

PIC / PC400 / PC600 / CL100 / CL300 / CL400 / CL500

# PLC Operating System Software manual

Version

# 104

PIC / PC400 / PC600 / CL100 / CL300 / CL400 / CL500

# PLC Operating System Software manual

1070 066 889-104 (92.10) GB



© 1987

by Robert Bosch GmbH,  
All rights reserved, including applications for protective rights.  
Reproduction or handing over to third parties are subject to our written permission.

Discretionary charge 30.– DM

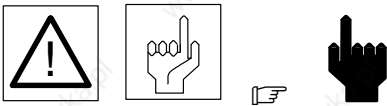


## Reading help

Read this manual before you start using the software. Keep this software manual in a place where it is always accessible for all users.

These software instructions are intended for use by PLC programmers, and knowledge of the MS DOS operating system is required. For programming a controller you also need to know the controller commands.

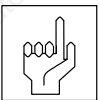
Please support us in improving this manual, and use the form at the back of the manual for your suggestions.



These symbols are used throughout this manual to indicate the following.



This symbol is used whenever an insufficient or lacking compliance with instructions can result in **personal injury**.



This symbol is used whenever an insufficient or lacking compliance with instructions can result in **damage to equipment or files**.



This symbol is used to draw the attention of the reader to special points.

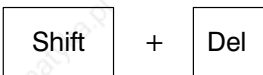


This symbol is used to point out changes in diagrams.

## Symbols used



Frequently, input has been simplified in that only one key has to be pressed. This symbol indicates which **key** on the keyboard should be pressed.



If there is a **plus sign +**, between two or more keys, these must be pressed **simultaneously**.



This sign indicates that an activity is being described which is to be performed by the reader, e.g.:

★ Insert disk 1 into the floppy disk drive.

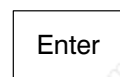


This symbol always comes at the beginning of a **PLC programming example**, e.g.:

 U            -BEDIN1                            ;Step-on condition 1



This symbol indicates that something must be entered. The text to be entered is then given after the symbol. If the entry has to be adapted to your requirements, the text after the symbol will be in *italics*. The **entry** must be confirmed by pressing



Example:



TYPE *file name* | MORE



This screen symbol is always followed by the **reaction of the screen** to your entries. When your text is adapted to your requirements it is shown in *italics*.

Example:



Save file *file name*.TXT?                    Yes / No



## Safety instructions



**Test each new program before operating the system!**



**All PLC programs produced to date can be edited with the new version 3.0. PLC programs which have been edited with the new version 3.0 can no longer be read or edited with older versions!**



**In this description the floppy disk drive is always drive A and the hard disk is always drive C.**



**In section A.3 Alterations, changes to the E5 edition are listed.**



**The chapters of the E5 edition**  
**8 IBM-AT03 compatible PC,**  
**9 Software installation and**  
**10 Professional integrator**  
**are now contained in the new Technical Documentation:**  
**PLC/DESI Utilities**  
**Professional Integrator**  
**Installation instructions**  
**P.-Nr. 4308**



**The PLC utilities can only be run on computers with the Intel processors 80 286, 80 386 and 80 486.**



**The following versions of the MS DOS operating system have been approved for use with PLC utilities:**  
**MS DOS Version 3.21**  
**MS DOS Version 3.30**  
**MS DOS Version 5.0**





**Contents**

**1 User guide and main menu**

**2 Programming**

**3 Editor**

**4 Monitor**

**5 Loader**

**6 Lister**

**7 Key functions**

**A Appendix**





**Contents**

	Page
<b>1</b>	<b>User guide and main menu ..... 1–1</b>
1.1	Overview ..... 1–1
1.2	Starting PLC utilities ..... 1–3
1.3	Exiting PLC utilities and parking the hard disk ..... 1–6
1.4	Structure of PLC utilities ..... 1–8
1.5	Description of the main menu ..... 1–11
1.6	Using the Command utility ..... 1–17
1.7	Entering defaults ..... 1–19
1.8	Using the pull–up menus ..... 1–27
1.9	Using the TAB menu ..... 1–30
1.10	Description of the help function ..... 1–32
1.11	Editing functions ..... 1–35

## Illustrations

Fig.		Page
1-1	Software dongle .....	1-3
1-2	Professional integrator .....	1-4
1-3	Main menu .....	1-5
1-4	Function key bar .....	1-6
1-5	Professional integrator .....	1-6
1-6	General screen layout .....	1-8
1-7	Main menu .....	1-9
1-8	Structure of PLC utilities .....	1-10
1-9	Main menu .....	1-11
1-10	Function blocks of the main menu .....	1-12
1-11	Help function .....	1-15
1-12	Function key bar .....	1-16
1-13	Input line .....	1-17
1-14	Defaults function block .....	1-19
1-15	Defaults .....	1-21
1-16	Project directory .....	1-22
1-17	Subdirectories for ZS0 and ZS1 .....	1-23
1-18	Editor menu .....	1-27
1-19	Search menu .....	1-27
1-20	Editor menu .....	1-28
1-21	Search menu .....	1-28
1-22	Search string .....	1-28
1-23	Search selection menu .....	1-29
1-24	File name .....	1-30
1-25	Selection menu .....	1-30
1-26	Help and pull-up menus .....	1-34



# 1 User guide and main menu

## 1.1 Overview

This software handbook describes PLC utilities. This description applies to the **Version 3.0x**.

PLC utilities used to program the following Bosch controllers:

- **PIC**
- **PC400**
- **PC600**
- **CL100**
- **CL300**
- **CL500**

In chapters **1 User guide and main menu**, **3 Editor**, **4 Monitor**, **5 Loader** and **6 Lister** the individual functions of PLC utilities are described.

Chapter **2 Programming** depicts the general programming sequence on the basis of an example program. This chapter offers a rapid introduction to programming the PLC.

**The description of software installation and the professional integrator can be found in the following technical documentation:**

**PLC/DESI Utilities  
Professional Integrator  
Installation instructions  
P.–Nr. 4308**



This chapter provides a user guide and describes the main menu PLC utilities. It contains the following information:

- **Starting** PLC utilities, section 1.2.
- **Exiting** PLC utilities, section 1.3.
- **Parking the hard disk**, section 1.3.
- **Structure** of PLC utilities, section 1.4.
- **Description** of the **main menu**, section 1.5.
- Using the utilities **Command**, section 1.6.
- Entering **defaults**, section 1.7.
- Operating the **pull-up menu**, section 1.8.
- Operating the **TAB selection menu**, section 1.9.
- **Description** of the **help function**, section 1.10.
- **Editing functions**, section 1.11.

PLC utilities user guide supports the user with:

- Uniform operation procedure for all utilities.
- Division of screen displays into function blocks.
- Comprehensive help function.
- Menu control via function keys and pull-up menus.
- Logging of activities when assigning, linking, loading, etc.
- Project-specific storage of defaults.
- Segmenting of programs into networks



## 1.2 Starting PLC utilities

First of all, install the PLC utilities according to the instructions in the Technical Documentation:

### **PLC/DESI Utilities**

### **Professional Integrator**

### **Installation instructions**

### **P.–Nr. 4308**

Remember to copy the **AUTOEXEC.BAT** and **CONFIG.SYS** files into the **main directory**, or, if necessary, to change your existing files.

- ★ Plug the **software dongle** into the **parallel** interface.

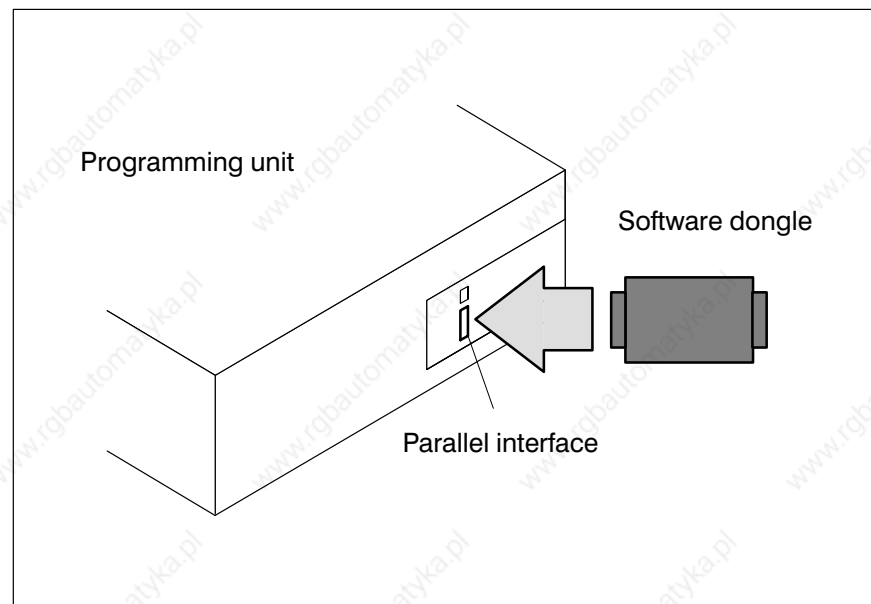


Fig. 1–1 Software dongle

- ★ Please **remove** any **floppy disk** which may be present in the disk drive.
- ★ **Switch on** programming unit.



C:\>



PROFI

The **Professional integrator** program is loaded and the menu is displayed, see Fig. 1–2.

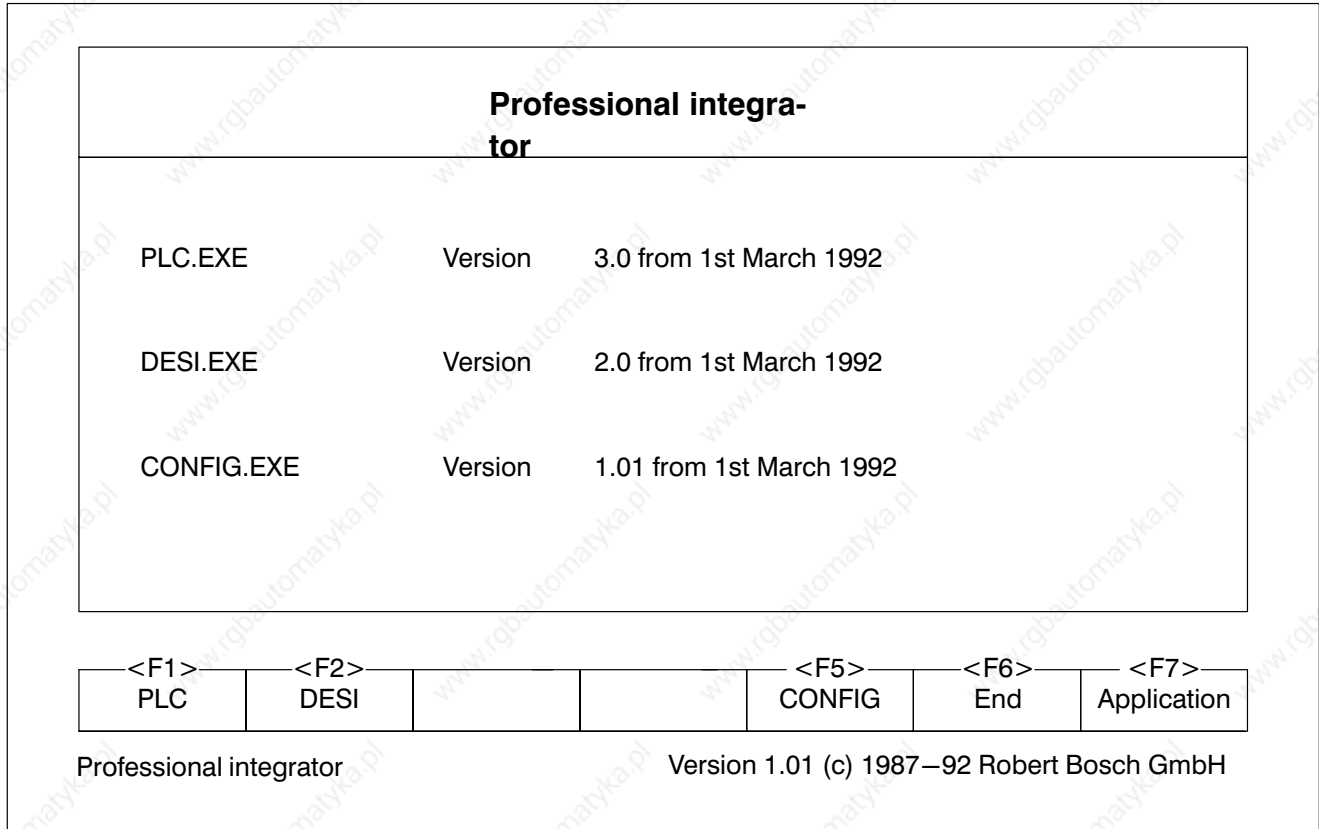


Fig. 1-2 Professional integrator

★

F1

## PLC

PLC utilities are loaded and the **main menu** is displayed, see Fig. 1-3.



F10

## Help.



C: \PG \

Disk	Name	Type	Length
Select>			

Information

> Disk Info <

0 Files

13797376 Bytes free

7512064 Bytes used

21309440 Bytes in total

---

>> Memory Info <<

123504 Bytes free

531856 Bytes used

655360 Bytes in total

---

>> Extended Memory Info <<

HIMEM driver active 2.77

196608 Bytes (XMS) free

in 32 memory blocks

Select info

PG >>>

---

Time

01.03.1992

13:13:13

<F10>

Help

<F1>	<F2>	<F3>	<F4>	<F5>	<F6>	
<b>Command</b>	Editor	Monitor	Loader	Lister	End	

PLC utilities Version 3.0 (c) 1987–92 Robert Bosch GmbH

Fig. 1–3 Main menu

Use

F1

to

F5

to select a utility.

## 1.3 Exiting PLC utilities and parking the hard disk

The **function key bar** is displayed in the bottom part of the screen in the main PLC utilities menu, Fig. 1–4.

