables, when displayed for the first time (when the mask is activated), are treated in the same way as one-time output variables, i. e. the same representation options are available. This includes the scaling effect.

Scaling is performed on the value in the PLC.

Input variables can be modified in the terminal by means of Editors. Their functionality is determined by the type of Editor that is selected.

The Editors that are supported for variable input are the same as those supported for variable output. Editing of the variables is subject to certain conditions which must take into consideration by the user when creating the project, such as the password protection and the external data release, for example.

The following must be observed when entering timers and counters:

Timer

When writing to a timer variable, any scaling that may have been performed is first reversed to give the time value to the base 10 ms again. The BCD-coded value is then calculated so that the lowest possible time base is used.

Avoid writing to a timer word in the PLC, as this has uncontrolled results on the control bits.

Counter

When writing to a counter variable, any scaling that may have been performed is first reversed. The BCD-coded value is then computed and transferred to the PLC. **Avoid writing to a counter word in the PLC, as this has uncontrolled results on the control bits.**

Formula for Scaling Input Values

Input values transferred by the PLC are processed according to *formula 1* before being displayed. After the values have been edited, they are processed in accordance with the inverse function (*formula 2*) before being transferred back to the PLC. The inverse function will automatically be generated in the terminal. Operands must be defined by the user on the basis of the values in the PLC including the input value.

$$\boxed{\text{PLC value}} = \frac{\boxed{\text{Input value}} \times \text{Summand}}{\text{Factor}} \times \text{Divisor}$$

Fig. 8: Formula 2

During conversion, the last digit is automatically rounded. This must be kept in mind when defining the upper limit.

$$(\text{Input value} \times \text{Factor}) < (\text{Pos. Upper Limit} - \text{Divisor} / 2)$$

Fig. 9: Formula 3

Plausibility Check

A plausibility check is performed on all input variables. As a part of this check, the entered value is validated against the upper and lower limits specified in the variable definition. System messages are generated if the entered value is outside these limits. In this case, the invalid value will not be transferred to the controller and the previous (valid) value will be retained.

If the system message texts 'Value too large' and 'Value too small' are deleted, the following applies:

- if the value drops below the limit, the value corresponding to the lower limit is used
- if the value exceeds the upper limit, the value corresponding to the upper limit is used.

A system message is not output during this process.

Help Text for Input Variables

A screen-sized help text can be created for each input variable. This help text can be displayed in the Editor by pressing the Help key. Important information to be inserted in a help text may include the upper and lower limits, the impact of variables on the process to be carried out, or any interactions with other variables.

If a variable-specific help text is not defined, the default help text will be displayed instead. This is also a screen-sized text which can be edited in the programming software. If a default help text is not defined, a blank mask will be displayed.

4.6.3 System Variables

System variables can be used to control terminal-internal functions. System variables can be referenced to function keys and soft keys. They can also be used in masks as normal variables for input and output processes.

The following applies if a system variable is linked to a function key or soft key:

- **do not** link a change of mask function and a system variable to the **same key**
- it is not necessary that the same function key or soft key is used to set (1) and reset (0) a system variable
- jogging mode is obtained by using the same function key or soft key to set (1) and reset (0) a system variable.

Do not include the names of system variables in the variable list. If the name of a system variable is referenced to a PLC address, it will lose its terminal-internal function.

The following subchapters list the system variables, organized into separate groups of functionality, and describes their functions.

4.6.3.1 Basic Functions

IntEraseEprom

Allows the user description to be erased from the Flash memory and prepares the terminal for a new download via interface channel SER2. There will be no communication with any connected PLC during this process. Writing of a '1' to this variable erases the user description unconditionally. Subsequently, the variable is automatically reset to '0'.

Data type:	Numeric
Editor:	Selection text , positive decimal number
Output:	---
Possible values:	(0) Inactive Initial state
	(1) Active Erases the user description

MainVersion

Allows the user to display the current firmware version of the operating terminal.

Data type:	Alphanumeric
Editor:	---
Output:	Alphanumeric, 8 characters
Possible values:	Format specified by manufacturer

ComVersion

Is used to display the type and version number of the current communication protocol.

Data type:	Alphanumeric
Editor:	---
Output:	Alphanumeric, 8 characters
Possible values:	Format specified by manufacturer

UserVersion

Indicates the version number of the project description to the operator. The user enters this number into the programming system during programming time.

Data type: Numeric
Editor: ---
Output: Numeric, 3 digits
Possible values: ---

Boot

This variable can be used to initiate the terminal to boot (system restart). The variable is suitable for integrating the setup function into the regular operator guidance, making it possible to modify terminal parameters which require a subsequent reinitialization (system restart) in any I/O mask. The terminal is booted if a '1' is written to this variable. Subsequently, the variable is automatically reset to '0'.

Data type: Numeric
Editor: Selection text , positive decimal number
Output: ---
Possible values: (0) Inactive Initial state
 (1) Active System restart

LCDDContrast

Is used to adjust the contrast for terminals with liquid crystal displays.

Data type: Numeric
Editor: Integer
Output: ---
Possible values: -127 to +127 (to be limited further based on terminal type)

LCDBackground

Is used to adjust the background of the display. This variable is only relevant for operating terminals fitted with a corresponding display.

Data type: Numeric
Editor: Selection text , positive decimal number
Output: ---
Possible values: (0) Normal representation
 (1) Inverse background

LCDBackLight

Is used to adjust the background lighting of the display. This variable is only relevant for operating terminals fitted with a corresponding display.

Data type: Numeric
Editor: Any
Output: ---
Possible values: (0) Background lighting OFF
 (1 to x) Background lighting ON (dimmed)

TurnOnTemp

Is used to ensure that the liquid crystal display of the operating terminal is not turned on until a certain ambient temperature is reached. This variable is only relevant for operating terminals provided with a corresponding functionality in conjunction with the corresponding display.

Data type:	Numeric
Editor:	Selection text , positive decimal number
Output:	---
Possible values:	(0) LCD-display OFF (1 to x) value of the temperature at which the LCD-display is turned ON

OsLanguage

With multilingual user descriptions, this variable is used for online language selection.

Data type:	Numeric
Editor:	Selection text
Output:	---
Possible values:	(0) First language (n) nth language

4.6.3.2 Communication SER1**ComDataLenA**

Defines the number of data bits for the communication via SER1.

Data type:	Numeric
Editor:	Selection text , positive decimal number
Output:	---
Possible values:	(0) 5bits/Char (1) 6bits/Char (2) 7bits/Char (3) 8bits /Char

ComParityA

Determines the creation of and check method of the parity for the communication via SER1.

Data type:	Numeric
Editor:	Selection text , positive decimal number
Output:	---
Possible values:	(0) No Parity (1) Odd Parity (2) Even Parity.

ComStopBitsA

Defines the number of stop bits for the communication via SER1.

Data type:	Numeric
Editor:	Selection text , positive decimal number
Output:	---
Possible values:	(0) 1 bit
	(1) 1.5 Bits
	(2) 2 bits

ComBaudrateA

Defines the baud rate for the communication via SER1.

Data type:	Numeric
Editor:	Selection text , positive decimal number
Output:	---
Possible values:	(0) 300 Bd
	(1) 600 Bd
	(2) 1200 Bd
	(3) 2400 Bd
	(4) 4800 Bd
	(5) 9600 Bd
	(6) 19200 Bd
	(7) 38400 Bd

ComHandshakeA

Specifies the type of handshake for the communication via SER1.

Data type:	Numeric
Editor:	Selection text , positive decimal number
Output:	---
Possible values:	(0) No handshake
	(1) RTS/CTS hardware handshake
	(2) XON/XOFF software handshake

ComDefaultA

This variable can be used to program new parameters for SER1.

Data type:	Numeric
Editor:	Selection text , positive decimal number
Output:	---
Possible values:	(0) Inactive Initial state
	(1) Active The data entered into the associated system variables are accepted as the new parameters.
	(2) Active The data stored in the Flash memory by the programming software are accepted as the new parameters.

ComTimeout

Specifies the monitoring period for SER1. The value (0) deactivates the timeout monitoring function.

Data type:	Numeric
Editor:	Positive decimal number
Output:	---
Possible values:	(0) Inactive Initial state (1 to 65535) Timeout monitoring active (time in ms)

ComRetryTimeout

Specifies the period of time, in milliseconds, that the terminal allows to elapse after an attempt to establish a connection via SER1 failed, before making another attempt. This allows the time period required for the PLC-specific powerup phase to be bridged, thereby avoiding the generation of an error message.

Data type:	Numeric
Editor:	Positive decimal number
Output:	---
Possible values:	0 to 65535 ms

ComSlaveNr

Contains the slave number used for the device on the network. This address can be used to address the operating terminal on the bus.

Data type:	Numeric
Editor:	Positive decimal number
Output:	Positive decimal number
Possible values:	0 to 255

4.6.3.3 Error Statistics SER1**ComParityCount**

Error counter for monitoring of parity errors on SER1 to the PLC. Is erased on every download.

Data type:	Numeric
Editor:	(Possible)
Output:	Positive decimal number
Possible values:	0 to 65535

ComOverrunCount

Error counter for monitoring of overrun errors on SER1 to the PLC. Is erased on every download.

Data type:	Numeric
Editor:	(Possible)
Output:	Positive decimal number
Possible values:	0 to 65535

ComFrameCount
Error counter for monitoring of framing errors on SER1 to the PLC. Is erased on every download.

Data type: Numeric
Editor: (Possible)
Output: Positive decimal number
Possible values: 0 to 65535

ComStatisticsTab
The system variable points to the beginning of the protocol statistics table with 6 elements of 4 bytes each.

Data type: Positive decimal number
Editor: ---
Output: Table
Possible values: ---

ComErrorTab
The system variable points to the beginning of a table with 16 elements which contain the most recent 16 communication errors.

Data type: Positive decimal number
Editor: ---
Output: Table
Possible values: ---

ComSubcodeTab
The system variable points to the beginning of a table with 16 elements which contain the subcodes of the communication errors.

Data type: Positive decimal number
Editor: ---
Output: Table
Possible values: ---

4.6.3.4 Communication SER2

ComDataLenB
Defines the number of data bits on SER2.

Data type: Numeric
Editor: Selection text , positive decimal number
Output: ---
Possible values (0) 5 bits/char
 (1) 6 bits/char
 (2) 7 bits/char
 (3) 8 bits/char

ComParityB

Determines the creation of and check method for the parity bit for SER2.

Data type:	Numeric
Editor:	Selection text , positive decimal number
Output:	---
Possible values:	(0) No parity (1) Odd parity (2) Even parity

ComStopBitsB

Defines the number of stop bits on SER2.

Data type:	Numeric
Editor:	Selection text , positive decimal number
Output:	---
Possible values:	(0) 1 bit (1) 1.5 bits (2) 2 bits

ComBaudrateB

Defines the baud rate on SER2.

Data type:	Numeric
Editor:	Selection text , positive decimal number
Output:	---
Possible values:	(0) 300 Bd (1) 600 Bd (2) 1200 Bd (3) 2400 Bd (4) 4800 Bd (5) 9600 Bd (6) 19200 Bd (7) 38400 Bd (8) 57600 Bd

ComHandshakeB

Specifies the type of handshake on SER2.

Data type:	Numeric
Editor:	Selection text , positive decimal number
Output:	---
Possible values:	(0) No handshake (1) RTS/CTS hardware handshake (2) XON/XOFF software handshake

ComDefaultB

This variable can be used to program new parameters for SER2.

Data type:	Numeric	
Editor:	Selection text , positive decimal number	
Output:	---	
Possible values:	(0)	Inactive Initial state
	(1)	Active The data entered into the associated system variables are accepted as the new parameters.
	(1)	Active The data stored in the Flash memory by the programming software are accepted as the new parameters.

4.6.3.5 Real-Time Clock

RTCSec
This variable is used to display and set the seconds of the clock.

Data type:	Positive decimal number
Editor:	Positive decimal number
Output:	Positive decimal number
Possible values:	0 to 59

RTCMin
This variable is used to display and set the minutes of the clock.

Data type:	Positive decimal number
Editor:	Positive decimal number
Output:	Positive decimal number
Possible values:	0 to 59

RTCHour
This variable is used to display and set the hours of the clock.

Data type:	Positive decimal number
Editor:	Positive decimal number
Output:	Positive decimal number
Possible values:	0 to 23

RTCDay
This variable is used to display and set the day of the month.

Data type:	Positive decimal number
Editor:	Positive decimal number
Output:	Positive decimal number
Possible values:	0 to 31 (varies from month to month, clock corrects incorrect entries on the next change of date)

RTCMonth
This variable is used to display and set the month of the year.

Data type:	Positive decimal number
Editor:	Positive decimal number
Output:	Positive decimal number
Possible values:	0 to 12

RTCYear

This variable is used to display and set the year.

Data type:	Positive decimal number
Editor:	Positive decimal number
Output:	Positive decimal number
Possible values:	0 to 99 (only the year and the decade are influenced)

RTCDayofWeek

This variable is used to display and determine the day of the week. The variable assumes the values 0 to 6. This is a modulo counter. A coded text should be used for the display. The assignment and starting point can be specified as needed.

Data type:	Positive decimal number
Editor:	Selection text
Output:	Selection text
Possible values:	0 to 6

RTCDateFmt

Is used to enter the format according to which the date is to appear in displayed messages.

Data type:	Numeric
Editor:	Selection text , positive decimal number
Output:	---
Possible values:	(0) European DD MM YY (1) US MM DD YY (2) JAPAN JJ MM DD

RTCYear2000

This variable is used to display and set the year in a 4-digit format.

Data type:	Numeric
Editor:	Selection text , selection image, alphanumeric, decimal
Output:	---
Possible values:	0 to 9999

4.6.3.6 Serial Message System**RepmanSortCrit**

Is used to enter the sorting criterion for the serial message output.

Data type:	Numeric
Editor:	Selection text , positive decimal number
Output:	---
Possible values:	(0) By priority of message number (1) By time of arrival (newest first) (2) By time of arrival (oldest first)

ClearRepBuf

Is used to erase the entire serial message buffer. Optionally direct control via function keys, soft keys or Editor. If necessary, passwords or special masks should be implemented to control erasure of the message buffer.

Data type:	Numeric
Editor:	Selection text , positive decimal number
Output:	---
Possible values:	(0) Maintain data (1) Erase message buffer

RepmanRepPrint

Switches the output of messages via the printer on and off.

Data type:	Numeric
Editor:	Selection text , positive decimal number
Output:	---
Possible values:	(0) Off (1) On

RepoutNr

Switches the output of the message number in the message mask on and off.

Data type:	Numeric
Editor:	Selection text , positive decimal number
Output:	---
Possible values:	(0) Off (1) On

RepoutDate

Switches the output of the date of the message in the message mask on and off.

Data type:	Numeric
Editor:	Selection text , positive decimal number
Output:	---
Possible values:	(0) Off (1) On

RepoutTime

Switches the output of the time of the message in the message mask on and off.

Data type:	Numeric
Editor:	Selection text , positive decimal number
Output:	---
Possible values:	(0) Off (1) On

RepoutAnzYear

Determines the number of digits that is to be used to represent the year.

Data type:	Numeric
Editor:	Selection text , selection image, decimal
Output:	---
Possible values:	(0) year represented by 2 digits (1) year represented by 4 digits

RepoutRepText

Is used to display the most current serial message. The message is displayed in the same way as it would appear in the message mask, i.e. in accordance with the selected message parameters.

Data type:	Alphanumeric
Editor:	---
Output:	Alphanumeric
Possible values:	Any

RepoutRepText21

Is used to display the most current message of the serial message system - from the 21st digit with a variable length (20, 40 or 60). The message is displayed in the same way as it would appear in the message mask, i.e. in accordance with the selected message parameters.

Data type:	Alphanumeric
Editor:	---
Output:	Alphanumeric
Possible values:	Any

RepoutRepText41

Is used to display the most current message of the serial message system - from the 41st digit with a variable length (40 or 60). The message is displayed in the same way as it would appear in the message mask, i.e. in accordance with the selected message parameters.

Data type:	Alphanumeric
Editor:	---
Output:	Alphanumeric
Possible values:	Any

RepoutRepText61

Is used to display the most current message of the serial message system - from the 61st digit. The last 20 characters of a message are displayed. The message is displayed in the same way as it would appear in the message mask, i.e. in accordance with the selected message parameters.

Data type:	Alphanumeric
Editor:	---
Output:	Alphanumeric
Possible values:	Any

4.6.3.7 Parallel Message System

RepmanSortCritP

This variable indicates to the terminal the sorting criterion according to which the messages are to be displayed. The default value is defined in the user description. This variable can be adapted during editing processes at a later date.

Data type:	Numeric
Editor:	Selection text , positive decimal number
Output:	---
Possible values:	(0) By priority of message number (1) By time of arrival (newest first) (2) By time of arrival (oldest first)

RepoutNrP

This variable indicates to the terminal whether or not the message number is to be displayed along with the message text of messages in the parallel message system. The default value is defined in the user description. This variable can be adapted during editing processes at a later date.

Data type:	Numeric
Editor:	Selection text , positive decimal number
Output:	---
Possible values:	(0) Off (1) On

RepoutDateP

This variable indicates to the terminal whether or not the date is to be displayed along with the message text of messages in the parallel message system. The default value is defined in the user description. This variable can be adapted during editing processes at a later date.

Data type:	Numeric
Editor:	Selection text , positive decimal number
Output:	---
Possible values:	(0) Off (1) On

RepoutTimeP

This variable indicates to the terminal whether or not the time is to be displayed along with the message text of messages in the parallel message system. The default value is defined in the user description. This variable can be adapted during editing processes at a later date.

Data type:	Numeric
Editor:	Selection text , positive decimal number
Output:	---
Possible values:	(0) Off (1) On

RepoutAnzYearP

Determines the number of digits that is to be used to represent the year.

Data type:	Numeric
Editor:	Selection text , selection image, alphanumeric, binary, decimal, hexadecimal
Output:	---
Possible values:	(0) year represented by 2 digits (1) year represented by 4 digits

RepoutRepTextP

The terminal uses this variable to display the most current message of the parallel message system.

Data type:	Alphanumeric
Editor:	---
Output:	Alphanumeric
Possible values:	Any

RepoutRepTextP21

Is used to display the most current message of the parallel message system - from the 21st digit with a variable length (20, 40 or 60). The message is displayed in the same way as it would appear in the message mask, i.e. in accordance with the selected message parameters.

Data type:	Alphanumeric
Editor:	---
Output:	Alphanumeric
Possible values:	Any

RepoutRepTextP41

Is used to display the most current message of the parallel message system - from the 41st digit with a variable length (40 or 60). The message is displayed in the same way as it would appear in the message mask, i.e. in accordance with the selected message parameters.

Data type:	Alphanumeric
Editor:	---
Output:	Alphanumeric
Possible values:	Any

RepoutRepTextP61

Is used to display the most current message of the parallel message system - from the 61st digit. The last 20 characters of a message are displayed. The message is displayed in the same way as it would appear in the message mask, i.e. in accordance with the selected message parameters.

Data type:	Alphanumeric
Editor:	---
Output:	Alphanumeric
Possible values:	Any

4.6.3.8 Printer Control**StopPrint**

When the system variable is activated, the print process currently in progress is terminated.

Data type: Numeric
Editor: Selection text , positive decimal number, soft key,
 function key
Output: ---
Possible values: (0) Initial state
 (1) Terminate print process

BlockPrint

Upon activation of this system variable, the block selected in the message mask is printed.

Data type: Numeric
Editor: Selection text , positive decimal number, soft key,
 function key
Output: ---
Possible values: (0) Initial state
 (1) Print block

PrintAllRep

Upon activation of this system variable, the entire serial message memory is printed. The output is as predefined.

Data type: Numeric
Editor: Selection text , positive decimal number, soft key,
 function key
Output: ---
Possible values: (0) Initial state
 (1) Formatted printout
 (2) Full-length printout

PrintAllState

Upon activation of this system variable, the entire parallel message memory (status messages) is printed.

Data type: Numeric
Editor: Selection text , positive decimal number, soft key,
 function key
Output: ---
Possible values: (0) Initial state
 (2) Print message buffer

BlockPrintLong

Upon activation of this system variable, the entire area selected in a message mask is printed in full length. The settings chosen at programming are ignored.

Data type:	Numeric
Editor:	Selection text , positive decimal number, soft key, function key
Output:	---
Possible values:	(0) Initial state (1) Print selected area

4.6.3.9 Menu Control / Keys**NewMask**

If a mask number is written to this system variable, the system will switch to the corresponding mask.

Data type:	Numeric
Editor:	Selection text, selection field
Output:	---
Possible values:	1 to 9999

VarTablenR0

Generates a consecutive numbering in tables beginning with the number '0'. This system variable is also used to output constant table texts.

Data type:	Numeric
Editor:	---
Output:	Selection text, positive decimal number
Possible values:	0 to n

VarTablenR1

Generates a consecutive numbering in table beginning with the number '1'. This system variable is also used to output constant table texts.

Data type:	Numeric
Editor:	---
Output:	Selection text, positive decimal number
Possible values:	1 to n

HardCopy

Is used to upload a hard copy in PCX or ASCII format via interface X3 to the connected PC.

Data type:	Numeric
Editor:	Selection text, positive decimal number, soft key, function key
Output:	---
Possible values:	(0) Initial state (1) Activate hard copy (always currently displayed screen)

TabLeft

In tables, this system variable can be used to shift to the column on the left.

Data type:	Numeric
Editor:	Soft key, function key
Output:	---
Possible values:	(0) Initial state (1) To the left by one column

TabRight

In tables, this system variable can be used to shift to the column on the right.

Data type:	Numeric
Editor:	Soft key, function key
Output:	---
Possible values:	(0) Initial state (1) To the right by one column

TabPgUp

This system variable enables soft keys to be used in tables to shift from page to page (in an upward direction).

Data type:	Numeric
Editor:	Soft key, function key
Output:	---
Possible values:	(0) Initial state (1) Move up one page

TabPgDn

This system variable enables soft keys to be used in tables to shift from page to page (in a downward direction).

Data type:	Numeric
Editor:	Soft key, function key
Output:	---
Possible values:	(0) Initial state (1) Move down one page

Shift

When the system variable is set, alphanumerical characters can be entered. When the keys on the numerical keypad are pressed, the associated alphabetical letters are automatically provided. By pressing repeatedly, you can proceed through the associated letters. Only uppercase letters are provided.

Data type:	Numeric
Editor:	Soft key, function key
Output:	---
Possible values:	(0) Initial state, numbers only (1) Shift mode 1 active, uppercase letters

<u>Key</u>	<u>Letters (Characters)</u>
Point	: ? ! .
Minus	\ * / -
Plus	< = > +
Zero	() ° 0
One	S T U 1
Two	V W X 2
Three	Y Z % 3
Four	J K L 4
Five	M N O 5
Six	P Q R 6
Seven	A B C 7
Eight	D E F 8

ShiftCase

When the system variable is set, alphanumeric characters can be entered. When the keys on the numerical keypad are pressed, the associated alphabetical letters are automatically provided. By pressing repeatedly, you can proceed through the associated letters. Uppercase and lowercase letters are provided.

Data type:	Numeric
Editor:	Soft key, Function key
Output:	---
Possible values:	(0) Initial state, numbers only (1) Shift Mode 2 active, lowercase and uppercase letters

<u>Key</u>	<u>Letters (Characters)</u>
Point	: ? ! .
Minus	\ * / -
Plus	< = > +
Zero	() ° 0
One	S T U s t u 1
Two	V W X v w x 2
Three	Y Z % y z % 3
Four	J K L j k l 4
Five	M N O m n o 5
Six	P Q R p q r 6
Seven	A B C a b c 7
Eight	D E F d e f 8

KeyCursLeft

This system variable allows soft keys to be used as a *Cursor left* key.

Data type:	Numeric
Editor:	Soft key, function key
Output:	---
Possible values:	(0) Initial state (1) Treat soft key as a <i>Cursor left</i> key.

KeyCursRight

This system variable allows soft keys to be used as a *Cursor right* key.

Data type:	Numeric
Editor:	Soft key, function key
Output:	---
Possible values:	(0) Initial state
	(1) Treat soft key as a <i>Cursor right</i> key

KeyCursUp

This system variable allows soft keys to be used as a *Cursor up* key.

Data type:	Numeric
Editor:	Soft key, function key
Output:	---
Possible values:	(0) Initial state
	(1) Treat soft key as a <i>Cursor up</i> key

KeyCursDown

This system variable allows soft keys to be used as a *Cursor down* key.

Data type:	Numeric
Editor:	Soft key, function key
Output:	---
Possible values:	(0) Initial state
	(1) Treat soft key as a <i>Cursor down</i> key

KeyHome

This system variable allows soft keys to be used as a *Cursor home* key.

Data type:	Numeric
Editor:	Soft key, function key
Output:	---
Possible values:	(0) Initial state
	(1) Treat soft key as a <i>Cursor home</i> key

KeyHelp

This system variable allows soft keys to be used as a *Help* key.

Data type:	Numeric
Editor:	Soft key, function key
Output:	---
Possible values:	(0) Initial state
	(1) Treat soft key as a <i>Help</i> key

KeyDot

This system variable allows soft keys to be used as a *Decimal point* key.

Data type:	Numeric
Editor:	Soft key, function key
Output:	---
Possible values:	(0) Initial state (1) Treat soft key as a <i>Decimal point</i> key

KeyClear

This system variable allows soft keys to be used as a *Delete* key.

Data type:	Numeric
Editor:	Soft key, function key
Output:	---
Possible values:	(0) Initial state (1) Treat soft key as a <i>Delete</i> key

Key0

This system variable allows soft keys to be used as the key *0 (zero)*.

Data type:	Numeric
Editor:	Soft key, function key
Output:	---
Possible values:	(0) Initial state (1) Treat soft key as the key <i>0</i>

Key1

This system variable allows soft keys to be used as the key *1*.

Data type:	Numeric
Editor:	Soft key, function key
Output:	---
Possible values:	(0) Initial state (1) Treat soft key as the key <i>1</i>

Key2

This system variable allows soft keys to be used as the key *2*.

Data type:	Numeric
Editor:	Soft key, function key
Output:	---
Possible values:	(0) Initial state (1) Treat soft key as the key <i>2</i>

Key3

This system variable allows soft keys to be used as the key 3.

Data type: Numeric
Editor: Soft key, function key
Output: ---
Possible values: (0) Initial state
 (1) Treat soft key as the key 3

Key4

This system variable allows soft keys to be used as the key 4.

Data type: Numeric
Editor: Soft key, function key
Output: ---
Possible values: (0) Initial state
 (1) Treat soft key as the key 4

Key5

This system variable allows soft keys to be used as the key 5.

Data type: Numeric
Editor: Soft key, function key
Output: ---
Possible values: (0) Initial state
 (1) Treat soft key as the key 5

Key6

This system variable allows soft keys to be used as the key 6.

Data type: Numeric
Editor: Soft key, function key
Output: ---
Possible values: (0) Initial state
 (1) Treat soft key as the key 6

Key7

This system variable allows soft keys to be used as the key 7.

Data type: Numeric
Editor: Soft key, function key
Output: ---
Possible values: (0) Initial state
 (1) Treat soft key as the key 7

Key8

This system variable allows soft keys to be used as the key *8*.