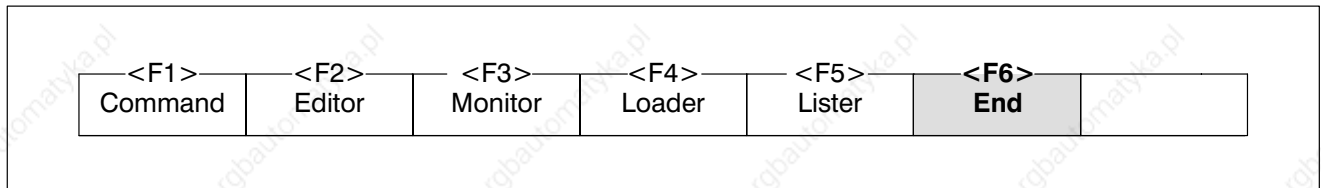
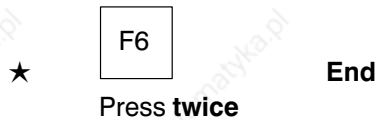


Fig. 1–4 Function key bar



The **Professional integrator** program is called up.

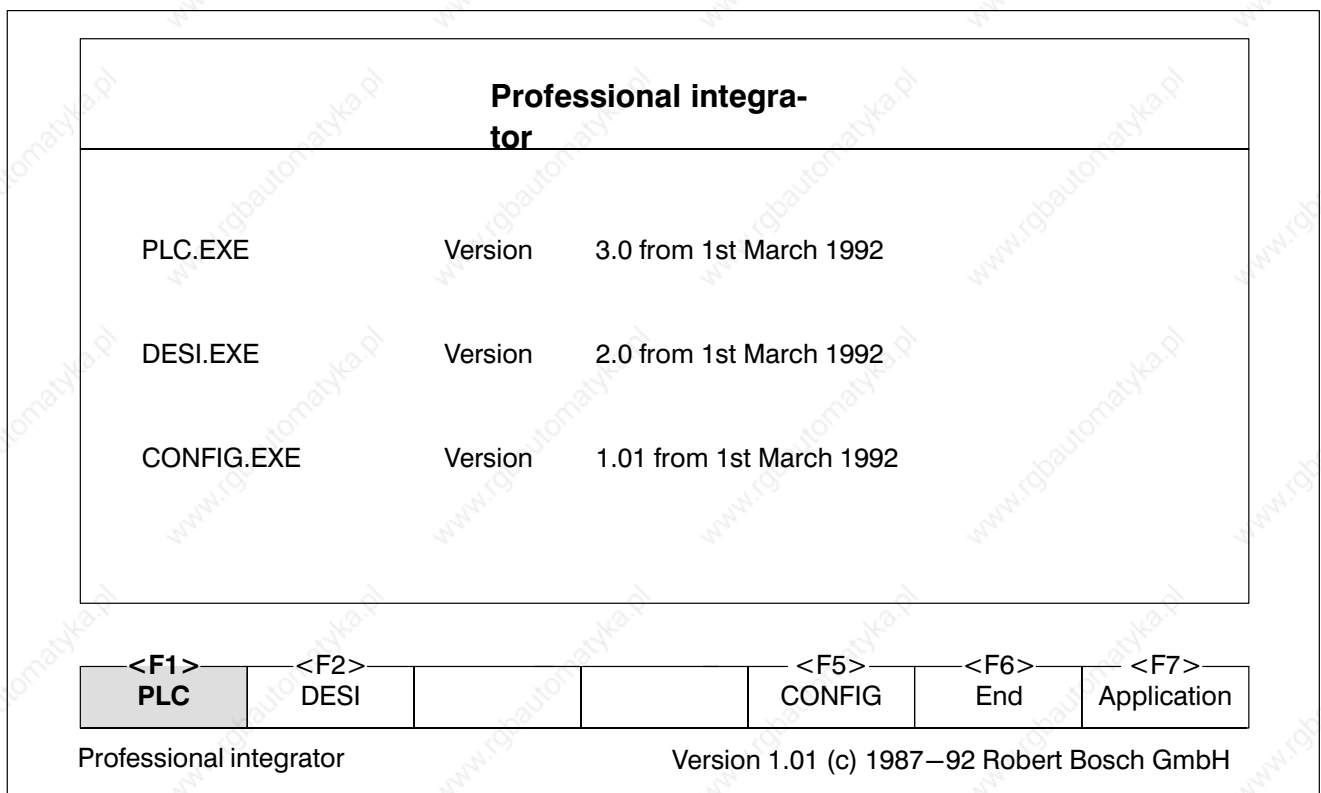


Fig. 1–5 Professional integrator





**Returning to MS DOS**

To continue your work in the MS DOS utilities, press:



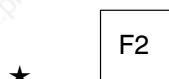
**Return to MS DOS.**



You are now in the MS DOS utilities.

**Parking the hard disk**

If you wish to transport the programming unit, you must first secure the hard disk:



**Park hard disk**

★ Switch off the programming unit at the mains.



**You must also protect the floppy disk drive from damage by inserting a shipping card or a floppy disk!**

## 1.4 Structure of PLC utilities

The user guide system is based on the following general **screen layout**.

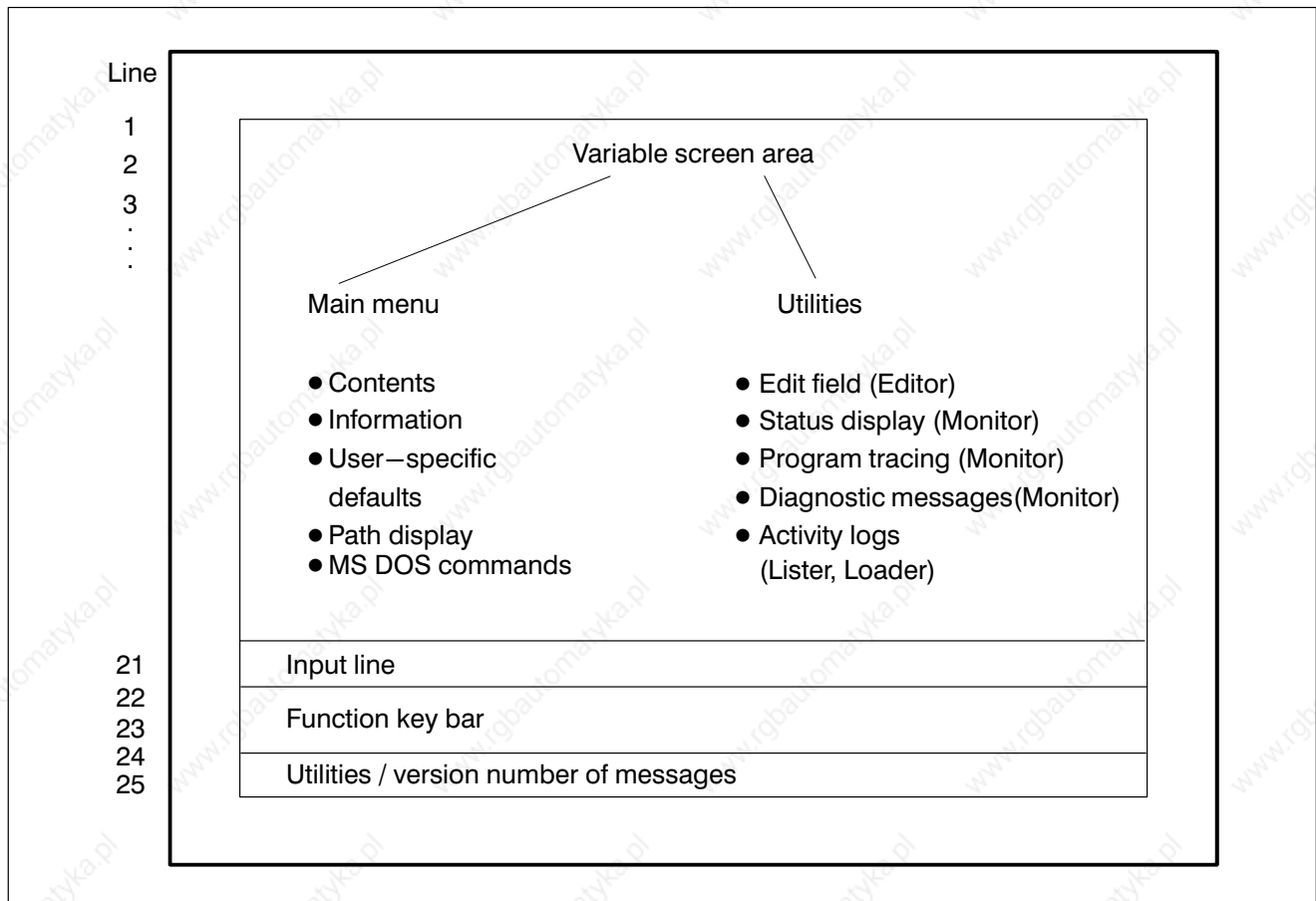


Fig. 1-6 General screen layout



The **main menu** is displayed when PLC utilities are called.

The screenshot shows the main menu interface with the following components:

- Path:** C: \PG \
- File List Table:**

Disk	Name	Type	Length
Select>			
- Information Panel:**
  - > Disk Info <
    - 0 Files
    - 13797376 Bytes free
    - 7512064 Bytes used
    - 21309440 Bytes in total
  - >> Memory Info <<
    - 123504 Bytes free
    - 531856 Bytes used
    - 655360 Bytes in total
  - >> Extended Memory Info <<
    - HIMEM driver active 2.77
    - 196608 Bytes (XMS) free
    - in 32 memory blocks
- Right Panel:**
  - Select Info: PG >>>
  - Time: 01.03.1992 13:13:13
  - Help: <F10> Help
- Function Key Bar:**

<F1> Command	<F2> Editor	<F3> Monitor	<F4> Loader	<F5> Lister	<F6> End	
--------------	-------------	--------------	-------------	-------------	----------	--
- Footer:** PLC utilities Version 3.0 (c) 1987–92 Robert Bosch GmbH

Fig. 1–7 Main menu

The utilities are listed in the **function key bar**.

- **Command**
- **PLC Editor program**
- **PLC Monitor program**
- **PLC Loader program**
- **PLC Lister program**

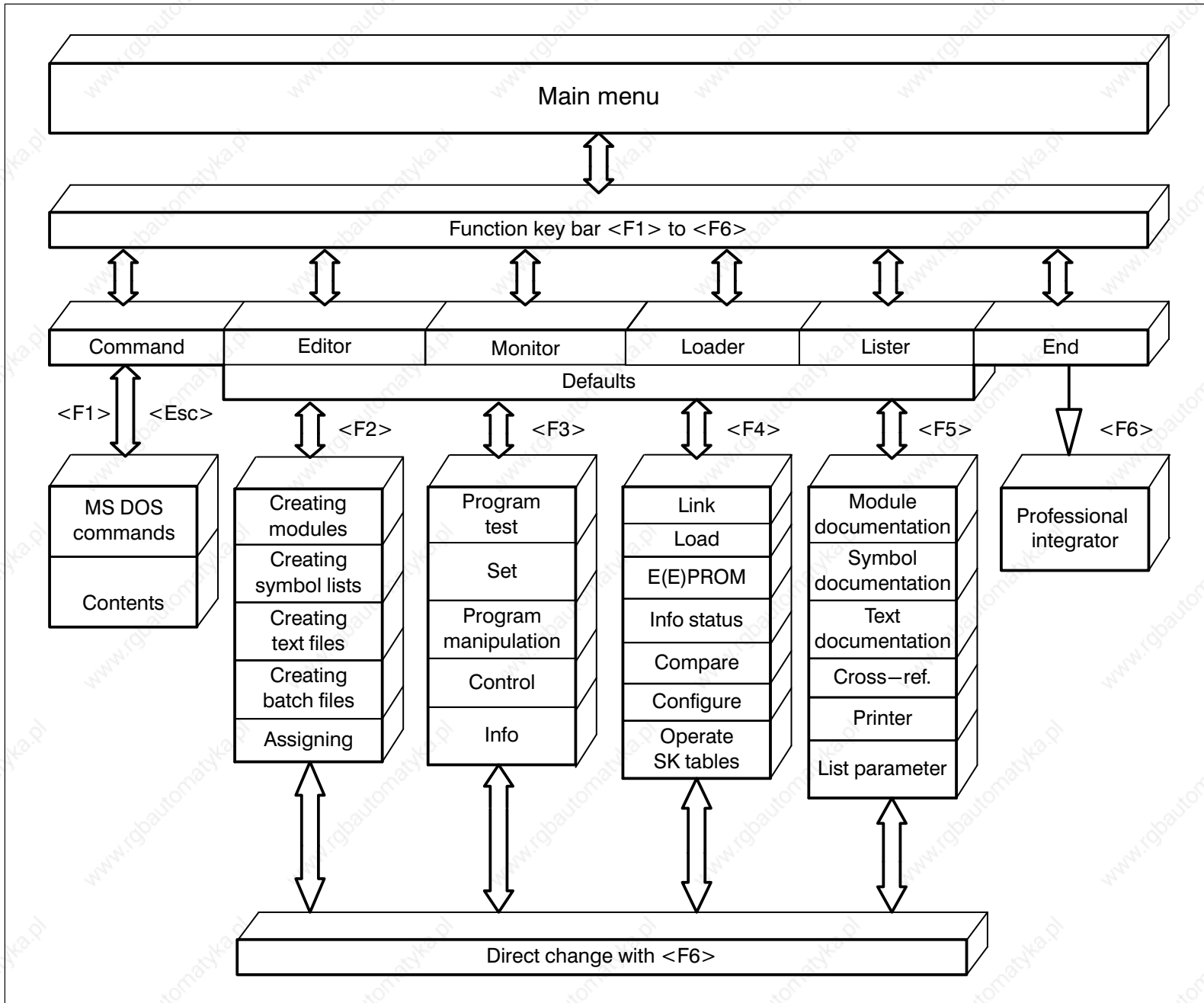
The PLC utilities program comprises various utility programs that are similar in structure and closely interrelated, see Fig. 1–8.



F10

**Help.**

Fig. 1-8 Structure of PLC utilities





## 1.5 Description of the main menu

The **main menu** is displayed when PLC utilities are called. This section describes the structure and operation of the main menu.

In the main menu you set the **defaults** for the **screen display** and select the utilities program.

C:\PG \

Disk	Name	Type	Length
Select>			

Information

> Disk Info <

0 Files

13797376 Bytes free

7512064 Bytes used

21309440 Bytes in total

---

>> Memory Info <<

123504 Bytes free

531856 Bytes used

655360 Bytes in total

---

>> Extended Memory Info <<

HIMEM driver active 2.77

196608 Bytes (XMS) free

in 32 memory blocks

Select info

PG >>>

---

Time

01.03.1992

13:13:13

<F10>

Help

<F1>  
**Command**

<F2>  
Editor

<F3>  
Monitor

<F4>  
Loader

<F5>  
Lister

<F6>  
End

PLC utilities Version 3.0 (c) 1987-92 Robert Bosch GmbH

Fig. 1-9 Main menu

The main menu is split into **function blocks**, see Fig. 1-10.

The function blocks **Path**, **Contents**, **Function key bar** and **Version/messages** are available in all utility programs.

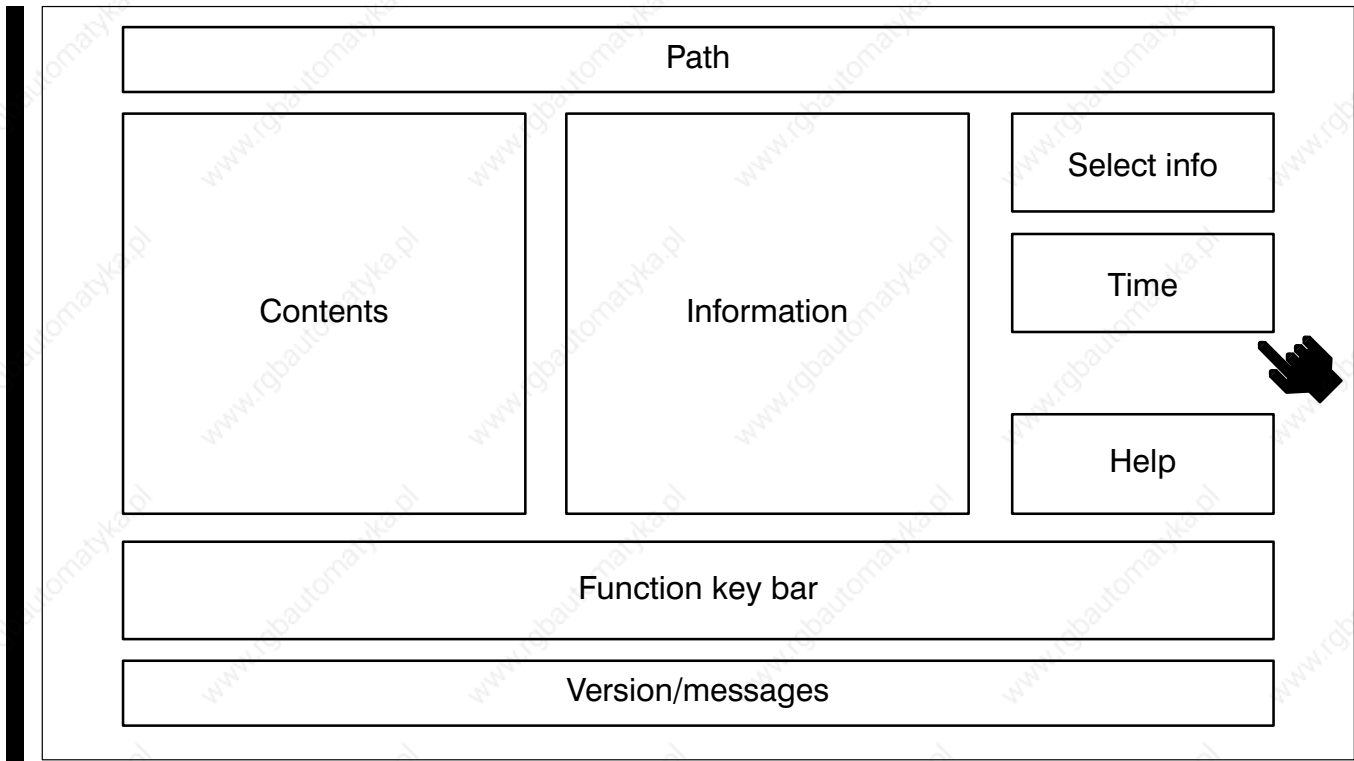



Fig. 1–10 Function blocks of the main menu

## Path

The first line displayed on the screen shows the **path** to the **subdirectory** you are currently in or whose contents are being displayed on the screen, e.g.:

 C:\PG\

This means you are in the **PG subdirectory** which branches off from the **main or root directory**.



**You will find further information on file management in MS DOS documentation.**

## Contents

The **contents** function block displays the contents of the drive selected. The **disk name** (<Volume>), if available, is shown in the top line. Depending on the directory structure the next lines show a **parent directory** (<Parent> directory) and any **subdirectories** (<Subdir> –ectories) at the next level. This is followed by a list of files.



In the **Editor**, **Monitor**, **Loader** and **Lister** utilities the display in the function block **Contents** is always **project-related**. You **cannot** leave the **project directory set** (<PRODIR>) in the defaults and go into the **parent directory** (<PARENT>).

In the utility **Command** it is possible to select **any desired directory**, see section 1.6.

## Scroll contents

**To start**

Home

**Page up**

PgUp

**Selection bar up a line**

↑

**Selection bar down a line**

↓

**Page down**

PgDn

**To end**

End

The following is only possible in the **Command** utility, see also section 1.6:

+

**Select**

The desired **directory** is selected by positioning the selection bar at the line containing the **directory name** (<PARENT>, <SUBDIR> or <PRODIR>).

## Copy

**A file name is copied**

- **to the input line** (see section 1.6) or
- **to the defaults** (see section 1.7),

by positioning the selection bar at the line containing a **file name**, and using the

+

key.

## Information

The function block **Information** displays information about the programming unit (**PG**) or about the files displayed in the contents (**file**). The selection is made in the function block **Select info**.

The following **PG information** is displayed.

- Drive info: (Disk Info)
  - Number of files in current directory
  - Free drive capacity
  - Used drive capacity
  - Total drive capacity
- Main memory info: (Memory Info)
  - Free RAM capacity
  - Used RAM capacity
  - Total RAM capacity
- Extended memory info:
  - Display of active HIMEM driver with version
  - Free capacity in extended memory

**HIMEM.SYS drivers from Version 2.77 may be used. See also MS DOS documentation and technical documentation:**

**PLC/DESI Utilities  
Professional Integrator  
Installation instructions  
P.–Nr. 4308**

If **file information** is selected, the date and time of the last save is also displayed for each file.

## Select info

In the function block **Select info**, select the **PG information** or **File information** display, see function block **Information**.

**Open select**



TAB key

**Scroll the defaults in the selected field**



**Close select**







Help F10

★ F10 Help

C: \PG \		PG help	01/09/001
Disk	Name	<p>PLC Help Function</p> <p>Welcome to the Help function for PLC utilities.</p> <p>With the key</p> <p style="margin-left: 100px;">&lt;F10&gt;</p> <p>you can request detailed help information for the current situation at any time.</p>	
<b>Select &gt;</b>		PgDn	Exit with Esc
		<span style="border: 1px solid black; padding: 2px;">&lt;F10&gt; Help</span>	
<F1> Command	<F2> Editor	<F3> Monitor	<F4> Loader
		<F5> Lister	<F6> End
PLC utilities		Version 3.0 (c) 1987–92 Robert Bosch GmbH	

Fig. 1–11 Help function

The **help function** can be called up at any time and in any utility. Only the main menu displays the help function as an explicit reminder.

You will receive assistance tailored to the current operating situation. The help is displayed in a window and the current screen is retained. The assistance always relates to the **current cursor position**. For further information refer to section 1.10 **Description of the help function**.

★ F10 or Esc Exit help

## Function key bar

The **function key bar** lists the utilities. Press the appropriate function key to call up the desired utility.

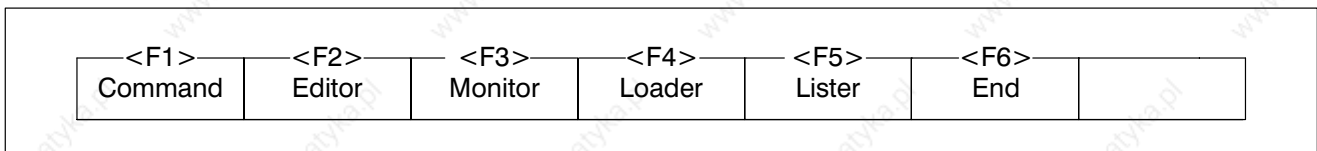


Fig. 1–12 Function key bar

- **Command**

F1

- **PLC Editor program**

F2

- **PLC Monitor program**

F3

- **PLC Loader program**

F4

- **PLC Lister program**

F5

Use

F6

**End**

to **exit** the PLC utilities, see section **1.3 Exiting PLC utilities and parking the hard disk**.

## Version/messages

The bottom line first shows the software version number.

During program execution this line displays:

- Utility name
- System queries about file management
- System messages
- Error messages



## 1.6 Using the Command utility

The **Command** utility is used to

- **enter MS DOS commands** in the **input line** and
- **to display the contents independently of the project.**

★  **Command**

This activates the function block **input line**, see Fig. 1–13.

### Input line

At the beginning of the input line is the **drive name** of the selected drive together with a **prompt symbol**.

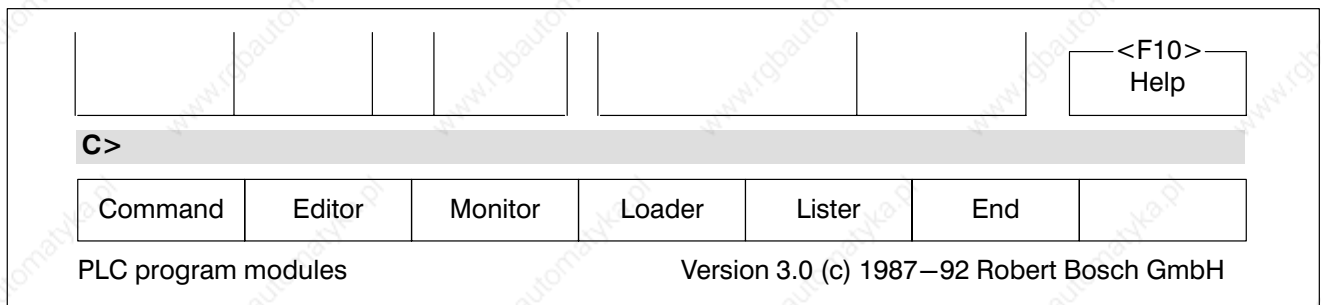


Fig. 1–13 Input line

You may enter any MS DOS commands in this line, e.g.:

 **TREE**

★  **Return**

★  **Exit command**

## Contents

**Any** directory you wish can be displayed in the function block **Contents** in the Command utility.

**To start**

Home

**Page up**

PgUp

**Selection bar up a line**

↑

**Selection bar down a line**

↓

**Page down**

PgDn

**To end**

End

**Copy file name** to the **input line** for any MS DOS command, when the selection bar is on a line with a **file name** using the key

+

**Select** the corresponding **directory** when the selection bar is on a line with a **directory name** (<PARENT>, <SUBDIR> or <PRODIR>) using the key

+

**Copy** a file name into the **input line** when the select bar is on a line with a **file name**, using the key

+

F10

**Help.**





## 1.7 Entering defaults

This section describes the common defaults for the **Editor**, **Monitor**, **Loader** and **Lister** utilities.

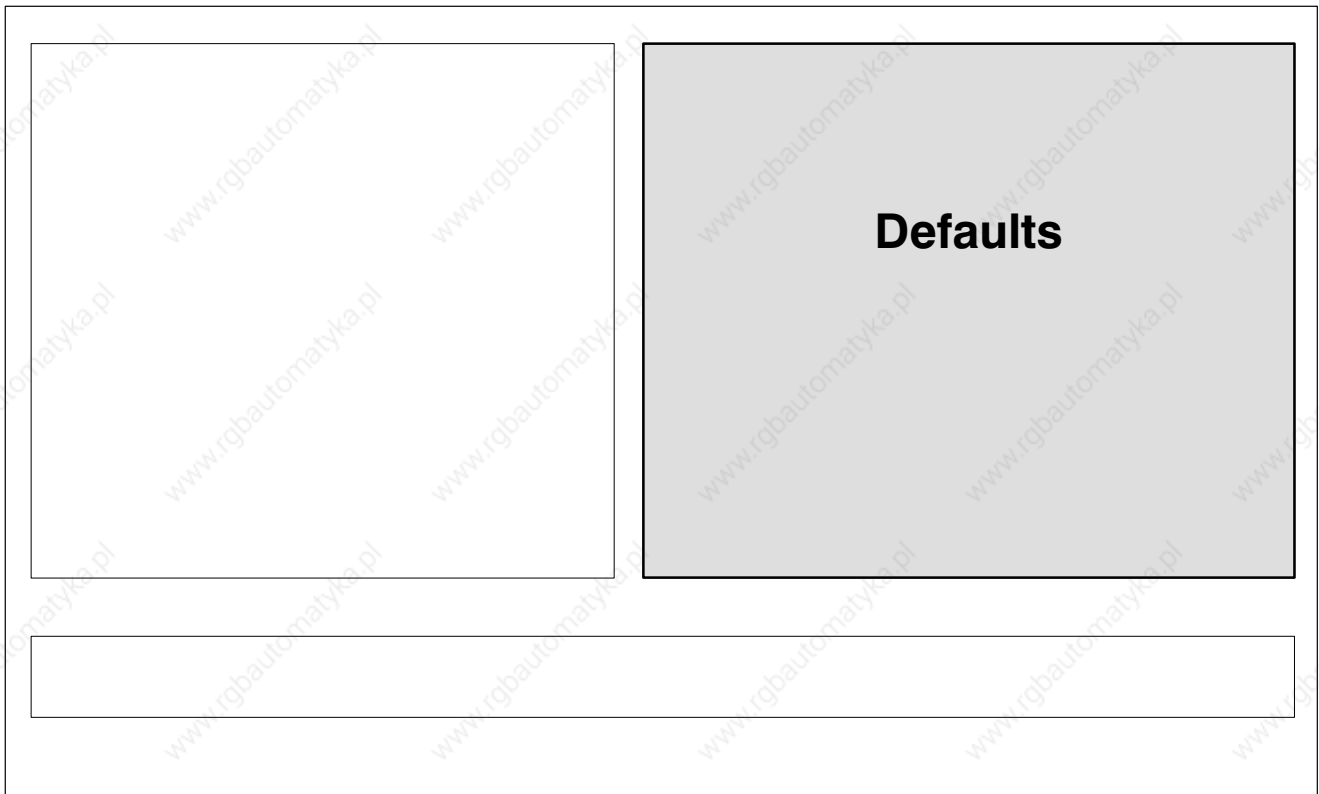


Fig. 1–14 Defaults function block

From the main menu, use the function keys



to call one of the utility programs. The screen will display the function block **defaults**. In this function block, the project name and file name have to be entered before the utility program can be started.