Data type:	Numeric
Editor:	Soft key, function key
Output:	---
Possible values:	(0) Initial state (1) Treat soft key as the key <i>8</i>

Key9

This system variable allows soft keys to be used as the key *9*.

Data type:	Numeric
Editor:	Soft key, function key
Output:	---
Possible values:	(0) Initial state (1) Treat soft key as the key <i>9</i>

KeyPlus

This system variable allows soft keys to be used as a *Plus* key.

Data type:	Numeric
Editor:	Soft key, function key
Output:	---
Possible values:	(0) Initial state (1) Treat soft key as a <i>Plus</i> key

KeyMinus

This system variable allows soft keys to be used as a *Minus* key.

Data type:	Numeric
Editor:	Soft key, function key
Output:	---
Possible values:	(0) Initial state (1) Treat soft key as a <i>Minus</i> key

KeyEnter

This system variable allows soft keys to be used as an *Enter* key.

Data type:	Numeric
Editor:	Soft key, function key
Output:	---
Possible values:	(0) Initial state (1) Treat soft key as an <i>Enter</i> key

KeyEdit

This system variable allows soft keys to be used as a *Release* key.

Data type:	Numeric
Editor:	Soft key, function key
Output:	---
Possible values:	(0) Initial state (1) Treat soft key as a <i>Release</i> key

4.6.3.10 Password

MskchgPasswd

Is used to enter the password into a mask.

Data type:	Numeric, alphanumeric
Editor:	Alphanumeric
Output:	---
Possible values:	11 characters

MskchgResPasswd

Is used to erase the password currently entered; resets the access authorization. The access levels are reset on every power-up.

Data type:	Numeric
Editor:	Selection text , positive decimal number, soft key, function key
Output:	---
Possible values:	(0) Initial state (1) Reset access authorization

ChangePasswd

This system variable can be used to modify the passwords in the terminal.

Data type:	Numeric, alphanumeric
Editor:	Alphanumeric
Output:	---
Possible values:	11 characters

FlashPasswd

This system variable can be used to reset the passwords to the code values specified in the programming software. Useful in the case of a loss of the passwords. Make sure to make a note of the master password first.

Data type:	Numeric
Editor:	Selection text , positive decimal number, soft key, function key
Output:	---
Possible values:	(0) Initial state (1) Reset passwords

PasswdInactive

Is used to deactivate the password protection. The system variable is battery-backed. The most recent setting is retained when the device is turned off.

Data type:	Numeric
Editor:	Selection text , positive decimal number
Output:	---
Possible values:	(0) Password protection active (initial state when first initialization) (1) Password protection inactive (edit and view level = 255)

4.6.3.11 Recipes

SelectDSNr

This variable contains the number of the active data set. To edit the variable, the associated Selection Text Editor must be used.

Data type:	Numeric
Editor:	Selection text
Output:	Positive decimal number
Possible values:	0 to 250

SelectDSName

This variable contains the name of the active data set. To edit the variable, the associated Selection Text Editor must be used.

Data type:	Alphanumeric
Editor:	Selection text
Output:	Alphanumeric
Possible values:	15 characters

DestDSNr

When copying data sets, this variable contains the number of the destination data set.

Data type:	Numeric
Editor:	Positive decimal number
Output:	---
Possible values:	1 to 250

DSCopy

This variable can be used to copy the active data set to the destination specified in **DestDSNr**.

Data type:	Numeric
Editor:	Selection text , positive decimal number, soft key, function key
Output:	---
Possible values:	(0) Initial state (1) Copies to destination in DestDSNr (2) Copying and automatic search for a free data set (3) Copies to destination in DestDSNr and overwrites any existing data set.

DSDelete

This variable can be used to delete the active data set. The first data set from the same recipe will become the new active data set.

Data type:	Numeric
Editor:	Selection text , positive decimal number, soft key, function key
Output:	---
Possible values:	(0) Initial state (1) Deletes the data set

ActDSName

This variable contains the name of the current data set. It can be read and in the case of RAM-data sets, it can additionally be written.

Data type:	Alphanumeric
Editor:	Alphanumeric
Output:	Alphanumeric
Possible values:	15 characters

SelectRezeptNr

This variable contains the active recipe. The variable can also be modified outside the recipe mask.

Data type:	Numeric
Editor:	Numeric, Selection text
Output:	Numeric, Selection text
Possible values:	1 to 250

DSDownload

This variable can be used to write the active data set to the controller.

Data type:	Numeric
Editor:	Selection text , positive decimal number, soft key, function key
Output:	---
Possible values:	(0) Initial state (1) Writes the data set to the controller

DSDnloadBreak

This variable can be used to terminate a data transfer to the controller currently in progress.

Data type:	Numeric
Editor:	Selection text , positive decimal number, soft key, function key
Output:	---
Possible values:	(0) Initial state (1) Terminates data set transfer

DSDnloadState

This variable can be used to monitor the data transfer to the controller.

Data type:	Numeric
Editor:	---
Output:	Selection text
Possible values:	(0) Initial state
	(1) Request for data transfer issued, but not yet released by the controller
	(2) Data transfer active

LoadDSName

This variable contains the name of the last data set which has been transferred to the controller. If the data set has already been deleted, question marks are displayed.

Data type:	Alphanumeric
Editor:	---
Output:	Alphanumeric
Possible values:	15 characters

StartSave

This variable can be used to transfer data sets to the PC.

Data type:	Numeric
Editor:	Selection text
Output:	---
Possible values:	(0) Initial state
	(1) Transfer one data set to the PC
	(2) Transfer all data sets of a recipe to the PC
	(3) Transfer all data sets in the terminal to the PC

SaveState

During a transmission to the PC, this variable indicates the current status of the transmission process.

Data type:	Numeric
Editor:	---
Output:	Selection text
Possible values:	(0) Initial state
	(1) One data set is transferred to the PC
	(2) All data sets of a recipe are transferred to the PC
	(3) All data sets in the terminal are transferred to the PC

StartRestore

This variable controls the download from the PC to the Terminal.

Data type:	Numeric
Editor:	Selection text
Output:	---
Possible values:	(0) Initial state (1) The terminal switches to the ready-to-receive state (2) The terminal terminates a transmission currently in progress.

RestoreState

This variable indicates the status of the transmission from the PC to the terminal.

Data type:	Numeric
Editor:	---
Output:	Selection text
Possible values:	(0) Initial state (1) Data transmission in progress

RestoreLineNr

This variable indicates the current line number of the data set file. It is used to verify the progress of the receive process and in the case of an error, for error localization.

Data type:	Numeric
Editor:	---
Output:	Numeric
Possible values:	1 to 255

StartRezPrint

This variable can be used to print data sets.

Data type:	Numeric
Editor:	Selection text
Output:	---
Possible values:	(0) Initial state (1) Starts data set printout (2) Terminates print process

RezPrintState

When printing data sets, this variable indicates the current printing status.

Data type:	Numeric
Editor:	---
Output:	Selection text
Possible values:	(0) Initial state (1) Data set printout in progress

StartUpload

This variable can be used to read, for the active recipe, a data set from the controller and to store it in the terminal.

Data type:	Numeric
Editor:	Selection text
Output:	---
Possible values:	(0) Initial state (1) Variables are read individually from their specified addresses (2) Variables are read as a block from the buffer specified for the recipe (3) Variables are read individually from their specified addresses. A free data set is automatically being searched for. If no free data set is available, system message 18 is displayed. (4) Variables are read as a block from the buffer specified for the recipe. A free data set is automatically being searched for. If no free data set is available, system message 18 is displayed.

UploadDSNr

This variable specifies the data set number to which the uploaded data set is to be written in the operating terminal.

Data type:	Numeric
Editor:	Numeric
Output:	---
Possible values:	1 to 250

UploadState

When uploading data sets to the operating terminal, this variable indicates the upload status.

Data type:	Numeric
Editor:	---
Output:	Selection text
Possible values:	(0) Initial state (1) Upload of data set in progress

4.6.3.12 Running Time Meter

Counter1
Counter2
Counter3
Counter4
Counter5
Counter6
Counter7
Counter8

Is used in conjunction with the start/stop function of the associated bit as a running time meter. The counter is incremented while the bit is set.

Data type:	Numeric
Editor:	Positive decimal number
Output:	Positive decimal number
Possible values:	0 to 4.294.967.295

4.6.3.13 Loop-Through Operation

Pg2Sps
Activates and deactivates the loop-through operation (toggle function). This function must be provided by the PG protocol.

Data type:	Numeric
Editor:	Selection text , positive decimal number
Output:	---
Possible values:	(0) Initial state (2) First 1 activates, second 1 deactivates the loop-through operation

Pg2SpsState
Indicates the status of the loop-through operation.

Data type:	Numeric
Editor:	---
Output:	Selection text
Possible values:	(0) Initial state (OFF) (1) Issue request for loop-through operation (2) Loop-through operation possible (3) Loop-through operation active

4.6.3.14 Loadable Font

ChrsetName

Is used to display the current font name.

The name of the font is limited to a maximum length of 8 characters.

Data type:	Alphanumeric
Editor:	---
Output:	---
Possible values:	(Standard) Display using terminal font (Font name) Display using own font

4.6.3.15 Maintenance

User1**User2****User3****User4****User5**

Universal variables that permit the user to store information in the terminal. The data are stored in the battery-backed RAM.

Data type:	Any
Editor:	Any
Output:	Any
Possible values:	16 bit

LCDADCInput

Is used to read the current input value from the AD-converter that is used for LCD contrast control. This value is only designed to be used for diagnosis purposes.

Data type:	Numeric
Editor:	Any
Output:	---

LCDDACOutput

Is used to read the current output value from the DA-converter that is used for LCD contrast control. This value is only designed to be used for diagnosis purposes.

Data type:	Numeric
Editor:	Any
Output:	---

Break

The current Editor is interrupted and the values entered are not transferred to the controller. Trigger the Break function with a function key in a soft key.

Data type:	Numeric
Editor:	Function key, soft key, selection text, decimal number, binary number, hexadecimal number
Output:	---
Possible values:	(0) Initial state (1) Editing process is terminated

4.6.3.16 Editors

EditInvers

Specifies if the inverse representation is to be used for the Editor for inputs.

Data type:	Numeric
Editor:	Any
Output:	---
Possible values:	(0) Normal representation (1) Inverse representation

EditEnter

Specifies the behavior of the Editor when the Data Release key is pressed.

Data type:	Numeric
Editor:	---
Output:	---
Possible values:	(0) Editor proceeds to the next input field (default setting) (1) Editor remains at the current position. To proceed, the cursor must be used.

StatePerm

Specifies the status of the status LED in the Data Release key.

Data type:	Numeric
Editor:	Any
Output:	0, 1, 2
Possible values:	(0) Status LED OFF (1) Status LED ON (2) Status LED FLASHING

4.6.3.17 Help

StateHelp

Specifies the status of the status LED in the Help key.

Data type:	Numeric
Editor:	Any
Output:	0, 1, 2
Possible values:	(0) Status LED OFF (1) Status LED ON (2) Status LED FLASHING

Message

In the case of a system error, the number of the system message is written to this variable.

Data type:	Numeric
Editor:	Any
Output:	---
Possible values:	(0) Initial state (1 to 29) System message number

QuitMessage

Acknowledges the system message that is currently displayed by the system variable **Message**.

Data type:	Numeric
Editor:	Any
Output:	---
Possible values:	(0) Initial state (1) Acknowledge

4.6.4 Editors

Various Editors are available in I/O masks for the various data types. With the term Editor, we refer to program parts designed to provide a convenient method of editing input variables. This sequence control required for the editing process is completely integrated into the terminal. An additional 'software' in the controller is not required for this purpose.

The Editors are influenced by the:

- data release (particularly from the connected controller)
- cyclic value refreshing
- the help system and
- the plausibility check.

The Editors are operated with the aid of editing and control keys. For details on the respective functions consult the corresponding Editor description.

The variables of the controllers can be modified using the Editors. Upon entry, the numerical variables are checked for plausibility. The limits are determined by the user in the variable definition. It is also possible to include a variable-specific help text. This text may, for example, provide a description of the variable function and its valid range of values.

Variables are entered as follows:

The Data Release key will allow the Editor to be activated if the I/O mask contains an editable value. When in the editing mode, the status LED in the Data Release key lights up and the cursor points at the first editable variable. This value can now be edited by using the numerical keyboard and/or the Plus/Minus keys. The value is stored by pressing the Enter key. The value is checked for plausibility during this process. If an error is detected, the status LED in the Help key will begin to flash.

A flashing status LED in the Help key indicates a malfunction to the user. A description of the malfunction will be displayed if the Help key is pressed. In the case of an error, the Enter key can not be used to exit the editing field.

The control keys, on the contrary, can be used to exit an editing field even if the value is invalid. In this case, however, the last value which was entered will be discarded and the old value (original value) will be restored.

In the editing mode, control keys can be used to select editable variables. If the Data Release key is pressed again, the editing mode will be exited and the status LED will go off.

The following Editors are currently available for selection in the variable definition: The functions of the keys are identical for all numerical Editors.

Key Functions in the Numerical Editor

Key: Cursor up	Moves the cursor up on the display by one editable variable and selects it in the process. After the cursor reaches the editable variable at the top, the variable at the bottom is selected next.
Key: Cursor down	Moves the cursor down on the display by one editable variable and selects it in the process. After the cursor reaches the editable variable at the bottom, the variable at the top is selected next.
Key: Cursor left	Moves the cursor within the editable variable to the left by one position until it reaches the end of the field.
Key: Cursor right	Moves the cursor within the editable variable to the right by one position until it reaches the end of the field.
Key: Plus	1. Variable currently selected: Value is deleted, a new value can be entered 2. Cursor has been moved within a positive value: No change 3. Cursor has been moved within a negative value: Deletes the negative sign
Key: Minus	1. Variable currently selected: Value is deleted, negative sign is entered at the least significant position, a new value can be entered 2. Cursor has been moved within a positive value: Negative sign is placed in front of the value 3. Cursor has been moved within a negative value: No change
Key: Delete	Deletes the position where the cursor is currently located (the sign also).

4.6.4.1 Decimal Number Editor

The Decimal Number Editor is a numerical Editor, optionally for positive decimal numbers only or for positive/negative decimal numbers.

In addition, the following variable types can be selected for decimal numbers:

- Standard Type Integers / fixed point numbers
- Timer Timer values in S5 format
- Counter Counter values in S5 format
- BCD Number Decade switch / incremental Editor

The number of digits (field length) is limited to the highest displayable 4-byte number. Decimal numbers can be output with leading zeros. Fixed point numbers can be represented by specifying a certain number of post-decimal places (fractional digits).

Other selectable options are scaling factors, limits, message transfer mode and display attributes.

This Editor can be used to access bit, byte, word and Lword variables. The maximum limits are determined by the memory map.

The variable types Timer and Counter are represented and can be edited on the operating terminal in the same way as the variable type Standard Type.

The variable type BCD-Number has a special feature.

The Editor for BCD-numbers is also referred to as the 'Decade Switch'.

With scaled variables, the value in the PLC changes by +1/-1, although the input value which is actually displayed also depends on the defined scaling factor! This means that special care must be taken whenever scaling is required. The Editor is also suitable for making fine adjustments or as a decade switch with decimal carry-over.

Possible Data Types:

- Positive integers (no entry of signs)
- Integers
- Positive fixed point numbers (no entry of signs)
- Fixed point numbers

Key Functions in the BCD-Number Editor

Keys: 1 to 9	1.)	Standard
		Direct input of numerical values
	2.)	Mix-Mode
Key: Plus		Direct input of numerical values
	3.)	Incremental
		No Function
Key: Minus	1.)	Standard
		No Function
	2.)	Mix-Mode and Incremental
Key: Cursor left		Increases the value at the cursor position and influences the more significant digits when the range is exceeded
		(with a dynamic repeat function)
	1.)	Standard
Key: Cursor right		No Function
	2.)	Mix-Mode and Incremental
		Reduces the value at the cursor position and influences the more significant digits when this value falls below the range (with a dynamic repeat function)

4.6.4.2 Floating Point Number Editor

The Floating-Point Number Editor supports Lword variables stored in IEEE format. This variable type is not supported by every controller type. In this case, an alternative would be to resort to fixed point numbers in Lword format. Scaling factors are also suitable for displaying precise fractional numbers.

The functions of the control and editing keys are identical to those of the Decimal Number Editor.

4.6.4.3 Hexadecimal Editor

The Hexadecimal Editor permits inputs of hexadecimal values. The sign keys have a special function here. Since there are no alphanumerical keys, alphanumerical characters can be entered with the sign keys instead.

Key Functions in the Hexadecimal Editor

Keys: 1 to 9	Direct entry of numerical values
Key: Plus	Enters the ASCII-characters 0 to 9, A to F in ascending order at the selected position
Key: Minus	Enters the ASCII-characters F to A, 9 to 0 in descending order at the selected position
Key: Cursor left	Moves the cursor within the editable variable to the left by one position until it reaches the end of the field.
Key: Cursor right	Moves the cursor within the editable variable to the right by one position until it reaches the end of the field.

4.6.4.4 Alphanumerical Editor

Entering alphanumerical characters with the numerical keyboard is possible by means of a special Incremental Editor. This Editor permits text strings that consist of lower and upper-case alphanumerical characters to be edited.

Key Functions in the Alphanumerical Editor

Keys: 1 to 9	Direct entry of the numbers 0 to 9.
Key: Plus	Enters the ASCII-characters 0 to 9, A to Z, a to z in ascending order at the selected position.
Key: Minus	Enters the ASCII-characters z to a, Z to A, 9 to 0 in descending order at the selected position.
Key: Cursor left	Moves the cursor within the editable variable to the left by one position until it reaches the end of the field.
Key: Cursor right	Moves the cursor within the editable variable to the right by one position until it reaches the end of the field.
Delete	Deletes the character at the cursor position. As the character is deleted, the text to the right of the cursor is moved to the left by one character.

Field Type 'Password'

In conjunction with the field type Password, the Alphanumerical Editor allows hidden entry of numerical values for the password. The field type Password has the same functions as the standard Alphanumerical Editor except that the characters do not appear when they are typed in. The entered character is represented by a letter 'X'. Entries are made in insert mode. The function of the field type **Password** is linked to the system variable **MskchgPasswd**. To be able to enter the password such that it also appears when it is entered, an alphanumerical variable without the field type Password must be used and linked to the system variable **MskchgPasswd**.

4.6.4.5 Selection Text Editor

The Selection Text Editor offers texts contained in a text list for selection. Every text in a text list is assigned to a numerical value. In the case of input variables, the numerical value associated with the corresponding text is written to the PLC variable. For output variables, the text associated with the numerical value is displayed on the operating terminal.

Key Functions in the Selection Text Editor

Key: Plus	Selection in ascending order (after the final value is reached, the value at top of the text list is selected next.)
Key: Minus	Selection in reversed order (after the first value is reached, the value at the bottom of the text list is selected next.)

4.6.4.6 Selection Image Editor

The Selection Image Editor offers images contained in an image list for selection. A numerical value is assigned to every image in an image list. In the case of input variables, the numerical value associated with the corresponding image is written to the PLC variable. For output variables, the image associated with the numerical value is displayed on the operating terminal.

Key Functions in the Selection Image Editor

Key: Plus	Selection in ascending order
Key: Minus	Selection in reversed order

4.6.4.7 Table Editor

Definition of Terms:

Number of rows:	Number of rows in the table that appear on the screen.
Number of elements:	Number of variables that are stored in the PLC for every column.
Table offset:	Offset of the variable represented in the first row in relation to the top of the table. When a mask is refreshed, the table offset is always set to 0. After paging down once, it is for example equal to the number of rows. The table offset is always the same for every column.
Single variable:	Every variable in a mask located outside of the table.
Column variable:	Every variable placed in a table creates one column.
Mask variable:	Column variable or single variable.
Row variable:	Every row in the column of a table contains one row variable.
Last row variable:	Row variable in the bottom row of the table with the maximum table offset.
First row variable:	Row variable in the top row of the table with a table offset of 0.

Functional Description:

- Every row variable is read from the PLC again just before being released for editing.
- Input and cyclic output variables in the table are also continuously updated. An exception to this is the row variable that is currently being edited. One-time output variables are updated whenever the mask is refreshed (for example, when the Help key is released) or whenever the table offset changes.
- Cursor keys and soft keys linked to system variables can be used to move from one variable to another. The order of movement between the mask variables corresponds to the order in which the variables were created in the mask (single variables) and in the table field. The following table illustrates the system variables (and their functions) that are relevant for navigating in table fields.

System Variable	Function in the Table Field	Key/Position
TabPgDn	Page Down	Function Key / Soft Key
TabPgUp	Page Up	Function Key / Soft Key
TabLeft	Shifts to the next column on the left When in the leftmost column, shifts to the previous mask variable	Function Key / Soft Key
TabRight	Shifts to the next column on the right When in the rightmost column, shifts to the next mask variable	Function Key / Soft Key
VarTablenR0	Displays row number beginning with 0	On the left side of the table field
VarTablenR1	Displays row number beginning with 1	On the left side of the table field

- Once the data release has been set (the status LED in the Data Release key lights up), the variables in a table field can be viewed and edited by means of the *Cursor up*, *Cursor down*, *Enter*, *Page up* and *Page down* keys. If the data release has not been set (the status LED in the Data Release key is off), the variables can only be viewed (using the *Page down* and *Page up* keys) and not edited.
- It must be possible to read all of the variables in one column of a table with one read request. When using 4-byte variables, this can result in a restriction of the number of rows.

Key Functions in the Table Editor

- | | |
|----------------|--|
| Key: Cursor up | Shifts to the previous mask variable if the cursor is located at a single variable or at the first row variable.
Shifts to the previous row variable if the cursor is located at a variable other than the first row variable. To be able to scroll row-by-row, the cursor must already be positioned at the row variable at the top.
When shifting to a new column variable, the cursor is placed at the bottom row variable (i.e. the last row variable).
When shifting to a single variable, the row number remains unchanged. |
|----------------|--|

Key: Cursor down	Shifts to the next mask variable if the cursor is located at a single variable or at the last row variable. Shifts to the next row variable if the cursor is located at a variable other than the last row variable. To be able to scroll row-by-row, the cursor must already be positioned at the bottom (not the last) row variable. When shifting to a new column variable, the row number is set to 0 and the cursor is placed at the top row variable (i.e. the first row variable). When shifting to a single variable, the row number remains unchanged.
Soft key: TabLeft	Shifts to the next column on the left in the same row. Shifts to the previous mask variable if the cursor is located in the column at the left-most position.
Soft key: TabRight	Shifts to the next column to the right in the same row. Shifts to the next mask variable if the cursor is located in the column at the right-most position.
Key: Page up	Reduces the table offset by the number of rows scrolled up. The row number is reduced by the corresponding number of rows (otherwise the remaining number of rows). Once the row number has also reached the minimum value, the key is no longer effective. Shifting to another mask variable in particular is not possible.
Key: Page down	Increases the table offset by the number of rows scrolled down. The row number is increased by the corresponding number of rows (otherwise the remaining number of rows). Once the row number has also reached the maximum value, the key is no longer effective. Shifting to another mask variable in particular is not possible.
Key: Enter	Function is the same as that of the Cursor down key, but the variable is additionally written to the PLC.

4.6.5 External Data Release

The PLC can prevent the input of variable values by the operator using the external data release. To be able to make use of this data release function, the following variables must have been created:

- a variable for the Read coordination byte
- a variable for the poll area.

When the Data Release key is pressed in an I/O mask, the terminal will enable the editing mode. If the controller has set the external data release, the status LED in the Data Release key lights up. The external data release is controlled by bit 0 in the first byte of the cyclic poll area of the controller (see chapter Poll Area). Data editing is inhibited whenever the PLC sets this bit to 0. In this case, the status LED in the Data Release key will begin to flash.

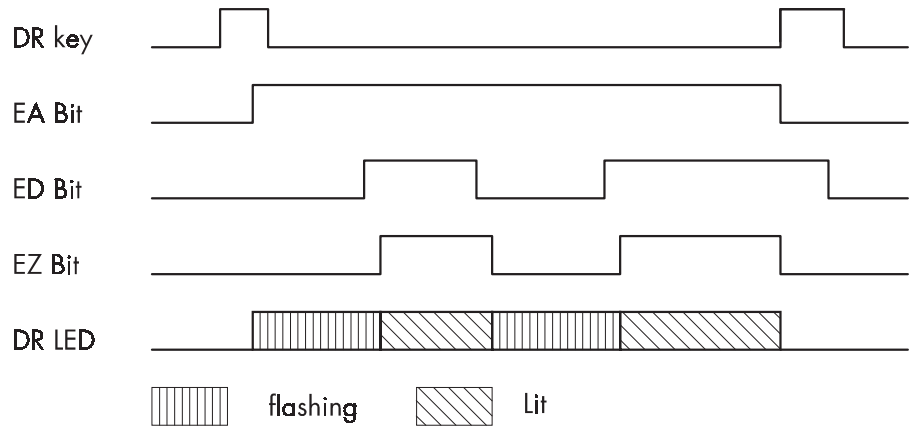


Fig.10: Pulse diagram - external data release

DR Key	Data Release key
EA-Bit	Editing request bit in the Read coordination byte
ED-Bit	External data release in the Write coordination byte
EZ-Bit	Editing status bit in the Read coordination byte
DF-LED	Status LED in the Data Release key

A flashing status LED in the Data Release key always indicates that no external data release has been set. Editing is allowed if the PLC has set bit 0 to 1. In this case, the status LED in the Data Release key will light up. A blinking cursor will appear in the first entry field in the mask. The Editor specified by the user will be activated, depending on the selected variable.

External data release handling:

- The PLC controls the external data release by switching the ED bit in the Write coordination byte.
- If the Data Release key is pressed, the EA bit in the Read coordination byte will be set and the status LED in the Data Release key will begin to flash.
- Approximately 100 ms later, the terminal will read the Write coordination byte from the controller. Editing is allowed if the external data release has been set (the ED bit in the Write coordination byte is set). In this case, the status LED in the Data Release key will light up and the EZ bit in the Read coordination byte will be set.

Editing is not allowed if the external data release has not been set (the ED bit in the Write coordination byte is not set). In this case, the status LED in the Data Release key will continue to flash.

- During the editing process, the PLC can inhibit data entry at any time via the ED bit. Whenever the operating terminal detects that the ED bit has been reset, data input is inhibited, the status LED in the Data Release key begins to flash and the EZ bit is also reset.

Whenever the operating terminal detects that the ED bit has been set, data input is allowed, the status LED in the Data Release key lights up and the EZ bit is also set.

- When the Data Release key is pressed again to exit the Editor, the last value that was edited is confirmed (i.e. it is transferred to the controller); the EA bit and EZ bit in the Read coordination byte are deleted.

4.6.6 PLC-Handshake

Handshake handling:

- If the terminal writes to a variable for which the PLC handshake option has been selected, the RA bit (refresh request bit) in the Read coordination byte will be set and data entry inhibited.
- The PLC can now, for example, provide a new data set for other input variables. The PLC will then set the RQ bit (refresh acknowledgement bit) in the Write coordination byte.
- Once the operating terminal detects that the RQ bit has been set, the terminal reads every input variable contained in the currently selected mask from the PLC again, then allows data inputs again and resets the RA bit.
- If the RQ bit is already set when writing to the variable for which the PLC handshake option has been selected, then the other input variables will be refreshed immediately and data entry will not be inhibited.