To create a new **PLC project** you will first have to enter the necessary information in the defaults. The utilities automatically take over the defaults of the project.

When you exit PLC utilities by pressing the function key

**F6**

**End,**

the project defaults will be saved and are available the next time the project is called up. Enter the file name under which **your** defaults are to be saved in the **project status** line.

## Defaults

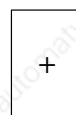
**Open defaults**



**Close defaults**



**Take over directory name or file name**



**Scroll through lines**



**Scroll through defaults of the selected field**



**F10**

**Help.**





C: \ PG \

Disk	Name	Type	Length
Select >	Example	300	<PRODIR>

<b>Project name</b>		
Controller type	CL300	>>>
Control unit	ZS0	>>>
Project status	PROJECT.PLC	
Module file name		
Symbol file name		
Program file name		
Text file name		

Command	Editor	Monitor	Loader	Lister	End	
---------	--------	---------	--------	--------	-----	--

PLC utilities Version 3.0 (c) 1987–92 Robert Bosch GmbH

Fig. 1–15 Defaults

**Open defaults****Project name**

A project unites all files which belong to one controller:

- **Project status**
- **Module files**
  - Organisation modules
  - Utilities
  - Library modules
- **Symbol file**
- **Program file**
- **Operand field file**
- **SK table**
- **Print files**
- **Text files**

A separate **subdirectory** is created for each project.

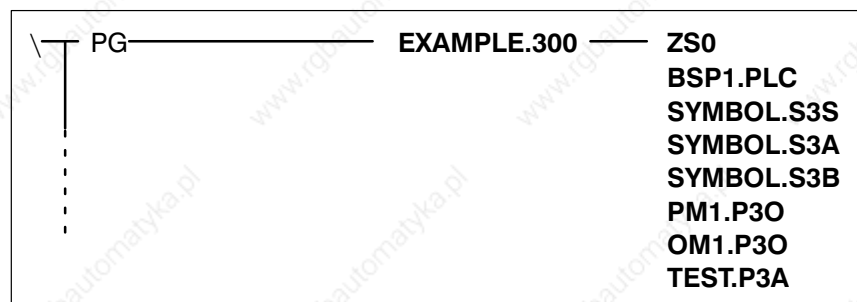


Fig. 1–16 Project directory

The **project name** may have a maximum of 8 characters.

In the **Contents** function block, all existing projects are displayed.

The project name can

- be entered manually



EXAMPLE

or

- can be automatically taken over from the contents using

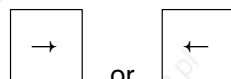


The project name is automatically supplemented with the **controller type**,  
e.g.:

EXAMPLE.300, TEST.100.

## Controller type

Selection of controller type:



or

PIC, PC400, PC600, CL100, CL300 or CL500



**Control unit ZS**

This line can **only** be changed on the **CL500**. You have to select one of the **4** control units (ZS0, ZS1, ZS2 and ZS3). For every control unit the corresponding module, symbol and program files are stored in a separate **subdirectory**.

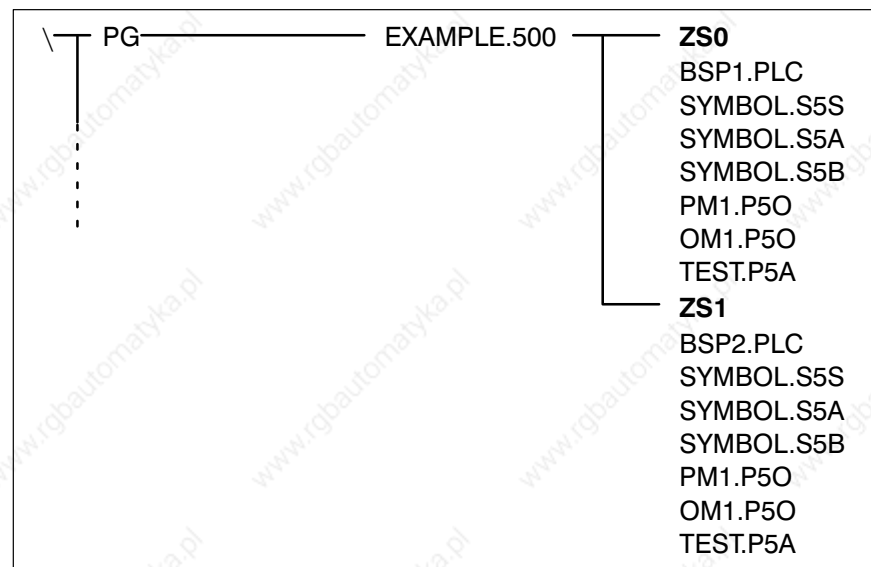


Fig. 1–17 Subdirectories for ZS0 and ZS1

**Project status**

Here you enter the **file names** to save **your** defaults. If several users are working in a network on **one project**, every user can save his special defaults with his file name.

In the **Contents** function block, all existing project status file names are displayed.

The file name can be

- entered manually



BSP1

or

- can be automatically taken over from the contents using



The file name automatically receives the **file type .PLC**.

## Module file name

In this line you enter the name of the **module file** you wish to process.

In the **Contents** function block all existing module files are displayed.

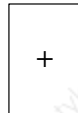
The file name can be

- entered manually



or

- automatically taken over from the contents with



The file name automatically receives a **file type**:

- **.PxT** for the file which is **not** assigned.
- **.PxO** for the correctly **assigned** file.

x = **controller type**, e.g.: 1 = CL100, 5 = CL500.

## Symbol file name

Enter the file name for the **symbol file** of the project. The symbol file contains:

- The list of the **module files**.
- The **data modules**.
- The **symbolic operand descriptions** (symbolic addresses).

In the **Contents** function block all existing symbol files are displayed.



The file name can be

- entered manually



*SYMBOL*

or

- automatically taken over from the contents with



The file name automatically receives a **file type**:

- **.SxS** for the contents of the **symbol file**
- **.SxA** for the assignment of the **absolute addresses**
- **.SxB** for the assignment of the **symbolic addresses**

x = **controller type**, e.g.: 1 = CL100, 5 = CL500.

### Program file name

Here you enter the file name for the operable PLC program. This program file name is used when linking and loading.

After linking, the program file contains all modules and data modules which form the control program according to the symbol file.

In the **Contents** function block all existing program files are displayed.

The file name can be

- entered manually



*ANLAGE1*

or

- automatically taken over from the contents with



The file name automatically receives the **file type .PxA**.

x = **controller type**, e.g.: 1 = CL100, 5 = CL500.

## Text file name

PLC utilities also contains a text editor for any ASCII files you wish, see section **3.10 Text file editor**.

The text editor is used for:

- Changing the **AUTOEXEC.BAT** or **CONFIG.SYS** file.
- Viewing and processing print files on screen which were created using the **PLC Lister program**.
- Processing a **header file** for the program print-out.
- Creating a **batch file**.

Enter the text file name in this line of defaults. The **file name** can be up to 8 characters long. The **file type** may have a maximum of 3 characters.

The file name and type can be

- entered manually



or

- automatically taken over from the contents with



**Close defaults**

## Storing defaults

When you exit PLC utilities with



**End**

the defaults are automatically saved in the file **PROJECT.PLC**. You specified this file name in the defaults on the **project status** line.



**Help.**



## 1.8 Using the pull-up menus

The user guide system is supported by **pull-up menus**. The pull-up menu display the **function keys** relating to the **Commands**. The screen only displays function keys whose corresponding command is permitted in the given situation.

The following examples illustrate the use of the pull-up menus. Try to reproduce these examples.

### 1st Example

Go to the end of the file in **Editor**.

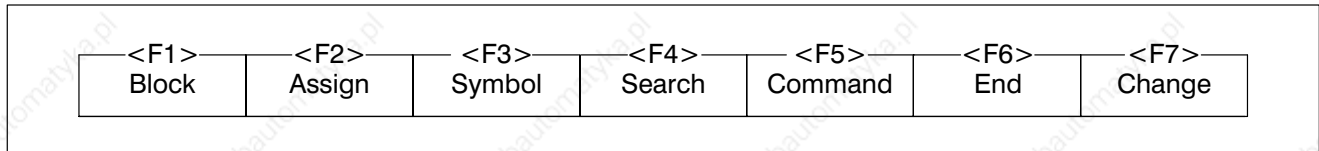


Fig. 1-18 Editor menu

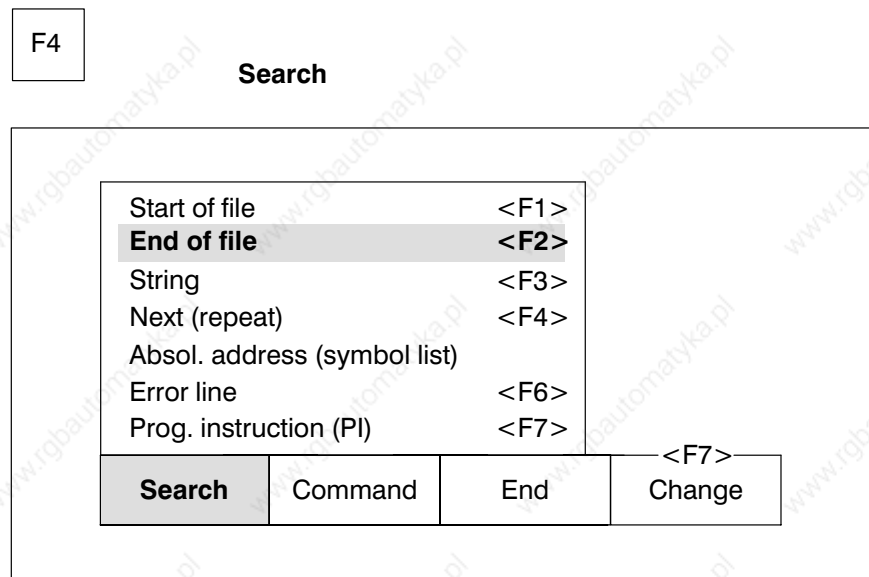
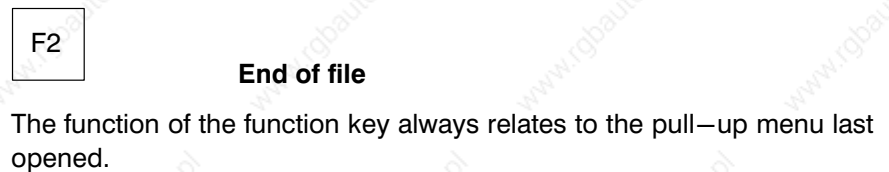


Fig. 1-19 Search menu



## 2nd Example

Search for character string **0.7**.

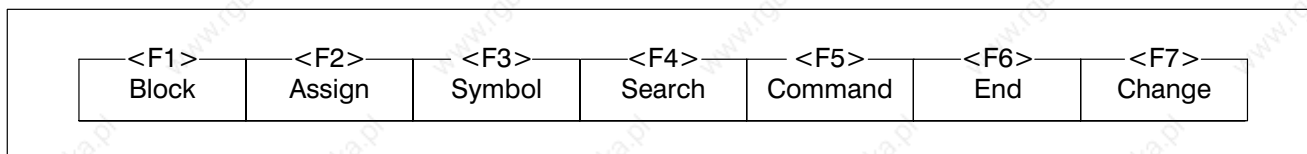


Fig. 1–20 Editor menu

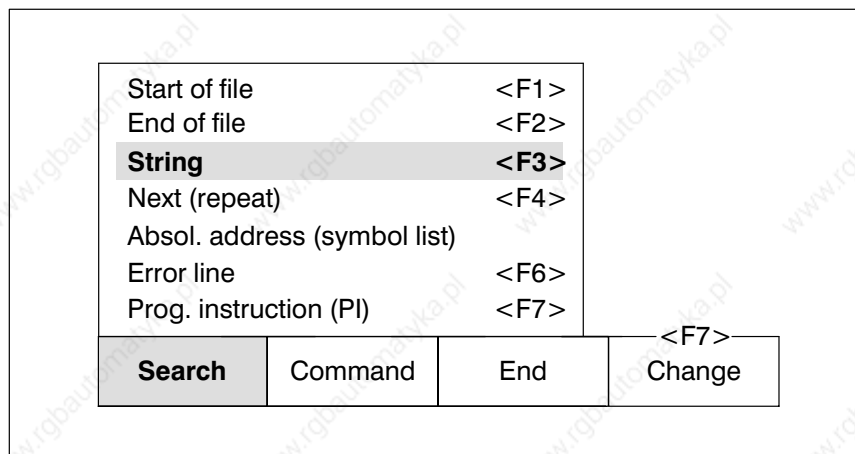


Fig. 1–21 Search menu

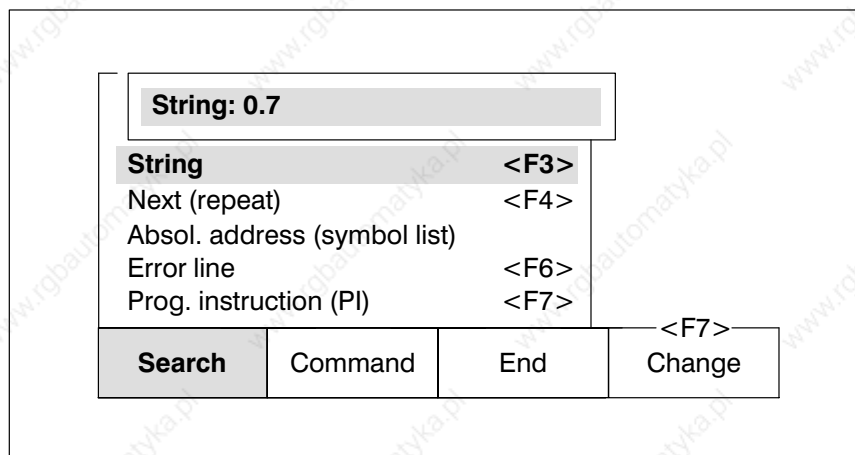


Fig. 1–22 Search string



0.7

Cancel an incorrect character with



or



(backspace).