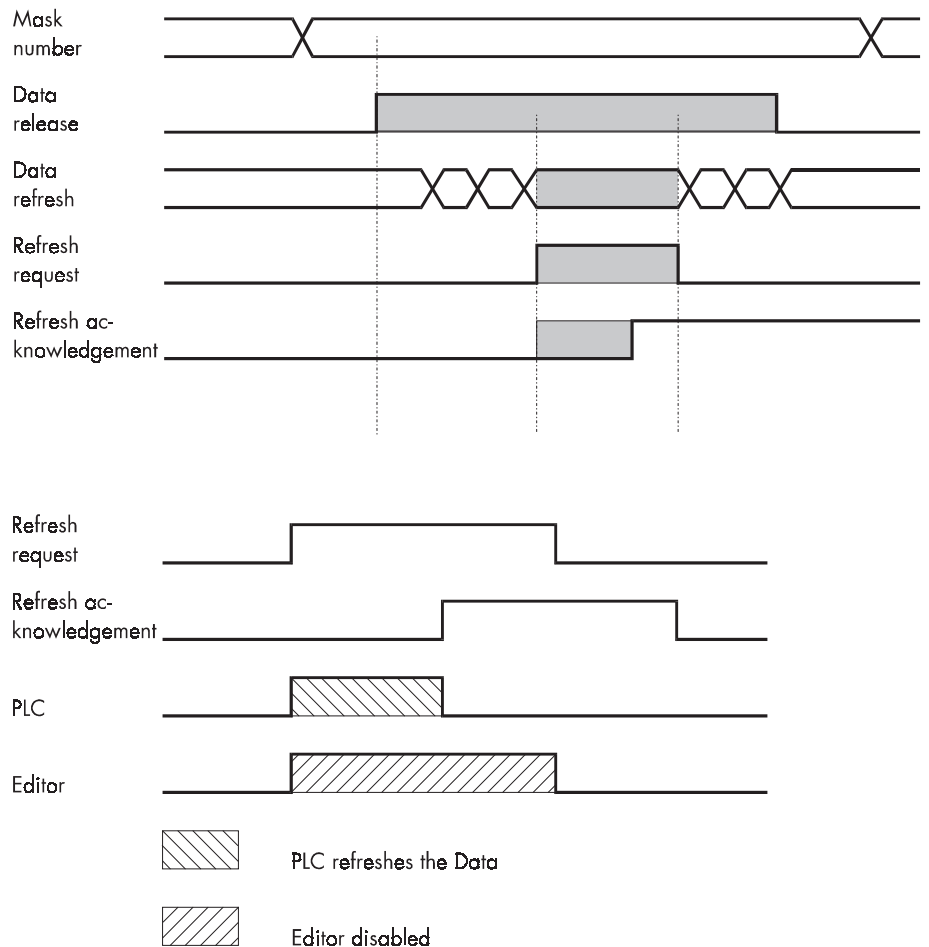


Fig. 11: Pulse diagram - PLC handshake

4.6.7 Refreshing One-Time Output Data

To refresh input or one-time output variables (variables are read only once and are displayed) that a mask contains, simply reactivate the mask (external mask selection).

4.6.8 Modified Data

The conditions that determine the transfer of modified data from the operating terminal to the controller are preset by the user in the variable definition. The following options are available, allowing the user to select the point of time at which the data are transferred:

- with the Enter key (standard)
- with the +/- keys or the Enter key (Incremental Editor)
- transfer continuously (upon modification), i.e. every intermediate status is transferred

In the Incremental Editor, every intermediate value is transferred.

The option to **transfer continuously** (upon every modification) is limited to values that are within the defined limits. If a value is entered that is outside the upper and lower limit, the operator will get a system message to this effect.

If a value with multiple digits is entered that

- exceeds the upper limit
the digit entered most recently will not be transferred to the controller.
- falls below the lower limit,
the previously valid value will be restored.
- falls below the lower limit (until the last digit has been entered) and then exceeds the upper limit (after the last digit has been entered),
the previously valid value will be restored.

4.6.8.1 Input Plausibility Check

The plausibility check is performed on every editable numerical variable. This requires that an upper and lower limit is specified in the variable definition. These limits are displayed in the corresponding output format. System messages are generated when invalid values are entered. Further guidance can be found in the chapters Editors and System Messages.

4.7 Graphics

4.7.1 Images

Any program installed by the user can be used by the programming software as a server for images.

Images can principally be made use of in two ways:

- 1.) Using existing images: The images created in another program can be selected. A copy of the source image will then be stored in the project database and converted to a pixel graphics of the specified dimensions.
- 2.) Creating new images: The program to be used to create the new image can be launched from within the programming system. Once the image has been created, the server is exited and the image is stored in the project database. To display the image in the programming system and on the operating terminal, the image is converted into a pixel graphics of the specified dimensions.

By double-clicking the image in the programming system with the mouse, the server program used to create the image is opened automatically and the image can be edited. The scope of functions available to design the images depends on the features provided by the server program. The placement of images in the programming system can be pixel or grid oriented depending on the operating terminal type.

4.7.2 Graphics on Operating Terminals

Graphics are linked into the user description in the form of static background images for each I/O mask, dynamic variables on a bar chart, dotted lines on a curve or selection images. These basic elements allow the user to design a customized, full-graphics user interface. The curves (trendlines), bar charts and selection images are updated cyclically, so that a simple kind of animation is also possible.

4.7.2.1 Background Images

Background images can be created for masks on the terminal with the aid of graphical objects. Up to ten different background images can be specified for each mask. Background images are defined by means of the name and its position in the mask. You can also select display attributes for each background image. The following attributes are available: normal, inverse or flashing. The size of the displayed image is determined by the definition of the graphical object.

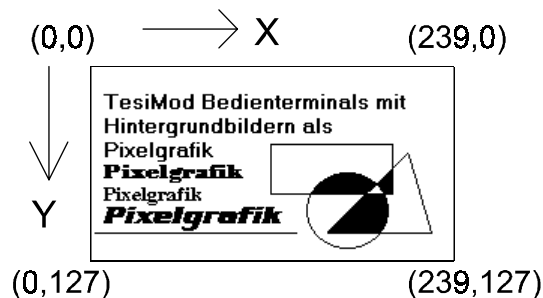


Fig. 12: Example of coordinates

Background images are output on the display one at a time. Their order is determined in the programming system by the order in which they are entered. The images are copied to the display memory in the selected mode. The maximum resolution and size of the bitmaps for background images are equivalent to those of the type of display that is used. Splitting the images into modular segments has the advantage that they can be reused within the system, which above all enables memory space to be optimized. Background images can be used for all static applications, such as:

- Fixed display arrangement
- Representing the soft key row as icons
- Coordinate systems
- Information about production processes
- Static images or icons
- Representing overviews of plants or processes

If several different background images have been defined for a mask, they can also overlay one another. The write-over mode to be used for this process can be specified.

4.8 Recipes

Various logically related variables can be organized into units known as recipes. Unlike mask variables, recipe variables are not transferred to the controller immediately after being entered, but are stored in the terminal as data sets. These data sets are protected against power failure. The data sets can be loaded to the controller as a unit as and when required.

The maximum number of recipes that can be created at programming time is 250. For each recipe, up to 250 data sets can be created. The data sets can either be created at programming time and be stored in the operating terminal's Flash memory together with the user description or can be entered online on the operating terminal and are then stored in the battery-backed RAM. Data sets stored in the Flash memory must be copied to the RAM first before they can be edited. Data sets that have been edited remain in the battery-backed RAM.

Example for the usage of recipes:

Settings of a machine for manufacturing various products:

The product <i>clamp</i> requires:	Variable	Value	Unit
	Material	ST37-3	
	Feedrate	25.00	mm/s
	Setpoint Value Axis 1	43.50	mm
	Setpoint Value Axis 2	56.30	mm
	Cutting Angle	30	°
The product <i>shaft</i> requires:	Cutting Speed	110	mm/s
	Variable	Value	Unit
	Material	X20Cr13	
	Feedrate	20.00	mm/s
	Setpoint Value Axis 1	45.60	mm
	Setpoint Value Axis 2	51.20	mm
	Cutting Angle	45	°
	Cutting Speed	76	mm/s

The variables Material, Feedrate, Setpoint Value Axis 1, Setpoint Value Axis 2, Cutting Angle and Cutting Speed can be organized into the recipe 'Machine Settings for Products'. The variables Feedrate, Setpoint Value Axis 1 and Setpoint Value Axis 2 are defined as floating point numbers or fixed point numbers while the variable Cutting Angle is defined as an integer and the variable Material as a selection text.

The values for manufacturing the products *Clamp* and *Shaft* must be stored as data sets. Whenever another product is to be manufactured, the data set of the product to be manufactured next can be loaded into the controller.

The following check list contains all of the elements that are required and useful for creating and handling a recipe with data sets:

- The recipe itself (texts and variables)
- Data sets with data set number, data set name and variable offset
- I/O mask for the recipe
- Recipe field in the mask
- Recipe buffer (address for the data area in the controller)
- Variable Data Set Number for Transfer from the Terminal
- Variable Recipe Number for Transfer from the Terminal
- Variable Data Set Number for Request from the Controller
- Variable Recipe Number for Request from the Controller
- System variables:

Variable Name	Linkage Partner	Description
SelectDSNr	Selection Text/Decimal Number	Display/Selection of the Data Set Number
SelectDSName	Selection Text Variable	Display/Selection of the Data Set Name
DestDSNr	Positive Decimal Number	Destination Data Set Number for Copy Process
DSCopy	Soft Key/Selection Text Variable	Activation of Data Set Copy Process
DSDDelete	Soft Key/Selection Text Variable	Deletion of Data Set
DSDownload	Soft Key/Selection Text Variable	Loading of Data Set to Controller
ActDSName	Alphanumerical Variable	Entering the Name for RAM-Data Set
SelectRezeptNr	Selection Text/Decimal Number	Display/Selection of the Recipe Number
TabPgUp	Soft Key	Page up
TabPgDn	Soft Key	Page down
Break	Soft Key	Cancel input

4.8.1 Structure of a Recipe

A recipe comprises a maximum of 255 variables. In addition, up to 255 explanatory texts can be programmed. The variables and texts can be spread out over a maximum of 255 lines (with each line stretching across the entire width of the screen). A help text can be programmed for every variable.

The recipe is displayed in a recipe field, within an I/O mask, that extends over the entire width of the screen. The height of the recipe field can be as small as one line or as large as the entire height of the screen. The Page up / Page down keys and the cursor keys can be used to scroll through long recipes in the recipe field.

All one-line formats and Editors that are available in I/O masks can also be used for recipe variables. Multiple-line formats can not be used (for example, multiple-line selection fields, tables, etc.). In addition, neither variables nor texts can be displayed with the zoom option.

4.8.2 Processing Recipes and Data Sets

The majority of the operations described below refer to the active data set. In order to activate a data set, first select the recipe to which it belongs and then the data set itself. How to select recipes and data sets is explained in the next two sections.

4.8.2.1 Selecting a Recipe

All recipes are assigned a number from 1 to 250 when they are programmed. You can select a recipe as follows:

- By means of a fixed assignment between the recipe and a mask. Whenever this mask is opened, the recipe window will contain the recipe that was specified when the mask was originally programmed. If a fixed assignment to a recipe has not been defined when the mask with the recipe window was programmed, the last recipe that was processed appears in this window when the mask is opened.
- By means of the system variable **SelectRezeptNr**. This variable can be edited using any Editor. It is a good idea, however, to use a selection text and assign meaningful names to each recipe number.

4.8.2.2 Selecting a Data Set

Data sets can be assigned both a number from 1 to 250 and a name.

The data set numbers and names are allocated when the data sets are created, in other words either in the programming system for data sets stored in the Flash memory or on the terminal in the case of data sets stored in the RAM. The maximum data set name length is 15 characters. Data set names need not necessarily be unique (though it is recommended that they are).

A data set can be selected in one of the following ways:

- By selecting a new recipe. The associated data set with the lowest number is then selected for it automatically.
- By means of the system variable **SelectDSNr**. This variable can only be edited as a selection text. In this case, only the numbers of those data sets that are available for the active recipe are displayed.
- By means of the system variable **SelectDSName**. This variable can only be edited as a selection text. In this case, only the names of those data sets that are available for the active recipe are displayed.

4.8.2.3 Copying a Data Set

Only the active data set can be copied. To do so, the number of the destination data set is written to the system variable **DestDSNr** and then the value 1 to the system variable **DSCopy**.

The following conditions must be fulfilled in order for the data set to be copied successfully:

- The number of the destination data set must be between 1 and 250 (DSCopy = 2 searches for a free data set, while DSCopy = 3 overwrites the existing data set).
- There must not already be a data set with the same number for the active recipe (unless DSCopy is set to 3).
- The active data set can not be edited at the same time.
- There must be enough free RAM on the terminal.

If any of these conditions is not satisfied, the data set is not copied and a system error message is output.

The destination data set becomes the active data set after it has been copied.

After it has been copied, the name of the destination data set consists merely of blanks. A new name can be defined with the system variable **ActDSName**.

4.8.2.4 Deleting a Data Set

Only the active data set can be deleted. To do so, the value 1 is written to the system variable **DSDelete**.

The following conditions must be fulfilled in order for the data set to be deleted successfully:

- The active data set can not be edited at the same time.
- The data set must be stored in the RAM.

If any of these conditions is not satisfied, the data set is not deleted and a system error message is output.

After the deletion, the data set with the lowest number in the current recipe becomes the active data set.

4.8.2.5 Modifying a Data Set

The active data set can be modified, providing it is stored in the RAM.

To change the contents of a data set, the variables must be edited in the recipe window. Note, however, that the new values are not written in the data set as soon as the Enter key is pressed, but are first stored in a temporary buffer. The Data Release key must then be pressed in order to accept them. If the new values are not to be accepted, the system variable **Break** can be set to 1 to discard the contents of the buffer. It is a good idea to program one of the soft keys to this variable, in order to save time.

Another data set can not be selected until the buffer contents has either been accepted or discarded.

If the controller changes to a different mask while a data set is being modified, or if the external data release is cancelled again before the Data Release key is pressed, the buffer contents will likewise be discarded.

The modified data set is not transferred to the controller automatically. An explicit command from the operator or the controller is necessary first.

4.8.3 Data Set Transfer to / from a Controller

4.8.3.1 Transfer to a Controller

A data set can be transferred to the controller in the following ways:

- The active data set can be transferred to the controller by setting the system variable **DSDownload** to 1.
- The controller can issue a request for any data set. It first sets the variable addresses for the recipe and data set numbers to the required values. It then writes the value **7FFBH** to the address of the serial signaling channel (see section 4.22 *Cyclic Poll Area*).

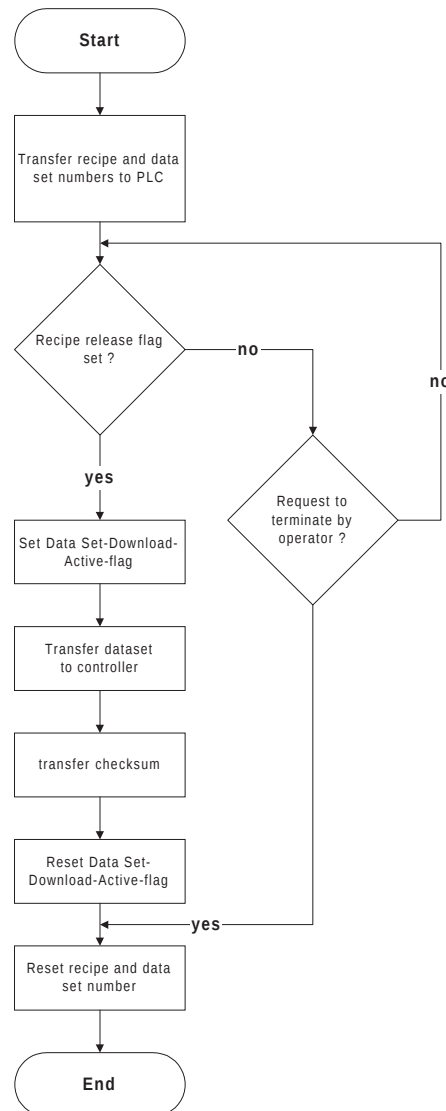


Fig. 13: Data transfer sequence to the controller

The sequence of a terminal-initiated data set transfer is recapitulated below in chronological order.

The process is described from the point of view of the terminal:

1. Transfers the recipe number to the variable addresses reserved for recipe numbers.
2. Transfers the data set number to the variable address reserved for data set numbers.
3. Waits until the data set transfer is released by the controller. For this process, the controller must set the Data Download Release bit (DDF bit) in the Write coordination byte to logical 1.
Until this happens, the transfer can be cancelled again by setting the system variable **DSDnloadBreak** to 1. Once the transfer has been released by the controller, it can not be cancelled again by resetting the Data Download Release bit in the Write coordination byte.

4. Sets the Data Download Active bit in the Read coordination byte.
This bit can be used by the controller to identify an incomplete data set transfer (communication interrupted, power failure). An incomplete transfer is, however, repeated the next time the terminal is booted.
5. Transfers the data set to the recipe buffer defined in the controller. A separate recipe buffer can be programmed for each recipe. The recipe buffer must be at least one byte larger than the data set (on account of the checksum) and consist of an even number of bytes (in other words, an extra byte must be added if necessary).
6. Transfers the checksum. The byte-wise XOR checksum (for the complete data set) is written to the address that follows the last byte of the data set. It provides the controller with additional protection against errors caused by incompletely transferred data sets.
7. Resets the Data Download Active bit in the Read coordination byte.
8. Resets the recipe and data set numbers to the variables that have been programmed in the controller for this purpose. The recipe number is set to 0 and the data set number to 255.

The various steps in this process can be monitored by the operator with the aid of the system variable **DSDnloadState**.

The name of the last data set to have been loaded in the controller is stored in the terminal for each recipe. This name can be displayed for the active recipe with the system variable **LoadDSName**. If a data set has not yet been loaded in the controller for this recipe, or if the data set which was loaded in the controller has already been deleted, a question mark appears on the display instead.

4.8.3.2 Transfer from a Controller

Certain applications may require that individual data set values in the controller are modified (for example, teaching) and that the modified data set is transferred back to the operating terminal. An option to transfer data sets from the controller to the terminal has therefore been provided.

A data set can be transferred from the controller to the terminal in either of the following ways:

- By writing a value to the system variable **StartUpload** to initiate a data set read process for the active recipe. If the value 1 is written to the system variable, the variables will be read individually from the addresses specified during variable programming. If the value 2 is written to the system variable, the variables will be read as a block from the data set buffer defined for the recipe. The number under which the data set is to be stored in the terminal can be specified via the system variable **UploadDSNr**. The default setting for this system variable is 0. If a destination data set number is not entered, the transferred data set will temporarily be stored under the number 0 and will then have to be copied to a valid number (1-250).

- The PLC issues a request for a data set read process. For this process, the controller writes the recipe number and data set number (the same as those for the transfer to the controller on request by the controller) of the destination data set to the variable specified for this purpose. Subsequently, one of the message numbers

7FFDH (variables are read from their specified addresses individually) or

7FFAH (variables are read from the defined data set buffer as a block)

must be transferred. After the data have been transferred successfully, the recipe number is overwritten with 0. If there is no more free memory available in the operating terminal or if the specified data set number already exists in the Flash memory, the recipe number will be overwritten with 255. The data set number will not be overwritten in either case.

If the newly read data set overwrites a data set that already exists in the RAM with the same number, its name will be used for the newly read data set. If a data set does not yet exist with that number, the name will merely consist of blanks.

The read process can be monitored by the operator with the aid of the system variable **UploadState**.

4.8.4 Transferring Data Sets to / from a PC

It is possible to transfer data sets to or from a PC via SER2, in order to back up the data sets that have been stored in the terminal, process the data or supply the terminal with new data sets.

It is also particularly important to back up the data sets if a new user description is loaded in the terminal, as all the data sets in the RAM are then deleted. If the recipe structure remains unchanged, however, they can be reloaded into the terminal again after the user description has been loaded. If changes have been made to the structure of any of the recipes (number of variables, position of the variables in the data set buffer, etc.), only the data sets of the other, unchanged recipes can be reloaded into the terminal.

The data sets are transferred in a format that can be edited using a Text Editor (see section 4.8.4.3 *Structure of the Data Set File*).

The parameters for the interface channel SER2 can be freely configured by means of the corresponding system variables. Merely make sure that the same parameters are set at the PC end. You can send or receive at the PC end with any suitable program, such as Windows Terminal (1).

4.8.4.1 Transfer to a PC

The transfer of data sets to the PC is initiated by writing a value to the system variable **StartSave**. The number of data sets that are transferred depends on the value that is written to the system variable. The following are valid values:

System variable value = 1: Only the active data set is transferred.

System variable value = 2: All of the data sets of the active recipe are transferred.

System variable value = 3: All of the data sets of all recipes are transferred.

The process can be monitored by the operator with the aid of the system variable **SaveState**.

4.8.4.2 Transfer from a PC

The terminal is placed to the Ready-to-Receive state when the system variable **StartRestore** is set to 1. The data sets can then be sent by the PC. The terminal recognizes the end of the data set transfer automatically by analyzing the data it has received. It then returns to its normal state.

To cancel the Ready-to-Receive state again without receiving data, the value of the system variable **StartRestore** must be changed to 2.

The system variable **RestoreState** indicates whether or not the terminal is ready to receive.

If a formatting error is detected in the received data, a system message to this effect is output and the receive process is terminated. The position of the formatting error can be localized, at least approximately, with the aid of the system variable **RestoreLineNr**. This variable contains the number of the last line to have been received.

Data sets can only be stored in the terminal if their structure is still identical to the data set structure specified for the recipe concerned in the user description. This can be checked by the terminal on the basis of a version number (see Structure of Data Set File). If a data set which is found to be invalid is received, it is rejected and a system message to this effect is output. The receive process is not terminated, however.

If a data set with the same number as the transferred data set is already stored in the Flash-Eprom, the newly received data set is rejected without any warning to the operator.

If a data set with the same number as the transferred data set is already stored in the RAM, a parameter setting in the received data (see Structure of Data Set File) determines whether or not the existing data set is overwritten. If it is not supposed to be overwritten, and another data set with the same number already exists in the terminal, the newly received data set is similarly rejected without any warning to the operator.

4.8.4.3 Structure of a Data Set File

The data sets transferred to the PC are generally stored in a file.

If this file is only used for backup purposes, the operator does not necessarily be familiar with its structure. In this case, the file can merely be transferred back to the terminal unchanged when it is needed.

If the data are to be processed further, for example, within the scope of production data acquisition, the operator should understand the structure of the file.

All of the data in the data set file are represented by a simple language specifically developed for this purpose.

The following are elements of this language:

Key words:	S + two further letters. They normally appear at the beginning of a line. Example: SDW or SFA
Decimal number:	Any number of the digits 0-9, preceded by a negative sign when required. Example: 999 or -1234567
Hexadecimal number:	H + any number of the digits 0-9 or letters A-F or a-f. Example: H999 or H123abCD4
Hexadecimal string:	C + any even number of the digits 0-9 or letters A-F or a-f. Example: C12 or CAAFF33
ASCII string:	Any string of characters enclosed between two back-slash characters (\) . Example: \This is one ASCII string\
Comment:	Any string of characters enclosed between two dollar signs (\$). Comments can be inserted in the data set backup file at any position and can stretch across several lines. Example: \$This is a comment\$

Any number of separators (blanks, tab characters or line feed characters) can be placed between these language elements.

The above-mentioned language elements are used to create a file with the following structure:

- **Start of file identifier**
- **Any number of data sets**
- **End of file identifier**

A data set consists of:

- **Data set header**
- **Any number of data set variables**
- **End of data set identifier**

Start of file identifier:

Key: SFA

Parameters: none (date and time are output by the terminal as a comment)

End of file identifier:

Key: SFE

Parameters: none

Data set header:

Key: SDK

Parameters: Recipe number, data set number, data set name (as an ASCII string), data set size in bytes, recipe version number, write-over identifier

Data set variables:

Key: SDW

Parameters: Offset of the variables in the recipe, variable size in bytes, value of the variables (as a hexadecimal string)

End of data set identifier:

Key: SDE

Parameters: none

Explanations:

Recipe Version Number:

On creating or changing the recipe description in the programming system, this version number is increased automatically whenever the structure of the data sets has changed. To be able to load a data set from the PC to the operating terminal, the downloaded version number and the version number stored in the operating terminal for the recipe involved must match. The downloaded data set will not be stored if they do not match.

Write-over identifier:

The value 1 means that the downloaded data set is to overwrite any data set with the same number that may already exist in the operating terminal. The value 0 means that the downloaded set is to be rejected if a data set with the same number already exists. Only those data sets can be overwritten that are not stored in the Flash memory, i.e. that were loaded into the terminal together with the user description.

4.8.5 Printing Data Sets

The data set printout can be started from both the operating terminal and the controller.

To be able to initiate a printout from the operating terminal, either the system variable **StartRezPrint** must be placed into a mask or a soft key must be assigned accordingly. The active data set can be printed via SER2 by writing the value 1 to the system variable.

Writing the value 2 to the same system variable will cancel the print process.

A heading including the recipe number, data set number and data set name will be printed at the beginning of each data set.

The status of the print process can be indicated via the system variable **RezPrint-State**.

To be able to control a print job from the controller, the data set number and recipe number must be entered into the appropriate variables first. The print job is then started by writing the value **7FF8H** to the address of the serial message channel. A value of **0** (zero) in the variable for the recipe number (for request from the terminal) will indicate that the data set is being printed.

If another print job is currently being printed so that the printer can not print the specified data set, the value **255** will be written to the variable for the recipe number (for request from the terminal).

4.8.6 Memory Requirements for Storing Data Sets

The RAM in the terminal that is not required by the system (approximately 110,000 bytes) is used to store messages as well as data sets that have been stored in the RAM.

The size of the message buffer is configurable. Each message takes up 24 bytes. This makes a total of 12,000 bytes for the default message buffer size (500 messages), so that a further 98,000 bytes are available for storing data sets.

Space is also needed to store the data set name and management information (additional 28 bytes per data set).

Example:
If the data set size is programmed as 22 bytes, a total of
 $98,000 / (22 + 28) = 1960$
data sets can be saved in the RAM (message buffer size: 500). Other, fixed programmed data sets can also be stored in the Flash-Eprom.

4.9 Message System

The message system is an integral part of the operating concept. Messages are reactions to events that enable these events to be communicated to the operator in an intelligible form. A distinction is made between internally and externally generated messages, depending on where the event occurred. The diagram below shows the structure of the message system.