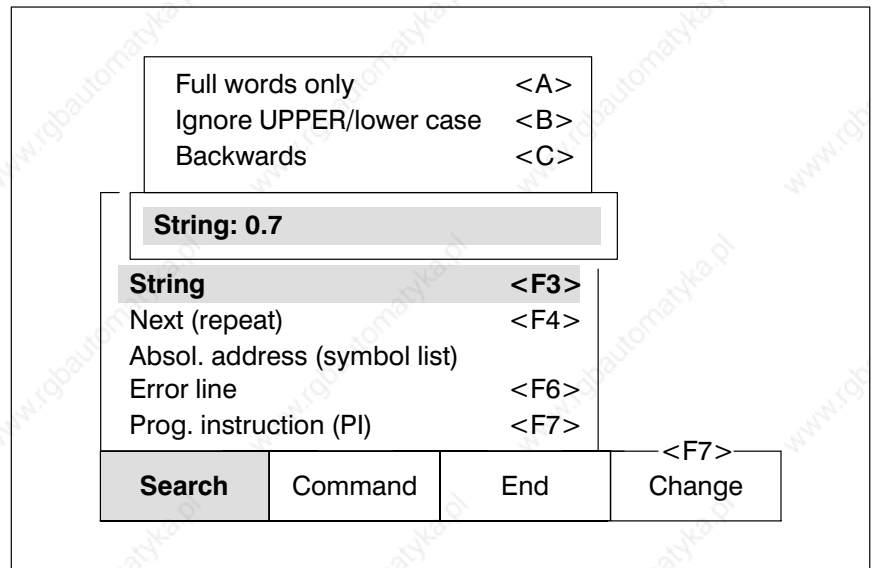


Fig. 1–23 Search selection menu

An additional menu is opened on screen. Use



or



to make your selection. The keys have a **toggle function**. Selected settings are highlighted by a bar. If a setting is **not** selected, then the **contrary statement** is true.



A



**Help.**



## 1.9 Using the TAB menu

For some entries you are required to select a file. So that you do not have to type in the file name every time, the file names are listed together in a selection menu.

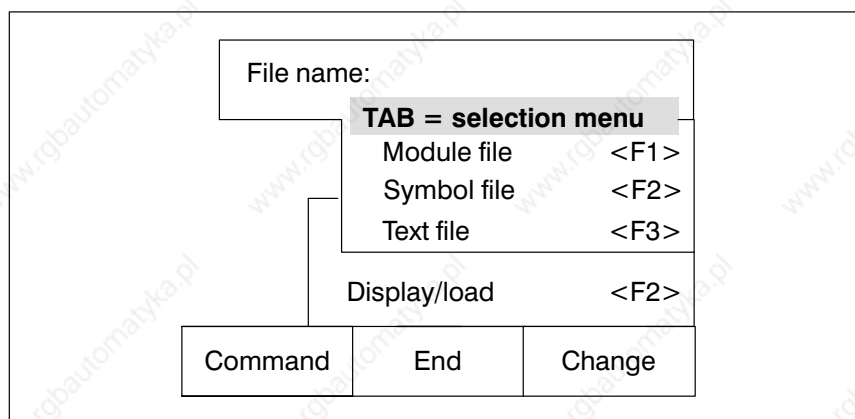


Fig. 1–24 file name



★

A menu with the available file names is opened.

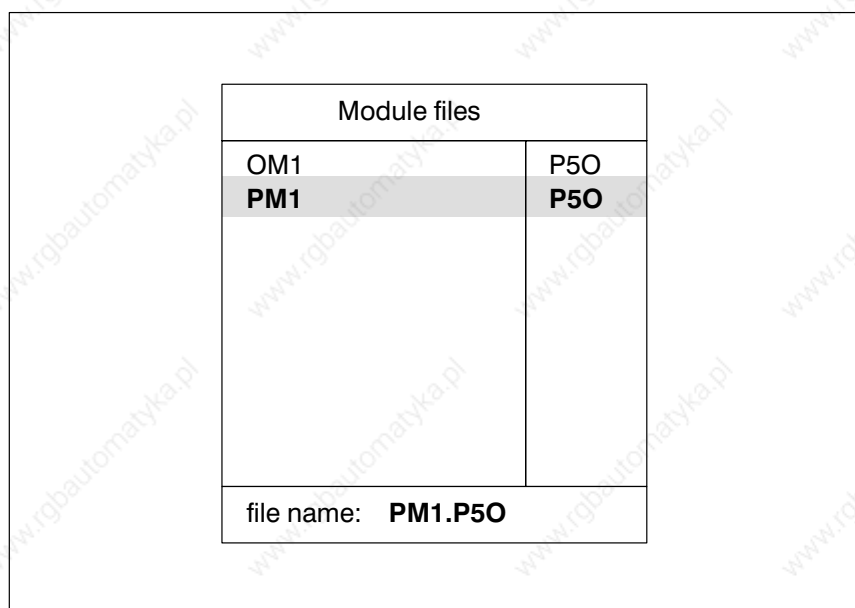


Fig. 1–25 Selection menu





★ Select file name:

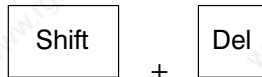


The file name is taken over in the bottom line.

★ Take over current file name:



Enter new file name:



★ Enter new name.



**Help.**



## 1.10 Description of the help function

The user guide to PLC utilities provides assistance in the form of information selected to suit your current situation. The current data is still largely visible even while help is being displayed.

<b>Call up help function</b>	<input type="text" value="F10"/>
<b>Exit help function</b>	<input type="text" value="Esc"/> or <input type="text" value="F10"/>
<b>Page up</b>	<input type="text" value="PgUp"/>
<b>Page down</b>	<input type="text" value="PgDn"/>

All other functions are **disabled** while the help function is active.

An example should illustrate the connection between the **position of the selection bar** in the pull-up menu and the information.

### Example

You are in the **main menu**.

★	<input type="text" value="F2"/>	<b>Editor, Defaults</b>
★	<input type="text" value="F2"/>	<b>Editor, Call</b>
★	<input type="text" value="Esc"/>	<b>Toggle between Edit and Command level</b>
★	Use	
	<input type="text" value="←"/> or <input type="text" value="→"/>	
		to move the selection bar to <b>Change</b> in the function key bar.
★	<input type="text" value="F10"/>	<b>Help</b>

You are given help on the keyword **Change**.




★  **Exit help function**

★  **Change**

★ Use



to move select bar to **Save** in the pull-up menu.

★  **Help**

You are given help on the keyword **Save**.

★  **Exit help function**

★ Use



to move the select bar to **Display / Load** in the pull-up menu.

★  **Help**

You are given help on the keyword **Display / Load**.

★  **Exit help function**

The following table shows the connection between the help given and the position of the selection bar in the pull-up menu.

Position of selection bar	Help on keyword																
<table border="1" style="width: 100%; text-align: center;"> <tr> <td>&lt;F4&gt; Search</td> <td>&lt;F5&gt; Command</td> <td>&lt;F6&gt; End</td> <td>&lt;F7&gt; <b>Change</b></td> </tr> </table>	<F4> Search	<F5> Command	<F6> End	<F7> <b>Change</b>	Change												
<F4> Search	<F5> Command	<F6> End	<F7> <b>Change</b>														
<table border="1" style="width: 100%; text-align: center;"> <tr> <td>&lt;F4&gt; Search</td> <td>&lt;F5&gt; Command</td> <td>&lt;F6&gt; End</td> <td>&lt;F7&gt; <b>Change</b></td> </tr> <tr> <td colspan="3"></td> <td> <table border="1" style="width: 100%; text-align: center;"> <tr> <td><b>Save</b></td> <td>&lt;F1&gt;</td> </tr> <tr> <td>Display / Load</td> <td>&lt;F2&gt;</td> </tr> </table> </td> </tr> </table>	<F4> Search	<F5> Command	<F6> End	<F7> <b>Change</b>				<table border="1" style="width: 100%; text-align: center;"> <tr> <td><b>Save</b></td> <td>&lt;F1&gt;</td> </tr> <tr> <td>Display / Load</td> <td>&lt;F2&gt;</td> </tr> </table>	<b>Save</b>	<F1>	Display / Load	<F2>	Save				
<F4> Search	<F5> Command	<F6> End	<F7> <b>Change</b>														
			<table border="1" style="width: 100%; text-align: center;"> <tr> <td><b>Save</b></td> <td>&lt;F1&gt;</td> </tr> <tr> <td>Display / Load</td> <td>&lt;F2&gt;</td> </tr> </table>	<b>Save</b>	<F1>	Display / Load	<F2>										
<b>Save</b>	<F1>																
Display / Load	<F2>																
<table border="1" style="width: 100%; text-align: center;"> <tr> <td>&lt;F4&gt; Search</td> <td>&lt;F5&gt; Command</td> <td>&lt;F6&gt; End</td> <td>&lt;F7&gt; <b>Change</b></td> </tr> <tr> <td colspan="3"></td> <td> <table border="1" style="width: 100%; text-align: center;"> <tr> <td>Save</td> <td>&lt;F1&gt;</td> </tr> <tr> <td><b>Display / Load</b></td> <td>&lt;F2&gt;</td> </tr> </table> </td> </tr> </table>	<F4> Search	<F5> Command	<F6> End	<F7> <b>Change</b>				<table border="1" style="width: 100%; text-align: center;"> <tr> <td>Save</td> <td>&lt;F1&gt;</td> </tr> <tr> <td><b>Display / Load</b></td> <td>&lt;F2&gt;</td> </tr> </table>	Save	<F1>	<b>Display / Load</b>	<F2>	Display / Load				
<F4> Search	<F5> Command	<F6> End	<F7> <b>Change</b>														
			<table border="1" style="width: 100%; text-align: center;"> <tr> <td>Save</td> <td>&lt;F1&gt;</td> </tr> <tr> <td><b>Display / Load</b></td> <td>&lt;F2&gt;</td> </tr> </table>	Save	<F1>	<b>Display / Load</b>	<F2>										
Save	<F1>																
<b>Display / Load</b>	<F2>																
<table border="1" style="width: 100%; text-align: center;"> <tr> <td>&lt;F4&gt; Search</td> <td>&lt;F5&gt; Command</td> <td>&lt;F6&gt; End</td> <td>&lt;F7&gt; <b>Change</b></td> </tr> <tr> <td colspan="3"></td> <td> <table border="1" style="width: 100%; text-align: center;"> <tr> <td><b>Module file</b></td> <td>&lt;F1&gt;</td> </tr> <tr> <td>Symbol file</td> <td>&lt;F2&gt;</td> </tr> <tr> <td>Text file</td> <td>&lt;F3&gt;</td> </tr> <tr> <td><b>Display / Load</b></td> <td>&lt;F2&gt;</td> </tr> </table> </td> </tr> </table>	<F4> Search	<F5> Command	<F6> End	<F7> <b>Change</b>				<table border="1" style="width: 100%; text-align: center;"> <tr> <td><b>Module file</b></td> <td>&lt;F1&gt;</td> </tr> <tr> <td>Symbol file</td> <td>&lt;F2&gt;</td> </tr> <tr> <td>Text file</td> <td>&lt;F3&gt;</td> </tr> <tr> <td><b>Display / Load</b></td> <td>&lt;F2&gt;</td> </tr> </table>	<b>Module file</b>	<F1>	Symbol file	<F2>	Text file	<F3>	<b>Display / Load</b>	<F2>	Loading and displaying module file
<F4> Search	<F5> Command	<F6> End	<F7> <b>Change</b>														
			<table border="1" style="width: 100%; text-align: center;"> <tr> <td><b>Module file</b></td> <td>&lt;F1&gt;</td> </tr> <tr> <td>Symbol file</td> <td>&lt;F2&gt;</td> </tr> <tr> <td>Text file</td> <td>&lt;F3&gt;</td> </tr> <tr> <td><b>Display / Load</b></td> <td>&lt;F2&gt;</td> </tr> </table>	<b>Module file</b>	<F1>	Symbol file	<F2>	Text file	<F3>	<b>Display / Load</b>	<F2>						
<b>Module file</b>	<F1>																
Symbol file	<F2>																
Text file	<F3>																
<b>Display / Load</b>	<F2>																
<table border="1" style="width: 100%; text-align: center;"> <tr> <td>&lt;F4&gt; Search</td> <td>&lt;F5&gt; Command</td> <td>&lt;F6&gt; End</td> <td>&lt;F7&gt; <b>Change</b></td> </tr> <tr> <td colspan="3"></td> <td> <table border="1" style="width: 100%; text-align: center;"> <tr> <td>Module file</td> <td>&lt;F1&gt;</td> </tr> <tr> <td><b>Symbol file</b></td> <td>&lt;F2&gt;</td> </tr> <tr> <td>Text file</td> <td>&lt;F3&gt;</td> </tr> <tr> <td><b>Display / Load</b></td> <td>&lt;F2&gt;</td> </tr> </table> </td> </tr> </table>	<F4> Search	<F5> Command	<F6> End	<F7> <b>Change</b>				<table border="1" style="width: 100%; text-align: center;"> <tr> <td>Module file</td> <td>&lt;F1&gt;</td> </tr> <tr> <td><b>Symbol file</b></td> <td>&lt;F2&gt;</td> </tr> <tr> <td>Text file</td> <td>&lt;F3&gt;</td> </tr> <tr> <td><b>Display / Load</b></td> <td>&lt;F2&gt;</td> </tr> </table>	Module file	<F1>	<b>Symbol file</b>	<F2>	Text file	<F3>	<b>Display / Load</b>	<F2>	Loading and displaying symbol file
<F4> Search	<F5> Command	<F6> End	<F7> <b>Change</b>														
			<table border="1" style="width: 100%; text-align: center;"> <tr> <td>Module file</td> <td>&lt;F1&gt;</td> </tr> <tr> <td><b>Symbol file</b></td> <td>&lt;F2&gt;</td> </tr> <tr> <td>Text file</td> <td>&lt;F3&gt;</td> </tr> <tr> <td><b>Display / Load</b></td> <td>&lt;F2&gt;</td> </tr> </table>	Module file	<F1>	<b>Symbol file</b>	<F2>	Text file	<F3>	<b>Display / Load</b>	<F2>						
Module file	<F1>																
<b>Symbol file</b>	<F2>																
Text file	<F3>																
<b>Display / Load</b>	<F2>																

Fig. 1-26 Help and pull-up menus

**1.11 Editing functions**

The processing of files is assisted by special key functions. These functions apply to the entire PLC utilities.