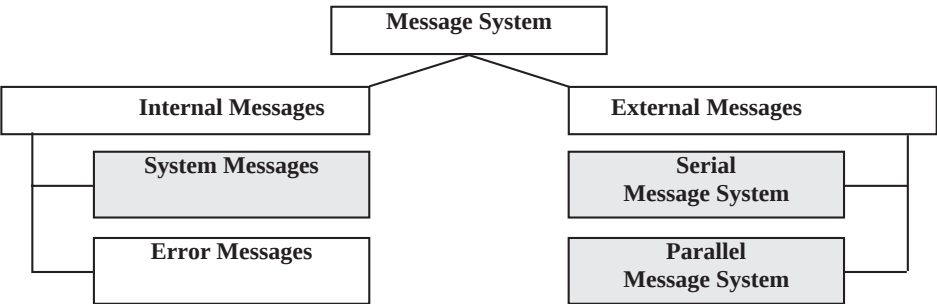


Fig. 14: Structure of the TesiMod message system

The grayed parts of the messages are freely configurable.

4.9.1 Internal Messages

Internal messages are all messages that are sent by the operating system to the operator. There is a further subdivision between system messages and error messages. The user (programmer) can not influence the generation of these messages.

4.9.1.1 System Messages

System messages are generated by the operating system as a result of internal plausibility checks. A system message is activated immediately after the corresponding event has occurred.

Pending system messages are signaled to the operator by a flashing Help key. If the Help key is pressed, the system message text will be displayed in its full length

for as long as the Help key is held down. If several system messages are pending at the same time, they will be displayed in order of their system numbers, whereby the system message number '1' represents the highest display priority.

The user has the option of editing system message texts with the programming software. The size of one screen is available for each message text. The system message text can be freely designed using the terminal-specific font. Additional character attributes or graphics are **not** possible.

The output of the texts is language-specific, i.e. if the operator guidance is multi-lingual, the system messages are displayed in accordance with the selected language.

System messages are assigned in the programming software via the system message number. The system message number stands for a predefined event. A brief info consisting of 20 characters is used to provide an explanation of the system number. The length of the texts is designed to allow them to be displayed directly on one line, even on the smallest display in the TesiMod operating terminal series. When a selection is made in the programming software, the brief description is provided next to the number as a message name. A newly created system contains the following system messages with brief infos:

System Message No.	Brief Info
1	Wrong format
2	Value too large
3	Value too small
4	Replace battery
5	Message overflow
6	New message
7	Message buffer full
8	Invalid mask no
9	Invalid message no
10	Print log invalid
11	Interface in use
12	Invalid password
13	Password unchanged
14	Overvoltage
15	Data set protected
16	Illegal data set
17	Data set unknown
18	Data set memory full
19	Data set active
20	Data set transfer
21	Password missing
22	Editing mode active
23	Data set file error
24	Data set format
25	(Floating point) Number invalid
26	Loop through active
27	No data set address
28	Recipe unknown
29	Data set download

System Message 1 'Wrong format'

This text indicates that an attempt has been made to enter an invalid data format into a variable field of the Numerical Editor. For example, the number of pre-decimal places entered exceeds the setting specified in the user description.

System Message 2 'Value too large'

This indicates that an attempt has been made to enter a value into a variable field of the Editor that exceeds the variable's upper limit. The upper limit is defined in the user description. If the text is deleted from the system message, no system message will be displayed if a value is entered that is too large. In this case, the maximum value that is valid is entered instead.

System Message 3 'Value too small'

This indicates that an attempt has been made to enter a value into a variable field of the Editor that falls below the variable's lower limit. The lower limit is defined in the user description. If the text is deleted from the system message, no system message will be displayed if a value is entered that is too small. In this case, the minimum value that is valid is entered instead.

System Message 4 'Replace battery'

This text is displayed when a test performed on the battery indicates that its capacity has fallen below the limit values. This capacity test is repeated every 60 minutes. To retain the recipe data and the data stored in the message memory, the battery must be replaced within 2-3 days of initial display of this message. To avoid loss of data when replacing the battery, the information in the chapter 1.9 *Battery* must be complied with. The same message appears when the battery is removed, switching the terminal off at this point will, however, result in the battery-backed data being lost.

System Message 5 'Message overflow'

This text indicates that the system is unable to process the external messages quickly enough. Upon display of this message, one message has already been lost.

System Message 6 'New message'

This text is displayed when the Help key is pressed and the terminal has received a new external message whose priority exceeds the programmed threshold value and no direct selector key has been assigned to the message mask.

System Message 7 'Message buffer full'

This text is displayed as a warning that the next external messages may overwrite the oldest or lowest-priority messages (depending on the configuration).

System Message 8 'Invalid mask no'

This text is displayed to indicate that a non-existing mask number has been transmitted by the PLC via the serial message channel.

System Message 9 'Invalid message no'

This text indicates that the PLC has attempted to activate a non-existing message number.

System Message 10 ... 'Print log invalid'

The operator or the PLC has attempted to activate a non-existing print log.

System Message 11'Interface in use'

SER2 is already being used by another print job. An attempt has been made to transmit different types of data to the printer at the same time (e.g. to print recipes and messages).

System Message 12'Invalid Password'

The operator has entered a password which does not exist in the password list. With this message, the previous access authorizations (view and edit level) are reset.

System Message 13'Password unchanged'

The operator did not enter the same new password two times in a row.

System Message 14'Overvoltage'

The system has detected that the supply voltage is too high. Switch the terminal off immediately to avoid damage. Check supply voltage.

System Message 15'Data set protected'

An attempt has been made to modify individual values of a data set stored in the flash or to delete the entire data set.

System Message 16'Illegal data set'

The data set number specified as the destination for the data set copy process exists already or is outside the valid range (for example, flash).

System Message 17'Data set unknown'

A data set has been selected whose number does not exist in the data set list.

System Message 18'Data set memory full'

An attempt has been made to create a new data set, but the data set memory is full.

System message 19'Data set active'

An attempt has been made to erase or to copy to the active data set or to select a data set even though the active data set is currently being edited.

System Message 20'Data set transfer'

An attempt has been made to initiate a data set transfer to the controller even though the previously initiated transfer has not yet been completed.

System Message 21'Password missing'

An attempt has been made to switch to a password-protected mask or to edit a password-protected mask without having entered a password with sufficient authorization.

System Message 22'Editing mode active'

An attempt has been made to perform a change of mask while the terminal was in the editing mode.

System Message 23'Data set file error'

The data set file loaded from the PC to the terminal contains a syntax error. The error can be localized by means of the line number system variable.

System Message 24'Data set format'

The size or internal version identifier of a data set loaded from the PC to the terminal and the corresponding values in the programming software do not match.

System Message 25'(Floating point) Number invalid'

The bit pattern read from the controller is not valid for a floating point number. The number is displayed as 0.0.

System Message 26'Loop-through active'

The selected action was not performed due to an active loop-through operation.

System Message 27'No data set address'

The addresses for the data set transfer did not exist at the time of the controller's request.

System Message 28'Recipe unknown'

An attempt has been made to select a recipe that does not exist in the terminal.

System Message 29'Data set download'

A data set transfer to the controller (download) has been initiated, but the Data Set Download Release bit in the Write coordination byte (bit 4) has not yet been set by the controller.

4.9.1.1.1 Suppressing the Display of System Messages

The operating concept provides you the option of suppressing the display of system messages. The display of a system message can be prevented by erasing the system message in the programming software.

Example:

System message 7 'Message buffer full' is to be suppressed. Older messages or messages with a lower priority are to be overwritten.

The system message text in the programming software is erased.

By suppressing the display of this system message, the user agrees that incoming messages automatically overwrite the oldest messages or those with the lowest priority once the message buffer is full.

4.9.1.2 Error Messages

The messages listed below are displayed by the operating system. The message texts are part of the operating system and are displayed in English. The size of the texts has been chosen in such a way that they can be displayed on every terminal. The output of these texts can not be suppressed and their contents can not be modified. The term 'error message' is used because the terminal does not operate in accordance with the true meaning of the TesiMod standard mode while these messages are displayed. In addition to true system errors, various conditions and processes are also described.

```
COMMUNICATION ERROR
CODE          X
SUBCODE:      X
RETRIES:      XXX
```

..... This message is generated for all types of protocol and interface errors. The error codes (CODE X) and SUBCODE (X) are protocol-specific and are listed in the respective application documentation. The connection with the communication partner has been interrupted. RETRIES displays the number of unsuccessful attempts to establish a connection. This number is incremented while the device is running. The number of retries depends on the protocol that is being used.

```
ADDRESS ERROR
```

..... This message may be displayed during a download. The S3 file addresses physical addresses in the terminal. The transmission is aborted as soon as invalid addresses are detected during this process. The starting address of the invalid line in the S3 file is specified in hexadecimal format.

```
FLASH MEMORY FAILURE
```

..... Is displayed during a download if the Flash-Eprom can not be programmed. This message indicates that the application memory is defective. The starting address of the invalid line in the S3 file is specified in hexadecimal format.

```
CHECKSUM ERROR
```

..... An error has occurred during transmission of the user description. The error has either occurred during the serial transmission or the S3 file contains invalid lines or no valid S3 file has been transmitted. Recompile the application description and attempt to retransmit.

```
BYTECOUNT OVERFLOW
```

..... Error during transmission of the user description. An error was detected in the S3 file of the application description. More bytes were received in one of the transmission lines than specified in the byte count.

```
FORMAT ERROR
```

..... Transmission format of the user description contains errors. The output file used has not been generated by this programming system. The transmitted file did not contain S0, S3 or S7 lines, no S3 format was used.

```
TURN POWER OFF
RESET DIP-SW 4
OTHERWISE ALL FLASH-
DATA WILL BE LOST
```

..... The mode selector switch S4 was at the 'on' position when the supply voltage for the terminal was switched on. The Flash data will be retained if the following instructions are complied with. Switch the terminal off, set S4 to 'off' position, switch terminal on - data will be retained and the terminal will function as before. If S4 is set to the off-position while power is on - data will be lost, terminal switches to the download mode!

```
DIFFERENT MASK VERS
          EPROM FLASH
VERSION   XXX   XXX
REVISION  XXX   XXX
```

..... The version of the programming system and the operating software of the terminal do not match. This error occurs if the wrong operating system version was selected for compilation of the user description. The two program versions must match.

```
DIFFERENT DRIV VERS
          EPROM FLASH
VERSION   XXX   XXX
REVISION  XXX   XXX
```

..... The protocol driver loaded via the programming system and the terminal's operating system do not match. The two program versions must match.

```
!!!!!! WARNING !!!!!
NONE DEFAULT PARA-
METERS ON SERIAL
PORT X2 USED
```

..... The parameters of SER1 were modified. To achieve an operational connection, both communication partners must be set to the new parameters. This message is used for informational purposes if the connection to the communication partner can not be established.

```
!!!!!! ERROR !!!!!
NO PROTOCOL-DRIVER
IN MASK DESCRIPTION
FOUND
```

..... The operating system can not find a protocol driver in the user description loaded into the terminal. Select a protocol, recompile the user description and activate another download.

```
!!!!!! ERROR !!!!!
PLC TYPE MISMATCH
BETWEEN TERMINAL
AND MASK-DESCRIPTION
```

..... The protocol selected in the programming system when creating the user description and the terminal hardware are not compatible. For example, the Interbus-S protocol driver has been loaded to a device with standard interfaces.

```
KEYBOARD ERROR
PLEASE RELEASE KEY
```

..... A self-test is performed when the terminal is switched on. This error message is generated if a key is pressed while the keyboard is being checked. Please release the key. If this message appears when no key is pressed, the message indicates a defective keyboard!

```
INITIALIZING
MESSAGE BUFFER
```

..... When the terminal is switched on, all messages in the terminal are sorted. This initialization process requires a certain length of time based on the number of stored messages. The message is always generated, but is only displayed for a very short time period or is not visible at all.

```
ERASE FLASH EPROM
```

..... Is displayed while the mask memory is being erased. All of the programmed data are erased at this point.

```
FLASH IS ERASED
```

..... This message indicates that the Flash has been erased. SER2 is initialized for the download mode.

```
DOWNLOAD 1
FLASH xxx kByte
```

..... The operating terminal indicates that it is ready for a download with a baud rate of 19200 Bd via SER2. A new project can now be loaded or new interface parameters for the transfer can be exchanged.

```
DOWNLOAD 2
```

..... The operating terminal indicates that it is ready for a download with the new interface parameters. If no data are received within 20 s, the operating terminal will return to the DOWNLOAD 1 condition.

```
AUTO REBOOT
```

..... The operating terminal will reboot within the next few seconds.

```
INITIALIZING
CPU    XX MHz
Flash XXX kByte
XXXXXXXX YYYYYYYY
```

..... The operating terminal reported its parameters during the startup process:

- CPU frequency in MHz
- Size of the Flash memory in Kbytes
- Eprom version number XXXXXXXX
- Loaded PLC driver YYYYYYYY

IDENTIFY MEMORY-TYP

..... The Flash memory type used is being identified.

!!! HIGH VOLTAGE !!!

..... The voltage applied to the operating terminal is too high. This message will not disappear until the specified supply voltage has been reached.

ERROR ASYNCHRONOUS
SERIAL I/O UNIT 0

..... Initialization of the serial interface (unit 0 or unit 1) failed.

KEYBOARD-MODUL
HARDWARE-VERSION
NOT CONFORM TO
DRIVER-VERSION

..... The program level of the keyboard card and the current firmware are not compatible. Retrofit the operating terminal. The subcode specifies the level of the keyboard card.

FIRMWARE UPDATE
SUCCESSFUL
AUTO REBOOT

..... Indicates a successful update operation. The operating terminal reboots automatically.

SYSTEM ERROR
CODE :
SUBCODE:
RETRIES:

..... A fatal error has been encountered. If this error message is generated, make a note of the firmware level and the hardware version and contact OMRON EUROPE B.V., Wegalaan 67-69, NL-2132 JD Hoofddorp, Tel.: +31 23 56 81 300.

UNEXPECTED INTERRUPT
NR =
IP =
CALL HOTLINE

..... An unexpected interrupt has occurred. Make a note of the interrupt number (NR) and the level of the program counter (IP) and contact OMRON EUROPE B.V., Wegalaan 67-69, NL-2132 JD Hoofddorp, Tel.: +31 23 56 81 300.

FLASH NOT ERASEABLE

..... Is displayed after the terminal has been switched on or prior to a download to indicate that the Flash-Eprom can not be erased.

WRONG S3-FILE

..... Is displayed at the beginning of a download to indicate that the S3 file is not the correct type for the terminal being used.

NO FLASH EPROM

..... This message is displayed to indicate that no Flash supported by the programming algorithm is found.

```
FLASH CHECKSUM ERROR
```

..... The user description stored in the FLASH contains errors. This error may occur at the end of a transmission, e.g. if the transmission is not complete or after a terminal is switched on that has a defective memory.

```
TERMINAL-TYP IS XXXX
```

..... An attempt has been made to load a S3 file which was intended for another terminal type. When this error occurs, the correct type for this terminal is displayed at the 'XXXX' position. Re-compile using this selection in the programming system.

```
MEMORY IS FLASH XXXK
```

..... An attempt has been made to load a S3 file which was created for a larger mask memory. The amount of memory space requested by the S3 file and the memory available in the terminal do not match. When this error occurs, the memory size available in the terminal is specified, in kBytes, at the 'XXX' position. This value must be specified in the programming system when compiling.

```
FATAL ERROR  
CODE : XXXXX  
SUBCODE: XXXXX  
CALL HOTLINE
```

..... An error message that should not occur, but which exists. The operating system of the terminal generates this error if proper operation is no longer possible due to a lack of plausibility. To be able to reconstruct the incident, we need to know the code and subcode number as well as the software versions of the operating system and programming software. Do not hesitate to call our hotline and we will help you.

```
FIRMWARE NOT CONFORM  
TO HARDWARE  
FIRMWARE 1  
HW_VERS. 2
```

..... If this error message is displayed, contact OMRON EUROPE B.V., Wegalaan 67-69, NL-2132 JD Hoofddorp, Tel.: +31 23 56 81 300. Before calling, make a note of the firmware and hardware version. The operating system of the operating terminal switches into an endless loop to prevent damage to the device.

```
DATASET STORAGE  
FAILURE
```

..... A checksum error has been detected when checking the memory areas of the recipe data sets. Either the battery or the RAM-memory is defective.

4.9.2 External Messages

External messages are generated by the connected controller and forwarded to the operating terminal as information on the monitored process. The user can choose two separate message systems. Depending on the requirements, message transfers to the operating terminal can be either serial or parallel, regardless of whether the messages are process or fault messages.

Messages can consist of the message text and a scaled and formatted variable. Every variable type available in the system is valid.

The information in the message memory can be used for statistical evaluations. The message is assigned between the terminal and the controller by means of a message number. The associated texts and variable specifications are stored in the terminal together with the user description. The function of a message and its contents are determined by the user when the user description is created in the programming system.

All of the external messages are stored in the message memory in chronological order or in order of priority. In addition parallel messages can optionally be stored in the serial message memory and as a result also evaluated statistically. If the message contains a variable, its value will be frozen in the message memory.

4.9.2.1 Structure of an External Message

An external message is made up of:

- a message number between 1 and 9999
- a message text with up to 255 characters
- one variable at maximum.

When creating a new application, existing messages (one or all) are reusable.

4.9.2.1.1 Assigning Message Numbers

With external messages, message numbers also determine the priority of a message, with message no. 1 being the highest and message no. 9999 being the lowest priority. It is not necessary that message numbers are assigned contiguously; this allows messages with related contents to be grouped.

The assignment of message numbers for status messages always starts with '1'. Make sure that the serial and parallel message system do not overlap. If the two message systems are to be independent from one another, ensure that the message numbers for the serial system start above those for the status messages. In addition, message numbers and mask numbers must be coordinated accordingly when the programming is carried out (see chapter 4.9.2.3.1 *Full-Page Message Output*).

The system allows status message texts to be used in the serial message system.

4.9.2.1.2 Message Buffer Size

The total message buffer size is designed for management of up to 3000 messages. Such a large number of data requires a corresponding computing capacity when the messages are sorted, resorted and initialized. Since this large number may not always be needed, the user has the option of setting the maximum system message buffer size to suit his own needs. The default buffer size allows 500 entries.

When determining the message buffer size, it must be considered that approximately 50 pages of paper are needed when the entire message buffer containing 3000 messages is printed!

The message buffer is output in the message mask. The message sorting criteria can be set via a system variable.

4.9.2.1.3 Message Texts, Variables

The maximum text length including a formatted variable is 255 characters. The programming system does not allow longer texts to be input. All characters that are available in the respective terminal can be used in standard size. One output variable can be included per message text. The variable output format is the same as that of one-time output variables (variables are transferred only once and are then displayed) in I/O masks. Individual messages can for example be modified with selection text or used for several statuses. The output format of the message line can be modified during operation in a configuration mask for the message mask. The capabilities for serial and parallel messages are the same.

Example: Complete message format:

No.	Date	Time	Text1	Variable	Text2
1234	25:08:92	11:30:00	Temperature	285 °C	at Meas. Pt.07

The message consists of the following elements:

1234	4-digit message number
25.08.92	Date, is stored in the terminal when the message is recorded by the terminal
11:30:00	Time, is stored in the terminal when the message is recorded by the terminal
Temperature	Text (1) preceding variable
285	Value of the variable at the time of the message
°C at Meas. Pt.07	Text (2) following variable

4.9.2.1.4 Sorting Messages

Messages can optionally be displayed in the message mask according to their time of arrival or priority. The desired option can be selected when the system is programmed. If both possible message systems are used, it is possible to select the settings separately. The settings are stored in the system variables **RepmanRepSortCrit** and **RepmanSortCritP**. They can be changed during operation on the terminal using a configuration mask, for example. If a configuration option is not offered on the operating terminal, the settings selected by the user will be used.

Sorting algorithm for the serial message system:

- 0 - by priority
- 1 - by time (newest first)
- 2 - by time (oldest first)

Sorting algorithm for the parallel message system:

- 0 - by priority
- 1 - by time (newest first)
- 2 - by time (oldest first)

4.9.2.1.5 Message Priority for Direct Display

The priority of a message is determined by its message number. The higher the message number, the lower the priority. The value that represents the upper limit for the message number that is to be signaled on arrival by a flashing LED or by outputting a system message can be entered, via the programming system, into the system parameters of the message system. See also 4.9.2.2.1 *Direct Selection of the Message Mask*.

If the value = 0 is entered, newly received messages will be not be signaled.

4.9.2.1.6 Printing the Message Memory

The memory contents of the serial and parallel message systems can be printed either in full or in part.

The entire contents of the **serial** message memory are printed if the system variable **PrintAllRep** is set to 1 (formatted printout) or 2 (full-length printout).

The entire contents of the **parallel** message memory are printed if a soft key that is linked to the system variable **PrintAllState** is pressed.

To print the message memory in part, the messages to be printed must be selected in the message mask. This is done by pressing the Data Release key in the message mask and selecting the messages in the message field using the Cursor up and Cursor down keys. The print job is started by pressing a soft key linked to the system variable **BlockPrint** (prints visible part of the selected block) or **BlockPrintLong** (prints messages of the selected block in full length).

The system variables can additionally be included in a configuration mask and be edited during operation of the terminal.

4.9.2.2 Message Mask, Status Message Mask

Message masks or status message masks are I/O-masks that contain a message field defined for serial or parallel messages. The key assignment for a message mask or status message mask is specifically designed to allow convenient navigation within the extremely large message memory.

- | | | |
|---------------------------|---------------|--|
| - Data release not active | Cursor up: | Moves the cursor up one message. The cursor can be moved upwards until the top message is reached. |
| | Cursor down: | Moves the cursor down one message. The cursor can be moved downwards until the bottom message is reached. |
| | Cursor left: | Moves the cursor up one screen (repeat function). The cursor can be moved upwards until the message at the top is reached. |
| | Cursor right: | Moves the cursor down one screen (repeat function). The cursor can be moved downwards until the message at the bottom is reached. |
| | Minus: | Moves the cursor to the bottom-most message. |
| | Plus: | Moves the cursor to the top-most message. |
| | Delete: | Inactive |
| | Data Release: | Enables the editing mode, provided the external data release has been set and the entered password has a sufficient access level. |
| | Enter: | The entire message at the cursor position will appear on the display. |
| - Data release active | Cursor up: | Selects individual messages from the current cursor position on upwards. |
| | Cursor down: | Selects individual messages from the current cursor position on downwards. |
| | Cursor left: | Selects messages in page-mode from the current cursor position on upwards until the topmost message is reached. No repeat function. |
| | Cursor right: | Selects messages in page-mode from the current cursor position on downwards until the bottommost message is reached. No repeat function. |
| | Delete: | Deletes all selected message entries. If no selection is made, the message at the cursor position will be deleted. |
| | Minus: | Selects all messages from the current cursor position down to the last message. |
| | Plus: | Selects all messages from the current cursor position up to the first message. |
| | Data Release: | Exits the editing mode, provided the external data release is still set. |

4.9.2.2.1 Direct Selection of the Message Mask

In the programming software, a function key can be linked to a message mask. This function key can then be used to switch to the message mask from within any mask. This provides another means to reach the message mask, in addition to the access via a selection menu. In this case, the arrival of new messages will be signaled by the integrated function key LED which will begin to flash (instead of the Help key). Direct access to the message mask can be obtained by pressing the flashing function key. If this function key is pressed again, the message mask will automatically be exited and the previous mask will reappear on the display.

4.9.2.2.2 Output Formats for Messages

The following information can be output with every external message:

- Message number
- Date
- Time of day
- Message text
- Value of a variable at the time the message was generated (if available only)

Various system parameters are available to influence the representation of a message in the message mask or the message printout. These parameters can be set in a configuration mask, provided such a mask is programmed. The selection or deselection of message elements is carried out by means of system variables.

Serial Messages	Parallel Messages	Influenced Element
- RepoutNr	RepoutNrP	Message number
- RepoutDate	RepoutDateP	Date
- RepoutTime	RepoutTimeP	Time
- RepoutAnzYear	RepoutAnzYearP	2 or 4 digit representation of the year

This allows the length of the message line to be influenced. Modifications to the output format have no impact on the stored information.

Possible output versions are as follows:

Complete message format:

No.	Date	Time	Text1	Variable	Text2
1234	25:08:92	11:30:00	Temperature	285 °C	at Meas. Pt.07

Versions:

1234 25:08:92 Temperature 285 °C at Meas. Pt.07
1234 11:30:00 Temperature 285 °C at Meas. Pt.07
1234 Temperature 285 °C at Meas. Pt.07
25:08:92 11:30:00 Temperature 285 °C at Meas. Pt.07
11:30:00 Temperature 285 °C at Meas. Pt.07
25:08:92 Temperature 285 °C at Meas. Pt.07
Temperature 285 °C at Meas. Pt.07

4.9.2.2.3 Zooming Messages

Messages are displayed in a one-line format in the message mask for the sake of clarity. In order to display a longer message in its full length, the message must first be selected and then the Enter key pressed.

Message mask line, for example on the NT18S-SF121B-E:

1234 25:08:92 11:30:00 The measuring point in

Zoomed view:

1234 25:08:92 11:30:00

**Measuring point 137 in the furnace has a
temperature of 285 °C**

The zoomed view remains active for as long as the Data Release key is held down. With smaller displays with 4 x 20 characters only the message text is zoomed. The terminal type that is to be used must be considered when the text is programmed, to ensure the lines are wrapped correctly.

4.9.2.2.4 Acknowledging Messages

Message acknowledgement in the controller can be carried out by means of variables. Various Editors or function keys (soft keys) are suitable for this purpose. The acknowledgement enables the controller to delete the message and verify the status of the process.

4.9.2.3 Serial Message System

A 2 byte compartment is used in the cyclic poll area for the transmission of serial messages. The byte order depends on the selected data type of the poll area (see Poll Area). The PLC stores a 16 bit message number in this transmission compartment. The operating terminal polls the entire poll area of the PLC at cyclic intervals and transmits the serial message in the process. Upon detecting a message (message number > 0), this message is stored in the internal message memory of the operating terminal and the compartment in the PLC is reset to zero. The value 0 in the compartment indicates to the PLC that the message has been fetched by the terminal. The polling time of the cyclic data area is configurable.

The same procedure is used to address external masks and message masks. Whenever the number transmitted corresponds to a mask number, this mask is displayed. If a mask and a message text exist for this number, the mask (message mask, full-page fault message text) is displayed and the associated message text is entered into the message memory.

Make sure that the message number is always entered in the serial data compartment as a 16 bit command. As a result of asynchronous processing of some data transfer protocols, evaluation of the message number may lead to problems if the message number has been entered with single-byte commands.

4.9.2.3.1 Full-Page Message Output

The full-page message is a combination of message processing and external mask selection. For a full-page message mask output, a mask and a message text must be programmed under the same number. The controller calls up the 'external mask' through the serial message channel. When it is called up, the mask is displayed and the associated message text is entered into the message memory. As the display contents can be chosen freely, it is possible to realize a message mask, a full-page error output or other contents types. To be able to return to the previous menu from here, at least one mask parameter must be programmed for 'Previous mask'. Message masks can also consist of several masks or even complete structures for troubleshooting.

It goes without saying that a separate, full-page help text can also be programmed for each full-page message.

4.9.2.3.2 Messages Directly to a Logging Printer

When serial messages are logged directly, the printer always runs synchronously. Every new message arriving via the serial message channel is printed immediately and is transferred to the message memory in parallel. Here, attention must be paid that the printer can only process one print job at one time. Every print request must be ended before any further print request is started by the system.

The output of the messages to a printer can be influenced by the system variable **RepmanRepPrint**.

The settings that apply when the formatted type of printout is selected are the same as those selected for the display of messages in the message mask.

The settings for the printout can be changed on the terminal during operation.

As the output consists of a pure text file, the message can also be read by a host computer or a PC. With a further system variable **PrintAllRepLong**, the full length of the message can be output.

4.9.2.3.3 Erasing the Message Memory Externally

The internal message memory of the serial message system can be erased externally, that is from the controller. To do this, a symbolic variable name for the deletion variable must be specified in the Message System option of the parameters file using the programming software. Two bytes are needed in the controller for the variable.

The terminal always checks the deletion variable in the controller once it has received the deletion sequence (deletion code **7FFE_H** via the serial message channel). The internal message buffer is deleted if the deletion variable contains the bit pattern **E216_H**. The deletion variable increases protection against unintentional deletion.

If deletion is not required, the variable should be reset or no symbolic name should be specified in the programming software.

4.9.2.3.4 Information about the Serial Message System

How do messages reach the terminal?

In the controller, a word variable, as a part of the cyclic poll area, is reserved for the message number. This variable is polled by the operating terminal. If the variable value is greater than zero, this indicates that this is a numeric value of a message, i.e. the message number. The message number is placed into the terminal's message memory and the variable in the controller is overwritten with zero. This is treated as the acknowledgement for the controller that the terminal has fetched the message. The next serial transfer of messages can now take place.

How is a new message recognized?

By flashing of the Help key or by a flashing function key, the terminal signals to the operator that a new message has been received. This visual indication only appears when a value has dropped below a limit (message priority for direct display).

How is the message mask configured?

The message mask is preceded by an I/O mask, the configuration mask, and output formats can be defined in it by means of various system variables.

How can the most current messages be output in any chosen I/O mask?

A system variable **RepoutRepText** is available which always contains the message most recently received or the one with the highest priority. The content is retained up to the next change. The output is always left-justified and as defined in the configuration menu. There are also the following system variables for complete or multiple-line output:

RepoutRepText21

RepoutRepText41

RepoutRepText61

See description of the system variables for more information.

Where are messages displayed?

In the message mask. The message mask can be reached via the node mask, the I/O mask, control keys or soft keys.

Owing to the fact that many control keys are used in the message mask, it can only be exited by pressing the Home key or function keys.

4.9.2.4 Parallel Message System (Status Messages)

The parallel message system complements the serial message system. The messages are transmitted in parallel and are evaluated in the terminal. During this process, the current message status is compared with the prior status in the terminal. These messages are automatically deleted from the memory once they are no longer pending. New messages are added to the memory. The current status of the messages can be output.

Date and time are included in every message to indicate the point of time at which a message has been generated.

The message buffer length is limited to a maximum of 256 bytes.

The length is to be set in the system parameters for the message system. There may be restrictions regarding the length depending on the protocol.

Status messages are retained in the message memory only as long as they are being reported by the controller.

Status messages can be transferred on a time or event controlled basis.

4.9.2.4.1 Number of Bytes for Status Messages

This parameter specifies the number of bytes to be transferred with the parallel message system. The size depends on the number of status messages. The absolute size also depends on the defined data type (address). For example, if a word address has been defined and the number of bytes is odd, then this number will automatically be rounded up.

4.9.2.4.2 Image of the Status Messages

For status messages, the starting address of the storage area containing the message information is entered here. The symbolic name is assigned to a fixed address. The assignment of message number to bit information depends on the data type of the controller variable. The data type has been processed in a protocol-specific manner since the structures of byte, word and Lword are not identical in the supported controllers. It is important that once an assignment method is chosen, it is not changed!

Byte-oriented assignment:

Byte address + 0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte address + 1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte address + 2	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte address + 3	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
:				:				
Byte address + n	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Message 1 = Byte 0, Bit 0

Message 2 = Byte 0, Bit 1

Message 3 = Byte 0, Bit 2

:

Message 8 = Byte 0, Bit 7

Message 9 = Byte 1, Bit 0

Message 10 = Byte 1, Bit 1

Word-oriented assignment:

Word address + 0	Bit 15	Bit 14	Bit 13	Bit 12	...	Bit 3	Bit 2	Bit 1	Bit 0
Word address + 1	Bit 15	Bit 14	Bit 13	Bit 12	...	Bit 3	Bit 2	Bit 1	Bit 0
Word address + 2	Bit 15	Bit 14	Bit 13	Bit 12	...	Bit 3	Bit 2	Bit 1	Bit 0
Word address + 3	Bit 15	Bit 14	Bit 13	Bit 12	...	Bit 3	Bit 2	Bit 1	Bit 0
:				:					
Word address + n	Bit 15	Bit 14	Bit 13	Bit 12	...	Bit 3	Bit 2	Bit 1	Bit 0

Message 1 = Word 0, Bit 0

Message 2 = Word 0, Bit 1

Message 3 = Word 0, Bit 2

:

Message 16 = Word 0, Bit 15

Message 17 = Word 1, Bit 0

Message 18 = Word 1, Bit 1

The same rules apply to other data formats.

4.9.2.4.3 Time-Controlled Transfer of the Status Message

The transmission of parallel messages is activated either in a time-controlled mode by the terminal or an event-controlled mode by the controller. The entire message buffer is transmitted at once.

Both transmission methods can be implemented at the same time. This permits longer polling intervals (every 5 to 10 s) to be defined. Critical or important messages can additionally be transferred on an event-controlled basis. The PLC is automatically polled for the status messages after the polling time has elapsed. The selected polling time should not be too short since the status message transfer can take up a longer amount of time (depending on the number of messages and the protocol).

No event-controlled transfer will take place if the polling time is set to 0,0 s.

4.9.2.4.4 Event-Controlled Transfer of the Status Message

The event-controlled transfer is activated by the controller by writing the event code **7FFF_H** to the serial message channel. This code causes the parallel message system to be updated to ensure the messages in the operating terminal's parallel message buffer are current.

To be able to use the event-controlled transmission (also for polling times of 0,0 s), the parameters file must contain the symbolic variable name and the length of the status messages.

4.10 Help System

A screen-sized help text can be assigned to each mask and editable variable. If no specific text is programmed, a default help text will be displayed instead. This default help text can be freely specified by the user through the programming system.

4.10.1 Default Help Text

If no help text has been defined for a particular mask, the default help text specified in the programming system is displayed. This text is always present. If no text is programmed, a blank mask is displayed instead.

4.10.2 Help Text For Masks

A help mask can be created for every programmed mask. The link between the current mask and help text is specified in the parameter settings.

All of the valid characters can be used to design the help mask. Help texts can be called up directly by means of the Help key. As long as no data release has been requested, the help text for the mask is displayed. Once the data release is requested, the help text for the variable is shown.

4.10.2.1 Help Text for the Message Mask

Help masks can also be created for messages masks. Help texts are only available for the mask. Any variables that may be embedded can not be linked to a help mask.

4.10.3 Help Text For Variables

The functionality is the same as that of help texts for masks. It is, in particular, possible to display the valid range of values. The link between the help mask and the variable is specified in the variable parameters.

4.11 Function Keys

Another important feature of the operating concept, in addition to the masks, are the function keys and their LEDs. Function keys are user-programmable. They can be used as direct selector keys for selecting more masks or as control keys for the machine. When used as control keys, the integrated LEDs provide feedback information.

Programming the function keys as direct selector keys allows fast, direct access to the masks as well as to entire menu structures.

If the operating terminal is fitted with parallel outputs, 8 function keys can be assigned to the outputs directly. The reaction time after pressing a key is approximately 30 ms. Before a function key signal is provided, the terminal debounces the key, thereby ensuring that it has actually been 'pressed'.

In the programming system, the combination of direct selection and control can be programmed for function keys and soft keys. Only the press codes of the keys should be evaluated in this mode of operation. This is because, depending on the length of time the key is pressed and the nature of the assigned mask, the stop code may have already changed!

4.11.1 Direct Selector Keys

Direct selector keys are function keys which have been programmed to directly call up a specific mask. Pressing this function key causes a new mask to be activated. This change of mask is **not** possible if the data release has been requested (status LED in the Data Release key is flashing or lights up) in a mask without automatic data release.

With direct selector keys, a speedy and convenient operation can be obtained.

4.11.2 Function Keys of the Controller

In addition to being programmed as direct selector keys, function keys can also trigger a function in the PLC. This requires that instead of a change of mask, the symbolic name of a controller variable is assigned to the function key. This assignment is performed in the user description. The function key can be defined to either 'set' or 'reset' a variable when pressed/released. If the key is assigned the function 'set' the specified value will be transmitted to the variable in the controller.

If the number 1 is entered as a value,

- a flag bit will be set to logical '1'
- a flag byte will be set to the value 01_H
- a flag word will be set to the value 0001_H
- a double word will be set to the value 00000001_H

For values greater than '1', at least a byte address must be specified for the variable.

If the number 3 is entered as a value,

- a flag byte will be set to the value 03_H
- a flag word will be set to the value 0003_H
- a double word will be set to the value 00000003_H

4.11.3 Soft Keys

In the TesiMod standard mode, soft keys are function keys that perform a mask-related function (they perform different functions in different masks). A description of the function that a soft key performs in a particular mask should be displayed in that mask. Several means are available to design this description such as images, background images, selection images, static texts and selection texts.

If a selection text is used for labeling a soft key, the function key can be used for multiple functions in one and the same mask.

The action to be performed is determined in the controller by linking the

- mask number
- number of the selection text
- variable value that is transferred with the soft key.

The number of keys that can be used as soft keys depends on the type of operating terminal used.

Example: A soft key (F1) is to be capable to switch a pump off and on in mask 10

1. Create a text list (pump) with two entries

Value	Text
0	Switch Pump OFF
1	Switch Pump ON

2. Define the variables

- Soft key labeling M 100.0
- Soft key status M 100.1
- Image of the mask MW 110

3. Create the mask (number 10)

- set up a controller variable (M 100.0) next to or above a function key:
link a selection text variable for cyclic output with the text list (pump)
- link the function key F1 of the mask with the variable Soft Key Status (M 100.1),
(set/reset)

4. Create the PLC program to perform the following: O 32.0 is to be used to control the pump
 - Evaluate mask number (MW 110); (must have the value 10)
 - Create the edge evaluation for M 100.1
 - Create the toggle function for pump output O 32.0
 - Set flag M 100.0 to 0 when the pump is on
 - Set flag M 100.0 to 1 when the pump is off

4.11.3.1 Reaction Time of Function and Soft Keys

Whenever function keys need to influence PLC variables, they are given highest priority when transferred via the protocol. The reaction times during the transfer procedure are protocol-specific and range from 60 to 120 ms. This is the period of time which elapses after a key has been pressed until an output is set or reset in the PLC. The reaction time varies depending on the protocol itself, the load on the protocol (cyclic data, etc.) and the cycle time of the PLC.

Note that reaction times can be influenced by the polling times of the variables, messages and images of the LEDs.

4.11.3.2 Control Keys as Function Keys

Control keys can alternatively be used as function keys to trigger certain actions in the PLC. They can be defined to carry out the same functions as function keys, i.e. they are capable of setting (value 1) or resetting (value 0) a variable or of assigning any values to it. The transfer procedure is independent of the mask parameter assignment. Thus, if a control key is to carry out a specific function in a mask, it should not be programmed as a 'mask selector key' at the same time (in other words, one which initiates a change of mask). The mask-specific evaluation is identical to that of the function keys.

4.11.4 Function Keys Controlling Parallel Outputs

Groups of 8 function keys can be linked to parallel outputs (semiconductor outputs). The keys are read in by the software, debounced and then mapped to the outputs. The reaction time to the outputs is around 30 ms. As the keys act on the PLC very quickly and independently of the protocol, they are ideal for controlling axes or for programming jogging mode.

The power output allows direct control of PLC inputs.

If a PLC variable has been programmed for the function key in addition to the output, it is of course also sent to the controller, though with a small time delay.

4.11.5 Status LEDs in the Function Keys

A 2-bit information is provided in the cyclic poll area for each status LED in a function key. One of the two bits is responsible for activating or deactivating the corresponding status LED while the other bit represents the flash attribute for the status LEDs. Status LEDs can only be influenced by the controller.

<u>Bit is on/off</u>	<u>Bit flashes</u>	<u>Status of the LED</u>
0	0	LED is off
1	0	LED is on
1	1	LED flashes
0	1	LED is off, flash bit can remain set

This is not true if

- a function key has been programmed for direct selection of a message mask (direct selector key)
- a value greater than 0 (zero) has been entered for the message priority

In this case, the LED in this function key can not be influenced by the PLC but is controlled entirely by the message system.

If the number of status LEDs provided by the operating terminal being used is smaller than could be controlled here, then the excessive bits have no function. For the arrangement of the bits for the individual status LEDs see the section on the cyclic poll area.

To reduce transfer times, the length of the poll area should be selected such that only the bytes required for the status LEDs are transferred.

4.12 System Parameters

All system parameters are set to default values.

They are loaded into the operating terminal together with the project created. The system parameters contain the value settings for the:

- General parameters
- Poll area
- Terminal clock
- Running time meter
- Message system
- Variant buffer
- Password management
- Printer interface
- Data set transfer
- Parallel outputs (optional).

4.12.1 System Parameters: General Parameters

The general parameters refer to all generally applicable functions of the operating terminal.

- Polling time for cyclic variables
- Screensaver settings
- Automatic download
- Symbolic addresses for:
 - Image of the mask number
 - Image of the DIP switch
 - Read coordination byte
 - Table index
 - Keyboard image

4.12.2 System Parameters: Poll Area

System parameters for the poll area include the symbolic address of the variable, the polling time and the size of the poll area.

4.12.3 System Parameters: Terminal Clock

System parameters that can be specified for the terminal clock are the symbolic addresses of the variable, the polling time, the transfer parameters (date, time of day, day of week) and the variables for setting the terminal clock from the controller end.

4.12.4 System Parameters: Running Time Meter

System parameters that can be specified for the running time meter are the addresses for the control byte and reset byte as well as the polling time.

4.12.5 System Parameters: Message System

System parameters that can be specified for the message system are the general parameters. These include the size of the message buffer and the message priority for direct display of a message.

Parameters that can additionally be specified for the serial message system are the parameters for the logging printer and the symbolic name of the variable which can be used to delete messages from within the controller.

Parameters that can be defined for the parallel message system are the size of the message buffer, the polling time and the symbolic name of the variable address.

Additional parameters for the serial and parallel message system are the sorting criteria and how the messages are to be displayed.

4.12.6 System Parameters: Variant Buffer

System parameters available for the variant buffer are the symbolic name of the variable and the size of the variant buffer.

4.12.7 System Parameters: Password Management

System parameters that can be defined for the password management are the passwords themselves, the corresponding authorization levels and the initialization values for the authorization levels.

4.12.8 System Parameters: Printer Interface

Parameters that are specified for the printer interface are the baud rate, parity, data bits, stop bits and the handshake.

4.12.9 System Parameters: Gateway

Setting of the gateway parameters applies only to operating terminals that have been fitted with the corresponding firmware.

Parameters that can be defined for these operating terminals are:

- Smallest possible slave number
- Largest possible slave number
- Polling time for text list
- Cache size
- Polling time for cache
- Variable for cache address
- Variable for network status address

4.12.10 System Parameters: Data Set Transfer

The following parameters are required to transfer data sets from the terminal to the controller:

- Variable for recipe number
- Variable for data set number

The following parameters are required to transfer data sets from the controller (on request):

- Variable for recipe number
- Variable for data set number

4.12.11 System Parameters: Parallel Outputs

System parameters for the parallel outputs are the symbolic name of the variable and the polling time of this variable. Every output can be enabled separately.

4.13 Version Number

The version number is reserved for the user. Valid values range from 0 to 255. The value is stored in a system variable. This system variable can be displayed by the user in any I/O mask.

This variable has no further function in the operating terminal. Online editing of the version number is not possible.

4.14 Running Time Meter

All operating terminals provide 8 running time meters to the user. In the programming software, enter a variable name for the control byte which constitutes the address at which the controller can influence the running time meters.

Every bit of this control byte represents one running time meter. If a bit is set to logical 1, the corresponding running time meter is incremented in accordance with the selected polling time cycle.

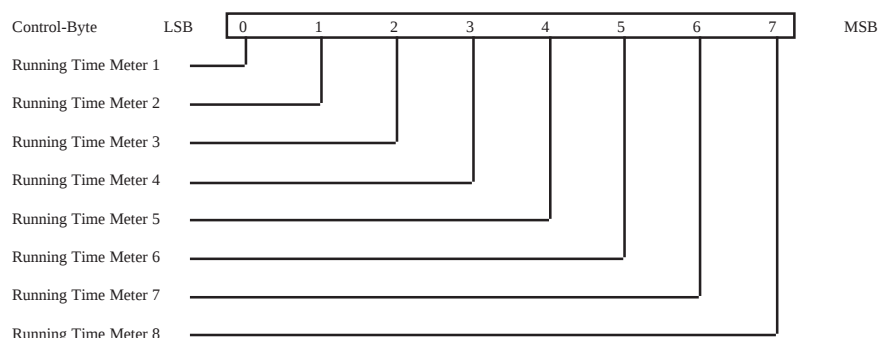


Fig. 15: Structure of the control byte

Example: A running time meter is to be set up for a maintenance interval of 50 hours.

Polling time for the counter: 60 seconds (the counter is increased by 1 every minute)

<u>Setting:</u>	System variable Counter1
	Decimal number
	4 digits; 1 post-decimal place (fractional digit)
	Only positive
	Factor 1; divisor 6, summand 0

The following is displayed on the terminal after 150 polling cycles:

Format:	$150 : 6 = 25$
Formatted display:	2.5 Hours

The precision in this example is +/- 6 minutes.

The variable 'Reset Byte' can be used to reset each running time meter from the controller. To do this, the bit for the running time meter in question must be set to logical 1.

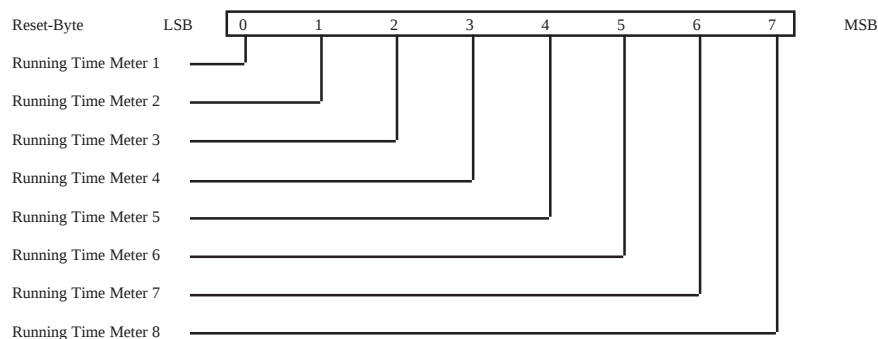


Fig. 16: Structure of the reset byte

4.15 Parallel Outputs

The parallel outputs of the operating terminal can be addressed directly with the function keys and by the controller. To be able to operate the parallel outputs from the controller, a variable must be defined for the control word. The word must be divided into two single bytes if the controller is only capable of accessing in byte mode. The byte order depends on the type of controller.

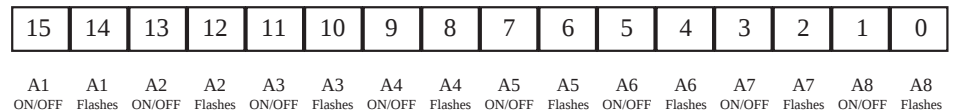


Fig. 17: Structure of the Control Word

The structure of the control word illustrates that one bit pair always controls one output. The following truth table applies to every bit pair:

0	0	Output OFF
1	0	Output ON
1	1	Output FLASHES

Fig. 18: Truth table for Parallel Outputs

In addition to the variable name, the polling time must be specified in the programming software to determine the cyclic intervals at which the control word is to be polled.

Outputs can also be addressed by means of function keys, provided this has been defined in the programming software. An output is being addressed (ON) as long as the function key is pressed.

The programming software checks if the function keys and the controller want to access the same outputs.

4.16 Screen Saver

Some terminals are fitted with a screen saver. This function monitors all outputs to the display. If the system detects that nothing is being output to the display, a timeout begins to elapse. After the timeout has elapsed, the display is blanked and the status LED in the Help key begins to flash. The display can be reactivated by pressing any key.

Activation of the screen saver can be made dependent on the display of cyclic variables.

The timeout can be defined in 0.1 second steps.

If the timeout is set to 0, the screen saver remains deactivated.

4.17 Image of the Mask Number

The mask number is a mask-specific code which the terminal writes to the controller variable entered here.

The terminal writes the mask number of the currently displayed mask into this variable whenever a new mask is activated. The variable is assigned in the variable list. In this list, a word address of the controller must be assigned to the symbolic name. The respective mask number can then be read from this address and be processed further as required. A 16-bit variable must be reserved for the mask number.

Also see the evaluation of the function keys and soft keys, etc.

4.18 Image of the Mode Selector Switch

In standard mode, the image of the mode selector switch (user mode switch) is transmitted to the controller after the initialization phase is complete. The user has the option of evaluating any unassigned DIP-switches in the controller. This allows the user to call up specific programs in the controller or to create queries in a service routine.

4.19 Terminal Clock

Every operating terminal is fitted with a real-time clock. The parameters of this real-time clock can be set in the system parameters.

Once per poll cycle, the real-time clock stores the current time, date and day of the week to the defined variable in the connected controller.

The connected controller itself can also transfer its current time, date and day of the week to the terminal.

This allows the time and clock in the controller, if any, to be synchronized or provides the real-time if controllers are not fitted with a clock.

Formats and contents of the variables must be identical with those in chapter 'Image of Date and Time'. To cause the terminal to read this variable, the controller needs to transfer the control code **7FF9_h** through the serial message channel.

4.19.1 Image of Date and Time

The time of day, the date and the day of week are stored in BCD-format. A maximum of 7 bytes of memory space is required. The address of the first byte is specified in the variable list. The data contents are defined as illustrated below:

Starting address +0 (Byte 0)	Y	Y	Year (0 to 99)
Starting address +1 (Byte 1)	M	M	Month (1 to 12)
Starting address +2 (Byte 2)	D	D	Day (1 to 31)
Starting address +3 (Byte 3)	h	h	Hour (0 to 23)
Starting address +4 (Byte 4)	m	m	Minute (0 to 59)
Starting address +5 (Byte 5)	s	s	Second (0 to 59)
Starting address +6 (Byte 6)	W	W	Weekday (0 to 6)

Fig. 19: Structure of the control byte for the time and date

The variable for the day of the week is calendar-independent and always runs modulo 6. The assignment of a number to a particular day of the week can be specified by the user when he sets the date and the text for the day.

Possible assignments in the text lists are:

- 0 Sunday
- 1 Monday
- 2 Tuesday
- 3 Wednesday
- 4 Thursday
- 5 Friday
- 6 Saturday

When entering the date, the user is then responsible for setting the variable for the day of the week to the correct value.

Example:

21.02.1997 (Friday)

Day of the week in accordance with table above: 5

4.20 Read Coordination Byte

The term **Read Coordination Byte** indicates that the controller only reads this byte. It is only written to by the operating terminal.

This byte is used for the handshake and data coordination with the controller. For this purpose, the terminal reports its current status to the controller in the symbolic name entered here. Each bit has its own specific function. The structure is as illustrated below:

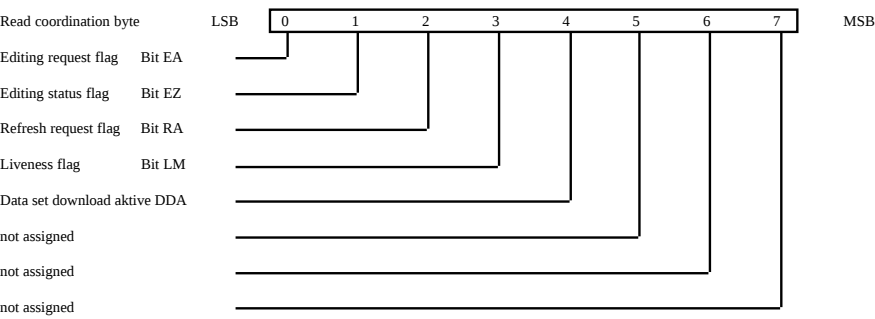


Fig. 20: Structure of the Read coordination byte

The Read coordination byte is used to pass the status of the operating terminal to the controller.

For the functionality of the individual bits refer to the respective chapters.

The terminal supports the <Read Coordination Byte> function only if the **Write Coordination Byte** has also been activated in the cyclic poll area. To activate both coordination bytes, the size of the poll area and the polling time must be specified during programming time. In addition, the addresses of the Read coordination byte and poll area must be specified in the variable list.

4.20.1 Editing Request Bit (Bit ‘EA’)

The editing request bit is used to signal to the controller that the value of a variable is to be modified on the terminal. To do so, the operator presses the Data Release key. The status LED in the Data Release key is flashing as long as the editing release has not been set by the controller. Once the controller has set the **External Data Release** bit in the **Write Coordination Byte** to logical 1, the status LED in the Data Release key remains lit and the value can be modified by the operator.

4.20.2 Editing Status Bit (Bit ‘EZ’)

Once the controller has set the editing release, the editing status bit is automatically set to logical 1 by the terminal. The bit is reset to logical 0 after the Enter key is pressed by the operator.

4.20.3 Refresh Request Bit (Bit 'RA')

The refresh request function is set up by using an input variable with the PLC-Handshake attribute. With this function, a data release for the next input variable in the same mask will not be set until a refresh acknowledgement has been received from the controller.

Upon request, the controller can then refresh the next variable (bit 'RA' = 1) before sending a refresh acknowledgement (bit 'RQ' in the Write coordination byte = 1), thereby ensuring that the value the operator works with is current.

4.20.4 Liveness Flag (Bit 'LM')

With some communication protocols it is not possible to check, from the controller end, whether or not the interface is in operating condition. This is why the liveness flag function has been introduced - a simple functionality that has proven to be very effective in practice.

Whenever the PLC wants to know whether a connection is still established, it writes a logical 1, and later a logical 0, to bit 3 of the **Write Coordination Byte**.

The operating terminal constantly monitors the liveness flag in the **Write Coordination Byte** and compares it with the status of the liveness flag in the **Read Coordination Byte**.

As soon as a discrepancy occurs, the operating terminal copies the **Liveness Flag Bit** from the **Write Coordination Byte** to the **Read Coordination Byte**.

The controller is now responsible for checking within a timeout whether both states are in agreement. The transfer times and polling times must be taken into account when defining the timeout.

4.20.5 Data Set Download Active (Bit 'DDA')

The **Data Set Download Active** bit remains set to logical 1 for as long as a data set is being transferred. The bit is reset after all of the data have been transferred. The controller can then work with the new values in the recipe.