**Cursor up****Cursor down****Cursor right****Cursor left****Jump to end of line****Jump to beginning of line****Delete current character****Delete character to left of cursor**

Backspace

**Delete whole line****Replace/Insert****F10****Help.**

For your notes:



**Contents**

	Page
<b>2</b>	<b>Programming ..... 2-1</b>
2.1	Main menu and defaults ..... 2-5
2.2	Edit module files ..... 2-8
2.3	Edit symbol file ..... 2-19
2.4	Assign module files/symbol file ..... 2-24
2.5	Load PLC program into the controller ..... 2-25
2.6	Test PLC program ..... 2-28

## Illustrations

Fig.		Page
2-1	Programming sequence .....	2-2
2-2	Configuration .....	2-3
2-3	Main menu .....	2-5
2-4	Defaults .....	2-6
2-5	Project directory EXAMPLE.300\ZS0 .....	2-7
2-6	OM1 .....	2-9
2-7	PM1, Network 1 .....	2-10
2-8	Network overview .....	2-11
2-9	Network overview with 5 networks .....	2-12
2-10	PM1, Network 2 .....	2-13
2-11	PM1, Network 3 .....	2-14
2-12	PM1, Network 4 .....	2-15
2-13	PM1, Network 5 .....	2-16
2-14	Network 1 in LD .....	2-17
2-15	Network 1 in FUD .....	2-18
2-16	OM form .....	2-19
2-17	I form .....	2-20
2-18	O form .....	2-21
2-19	DM form .....	2-23
2-20	Project directory before assignment .....	2-23
2-21	Project directory after assignment .....	2-24
2-22	Project directory after linking .....	2-26
2-23	PLC Monitor program .....	2-28
2-24	LD monitor .....	2-29
2-25	FUD monitor .....	2-30





## 2 Programming

This chapter describes programming with PLC utilities, with the help of an example containing all the important steps for programming a PLC program.

Before you program your controller with the example program in this chapter, you should familiarize yourself with chapter **1 User guide and main menu**.

For greater detail on programming your controller, please also consult the special device or software manuals of the PIC, PC400, PC600, CL100, CL300 or CL500 controllers.



**The following example applies to the CL300 controller.**

Programming takes the form of an instruction list.

The program consists of an organisation module **OM1**, a program module **PM1** and a symbol file **SYMBOL**.

The following steps are required for programming:

- **Defaults**
- Edit module file **OM1**
- Edit module file **PM1**
- Edit symbol file **SYMBOL**
- **Assign**
- Set **memory configuration**
- **Link** modules
- **Load** program into controller
- **Test** program

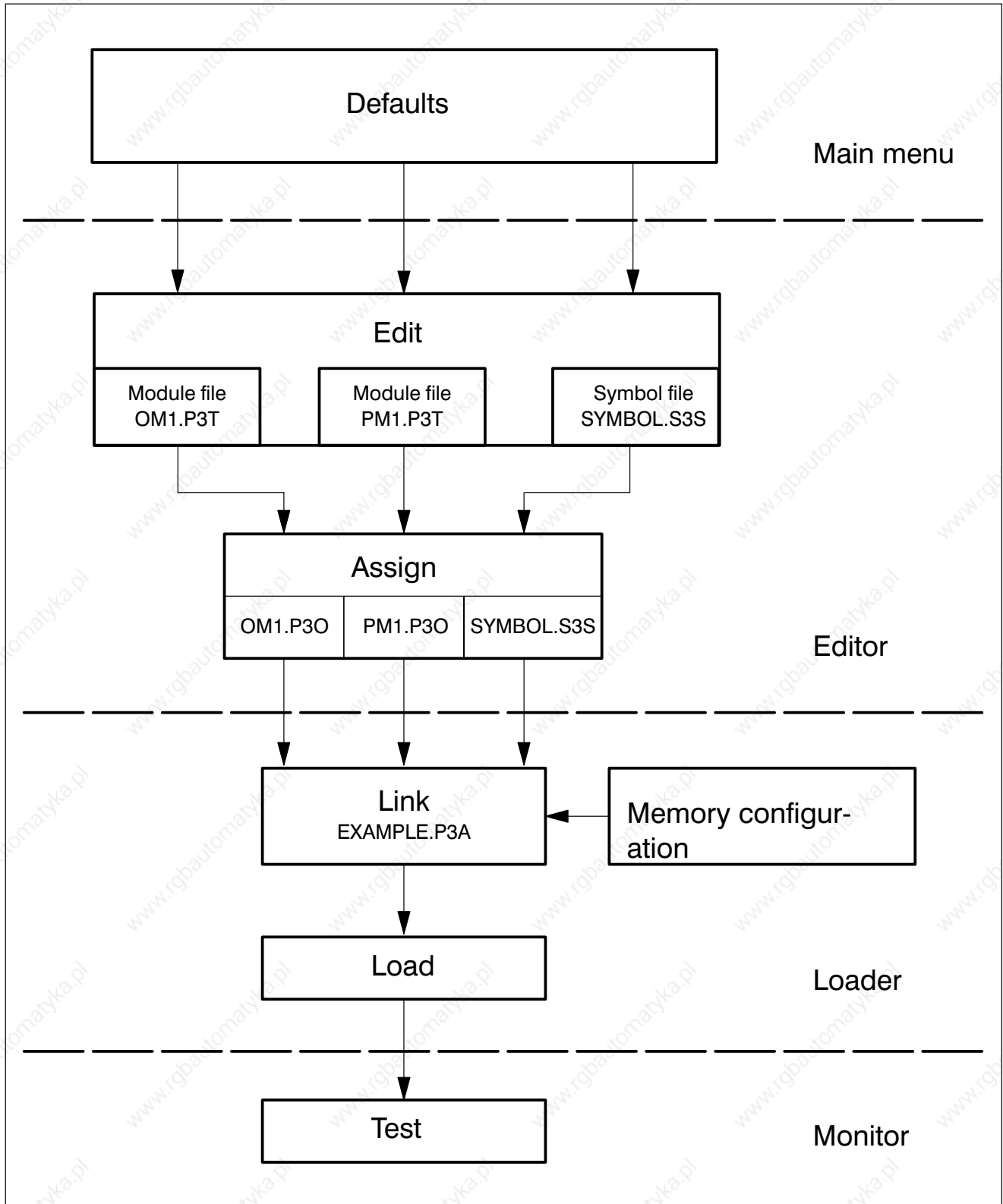


Fig. 2-1 Programming sequence



The following configuration is required for the programming example.

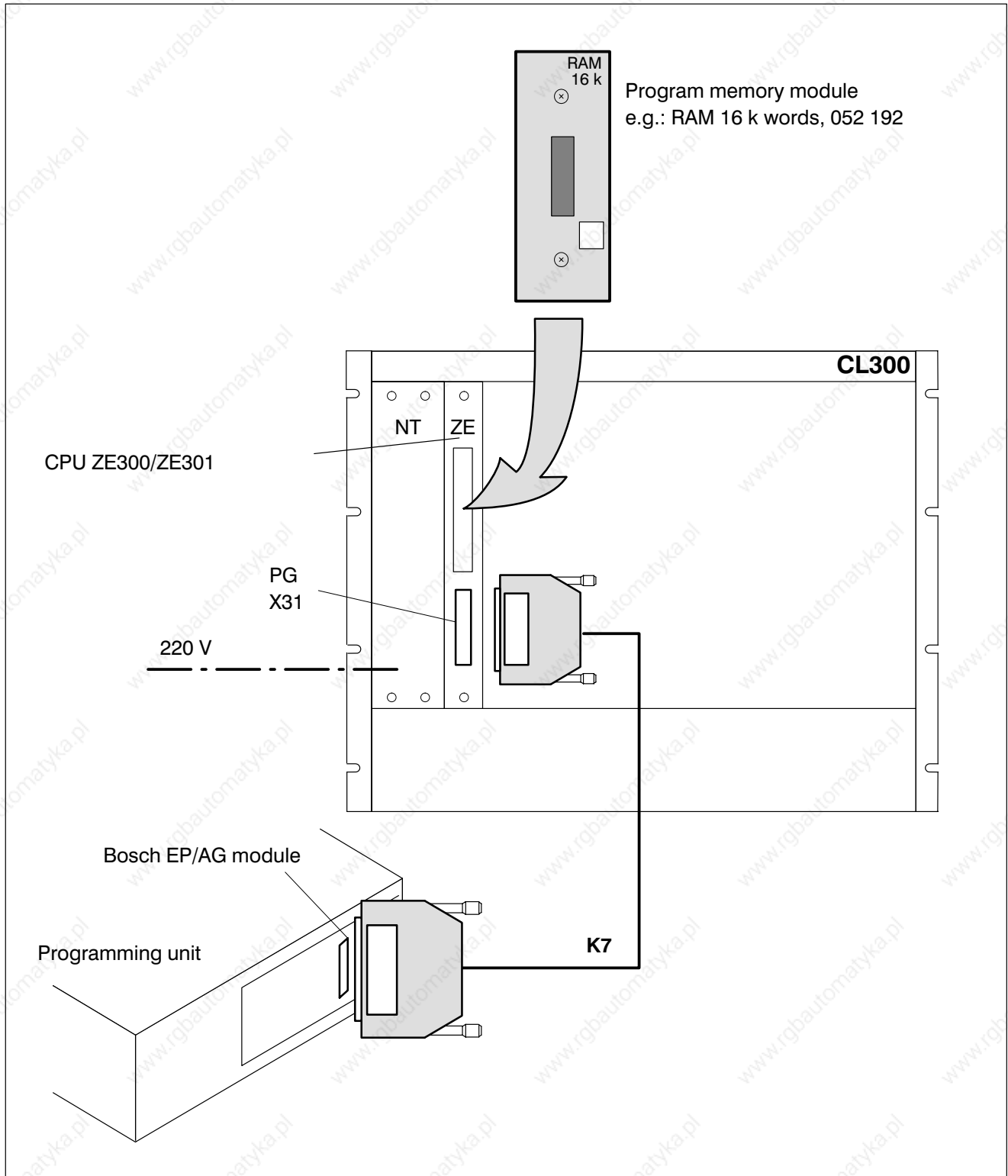


Fig. 2-2 Configuration



Create this small configuration at your own workplace, so that you can reconstruct all the steps of the example. You can also carry out this programming example in a modified form using a control unit other than a CL300.

**Do not load this example program into a running system!**



**2.1 Main menu and defaults**

- ★ Plug the **software dongle** into the **parallel interface**.
- ★ **Remove any disk** which may be in the floppy disk drive.
- ★ **Switch on** programming unit.



C:\>



PROFI



★

**PLC**

C:\PG \

Disk	Name	Type	Length
Select>			

Information

> Disk Info <

0 Files

13797376 Bytes free

7512064 Bytes used

21309440 Bytes in total

---

>> Memory Info <<

123504 Bytes free

531856 Bytes used

655360 Bytes in total

---

>> Extended Memory Info <<

HIMEM driver active 2.77

196608 Bytes (XMS) free

in 32 memory blocks

Select info

PG >>>

---

Time

01.03.1992

13:13:13

<F10>  
Help

<F1> Command	<F2> Editor	<F3> Monitor	<F4> Loader	<F5> Lister	<F6> End	
-----------------	----------------	-----------------	----------------	----------------	-------------	--

PLC utilities Version 3.0 (c) 1987–92 Robert Bosch GmbH

Fig. 2–3 Main menu

★



**Editor**

**Enter defaults**

★


**Call up defaults**

C:\PG\				<b>Project name</b>		
Disk C	Name	Type	Length	Controller type	CL300	>>>
<b>Select &gt;</b>				Control unit	ZS0	>>>
				Project status		
				Module file name		
				Symbol file name		
				Program file name		
				Text file name		
Command	Editor	Monitor	Loader	Lister	End	
PLC utilities			Version 3.0 (c) 1987–92 Robert Bosch GmbH			

Fig. 2–4 Defaults

**Project name**

**EXAMPLE**
**Controller type**

★

 Select the controller **CL300**.


★


 Create new project **EXAMPLE?**
**Yes/No**

★





**Project status**



PROJECT1.PLC

**Module file name**



OM1



OM1.P3T

**Symbol file name**



SYMBOL



SYMBOL.S3S

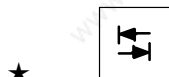
**Program file name**



TEST



TEST.P3A



★

**Exit defaults**

PLC utilities has created a new **project directory EXAMPLE.300** (PRO-DIR). In the project directory there is a subdirectory for the control unit **ZS0**. In the **PROJECT1.PLC** file the project status is saved with the defaults of the project.

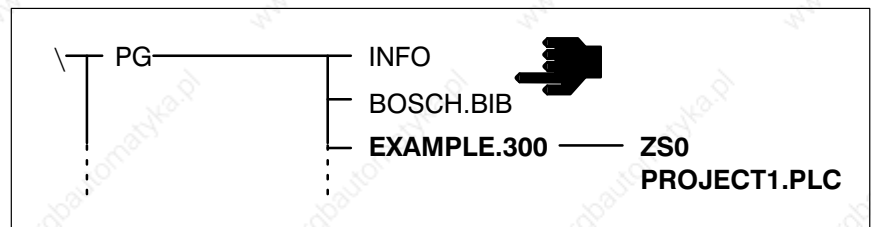


Fig. 2-5 Project directory EXAMPLE.300\ZS0



**Help.**



## 2.2 Edit module files

Create the **module files OM1** and **PM1** with the PLC Editor program.