4.21 Write Coordination Byte

The term **Write Coordination Byte** indicates that the controller writes to this byte. The operating terminal only reads this byte. Together with the **Read Coordination Byte**, this byte is used for the handshake and for data coordination with the controller. Here, the controller reports its current status to the terminal. The individual bits are independent of one another. The **Write Coordination Byte** is the first byte of the cyclic poll area. The structure is as illustrated below:

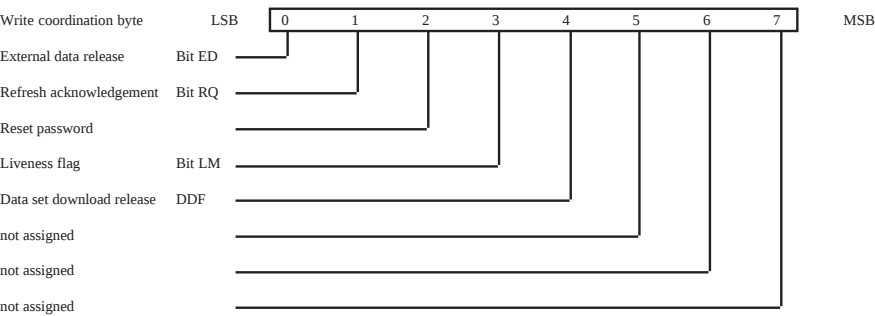


Fig. 21: Structure of the Write coordination byte

4.21.1 External Data Release (Bit ‘ED’)

The external data release can be used to determine, from the controller end, the point of time from which editing of the value of a variable is allowed on the terminal. A flashing Data Release key LED on the operating terminal indicates that an external data release has not yet been set. Editing of the value is therefore still inhibited. Once the PLC has set the data release bit to logical 1, the Data Release LED remains lit and a new value can be entered by the operator. After all modifications have been made, the operating terminal sets the editing status bit back to logical 0, thereby signaling that the editing process has been completed.

4.21.2 Refresh Acknowledgement (Bit ‘RQ’)

The refresh request bit is used to indicate to the controller that the values of the variables in the current mask are to be refreshed. After the values have successfully been transferred to the terminal, the controller sets the refresh acknowledgement bit to logical 1 to indicate that the process has been completed. See chapter 5.20.3 *Refresh Request Bit (Bit ‘RA’)*.

4.21.3 Resetting the Password

The password protection should be reactivated after the operator exits a password-protected access level. This can be forced by the controller by setting the **Reset Password** bit to logical 1.

4.21.4 Liveness Flag (Bit ‘LM’)

See chapter Liveness Flag of the **Read Coordination Byte** (5.20.4).

4.21.5 Data Set Download Release (Bit 'DDF')

The Data Set Download Release (DDF) bit allows the controller to determine the point of time from which transfer of a data set to the controller can begin.

4.22 Cyclic Poll Area

In addition to a random read and write access to controller variables, a 23-byte memory area is defined in the user description as the cyclic poll area.

The cyclic poll area consists of:

- 1 byte **Write coordination byte**
- 2 bytes **serial message channel** (low and high byte)
- a terminal-specific number of **control bytes** for the **status LEDs** in the function keys

The poll area is written to by the controller and polled by the operating terminal at cyclic intervals. During this process, the controller can trigger, control, enable or inhibit actions that are connected with the terminal. In addition, the liveness flag of the Write coordination byte can be used to verify whether or not the connection with the terminal is still established. Actions are triggered and serial messages are passed to the operating terminal via the serial message channel (for example, transfer of data sets, change of masks). The bits of the control bytes can be used to activate or deactivate the status LEDs in the function keys or to place them to the flashing mode.

The marginal conditions regarding the memory area for the poll area are:

- the PLC accesses the Write coordination byte and status LEDs in bit mode and the serial message channel in byte mode or word mode
- the terminal accesses in byte or word mode
- the memory area must be contiguous.

The symbolic name for the cyclic poll area is specified in the parameters for the poll area using the programming system. The assignment of the starting address of this area is defined in the variable list.

Note that access to byte and word structures is not identical. Once an access method is selected, it should be applied throughout.

4.22.1 Byte-Oriented

If the byte-oriented assignment method has been selected in the variable list for the cyclic data area, the data area will be allocated as shown below:

Example: The cyclic poll area is set to flag byte MB 12 in the programming system.

Access in the PLC is via:

Byte Address	MB	Description
Byte address +0	MB12	Write coordination byte
Byte address +1	MB13	Message channel low-byte
Byte address +2	MB14	Message channel high-byte
Byte address +3	MB15	Status LEDs 1 to 4
Byte address +4	MB16	Status LEDs 5 to 8
Byte address +5	MB17	Status LEDs 9 to 12
Byte address +6	MB18	Status LEDs 13 to 16
Byte address +7	MB19	Status LEDs 17 to 20
Byte address +8	MB20	Status LEDs 21 to 24
Byte address +9	MB21	Status LEDs 25 to 28
Byte address +10	MB22	Status LEDs 29 to 32

The size of the data area is 11 bytes.

The structure of the byte-oriented cyclic poll area is as shown below:

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte address +0	not used	not used	not used	DDF	LM	PL	RQ	ED
Byte address +1	serial message channel low-byte							
Byte address +2	serial message channel high-byte							
Byte address +3	LED1 on/off	LED1 flashing	LED2 on/off	LED2 flashing	LED3 on/off	LED3 flashing	LED4 on/off	LED4 flashing
Byte address +4	LED5 on/off	LED5 flashing	LED6 on/off	LED6 flashing	LED7 on/off	LED7 flashing	LED8 on/off	LED8 flashing
Byte address +5	LED9 on/off	LED9 flashing	LED10 on/off	LED10 flashing	LED11 on/off	LED11 flashing	LED12 on/off	LED12 flashing
Byte address +6	LED13 on/off	LED13 flashing	LED14 on/off	LED14 flashing	LED15 on/off	LED15 flashing	LED16 on/off	LED16 flashing
Byte address +7	LED17 on/off	LED17 flashing	LED18 on/off	LED18 flashing	LED19 on/off	LED19 flashing	LED20 on/off	LED20 flashing
Byte address +8	LED21 on/off	LED21 flashing	LED22 on/off	LED22 flashing	LED23 on/off	LED23 flashing	LED24 on/off	LED24 flashing
Byte address +9	LED25 on/off	LED25 flashing	LED26 on/off	LED26 flashing	LED27 on/off	LED27 flashing	LED28 on/off	LED28 flashing
Byte address +10	LED29 on/off	LED29 flashing	LED30 on/off	LED30 flashing	LED31 on/off	LED31 flashing	LED32 on/off	LED32 flashing
Byte address +11	LED33 on/off	LED33 flashing	LED34 on/off	LED34 flashing	LED35 on/off	LED35 flashing	LED36 on/off	LED36 flashing
Byte address +12	LED37 on/off	LED37 flashing	LED38 on/off	LED38 flashing	LED39 on/off	LED39 flashing	LED40 on/off	LED40 flashing
Byte address +13	LED41 on/off	LED41 flashing	LED42 on/off	LED42 flashing	LED43 on/off	LED43 flashing	LED44 on/off	LED44 flashing
Byte address +14	LED45 on/off	LED45 flashing	LED46 on/off	LED46 flashing	LED47 on/off	LED47 flashing	LED48 on/off	LED48 flashing

Fig. 22: Byte-oriented poll area

4.22.2 Word-Oriented

The structure of the word-oriented cyclic poll area is as shown below:

	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Word address + 0	not used	not used	not used	DDF	LM	PL	RQ	ED	not used	not used	not used	not used	not used	not used	not used	not used
Word address + 1	serial message channel low byte								serial message channel high byte							
Word address + 2	LED1 on/off	LED1 flashing	LED2 on/off	LED2 flashing	LED3 on/off	LED3 flashing	LED4 on/off	LED4 flashing	LED5 on/off	LED5 flashing	LED6 on/off	LED6 flashing	LED7 on/off	LED7 flashing	LED8 on/off	LED8 flashing
Word address + 3	LED9 on/off	LED9 flashing	LED10 on/off	LED10 flashing	LED11 on/off	LED11 flashing	LED12 on/off	LED12 flashing	LED13 on/off	LED13 flashing	LED14 on/off	LED14 flashing	LED15 on/off	LED15 flashing	LED16 on/off	LED16 flashing
Word address + 4	LED17 on/off	LED17 flashing	LED18 on/off	LED18 flashing	LED19 on/off	LED19 flashing	LED20 on/off	LED20 flashing	LED21 on/off	LED21 flashing	LED22 on/off	LED22 flashing	LED23 on/off	LED23 flashing	LED24 on/off	LED24 flashing
Word address + 5	LED25 on/off	LED25 flashing	LED26 on/off	LED26 flashing	LED27 on/off	LED27 flashing	LED28 on/off	LED28 flashing	LED29 on/off	LED29 flashing	LED30 on/off	LED30 flashing	LED31 on/off	LED31 flashing	LED32 on/off	LED32 flashing
Word address + 6	LED33 on/off	LED33 flashing	LED34 on/off	LED34 flashing	LED35 on/off	LED35 flashing	LED36 on/off	LED36 flashing	LED37 on/off	LED37 flashing	LED38 on/off	LED38 flashing	LED39 on/off	LED39 flashing	LED40 on/off	LED40 flashing
Word address + 7	LED41 on/off	LED41 flashing	LED42 on/off	LED42 flashing	LED43 on/off	LED43 flashing	LED44 on/off	LED44 flashing	LED45 on/off	LED45 flashing	LED46 on/off	LED46 flashing	LED47 on/off	LED47 flashing	LED48 on/off	LED48 flashing

Fig. 23: Word-oriented poll area

Example: The cyclic poll area is set to DW 21 in the programming system:

Word Address	DW	High Byte	Low Byte
Word address +0	DW21	Write coordination byte	Reserved
Word address +1	DW22	Message channel high-byte	Message channel low-byte
Word address +2	DW23	Status LEDs 1 to 4	Status LEDs 5 to 8
Word address +3	DW24	Status LEDs 9 to 12	Status LEDs 13 to 16
Word address +4	DW25	Status LEDs 17 to 20	Status LEDs 21 to 24
Word address +5	DW26	Status LEDs 25 to 28	Status LEDs 29 to 32

4.22.3 Image of the LEDs

The image of the LEDs enables the PLC to control the status LEDs in the function keys. Functions that can be set for each status LED are the functions ON, OFF and FLASHING. Whenever the controller sets a bit, the associated LED on the operating terminal is influenced accordingly.

It is important here that the size of the poll area and the polling time were set appropriately. Not defining these additional parameters may cause problems when the LED is addressed.

In the case of a function key that is to directly select a message mask, the status LED is influenced by the message system. The message system uses this status LED to indicate that a new message has been received but not yet acknowledged. To be able to influence the status LED in this function key from the controller, the message priority must be set to 0 (zero). See chapter 4.11.5 *Status LEDs in the Function Keys*.

0	0	Status LED OFF
1	0	Status LED ON
1	1	Status LED FLASHES

Fig. 24: Truth table for status LEDs

4.22.4 Serial Message Channel

The serial message channel is a part of the cyclic poll area and is used to transfer 16-bit information. The numbers of serial messages, selection of message masks, external selection of masks and transfer of control codes are made possible via this data channel.

The following handshake is used for the information transfer: the PLC stores a value (> 0) in this data word. This value is then transferred to the operating terminal which will write the value 0 into this data word again. This indicates to the PLC that it can now transfer the next value. The value is interpreted by the operating terminal and its function is executed.

Values can be:

- Message numbers
- Mask numbers (mask number + 8000_H)
- Control codes

4.22.5 Polling Time

The polling time is the frequency with which the variables for the cyclic poll area are to be read by the operating terminal. This setting is specified in the system parameters for the poll area. Polling of this variable includes the **Write Coordination Byte**, the **Serial Message Channel** and the **Image of the status LED**. With most protocols, settings of around half a second have worked well. If the setting for the cycle time is too short, the interface protocol can no longer meet the requirements resulting in poorer reaction times. There is no general recommendation since the options depend on the user description involved. Time settings should, however, be at least greater than 100 ms.

4.22.6 Size of the Poll Area

The maximum size is 23 bytes depending on the data type and type of terminal. The poll area size can be adapted to the area that is actually used if the image of the LED or part of this function is not to be used. The default size for all operating terminals is 12 bytes.

4.23 Control Codes

Control codes have already been mentioned in the previous sections which can be used to trigger certain actions or functions in the operating terminal. All of these actions are initiated by the controller by writing the desired control code to the serial message channel within the poll area.

An explanation of these control codes is provided below.

4.23.1 Triggering Data Set Printouts

The following hex code can be used from the controller to cause the current data set to be output to the printer connected to the operating terminal.

Hexadecimal code: **7FF8_H**

The operating terminal can indicate the status of the print process by writing either of two values to the address for the recipe number variable.

0 Data set printout ok.

255 Data set with the desired data set number can not be printed.

4.23.2 Setting the Clock in the Operating Terminal

The following hex code can be used from the controller to cause the real-time clock in the operating terminal to be set in accordance with the specifications in the defined control word.

Hexadecimal code: **7FF9_H**

4.23.3 Transferring Data Sets from the Controller to the Terminal

The following hex code can be used from the controller to initiate the transfer of a data set from the controller to the operating terminal. This requires that the recipe number and data set number are specified by the controller first.

The data sets are transferred as a block.

The following variables must be defined:

- Recipe number for the Request from Controller (System Parameters: Data Set Transfer)
- Data set number for the Request from Controller (System Parameters: Data Set Transfer)

Hexadecimal code: **7FFA_H**

4.23.4 Transferring Data Sets from the Terminal to the Controller

The following hex code can be used from the controller to initiate the transfer of a data set from the operating terminal to the controller. This requires that the recipe number and data set number are specified by the controller first.

The following variables must be defined:

- Recipe number for the Request from Controller (System Parameters: Data Set Transfer)
- Data set number for the Request from Controller (System Parameters: Data Set Transfer)

Hexadecimal code: **7FFB_H**

4.23.5 Transferring Data Sets from the Controller to the Terminal (Individually)

The following hex code can be used from the controller to initiate the transfer of a data set from the controller to the operating terminal. This requires that the recipe number and data set number are specified by the controller first.

The data sets are transferred individually.

The following variables must be defined:

- Recipe number for the Transfer from the terminal (System Parameters: Data Set Transfer)
- Data set number for the Transfer from the terminal (System Parameters: Data Set Transfer)

Hexadecimal code: **7FFD_H**

4.23.6 Refreshing the Message System

The following hex code can be used from the controller to cause the operating terminal to read in the most recent parallel messages.

This procedure can be used to achieve an event-controlled message system.

Hexadecimal code: **7FFF_H**

4.24 Cyclic Variables

The term Cyclic Variables refers to data which may change continuously while a mask is displayed, i.e. all types of ACTUAL VALUES. The terminal must therefore poll the controller for the current values at cyclic intervals. The time selected here specifies the intervals at which these mask values are refreshed.

In addition to the cyclic variables, the terminal also polls the controller for the current status of the Write coordination byte, the serial message channel and the LEDs (update). This process is either time or event controlled depending on the protocol. No polling will take place if the value = 0 is entered for the time period. If this variable is not polled, then an external control of the data release is not possible, for example.

4.25 Interface Parameters

The interface parameters selected in the programming system are stored in the mask memory. It is, however, possible to modify the data in the operating terminal at a later date. Values modified in the operating terminal are retained in the battery-backed RAM. When necessary, the original settings can be restored at any time. The parameters of the interface channel SER1 are determined by the protocol. Modifications to any of these parameters could result in communication failures. Values that are not typical are detected when the terminal is initialized and a message to this effect is generated.

The parameters of the interface channel SER2 are reserved for control of a printer and, when transferring data. This process requires that the settings for both devices must match. The selected parameters do not apply to the download. For the download, the system will automatically switch to the highest possible transmission rate.

4.26 Variable Definition

The variable definition contained in the user description defines the format, type, range of values, scaling and attribute of the representation. All controller-independent parameters are stored here. For editable variables, it is additionally possible to influence the Editor and the transfer mode to the controller. The variable definition contains the symbolic name of the variables, the format definition and, if necessary, the text lists in the case of selection text variables.

4.26.1 Variable List

The variable list contains the assignment of symbolic variable names to the destination hardware. Due to its design and proximity to the PLC, this file is always manufacturer-specific.

In the programming software, the variable list is a part of the database.

The variable list contains only the required manufacturer-specific user description parts. Its size is therefore small in relation to the entire project. A project can be quickly adapted to other PLC specifications by simply adapting the variable list to the PLC to be used.

A variable list can be created in the programming system in two ways.

First method:

As a variable is defined in a mask (with or without address information) it is entered into the variable list at the same time.

Second Method:

The first step can be to define the entries in the variable list or alternatively to insert them from any suitable program via the clipboard. The entries will immediately be available in the programming software on a global basis.

Third Method:

The elements of the variable list are imported from a Siemens SEQ file or a Bosch SXS file. Only the relevant data of the import files will be extracted and inserted into the variable list.

4.27 Application Programming

The operating concept is based on a mask structure that can be created by the user in accordance with the requirements and on system variables that can influence important functions in the operating terminal. Various types of masks are available to facilitate the creation of solutions for standard tasks.

Every mask structure initially consists of 4 types of masks whose contents are user-configurable.

Any I/O mask can be programmed as a system mask.

- Setup mask
- Start mask
- Password mask
- Main mask

After being switched on, the operating terminal is initialized and displays the startup mask during this process. The setup mask can be called up by pressing the Enter key while the startup mask is displayed. The setup mask can be exited by pressing the Enter key again and the startup mask will reappear.

The next mask of the operating guidance that is displayed is the Main Mask which represents the first mask of the user-programmed operator guidance. A menu with menu items is normally set up in the main mask to allow selection of lower-level masks. The user has the option of specifying a full-page help text for every mask which can be displayed by the operator with the Help key.

The functions of the system variables are briefly described in chapter 4.6.3. Some of these variables are mentioned in the following chapters to be able to explain the application programming.

4.27.1 Configuring the System

To run the programming software, you will need a PC that complies with the requirements for a computer that runs with Windows95/WindowsNT (1) operating system. One of the computer's serial interfaces (COM1 or COM2) is connected with the interface channel SER2 of the operating terminal using the download cable. The connection of the operating terminal to the PLC is established with an additional, controller-specific interface cable. Care must be taken to ensure that the correct type of interface cable is used and that the operating terminal is connected to a power source which complies with the technical specifications for the operating terminal.

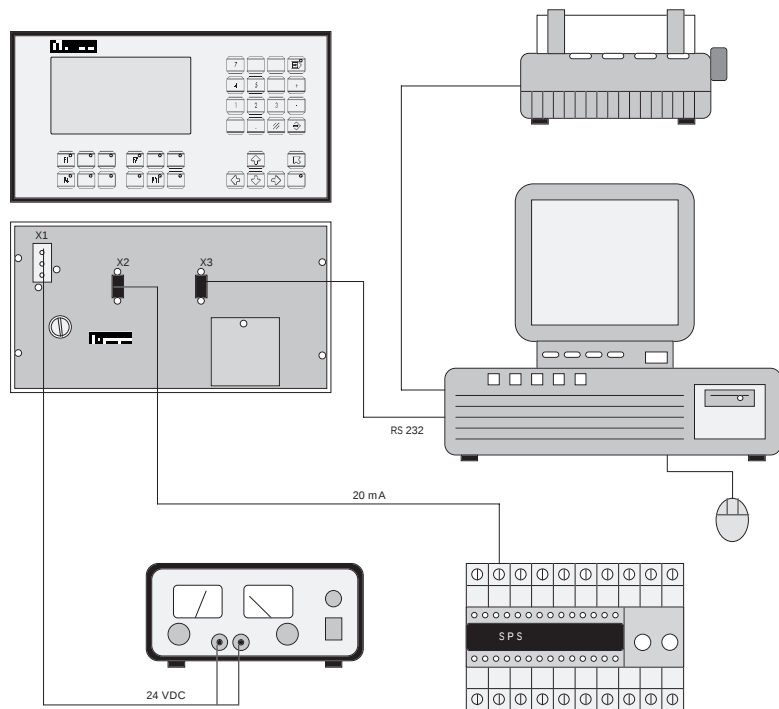


Fig. 25: System configuration

Project execution can begin once all preparatory measures have been taken. All functions provided by the operating terminal can be executed and tested with the configuration shown above. If operated without a PLC (see simulation without PLC), all operating functions with the exception of the PLC responses can be tested.

(1) Windows is a registered trademark of the Microsoft Corporation.

4.27.2 Programming Systems

The programming system provides all of the capabilities required to conveniently and quickly program the entire range of functions offered by an operating terminal. The programming system requires either Windows95 or Windows NT 4.0. It offers Windows-conforming operability in conjunction with all of the means provided by Windows (OLE, COPY/CUT/PASTE, WYSIWYG). The programming software provides a help system that can be reached with the Help key or the function key F1 from anywhere within the program (online help).

The programming system operates similarly to a database. All project-related data are collected in this database and are organized into a project.

This project can then be compiled into a S3 file and transferred to the terminal using the download function.

4.27.3 Project Description

The project description consists of:

- at least the user description (one per language)
- a variable list (controller-specific)
- the global data (parameters)
- and, if used, parts of the graphical library.

The user description contains all of the controller-independent information. Symbolic names are inserted for all controller-specific variables.

The assignment of these symbolic names to the hardware is specified in the variable list. This makes it possible to easily change to another controller manufacturer without having to recreate the masks, variable definition or operating functionality. The adaptation to a different controller type can be accomplished by simply replacing the variable list.

4.27.3.1 Multilingual Projects

The mask contents of a user description determines the language in which the texts are displayed. To make an operator interface available in multiple languages, the user description of each language is stored with a language number. The desired language can then be selected online via a system variable using the assigned language number.

The number of languages that can be stored is limited only by the size of the mask memory. Please note that it is not necessary that the length of the texts, the variables and the number of masks are identical in the various languages.

The number of the currently active language is stored in the operating terminal and is retained when the terminal is switched off. When booted again, the terminal defaults to the language which was active when the terminal was turned off. In the case of a loss of data, the language that corresponds to the number 0 is automatically displayed.

The language can easily be selected on the operating terminal using a selection text. The language can then be selected with the Cursor up and Cursor down keys.

Overview of the required steps:

1. Create a text list with the language name (Example: Deutsch, English, Française)
2. Create the mask
3. Create the variable in the mask:
OSLanguage, representation method, selection text, text list name as created in step 1.



Attention: Do not enter system variables in the variable list. This would result in the controller addresses being accessed instead of the internal operating terminal addresses.

4.27.3.2 Variants of a Project

The system variable **OSLanguage** also permits the management of variants. For example, if you design a machine with various options but do not want the operator of the machine to be distracted by these options, it is possible to select the appropriate variant online on the terminal.

Example:

VALUE	FUNCTION
0	Variant 1
1	Variant 2
2	Variant 3

The variants may differ in operating structure, the number of masks as well as the variables used. The advantage is that the user description must be maintained only once for the entire system.

Example:

A possible application is when a system is to be operated with the metric and Anglo-Saxon dimensions. In this case, the format and scaling of the variable inputs and outputs could be switched from millimeter to inches along with the language.

4.27.4 Project Documentation

The programming software also allows the generation of extensive documentation on the project. Detailed information on how to use variables, the scaling parameters, representation methods, mask contents as well as the operator guidance facilitate verification of the user description.

The documentation should cover any information necessary to recreate the user description in the event of a loss of data.

4.27.4.1 Creating Documentation

The programming system offers extensive dialogs that allow those elements to be selected that are to be documented.

The documentation can then be easily processed further using word processing programs or desktop publishing programs.

4.27.5 Project Back-up

A project is backed up by storing the source files onto various data mediums, as recommended by the PC manufacturers. For information on the available options refer to your PC's operator system manual.



In addition, we recommend that the projects are printed using the documentation functions.

4.27.6 Optimizing the Transmission Rate

The operating concept supports free access to various data types and data lengths.

Although, the extent to which the various protocols actually support this free access varies greatly. The firmware of the terminal optimizes the access method mask-specifically to achieve extremely short access times.

This method means that an attempt is made to transmit several variables to the display simultaneously. The user can actively contribute to this process by keeping the number of different variable types used in one mask to a minimum and by addressing variables of one type contiguously (small gaps are no problem!). This allows shorter refresh intervals and reduces the load on the interface.

4.28 Downloading the User Description

The download function describes how a new project is loaded into the terminal's Flash memory. Before a new project can be downloaded, the terminal must be placed into the download mode.

The download mode is activated by:

- Writing a '1' to the system variable **IntEraseEPROM** in an I/O mask.
- Switching off the supply voltage, setting the user-mode switch S4 to ON, turning the device back on and once the following message is displayed

```
1. TURN POWER OFF
2. RESET DIP-SW 4
OTHERWISE ALL FLASH-
DATA WILL BE LOST !!
```

switching the user-mode switch S4 to OFF again with the power on.

While programming, you may want to activate the automatic download function. This will automatically switch the operating terminal into the download mode whenever a download is started on the PC.

When the download function is activated, the user description currently stored in the terminal will be erased and the following information will appear on the display at the cursor position (1.1):

```
DOWNLOAD 1
FLASH xxx kByte
```

A new user description can now be transferred to the terminal using the Project Management dialog in the programming system. For this process, the PC must be connected to interface channel SER2 of the terminal using the download cable. To prevent the user description from accidentally being erased during operation, it is possible to assign an appropriately high access level to the mask with the down.

load function, so a password is required to access the mask. If the mask has been assigned to a control key, the operator will not even know that such a mask exists. Another method would be to prevent input by using the connected controller. The password protection function is not required in this case. With this method, the controller will provide data release for the mask number with the download function only, if specific conditions set out by the user are fulfilled.

If the user does not want to make use of these options, another alternative to the software solution via the system variable **IntEraseEPROM** would be to erase the old variable definition with the aid of the user-mode switch S4. This will require strict adherence to the routine to be executed. Subsequently the terminal will automatically activate the download operating mode.

To place the operating terminal to the download operating mode:

The Flash memory is erased once the system variable has been activated. The following message is displayed during this process:

```
ERASE FLASH EPROM
```

After the Flash memory has been erased, this is indicated by the following message:

```
FLASH IS ERASED
```

The terminal now automatically activates the download mode and the following message appears:

```
DOWNLOAD 1
FLASH xxx kByte
```

The terminal is now ready to receive a new user description via interface channel SER2 and to store it in the Flash memory. The progress of the data transfer will be indicated on the display by the characters '>>>>>', the number of characters will therefore change continuously!

After the transfer is complete, the following message is displayed:

```
READY
```

After the transfer, the terminal immediately starts the initialization phase and displays the startup mask of the user description.

Various error messages may appear on the display during the download or at the beginning of the initialization phase:

ADDRESS ERROR	Error in the address calculations of the compiler
FLASH MEMORY FAILURE	The loaded user description contains errors or is incomplete.
CHECKSUM ERROR	The user description contains errors, please re-compile.
BYTECOUNT OVERFLOW	The user description contains errors, please re-compile.
FORMAT ERROR (S0, S3, S7)	Sequences in the S3 file that was transferred contains errors.

The display of an error message indicates that the transferred user description is incomplete or contains errors. Switching the terminal off and on at this point will automatically reactivate the download operating mode.

4.28.1 Downloading with Windows

The programming software does not necessarily have to be used for downloading. Another option is to use the program Terminal that is available under MS Windows (group 'Accessories').

The following steps are required:

- 1. Connect COM1 or COM2 of the PC with the interface channel SER2 using the download cable
- 2. Double-click the Terminal icon under Windows
- 3. Set the parameters:

Menu:

Settings:

Communication: 19200Bd
7 bits
odd parity
xon/xoff
1 stop bit

Transfers:

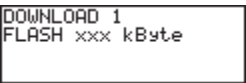
Send text file: (Enter file name) xxxxx.S3

4.28.2 Application Memory

The terminals come fitted with a 256 kByte Flash memory. Flash memory is non-volatile either as UV-EPROMs are, but flash memory can be erased and written without putting it out of the appliance. Therefore the application can easily and quickly be tested or setup to new conditions.

4.28.3 Loading an Application

To load the application, a PC must be connected to interface channel SER2 using the download cable. The terminal must then be placed to the download mode. The following message must appear on the display:



The terminal is now ready to receive a new application via interface channel SER2 and store it in the Flash memory.

Now the download of an error-free user description 'name S.3' must be activated using the Project Management dialog in the programming software.

The progress of the data transfer will be indicated on the display by the characters '>>>>>', the number of characters will therefore change continuously.

The download transmission can either be controlled by a beam in the dialog box. After the transmission is complete, the following is displayed



to indicate that the transfer has completed. After the transfer, the terminal immediately starts the initialization phase and displays the start mask of the user description.

4.28.4 Activating the Download Function using the Software

This requires that the terminal is placed to the download mode by activating an I/O mask and writing a '1' to the system variable **IntEraseEprom**.

In the event that you have decided not to use the system variable **IntEraseEprom**, the old variable definition must be erased with the user-mode switch S4. This will require strict adherence to the routine to be executed. Subsequently the terminal will automatically activate the download operating mode.

4.28.5 Activating the Download Function using the Hardware

If the system variable responsible for erasing the Flash memory can not be activated in the masks or is not available at all, the Flash memory can still be erased by switching the mode selector switch to a defined position. This will cause the terminal to activate the download mode.

Switch position to completely erase the Flash memory:

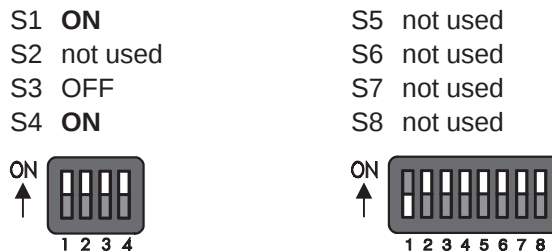


Fig. 26: User-Mode switch

The following message is displayed once power has been applied the terminal:

```
TURN POWER OFF
RESET DIP-SW 4
OTHERWISE ALL FLASH-
DATA WILL BE LOST
```

This message is designed to prevent accidental erasure of the user description. The user description will be preserved if this message is complied with.

Thus, turn off the power, reset the mode selector switch S4 and switch the terminal on again. The terminal will continue to work with the current user description.

Note that the contents of the Flash memory will be erased if the power is not turned off before the DIP switch S4 is reset.

4.28.6 Automatic Download Function

The Automatic Download option can be enabled by selecting the appropriate check box under 'System Parameters', 'General Parameters' in the programming system.

The user description that contains this function must initially be loaded into the terminal via a forced download (user-mode switch S4 at the ON position). Every download thereafter is automatically initiated by the PC wanting to load the S3 file of the new application into the terminal. A download can be forced when the operating terminal is in full operation.

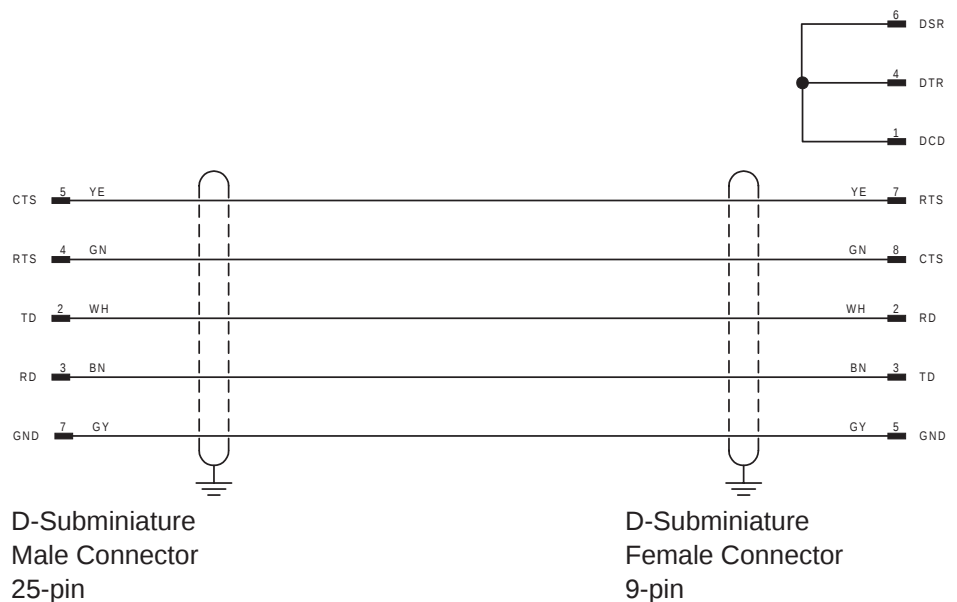
During this process, all

- message data in the message memory
- all RAM-data sets that have not been backed-up
- all system parameters that have been edited during operation (values of the system variable, interfaces, passwords, etc.) will be lost!

4.28.7 Download Cable

Operating Terminal
SER2

Personal
Computer



The shield is connected to the metal casing on both ends.

4.29 Simulation without the Controller

The terminal can be operated without a controller by setting the user-mode switch S3 to ON. The user-mode switch S3 will automatically set the external data release. It is possible to simulate the entire operator guidance including help texts. The variables are automatically assigned to a common variable address in the terminal. This allows the Editors and their limits to be tested. The values are, however, not retained. There is no communication with the controller.

SECTION 5

Transparent Mode

This section describes the transparent mode in detail. In transparent mode the PT reacts as an ANSI terminal.

5.1	Setting the Operating Mode	187
5.2	Start-up Processes	187
5.3	Communication in the Transparent Mode	188
5.3.1	Interface Parameters	188
5.3.2	Receive Buffer for SER1	188
5.4	Function Setup Menu	189
5.4.1	Adaptation of the Parameters in the Setup Menu	189
5.4.2	Example of a Properly Performed Modification!	189
5.4.3	Setup Menu NT4S-SF121B-E, NT4S-SF122B-E, NT4S-SF123B-E	190
5.5	Display	191
5.5.1	Character Set, Character Attributes	191
5.6	Keys	192
5.6.1	Key Codes for the Operating Terminals	192
5.7	Interface Control Characters	193
5.7.1	LED Codes for the Operating Terminals	194
5.7.2	Control Sequences for the Operating Terminals	194
5.8	Error Messages	197

5 Transparent Mode

Transparent mode is the name of an operation mode that lets the operating terminal function as a text monitor with printer interface. The controller must serve all characters and control sequences, that are needed to display text items and values. The operating terminal displays only fatal error messages (system messages). All parameters for the communication are set up in the setup mask.

5.1 Setting the Operating Mode

The user-mode switch must be used to select the transparent mode. The terminal must be off when selecting the mode of operation. To access the user-mode switch, open the enclosure on the rear side of the terminal. The terminal is factory-set to the standard mode! The positions ON/OFF are as indicated by the labelling on the switch.

Position of the switch for transparent mode:

S1	OFF	S5	not relevant
S2	not relevant	S6	not relevant
S3	OFF	S7	not relevant
S4	OFF	S8	not relevant

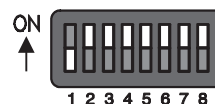


Fig. 27: User-Mode Switch

The selected mode of operation is activated once power is applied to the terminal.

All terminals are factory-set to standard mode!

5.2 Start-up Processes

The start-up processes are identical for all operating terminals. After the voltage has been applied, the following message will be displayed:



Fig. 28: Example of a start mask

The message will be displayed in an one-line, centered format or in case of smaller-sized displays in a two-line format. All LEDs will be activated while this message is being displayed, allowing the operator to perform a visual inspection. The message will be displayed for 2 sec. After this time period has elapsed, the message will be erased, the cursor will be positioned at the top left of the display (position 1, 1) and the characters received at interface channel SER1 will be interpreted and displayed.

Pressing the enter key immediately after the terminal has been switched on, while the prompt is being displayed, will activate the setup mask, containing the terminal-specific parameters, on the display. All editable parameters will be displayed within one screen. Editing of the interface parameters is possible after the data release key has been pressed, the status LED will be activated. The input via selection text prevents incorrect input. The data input is confirmed by pressing the data release key once more. The status LED will be deactivated. Pressing of the enter key will reboot the terminal with the current values.

Pressing the enter key before the prompt 'Press Enter to run Setup' is displayed, will result in the generation of the following error message during the keyboard test.



The self-test performed on the keyboard after having switched on the terminal has detected that a key is being pressed. Comply with the request and release the key. If the message is generated without any key being pressed, this indicates a defective keyboard!

5.3 Communication in the Transparent Mode

In transparent mode, the communication is carried out via interface channel SER2 exclusively. This is where the controller or the higher-level computer must be connected, taking into consideration the hardware configuration. Adaptation of the transmission parameters to the requirements must be carried out in the setup mask.

5.3.1 Interface Parameters

Unless otherwise specified, the following interface default parameters are used:

Interface	SER1	SER2
Baud rate	19200Bd	9600Bd
Data bits	8bit	7bit
Stop bits	1bit	1bit
Parity	none	odd
Handshake	Xon/Xoff	Xon/Xoff

The default parameters are stored in the flash memory and are therefore not lost.

5.3.2 Receive Buffer for SER1

The receive buffer of the communication interface SER1 comprises a size of 256 bytes.

5.4 Function Setup Menu

The setup menu allows selection of the following parameters:

- Parameters communication SER1
- Parameters communication SER2
- Date
- Time
- Default contrast

In addition, the version number of the operating system is displayed in the header line.

5.4.1 Adaptation of the Parameters in the Setup Menu

Within the setup mask, the parameters can be modified by means of an editor. Pressing of the data release key allows changing into the editor. Modifying of the data is possible once the status LED 'data release' has been activated. Modifying of the variable values is possible through the + key and - key or the numerical keys. The input must be confirmed by pressing the enter key. If supported by the display, the modified data will be displayed in an inverse format. To prevent incorrect or unintentional modifications, the input must be carried out in compliance with a pre-defined sequence which must be strictly adhered to.

5.4.2 Example of a Properly Performed Modification!

1. After having switched on the device, the enter key must be pressed immediately once the prompt is being displayed. The setup mask will appear on the display. This procedure must be carried out even if the parameters are to be viewed only.
2. It is possible to exit the menu by means of the enter key without modifying the menu. Upon pressing the data release key, the LED integrated in the key will be activated, thus signalling the editing mode within the transparent mode of operation.
3. The cursor will be positioned on the first editable variable and will be flashing. The cursor keys allow selection of the variable to be modified.
4. Numerical values may now be entered through the numerical keypad, texts are selected using the +,- keys.
5. The enter key must be pressed following each input. After a valid input, the cursor will appear on the next editable input field.
6. In case of interface parameters, the variable to be modified must be set to 'SETUP DATA', thereby preventing an accidental input. The selection 'SETUP DATA' must be made separately for the communication interface SER1 and the download/printer interface SER2.
7. Exiting of the editing mode is possible by pressing the data release key once more, the status LED will be deactivated; any modified values will still be displayed in an inverse format. Repeated corrections are possible using the data release key.

8. The enter key allows exiting of the setup menu. The terminal will be rebooted and the selected parameters will become effective. The entire process may be repeated, if necessary, it will not be necessary to switch the terminal off.

The data are stored in a battery-backed RAM memory. The battery is monitored such that a loss of data is prevented and a message will be displayed whenever the capacity of the battery is below the required level.

5.4.3 Setup Menu NT4S-SF121B-E, NT4S-SF122B-E, NT4S-SF123B-E

The setup masks for NT4S-SF121B-E, NT4S-SF122B-E and NT4S-SF123B-E are identical. This information is packed, thus permitting all parameters to be included into one mask.

```
XXXXXXXXX  YYYYYYYY
X2 300      N 5 1      N I
X3 300      N 5 1      N I
777777 888888LCD9999
```

1st line	XXXXXXXXX	Program Version	
	YYYYYYYYY	Driver Version	
2nd line	X2	Communication SER1	
(9600)	19200	Baud rate	(19200)
			(9600)
			(4800)
			(2400)
			(1200)
			(600)
			(300)
	N	Parity	(O)dd
			(E)ven
			(N)one
	5	Data bits	(8)bit
			(7)bit
			(6)bit
			(5)bit
	1	Stop bits	(1)bit
			(1.5)bit
			(2)bit
	N	Handshake	(N)one
			(R)TS/CTS
			(X)on/Xoff
	I	Acceptance	(I)nactive
			(S)etup Data
			(D)efault Values
3rd line	X3	Communication SER2	
	(Structure same as line 2)		
4th line	777777	Date	010100 to 311299
	888888	Time	000000 to 235959
	9999	Contrast	-125 to +125

5.5 Display

The first cursor position on the display is on the top left (X=1, Y=1). The number of permissible rows and columns varies with the terminal type.

	Rows:	Columns:
NT4S-SF121B-E Normal	4	20
NT4S-SF122B-E Normal	4	20
NT4S-SF123B-E Normal	4	20
NT15S-SF121B-E Normal	8	40
NT15S-SF121B-E Zoom	4	20
NT18S-SF121B-E Normal	16	40
NT18S-SF121B-E Zoom	8	20

5.5.1 Character Set, Character Attributes

Please note that it is not possible to address the extended ASCII character set in the range of 128_D to 255_D when selecting 7 data bits for the communication.

5.6 Keys

A start and stop code is transmitted to the controller for each key that is being pressed. Please consult the table of the terminal for the respective code being generated. The transmission of the characters is effected in compliance with the conditions of the selected interface parameters.

5.6.1 Key Codes for the Operating Terminals

Key	Start-code	Stop-code	NT4S-SF 121B-E	NT4S-SF 122B-E	NT4S-SF 123B-E	NT15S-SF 121B-E	NT18S-SF 121B-E
0	48	33	-	X	X	X	X
1	49	34	-	X	X	X	X
2	50	35	-	X	X	X	X
3	51	36	-	X	X	X	X
4	52	37	-	X	X	X	X
5	53	38	-	X	X	X	X
6	54	39	-	X	X	X	X
7	55	40	-	X	X	X	X
8	56	41	-	X	X	X	X
9	57	42	-	X	X	X	X
MINUS	111	125	X	X	X	X	X
DEC PT.	112	126	-	X	X	X	X
PLUS	113	124	X	X	X	X	X
DATA RELEASE	108	122	X	X	X	X	X
ENTER	109	121	X	X	X	X	X
DELETE	110	123	-	X	X	X	X
?-HELP	106	107	X	X	X	X	X
CURSOR UP	101	118	-	X	X	X	X
CURSOR DOWN	102	119	X	X	X	X	X
CURSOR LEFT	103	116	X	X	X	X	X
CURSOR RIGHT	104	117	-	X	X	X	X
HOME	105	120	-	X	X	X	X

Key	Start-code	Stop-code	NT4S-SF 121B-E	NT4S-SF 122B-E	NT4S-SF 123B-E	NT15S-SF 121B-E	NT18S-SF 121B-E
F1	65	66	x	x	x	x	x
F2	67	68	x	x	x	x	x
F3	69	70	x	x	x	x	x
F4	71	72	x	x	x	x	x
F5	73	74	-	x	x	x	x
F6	75	76	-	x	x	x	x
F7	77	78	-	-	x	x	x
F8	79	80	-	-	x	x	x
F9	81	82	-	-	-	x	x
F10	83	84	-	-	-	x	x
F11	85	86	-	-	-	x	x
F12	87	88	-	-	-	x	x
F13	89	90	-	-	-	x	-
F14	91	92	-	-	-	x	-
F15	93	94	-	-	-	x	-
F16	95	96	-	-	-	x	-
print	77	78	-	x	-	x	-
page up	79	80	-	x	-	x	-
page down	.	.	-	-	-	x	-
acknowledge	.	.	-	-	-	x	-

5.7 Interface Control Characters

Control Characters: Control characters are interpreted by the terminal and can not be displayed on the display!

<u>Character</u>		<u>Code</u>	<u>Function</u>
Backspace	BS	8	Cursor is moved to the left by one column.
Linefeed	LF	10	Cursor is moved downwards by one line.
Return	CR	13	Cursor is moved to the beginning of the line.
Xon	XON	17	Ready-signal during software handshake
Xoff	XOFF	19	Not ready until next XON
Escape	ESC	27	Start code for all control sequences
Delete	DEL	127	Deletes character beneath cursor position

5.7.1 LED Codes for the Operating Terminals

Status LED	Start-code	NT4S-SF 121B-E	NT4S-SF 122B-E	NT4S-SF 123B-E	NT15S-SF 121B-E	NT18S-SF 121B-E
DATA RELEASE ?-HELP	20 19	x x	x x	x x	x x	x x
F1	1	x	x	x	x	x
F2	2	x	x	x	x	x
F3	3	x	x	x	x	x
F4	4	x	x	x	x	x
F5	5	-	x	x	x	x
F6	6	-	x	x	x	x
F7	7	-	-	x	x	x
F8	8	-	-	x	x	x
F9	9	-	-	-	x	x
F10	10	-	-	-	x	x
F11	11	-	-	-	x	x
F12	12	-	-	-	x	x
F13	13	-	-	-	x	-
F14	14	-	-	-	x	-
F15	15	-	-	-	x	-
F16	16	-	-	-	x	-

5.7.2 Control Sequences for the Operating Terminals

Sequence	Description
<ESC>	Represents the ESC character.
<A>	Represents the line number.
	Represents the column number.
	<A> and correspond to a decimal number, specified through one or several ASCII characters. The limits are governed by the size of the display.
<An>	Represents a decimal number which selects a sub-function. Multiple sub-functions must be separated by semicolons.
<Pn>	Represents a numerical parameter. Denotes a decimal number.

NT4S-SF NT4S-SF NT4S-SF NT15S-SF NT18S-SF
121B-E 122B-E 123B-E 121B-E 121B-E

X	X	X	X	X	<ESC>[2J.....Erase screen, and cursor in line 1 / column 1.
X	X	X	X	X	<ESC>[K.....Erasing of the line: Deletes all characters to the right of the cursor position up to the end of the line, including the cursor position.
X	X	X	X	X	<ESC>[<A>;h.....Positioning of the cursor in line A of column B. A, B limited by the display.
X	X	X	X	X	<ESC>[<A>;H.....Positioning of the cursor in line A of column B. A, B limited by the display.
X	X	X	X	X	<ESC>[<Pn>A.....Cursor upwards: Moves the cursor up by the number of lines specified without changing the column. Within the uppermost line, the command will be ignored.
X	X	X	X	X	<ESC>[<Pn>B.....Cursor downwards: Moves the cursor down by the number of lines specified without changing the column. Within the last line on the bottom, the command will be ignored.
X	X	X	X	X	<ESC>[<Pn>C.....Cursor to the right: Moves the cursor to the right by the number of columns specified without changing the line. Within the column on the extreme right, the command will be ignored.
X	X	X	X	X	<ESC>[<Pn>D.....Cursor to the left: Moves the cursor to the left by the number of columns specified without changing the line. Within the column to the extreme left, the command will be ignored.
X	X	X	X	X	<ESC>&C.....Activate cursor.
X	X	X	X	X	<ESC>&D.....Deactivate cursor.
X	X	X	X	X	<ESC>[s.....Stores the cursor position: An additional ESC-sequence can be used to move the cursor back to the stored position.
X	X	X	X	X	<ESC>[u.....Restores the cursor position: Moves the cursor back to the stored position.
-	-	-	X	X	<ESC>&Z.....Characters are displayed in capitals (Zoom 2).
-	-	-	X	X	<ESC>&A.....Characters are displayed in normal size.
X	X	X	X	X	<ESC>[<A1>;<An>m.....Setting the character attributes. A1 to An corresponds to the attribute number
X	X	X	X	X	0 = deselects all attributes
-	-	-	X	X	4 = underline
X	X	X	X	X	5 = flashing
-	-	-	X	X	7 = inverse
X	X	X	X	X	<ESC>[<A1>;<An>q.....Activate LED. A1 to An corresponds to the number of the LED
X	X	X	X	X	A = 0 activate all LEDs
X	X	X	X	X	<ESC>&q<A1>;<An>:.....LED flashing. A1 to An corresponds to the number of the LED
X	X	X	X	X	A = 0 all LEDs flashing
X	X	X	X	X	<ESC>&p<A1>;<An>:.....Deactivate LED. A1 to An corresponds to the number of the LED
X	X	X	X	X	A = 0 deactivate all LEDs
X	X	X	X	X	<ESC>[6n.....Inquiry by the controller as to which terminal is connected. Response provided by the terminal:
X	-	-	-	-	Response <ESC>[02
-	X	-	-	-	Response <ESC>[05
-	-	X	-	-	Response <ESC>[10
-	-	-	X	-	Response <ESC>[15
-	-	-	-	X	Response <ESC>[20

NT4S-SF NT4S-SF NT4S-SF NT15S-SF NT18S-SF
121B-E 122B-E 123B-E 121B-E 121B-E

x	x	x	x	x	<ESC>&t.....Display time at cursor position. Format hh:mm:ss
x	x	x	x	x	<ESC>&d.....Display date at cursor position. Format TT.MM.JJ
x	x	x	x	x	<ESC>&JJMMTThhmmssi.....Set Date / Time. JJ = Year MM = Month TT = Day hh = Hours mm = Minutes ss = Seconds
x	x	x	x	x	<ESC>&u.....Transmit clock to interface X2. The response from the terminal to this request has the same format as the setting for the date/time above. Response: <ESC>&JJMMTThhmmssi
x	x	x	x	x	<ESC>[<A>;U.....Cyclic output of the time in line A and column B is started. The output occurs in accordance with the current character attributes.
x	x	x	x	x	<ESC>[0;0U.....Cyclic output of the time is stopped.
x	x	x	x	x	<ESC>&K.....Default value for the display's contrast setting.
x	x	x	x	x	<ESC>&+.....Increments the selected value by +1. The contrast becomes 'brighter'. The maximum total range of val- ues is ± 125 . This range of values is being limited to values appropriate for the respective display.
x	x	x	x	x	<ESC>&-.....Decrements the selected contrast value by -1. The contrast is 'dimmed'.

5.8 Error Messages

Any error messages generated by the terminal are principally displayed on the top left of the display. The error messages are generated by the terminal software in plain text. The error message is displayed once at the cursor position 1,1. The previous text will be overwritten. The message will not be deleted! - but will rather need to be overwritten by the user. This must be taken into consideration when creating the program. The following internal error text might be displayed:

LOW BAT	Indicates that the capacity of the battery to back data is falling below the required level. Upon initial display of this message, the battery should be replaced within 2-3 days! A battery test is performed after the terminal has been switched on and every 60 minutes thereafter.
FRAMING ERROR	The data format of the characters received at the interface does not conform with the selected interface parameters. Check number of data bits, stop bits and parity bit.
PARITY ERROR	A parity error has been detected regarding the characters received at the interface. This indicates a possible problem with the transmission line (character length, baud rate, interference, invalid interface parameter etc.)
OVERRUN ERROR	The terminal was unable to interpret and evaluate the characters received at the interface. This indicates a possible error regarding the handshake procedure between terminal and controller. Check hardware or software handshake.
TRANSMIT ERROR	On account of a blocked transmitter (no handshake enable signal), the terminal was unable to transmit the characters within the specified timeout period. This indicates an error regarding the handshake procedure between terminal and controller. Check hardware or software handshake.
KEYBOARD ERROR PLEASE RELEASE KEY	The self-test performed on the keyboard after having switched on the terminal has detected that a key is being pressed. Please comply with the message. Display of this message without any key being pressed indicates a defect in the keyboard!

SECTION 6

Linking to OMRON Controllers

This section describes in detail how to connect to an OMRON PLC using hostlink or NT-Link.

6.1	OMRON Host-Link	201
6.1.1	General Information	201
6.1.1.1	Hardware	201
6.1.1.2	Telegram Mode	201
6.1.1.3	Point-to-Point Connection	201
6.1.1.4	Multiple Point Connection	201
6.1.2	Protocol Parameters	201
6.1.2.1	Parameters of SER1	201
6.1.2.2	Additional Parameters	202
6.1.2.2.1	Parameter Maximum Waiting Time for Response	202
6.1.2.2.2	Parameter Delay until Connection Set-up	202
6.1.2.2.3	Parameter Monitor Mode	202
6.1.3	Data Types	202
6.1.3.1	Data Type Orientation	203
6.1.4	Poll Area	204
6.1.5	Parallel Message System	204
6.1.6	Connecting Cable SER1RS232 - OMRON Host-Link	204
6.1.7	Error Messages	205
6.2	OMRON NT-Link	205
6.2.1	General Information	205
6.2.1.1	Hardware	205
6.2.1.2	Telegram Mode	205
6.2.1.3	Point-to-Point Connection	206
6.2.1.4	Multipoint Connection	206
6.2.2	Protocol Parameters	206
6.2.2.1	Parameters of SER1	206
6.2.2.2	Additional Parameters	206
6.2.2.2.1	Parameter Maximum Waiting Time for Response	206
6.2.2.2.2	Parameter Delay until Connection Set-up	207
6.2.2.2.3	Parameter NT Module Number	207
6.2.2.2.4	Parameter Protocol Mode	207
6.2.2.2.5	Parameter Priority	207
6.2.2.2.6	Parameter Duplex Operation	207
6.2.2.2.7	Parameter PLC Type	207
6.2.3	Data Types	208
6.2.3.1	Data Type Orientation	209
6.2.4	Poll Area	210
6.2.5	Parallel Message System	210
6.2.6	Connecting Cable SER1 RS232c - OMRON NT-Link	210
6.2.7	Connecting Cable SER1 RS485 2-Wire - OMRON NT-Link	211
6.2.8	Connecting Cable SER1 RS485 4-Wire - OMRON NT-Link	211
6.2.9	Error Messages	212

6 Linking to OMRON Controllers

6.1 OMRON Host-Link

6.1.1 General Information

TesiMod operating terminals can be conveniently connected to all controllers that support the OMRON Host-Link protocol. The OMRON Host-Link protocol allows communication between one operating terminal and up to 32 controllers.

6.1.1.1 Hardware

The interface channel SER1 RS232 is used to connect the operating terminals to a OMRON controller.

6.1.1.2 Telegram Mode

The operating terminal always has a client functionality which means that data transfers are always initiated by the terminal.

6.1.1.3 Point-to-Point Connection

RS232 interfaces allow only point-to-point connections.
Therefore the operating terminals can be set to Single-Mode or Multiple-Mode. The selected mode must be equal to the mode set for the controller.
Most of the OMRON controllers are used with Multiple-Mode and cannot be adjusted

6.1.1.4 Multiple Point Connection

In the telegram mode Multiple-Mode the terminal can be connected to at least 32 controllers using multiple point connection.
In this case the terminal must be connected via the communication module NT-AL001 with the RS422-Bus.
A direct connection with the RS485 interface of SER2 of the terminals is not permissible due to a lack of terminator possibilities.