### Organisation module OM1

★  **Editor**

On the screen, the first empty network of the OM1 is displayed. Enter the network title.

★  **Command**

★  **Network**

★  **Edit title**



OM1, PM1 module call

The **program module PM1** is called in the organisation module OM1. Enter the following PLC program.

```
 ;Project EXAMPLE.300
;Organisation module OM1
```

```
CM      -PM1      ;module call -PM1
```

```
EP
```

Fig. 2–6 shows the completed OM1.





1
**OM1, module call of PM1**

```
;Project EXAMPLE.300
;Organisation module OM1

CM      -PM1      ;module call -PM1

EP
```

ZS0/OM1

PI: 2

RG: 0

Insert

IL mode

<b>&lt;F1&gt;</b> Block	<b>&lt;F2&gt;</b> Assign	<b>&lt;F3&gt;</b> Symbol	<b>&lt;F4&gt;</b> Search	<b>&lt;F5&gt;</b> Command	<b>&lt;F6&gt;</b> End	<b>&lt;F7&gt;</b> Change
----------------------------	-----------------------------	-----------------------------	-----------------------------	------------------------------	--------------------------	-----------------------------

PLC Editor program
Version 3.0 (c) 1987–92 Robert Bosch GmbH

Fig. 2–6 OM1

Change to program module PM1.

- ★  F7 **Change**
- ★  F2 **Display/Load**
- ★  F1 **Module file**
- OM1.P3T
- PM1
- Save module file **OM1.P3T?** **Yes / No**
- ★  Y


## Program module PM1

The PM1 should be divided between **5 networks**. On the screen, the 1st empty network of the PM1 is displayed. Enter the **network title** for the 1st network.

★  **Command**

★  **Network**

★  **Edit title**

 **PM1, AND connection of inputs**

Enter the PLC program for the 1st network.

A B -INPUT1  
 AN B -INPUT2  
 O B -INPUT5  
 = B -OUTPUT1

1	PM1, AND connection of inputs					
A	B	-	INPUT1			
AN	B	-	INPUT2			
O	B	-	INPUT5			
=	B	-	OUTPUT1			

ZS0/PM1		PI: 5	RG: 1	Insert	IL mode	
<F1>	<F2>	<F3>	<F4>	<F5>	<F6>	<F7>
Block	Assign	Symbol	Search	Command	End	Change

PLC Editor program Version 3.0 (c) 1987–92 Robert Bosch GmbH

Fig. 2–7 PM1, Network 1

The 1st network is now completed. To create the other networks, go to the **network overview**.



- ★  **Command**
- ★  **Overview/Detail**

Network overview

1	<b>PM1, AND connection of inputs</b>
---	--------------------------------------

ZS0/PM1		Insert		IL	
<F1>	<F2>	<F3>	<F4>	<F5>	<F6>
Block	Assign	Symbol	Search	Command	End
			<F7>		
			Change		

PLC Editor program Version 3.0 (c) 1987–92 Robert Bosch GmbH

Fig. 2–8 Network overview

Create the 2nd network:



Move the cursor to the right so that the 2nd network is inserted after the 1st network.



Enter the 2nd network title:



Load const. to A and after –OUTP10 transf., activate –DMTEST



**a new network opens at the end of the network title.**



Enter the network titles for the networks 3 to 5.



Jump, when  $-\text{INPUT2} = 1$



$-\text{INPUT2} = 0 \Rightarrow 01010101$  after  $-\text{DW10}$



**Terminate input of 5th network title without**



$-\text{INPUT2} = 1 \Rightarrow 00001111$  after  $-\text{DW10}$

Network overview	
1	PM1, AND connection of inputs
2	Load const. to A after $-\text{OUTP10}$ transf., activate $-\text{DMTEST}$
3	Jump, when $-\text{INPUT2} = 1$
4	$-\text{INPUT2} = 0 \Rightarrow 01010101$ after $-\text{DW10}$
5	$-\text{INPUT2} = 1 \Rightarrow 00001111$ after $-\text{DW10}$

ZS0/PM1		Insert			IL	
<F1>	<F2>	<F3>	<F4>	<F5>	<F6>	<F7>
Block	Assign	Symbol	Search	Command	End	Change

PLC Editor program Version 3.0 (c) 1987–92 Robert Bosch GmbH

Fig. 2–9 Network overview with 5 networks

Go into the 2nd network:

★ 3 times



★



**Command**



★ F5

**Overview/Detail**

Enter the PLC program for the 2nd network.

L BY K11110000B, A  
 T BY A, -OUTP10  
 CM -DMTEST

2	Load const. to A and after -OUTP10 transf., activate -DMTEST
L	BY K11110000B, A
T	BY A, -OUTP10
CM	-DMTEST

ZS0/PM1	PI: 9	RG: 1	Insert	IL mode		
<F1> Block	<F2> Assign	<F3> Symbol	<F4> Search	<F5> Command	<F6> End	<F7> Change

PLC Editor program

Version 3.0 (c) 1987-92 Robert Bosch GmbH

Fig. 2-10 PM1, Network 2

Jump to the 3rd network:

★ Control + PgDn

Enter the PLC program for the 3rd network.

A B -INPUT2  
 JPC -JUMP1

<b>3</b>	<b>Jump, when -INPUT2 = 1</b>					
A	B	-INPUT2				
JPC	-JUMP1					

ZS0/PM1	PI: 12	RG: 2	Insert		IL mode	
<F1>	<F2>	<F3>	<F4>	<F5>	<F6>	<F7>
Block	Assign	Symbol	Search	Command	End	Change

PLC Editor program Version 3.0 (c) 1987-92 Robert Bosch GmbH

Fig. 2-11 PM1, Network 3

Jump to the 4th network:



Enter the PLC program for the 4th network.

L BY K01010101B, A  
T BY A, -DW10

AN B -INPUT2  
JPC -JUMP0



<b>4</b>	<b>-INPUT2 = 0 =&gt; 01010101 after -DW10</b>					
L	BY	K01010101B,A				
T	BY	A,-DW10				
AN	B	-INPUT2				
JPC		-JUMPO				

ZS0/PM1	PI: 17	RG: 3	Insert	IL mode	
<F1>	<F2>	<F3>	<F4>	<F5>	<F6>
Block	Assign	Symbol	Search	Command	End
<F7>					<F7>
Change					Change

PLC Editor program Version 3.0 (c) 1987-92 Robert Bosch GmbH

Fig. 2-12 PM1, Network 4

Jump to the last network:



Enter the PLC program for the 5th network.

```


    -JUMP1
L   BY K00001111B,A
T   BY A,-DW10

    -JUMPO

EM
  
```

<b>5</b>	<b>-INPUT2 = 1 ==&gt; 00001111 after -DW10</b>					
	-JUMP1					
L	BY K00001111B,A					
T	BY A,-DW10					
	-JUMP0					
	EM					

ZS0/PM1	PI: 21	RG: 3	Insert	IL mode		
<F1>	<F2>	<F3>	<F4>	<F5>	<F6>	<F7>
Block	Assign	Symbol	Search	Command	End	Change

PLC Editor program Version 3.0 (c) 1987-92 Robert Bosch GmbH

Fig. 2-13 PM1, Network 5

The PM1 is now completed.



F10

**Help.**





We now want to have a look at the 1st network in the LD and FUD display.

- ★ Control + Home
- ★ F5 **Command**
- ★ F7 **Ladder diagram**

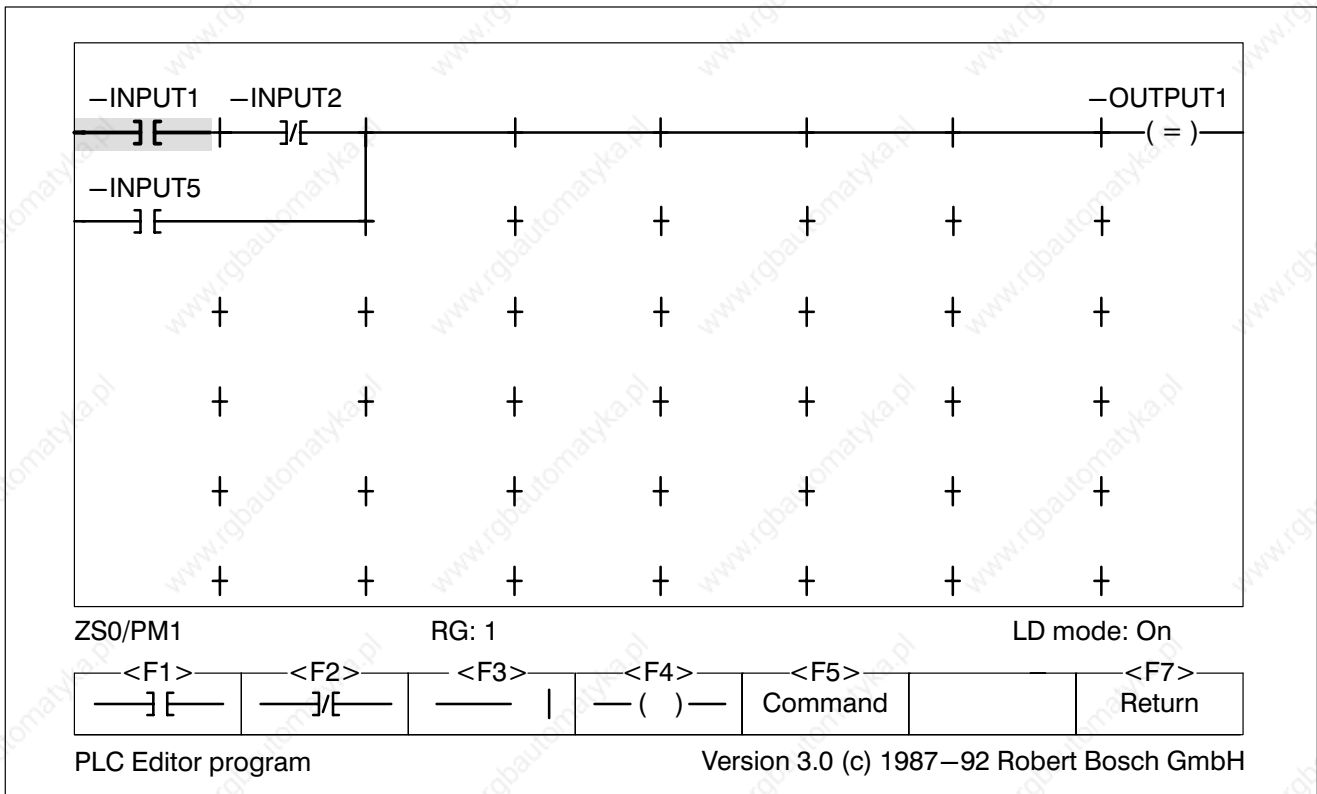


Fig. 2–14 Network 1 in LD

- ★ F7 **Return**
- ★ F5 **Command**
- ★ F8 **Function diagram**

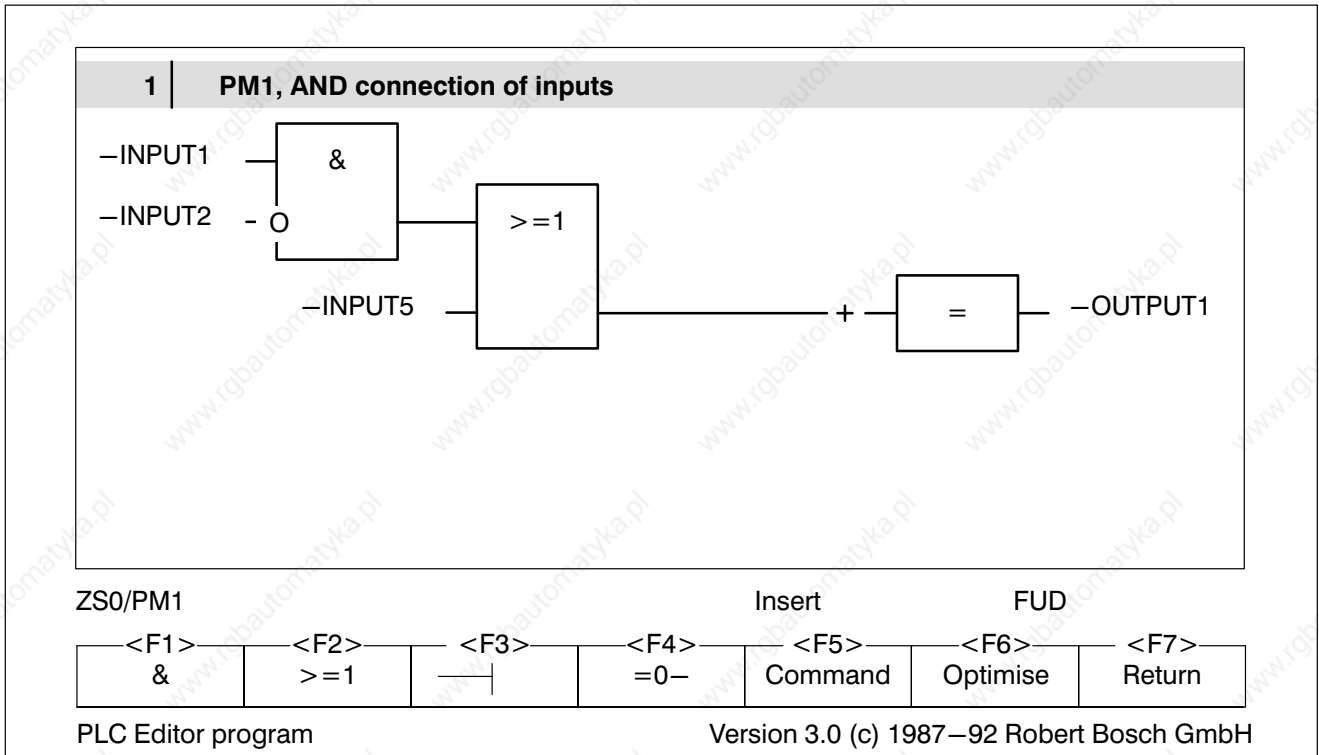


Fig. 2–15 Network 1 in FUD

Change to symbol file **SYMBOL.S3S**.