6.1.2 Protocol Parameters

6.1.2.1 Parameters of SER1

Parameters that can be defined for the communication interface SER1 are as follows:

Baud rate:	300, 600, 1200, 2400, 4800, 9600, 19200
Data bits:	7 , 8
Parity:	even , odd
Stop bits:	1 , 2
Handshake:	no handshake

Default values in the programming software are in **bold** print.

6.1.2.2 Additional Parameters

6.1.2.2.1 Parameter Maximum Waiting Time for Response

This parameter specifies the maximum time the operating terminal waits for a response from the PLC.

The valid range of values is 50 ms to 65535 ms.

The value 500 ms is specified by the programming software by default.

6.1.2.2.2 Parameter Delay until Connection Set-up

This parameter specifies the time the operating terminal waits before starting communication.

The valid range of values is 5 s to 255 s.

The value 5 s is specified by the programming software by default.

6.1.2.2.3 Parameter Monitor Mode

The PLC can be used in the RUN or MONITOR mode of operation.

Write access via the protocol is not possible in the RUN mode.

To be able to edit variables from the terminal, the PLC is switched to the Monitor mode whenever the operating terminal is booted.

6.1.3 Data Types

The following data types are available for direct user access.

The address ranges depend on the type of PLC.

The valid address ranges are specified below.

PLC Data Types	Mnemonic Code in Variable List	Address Range	Access
Intern Bit	IR	0.0 to 2555.15	Bit
Intern Word	IR	0 to 2555	Word
Intern Double Word	IRd	30 to 2554	Double Word
Link Bit	R	0.0 to 63.15	Bit
Link Word	LR	0 to 63	Word
Link Double Word	LRd	0 to 62	Double Word
Holding Bit	HR	0.0 to 99.15	Bit
Holding Word	HR	0 to 99	Word
Holding Double Word	HRd	0 to 98	Double Word
Auxiliary Bit	AR	0.0 to 511.15	Bit
Auxiliary Word	AR	0 to 511	Word
Auxiliary Double Word	ARd	0 to 510	Double Word
Data Memory Word	DM	0 to 9999	Word
Data Memory Double Word	DMd	0 to 9998	Double Word
Timer/Counter	TC	0 to 1023	Read-Only Word

6.1.3.1 Data Type Orientation

a) Alphanumerical Texts

These data are interpreted in the PLC-conforming byte order for words.

b) Floating Point

These data are interpreted in IEEE floating point format.

c) Binary Variables with a Length of 2 or 4 Bytes

Data with a length of 2 bytes are interpreted in the PLC-conforming byte order for words.

Data with a length of 4 bytes are interpreted in the PLC-conforming byte order for double words.

6.1.4 Poll Area

The poll area must reside in the data memory area and be addressed as a data memory word (DM).

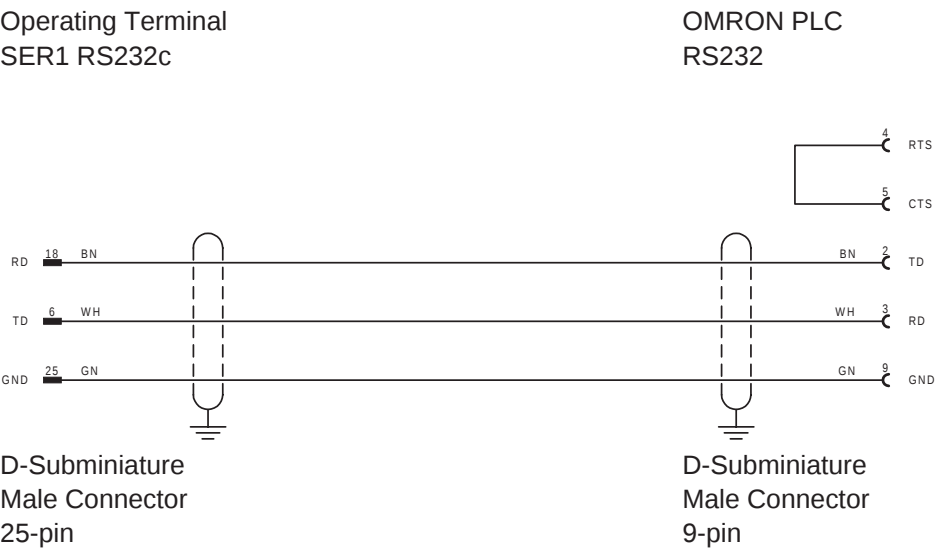
Example:
Word address + 0 DM 100
Word address + 1 DM 101

6.1.5 Parallel Message System

The parallel message system must reside in the data memory area and be addressed as a data memory word (DM).

Example:
Word address + 0 DM 100
Word address + 1 DM 101

6.1.6 Connecting Cable SER1RS232 - OMRON Host-Link



6.1.7 Error Messages

Code	50	E_HARDWARE	Communication error on hardware level
Code	60	E_PROTOKOL.....	Communication error on protocol level
	Subcode		
		10. E_RESPONSE_TIMEOUT...	No response from PLC
		20. E_RECEIVE_COUNT	Wrong number of data received
		30. E_FCS.....	Wrong checksum
Code	70	E_UNDEF_CODE.....	Undefined command
	80	E_RESPONSE_CODE.....	Error from PLC
	Subcode		
		01.....	Can not be executed when PLC is in RUN mode
		02.....	Can not be executed when PLC is in MONITOR mode
		13.....	Frame checksum error
		14.....	Command format error
		15.....	Wrong read or write areas
		16.....	Command not available
		18.....	Frame length error
		23.....	User memory is write-protected

6.2 OMRON NT-Link

6.2.1 General Information

All operating terminals can be conveniently connected to controllers that support the OMRON NT-Link protocol. The OMRON NT-Link protocol allows communication between one controller and up to 8 operating terminals. Since the Host-Link protocol is the default setting in OMRON controllers, this setting must be checked before using the NT protocol. See the controller's operating instructions.

6.2.1.1 Hardware

The SER1 RS232 or SER1 RS485 is used to connect the operating terminals to OMRON controllers.

6.2.1.2 Telegram Mode

With the NT-Link, a distinction is made between two protocols. The 1:1 protocol is used for point-to-point connections while the 1:N protocol is used for multipoint connections.

6.2.1.3 Point-to-Point Connection

The 1:1 protocol is used for point-to-point connections. This means that only one operating terminal can be connected with one controller. Interfaces that can be used with this connection type are the RS232 or the RS422 / RS485 interface. A 4-wire cable must be used in conjunction with the RS422 / RS485 interface.

6.2.1.4 Multipoint Connection

The 1:N protocol is used for multipoint connections. Up to 8 operating terminals can be connected to one controller.

The controller must be configured to address at least two operating terminals. The connection must be established using SER1 RS485. A 2-wire cable must be used to ensure a sufficient termination.

6.2.2 Protocol Parameters

6.2.2.1 Parameters of SER1

Parameters that can be defined for SER1 are as follows.

Different default values are predefined for the parameters depending on the protocol.

1:1 NT-Link

Baud rate: 300, 600, 1200, 2400, 4800, 9600, **19200**, 38400
 Data bits: 7, **8**
 Parity: even, **odd**
 Stop bits: **1**, 2
 Handshake: **no handshake**

1:N NT-Link

Baud rate: 300, 600, 1200, 2400, 4800, 9600, 19200, **38400**
 Data bits: 7, **8**
 Parity: even, **odd**
 Stop bits: **1**, 2
 Handshake: **no handshake**

Default values in the programming software are in **bold** print.

6.2.2.2 Additional Parameters

6.2.2.2.1 Parameter Maximum Waiting Time for Response

This parameter specifies the maximum time the operating terminal waits for a response from the PLC.

The valid range of values is 50 ms to 65535 ms.

The default value in the programming software is 500 ms.

6.2.2.2.2 Parameter Delay until Connection Set-up

This parameter specifies the time the operating terminal waits before starting communication.

The valid range of values is 5 s to 255 s.

The default value in the programming software is 5 s.

6.2.2.2.3 Parameter NT Module Number

This parameter is used to set the NT module number for the operating terminal. If multiple operating terminals are connected to one controller, then a separate number must be specified for each operating terminal. Otherwise, a reliable communication between the operating terminals and the controller is not possible.

Valid values that can be specified as a NT module number range from 0 to 7.

The default value is 0.

The NT module number can also be set via the operating terminal by means of the system variable **ComSlaveNr**.

6.2.2.2.4 Parameter Protocol Mode

Selectable protocol modes are either 1:1 or 1:N. Ensure that this parameter and the setting specified for the controller match.

6.2.2.2.5 Parameter Priority

This parameter applies only in conjunction with the 1:N protocol.

If the priority is turned off, the master allocates the token sequentially to each station. If the parameter Priority is turned on, the master allocates every second token to the operating terminal.

6.2.2.2.6 Parameter Duplex Operation

Select the parameter **full duplex** when using **SER1 RS232**.

The parameter **half duplex** applies only in conjunction with **SER1 RS485**.

6.2.2.2.7 Parameter PLC Type

Different data types are permitted depending on the controller type. Therefore, select either the PLC type C-Series or CV-Series.

6.2.3 Data Types

Data types for the C-series:

Mnemonic	Range	Address Width	Access
IR	0.0 to 511.15	Bit access	
IR	0 to 511	Word access	
IRd	0 to 510	Double-word access	
LR	0.0 to 63.15	Bit access	
LR	0 to 63	Word access	
LRd	0 to 62	Double-word access	
HR	0.0 to 99.15	Bit access	
HR	0 to 99	Word access	
HRd	0 to 98	Double-word access	
AR	0.0 to 27.15	Bit access	
AR	0 to 27	Word access	
ARd	0 to 26	Double word access	
DM	0.0 to 9999.15	Bit access	
DM	0 to 9999	Word access	
DMd	0 to 9998	Double word access	
EM	0.0 to 6143.15	Bit access	
EM	0 to 6143	Word access	
EMd	0 to 6142	Double-word access	
TC	0 to 511	Word access	read-only

Data types for the CV-series:

Mnemonic	Range	Address Width	Access
IR	0.0 to 2555.15	Bit access	
IR	0 to 2555	Word access	
IRd	0 to 2554	Double-word access	
GR	0.0 to 255.15	Bit access	
GR	0 to 255	Word access	
GRd	0 to 254	Double-word access	
AR	0.0 to 511.15	Bit access	
AR	0 to 511	Word access	
ARd	0 to 510	Double-word access	
DM	0 to 24575	Word access	
DMd	0 to 24575	Double-word access	
EM	0 to 32765	Word access	
EMd	0 to 32764	Double-word access	
TIM	0 to 1023	Word access	read-only
CNT	0 to 1023	Word access	read-only

6.2.3.1 Data Type Orientation

a) Alphanumerical Texts

These data are interpreted in the PLC-conforming byte order for words.

b) Floating Point

These data are interpreted in IEEE floating point format.

c) Binary Variables with a Length of 2 or 4 Bytes

Data with a length of 2 bytes are interpreted in the PLC-conforming byte order for words.

Data with a length of 4 bytes are interpreted in the PLC-conforming byte order for double words.

6.2.4 Poll Area

The poll area must reside in the data memory area and be addressed as a data memory word (DM).

Example:

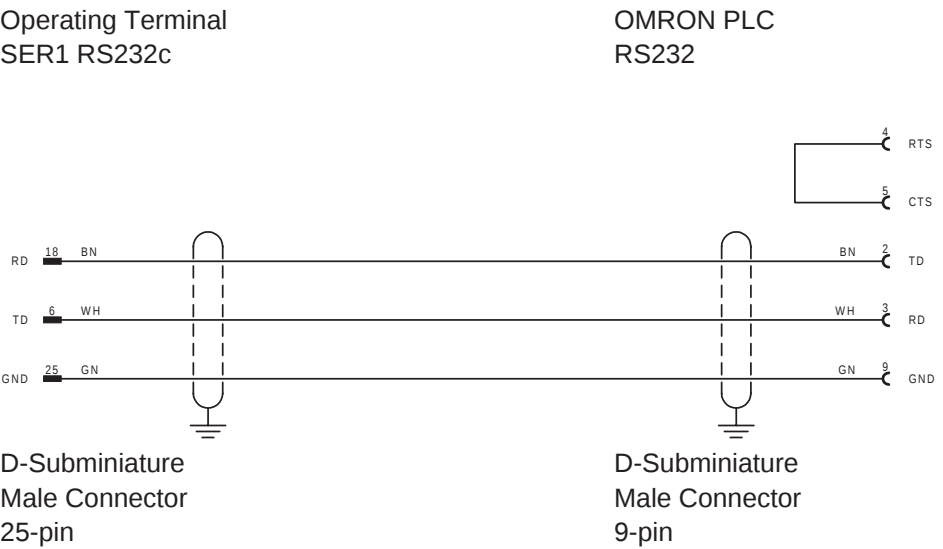
Word address	Reference	High-byte	Low-byte
Word address +0	DM100	Write coordination byte	Reserved
Word address +1	DM101	Message channel, high-byte	Message channel, low-byte
Word address +2	DM102	Function key LEDs 1 to 4	LEDs 5 to 8
Word address +3	DM103	Function key LEDs 9 to 12	LEDs 13 to 16
Word address +4	DM104	Function key LEDs 17 to 20	LEDs 21 to 24
Word address +5	DM105	Function key LEDs 25 to 28	LEDs 29 to 32

6.2.5 Parallel Message System

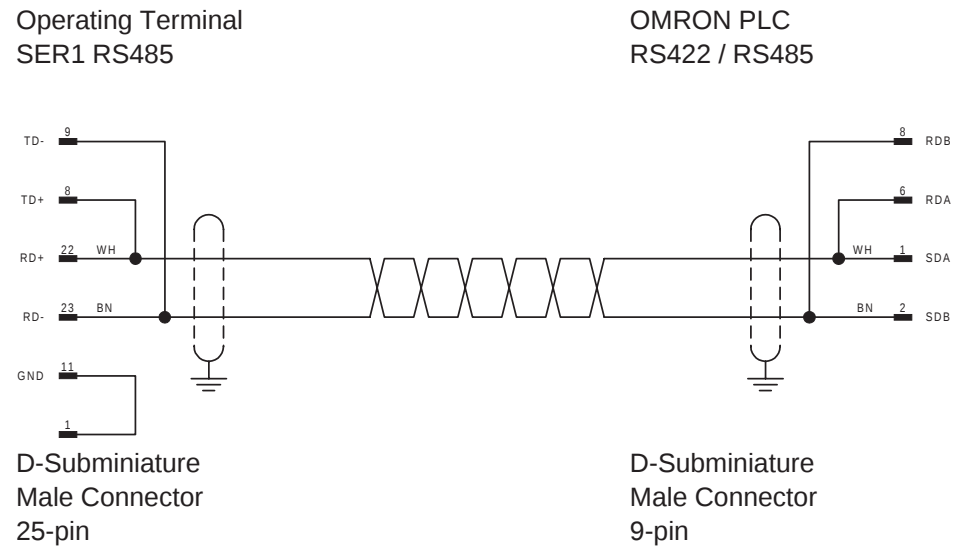
The parallel message system must reside in the data memory area and be addressed as a data memory word (DM).

Example:
Word address + 0 DM 100
Word address + 1 DM 101

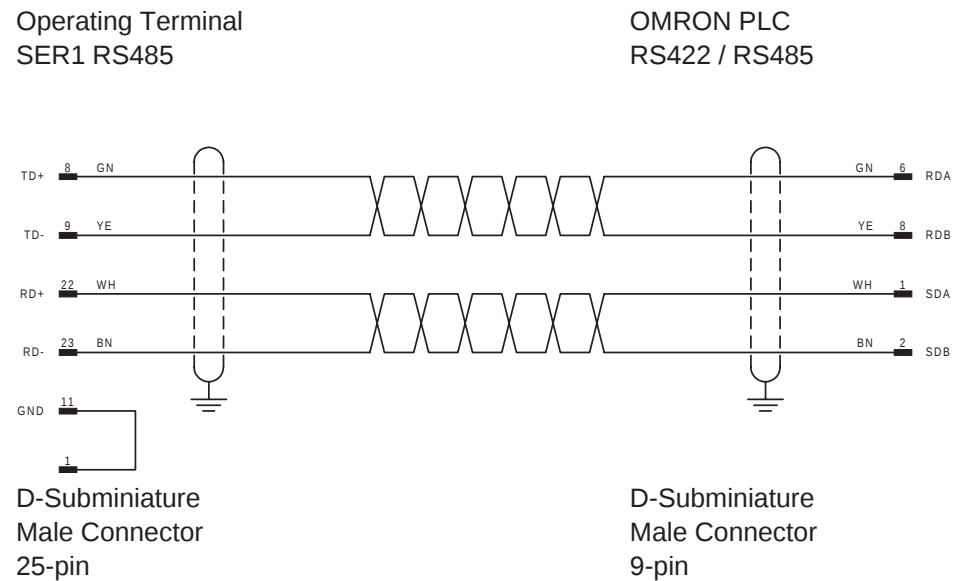
6.2.6 Connecting Cable SER1 RS232c - OMRON NT-Link



6.2.7 Connecting Cable SER1 RS485 2-Wire - OMRON NT-Link



6.2.8 Connecting Cable SER1 RS485 4-Wire - OMRON NT-Link



6.2.9 Error Messages

Code	50	E_HARDWARE	Communication error on the hardware level
		Subcode	
		3.....E_FRAME.....	Protocol frame error
		6.....E_PARITY	Parity error
		10.....E_NO_DEFINE	Poll area not defined
		16.....E_OVERRUN	Receive buffer overrun
		40.....E_SYS_ADDRESS.....	Access to unknown system address
Code	60	E_PROTOKOL.....	Communication error on the protocol level
		Subcode	
		10.....E_RESPONSE_TIMEOUT...	No response from PLC
		20.....E_WRONG_PROTOKOL....	Terminal setting = 1:1, Controller setting= 1:N
		30.....E_FCS.....	Wrong check sum
Code	70	E_UNDEF_CODE.....	Undefined command
	80	E_RESPONSE_CODE	Error from PLC
		Subcode	
		00.....	Error in check sum
		01.....	Undefined command
		03.....	Format contains error
		05.....	Parameters contain error
		Possible cause:	
		- wrong address	
		07.....	Status error

INDEX

A

Application Memory	43
Authorization levels.....	66

B

Battery	42
---------------	----

C

Character Attributes	39
Configuration mask	147, 188, 190
Control Keys	10
Cursor down	10
Cursor home	10
Cursor left	10
Cursor right	10
Cursor up	10

D

DIP switch S4	181, 183
Download cable	180

E

Error Message	188
Error messages	181
Error messages	141
ADDRESS ERROR.....	141
BYTECOUNT OVERFLOW	141
CHECKSUM ERROR.....	141
COMMUNICATION ERROR	141
DATASET STORAGE FAILURE	145
DIFFERENT DRIV VERS	142
DIFFERENT MASK VERS	142
DOWNLOAD	143
ERASE FLASH EPROM	143
!!!! ERROR !!!!!	142
FATAL ERROR	145
FIRMWARE NOT CONFORM	145
FLASH CHECKSUM ERROR.....	145
FLASH IS ERASED	143
FLASH MEMORY FAILURE	141
FLASH NOT ERASEABLE.....	144
FORMAT ERROR.....	141
INITIALIZING MESSAGE BUFFER	143
KEYBOARD ERROR	143
MEMORY IS FLASH XXXXK	145
NO FLASH EPROM	144
TERMINAL-TYP IS XXXX	145
TURN POWER OFF.....	142

UNEXPECTED INTERRUPT	144
!!!! WARNING!!!!	142
WRONG S3-FILE	144
External data release	121

F

Firmware.....	180
Formatted output of variables	73
Formula for scaling input values	84
Formula for scaling the output of variables.....	75
Front Panel Dimensions	23
NT15S-SF121B-E	23
NT18S-SF121B-E	24
NT4S-SF122B-E	22
NT4S-SF123B-E	23
Front View	7
NT15S-SF121B-E	7
NT18S-SF121B-E	8
NT4S-SF122B-E	5
NT4S-SF123B-E	6
Function Key Arrangement	13
NT15S-SF121B-E	13
NT18S-SF121B-E	13
Fuse.....	43

H

Help key	115
Help text	84
for input variables	84

I

Input variable	84
Counter	84
Timer	83
Interface Parameters	188

K

Key	115
Data Release	115
Key functions	119
editor for coded text	119
Key functions in the alphanumerical editor.....	118
Key functions in the BCD-number editor.....	117
Key functions in the hexadecimal editor	118
in the I/O mask	72
Key functions in the numerical editor	116

M	
Marginal conditions.....	169
Mask number	164
Mask types.....	70
Master password	68
Mode selector switch	63
Standard Mode.....	63
N	
Number of languages.....	178
O	
One-time output variables/cyclic output variables	73
P	
Password editor	118
Password protection.....	66
access level.....	66
authorization level	66
setup mask.....	68
view level	67, 68
Pin assignment.....	32
Plausibility	115
Plausibility check	84
input variable.....	84
Position of the Switch for Transparent Mode.....	187
Post-decimal places.....	73
R	
Range of values	72, 73
Refresh intervals	180
Representation of output variables	75
Representation with leading zeros	75
Running time meter	162
reset byte	162
S	
Scaled output of variables	74
SETUP DATA.....	189
Side View	27
NT15S-SF121B-E	27
NT18S-SF121B-E	28
NT4S-SF122B-E	25
NT4S-SF123B-E	26
Slide-in Identification Strips for the Function Keys	15
NT15S-SF121B-E	15
NT18S-SF121B-E	15
NT4S-SF122B-E	14
NT4S-SF123B-E	15
Special Keys	10
Status LED.....	121
data release	121
Symbolic name	164
System message.....	139
data set active.....	139
data set download	140
data set file error	139
data set format	140
data set memory full	139
data set protected	139
data set transfer	139
data set unknown.....	139
editing mode active.....	139
system message	139
illegal data set.....	139
System message.....	139
interface in use	139
invalid mask no.....	138
invalid message no	138
invalid password.....	139
loop-through active	140
message buffer full	138
message overflow	138
new message	138
no data set address	140
overvoltage	139
password missing	139
password unchanged.....	139
print log invalid	138
recipe unknown	140
replace battery	138
value too large	138
value too small	138
wrong format	138
(floating point) number invalid.....	140
System numbers	137
System variables.....	86
basic functions	86
Boot.....	86
ComVersion.....	85
handshake	172
IntEraseEeprom	85
LCDBackground.....	86
LCDBackLight.....	86
LCDContrast	86
MainVersion.....	85
OsLanguage	87
TurnOnTemp	87
UserVersion.....	86
communication area X2.....	88
ComBaudrateA.....	88
ComDataLenA.....	87
ComDefaultA	88

Index

ComHandshakeA.....	88	KeyEdit	105
ComParityA	87	KeyEnter	105
communication area x2	89	KeyHelp.....	102
ComRetryTimeout	89	KeyHome	102
communication area X2.....	89	KeyMinus	105
ComSlaveNr	89	KeyPlus	105
ComStopBitsA.....	88	Key0	103
ComTimeout	89	Key1	103
communication area X3.....	91	Key2	103
ComBaudrateB.....	91	Key3	104
ComDataLenB.....	90	Key4	104
ComDefaultB	91	Key5	104
ComHandshakeB.....	91	Key6	104
ComParityB	91	Key7	104
ComStopBitsB.....	91	Key8	105
editors	114	Key9	105
EditEnter	114	NewMask	99
EditInvers	114	Shift.....	100
StatePerm.....	114	ShiftCase	101
error statistics interface X2.....	90	TabLeft.....	100
ComErrorTab.....	90	TabPgDn.....	100
ComFrameCount	90	TabPgUp.....	100
ComOverrunCount	89	TabRight	100
ComParityCount.....	89	VarTablenR0	99
ComStatisticsTab.....	90	VarTablenR1	99
ComSubcodeTab.....	90	parallel message system.....	96
help	114	RepmanSortCritP.....	96
Message	114	RepoutAnzYearP	97
QuitMessage	115	RepoutDateP.....	96
StateHelp	114	RepoutNrP	96
loadable font	113	RepoutRepTextP	97
ChrsetName.....	113	RepoutRepTextP21.....	97
loop-through operation.....	112	RepoutRepTextP41.....	97
Pg2Sps	112	RepoutRepTextP61.....	97
Pg2SpsState	112	RepoutTimeP	96
maintenance	113	password	106
Break	113	ChangePasswd.....	106
LCDADCInput	113	FlashPasswd.....	106
LCDADCOutput	113	MskchgPasswd.....	106
User1	113	MskchgResPasswd.....	106
User2	113	PasswdInactive	106
User3	113	password protection	66
User4	113	edit level	66
User5	113	printer control	98
menu control / keys.....	99	BlockPrint	98
HardCopy	99	BlockPrintLong	99
KeyClear	103	PrintAllRep	98
KeyCursDown.....	102	PrintAllState	98
KeyCursLeft.....	101	StopPrint	98
KeyCursRight	102	real-time clock.....	93
KeyCursUp.....	102	RTCDateFmt	93
KeyDot.....	103	RTCDay.....	92

Index

RTCDayofWeek	93
RTCHour	92
RTCMin	92
RTCMonth	92
RTCSec	92
RTCYear	93
RTCYear2000	93
recipes	108
ActDSName	108
DestDSNr	107
DSCopy	107
DSDDelete	108
DSDnloadBreak	108
DSDnloadState	109
DSDownload	108
LoadDSName	109
RestoreLineNr	110
RestoreState	110
RezPrintState	110
SaveState	109
SelectDSName	107
SelectDSNr	107
SelectRezeptNr	108
StartRestore	110
StartRezPrint	110
StartSave	109
StartUpload	111
UploadDSNr	111
UploadState	111
running time meter	112
Counter1	112
Counter2	112
Counter3	112
Counter4	112
Counter5	112
Counter6	112
Counter7	112
Counter8	112
serial message system	94
ClearRepBuf	94
RepmanRepPrint	94
RepmanSortCrit	93
RepoutAnzYear	95
RepoutDate	94
RepoutNr	94
RepoutRepText	95
RepoutRepText21	95
RepoutRepText41	95
RepoutRepText61	95
RepoutTime	94

T

Technical Data	51
NT15S-SF121B-E	51
NT18S-SF121B-E	53
NT4S-SF122B-E	48
NT4S-SF123B-E	50
Transmission Parameters	188

U

User-Mode Switch	187
------------------------	-----

V

Variable type	72
ASCII	72
bit	72
byte	72
Lword	72
word	72

Revision History

A manual revision code appears as a suffix to the catalog number on the front cover of the manual.

Cat. No. V056-E1-1



Revision code

The following table outlines the changes made to the manual during each revision. Page numbers refer to the previous version.

Revision code	Date	Revised content
1	March 1999	Original production

OMRON

Authorized Distributor:

Regional Headquarters

OMRON EUROPE B.V.

Wegalaan 67-69

NL-2132 JD Hoofddorp

The Netherlands

Tel: +31 (0)23 56 81 300

Fax: +31 (0)23 56 81 388