- ★ F7 **Return**
- ★ F7 **Change**
- ★ F2 **Display / Load**
- ★ F2 **Symbol file**
- file name:
- SYMBOL**



**2.3 Edit symbol file**

The first form of the symbol file appears on the screen.

Enter the name of the **organisation module**.



OM1; Organisation module No. 1

Type	Module name ;Comment	R/E
OM 1	<b>OM1 ; ORGANISATION MODULE NO. 1</b>	<b>R</b>
OM 2		R
OM 3		R
OM 4		R
OM 5		R
OM 6		R
OM 7		R
OM 8		R
OM 9		R

ZS0/SYMBOL Insert

<F1> Block	<F2> Assign	<F3> Symbol	<F4> Search	<F5> Command	<F6> End	<F7> Change
---------------	----------------	----------------	----------------	-----------------	-------------	----------------

PLC Editor program Version 3.0 (c) 1987–92 Robert Bosch GmbH

Fig. 2–16 OM form

PgDn



Enter the name of the **program module**.



PM1; Program module No. 1

Scroll further to the **input form**.

★

★

Enter the inputs for the absolute addresses.

I0.0      INPUT1      Input No. 1, sensor switch On  
 I0.1      INPUT2      Input No. 2, limit switch Front  
 I0.4      INPUT5      Input No. 5, switch Manual



INPUT1      Input No. 1, sensor switch On



INPUT2      Input No. 2, limit switch Front

★  Twice



INPUT5      Input No. 5, switch Manual

Address	Symbol	Comment	Type
I 0.0	INPUT1	Input No. 1, sensor switch On	
I 0.1	INPUT2	Input No. 2, limit switch Front	
I 0.2			
I 0.3			
I 0.4	<b>INPUT5</b>	<b>Input No. 5, switch Manual</b>	
I 0.5			
I 0.6			
I 0.7			
I 1.0			
I 1.1			
I 1.2			
I 1.3			
I 1.4			
I 1.5			
I 1.6			
I 1.7			

ZS0/SYMBOL

Insert

<F1> Block	<F2> Assign	<F3> Symbol	<F4> Search	<F5> Command	<F6> End	<F7> Change
---------------	----------------	----------------	----------------	-----------------	-------------	----------------

PLC Editor program

Version 3.0 (c) 1987–92 Robert Bosch GmbH

Fig. 2–17 I form



Scroll further to the **output form**.

★

★

Enter the outputs for the absolute addresses O0.0 and O1.0.

Address	Symbol	Comment	Type
O 0.0	OUTPUT1	Motor On	
O 0.1			
O 0.2			
O 0.3			
O 0.4			
O 0.5			
O 0.6	OUTP10	Display	
O 0.7			
O 1.0			
O 1.1			
O 1.2			
O 1.3			
O 1.4			
O 1.5			
O 1.6			
O 1.7			

ZS0/SYMBOL

Insert

<F1> Block	<F2> Assign	<F3> Symbol	<F4> Search	<F5> Command	<F6> End	<F7> Change
---------------	----------------	----------------	----------------	-----------------	-------------	----------------

PLC Editor program

Version 3.0 (c) 1987–92 Robert Bosch GmbH

Fig. 2–18 O form

Jump further to data module DM0.

★

**Search**

★

**Absol. address (symbol list)**

Address:

DM0

Fill out the **data module header** and set the **data module length**.

★  **Symbol**

★  **Edit data module header**

Data module name:

 DMTEST

Comment:

 DM to be tested

EPROM RAM <A>

★ Select RAM.

★

DM length 0


 20

★ Enter the data words D0 to D18 as shown in Fig. 2–19.

**Help.**





DM		0	Name: DMTEST		Comment: DM to be tested		RAM/EPROM: R	
No.	Symbol	Type	S	Data field			F	
D 0	DW0	Word	N	20			D	
D 2	DW2	Word	N	09			D	
D 4	DW4	Word	N	85			D	
D 6	DW6	Ascii	N	Beatrice				
D 14	DW8	Word	Y	-10			D	
		<b>The result is entered in -DW10</b>						
D 16	DW10	Word	N				B	
D 18		Word	N				D	
D 20		Word	N				D	
D 22		Word	N				D	
D 24		Word	N				D	
D 26		Word	N				D	
D 28		Word	N				D	
D 30		Word	N				D	

ZS0/SYMBOL Replace

<F1> Block	<F2> Assign	<F3> Symbol	<F4> Search	<F5> Command	<F6> End	<F7> Change
---------------	----------------	----------------	----------------	-----------------	-------------	----------------

PLC Editor program Version 3.0 (c) 1987-92 Robert Bosch GmbH

Fig. 2-19 DM form



Enter

**inserts comment lines in the data module.**

Shift

+

Del

**deletes lines which contain errors.**

The comment line is only possible with assigned data lines and always belongs to the following data line.

The symbol file SYMBOL.S3S is thus completed.

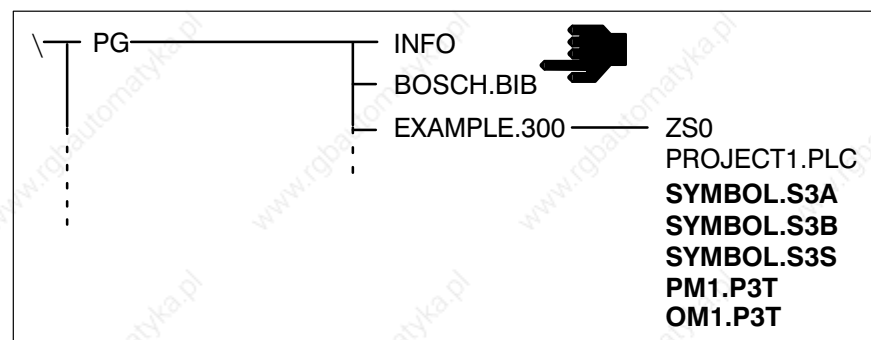


Fig. 2-20 Project directory before assignment

## 2.4 Assign module files/symbol file

The **absolute** and **symbolic addresses** from the **module files** and the **symbol file** must be assigned.

- ★  **Assign**
- ★  **Acc. to symbol file**
- ★  **Priority symbols**

The process is logged on the screen.

PLC utilities has created the following files.

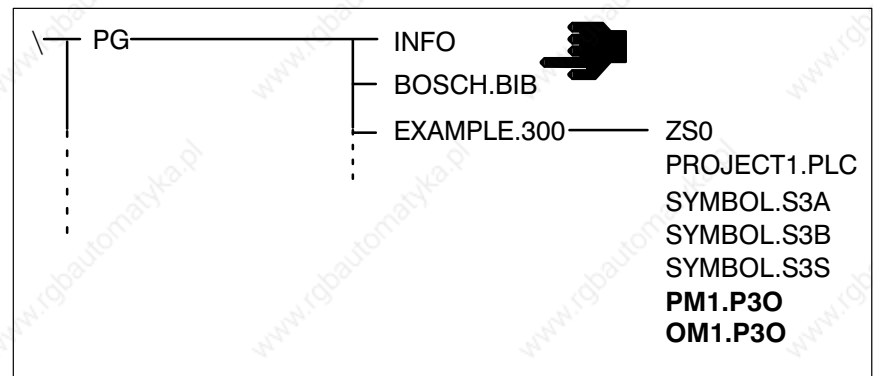


Fig. 2–21 Project directory after assignment

- ★  **Change**
- ★  **Save**
- Save **SYMBOL.S3S** symbol file? **Yes / No**
- ★



**2.5 Load PLC program into the controller**

★  **End**

★  **Loader**

**Configuration**

First of all the memory configuration of the PLC must be set.

★  **Configuration**

★ Start address: dec. 0 k

★

★ Length RAM: dec. 16 k

★

★

**Link**

★  **Link**

★  **Link all modules**

The program file **TEST.P3A** is linked. The process is logged on the screen.

After linking, the following files exist.

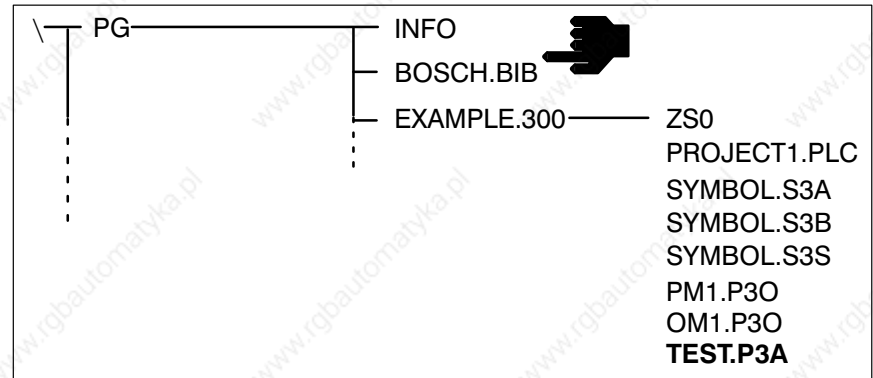


Fig. 2–22 Project directory after linking



F10

**Help.**



**Loading**



When you have created the controller configuration (see Fig. 2–2) at your workplace, you can load the program into the central processing unit.

**Do not load example program into a running system!**

★ Connect the connecting cable **K7** to the PLC programming interface of the **EP/AG module** and to the **central processing unit ZE300/ZE301**, see Fig. 2–2.

★ Switch on controller.



**Load**



**Load program into the controller**



**With reset rem. markers/operands**



Controller operating in RUN (Monitor)! Switch to STOP (Edit)? **Yes / No**



The PLC program is loaded. The process is logged on the screen.



Controller is in STOP (Edit)! Switch back to RUN (Mon.)? **Yes/No**



The creation of the PLC program with PLC utilities is thus completed.

Change to the PLC Monitor program.



**End**



**Monitor**

## 2.6 Test PLC program

The PLC program is tested in the **PLC Monitor program**.

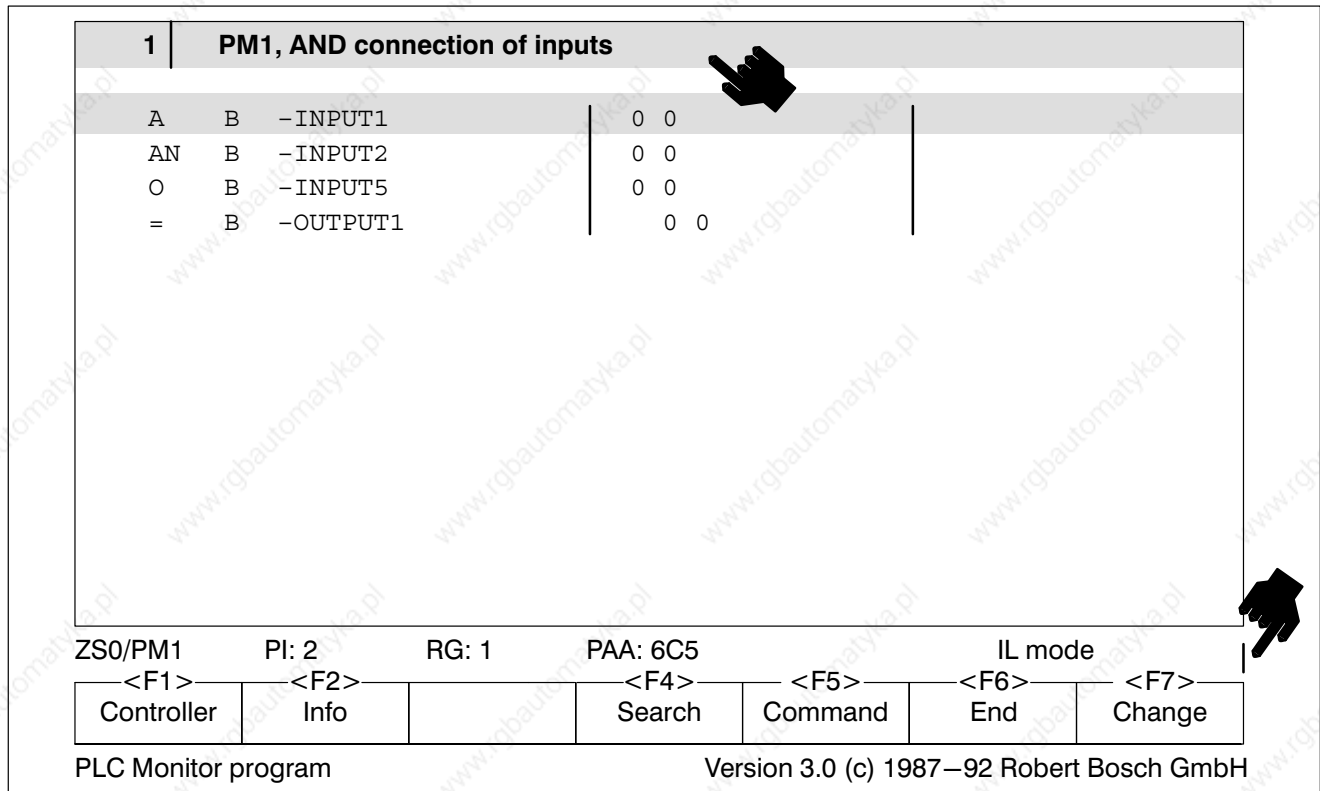


Fig. 2-23 PLC Monitor program



The status are refreshed on the screen only when the bar rotates around the right-hand margin of the information line.

### Setting

With the **set** command the individual bits are set and the function of the PLC program can be tested.

The input **-INPUT1** is set to 1. Output **-OUTPUT1** must then also be 1.

- ★ F7      **Change**
- ★ F2      **Display / Load**



- ★ F6 **Set**
- ★ F1 **Bit**
- Bit operand:
- INPUT1**
- ★ F1 **Set value 1**
- ★ F5 **Command**
- ★ F7 **Ladder diagram**

The monitor display is updated. We would now also like to look at the result in the LD and FUD display.

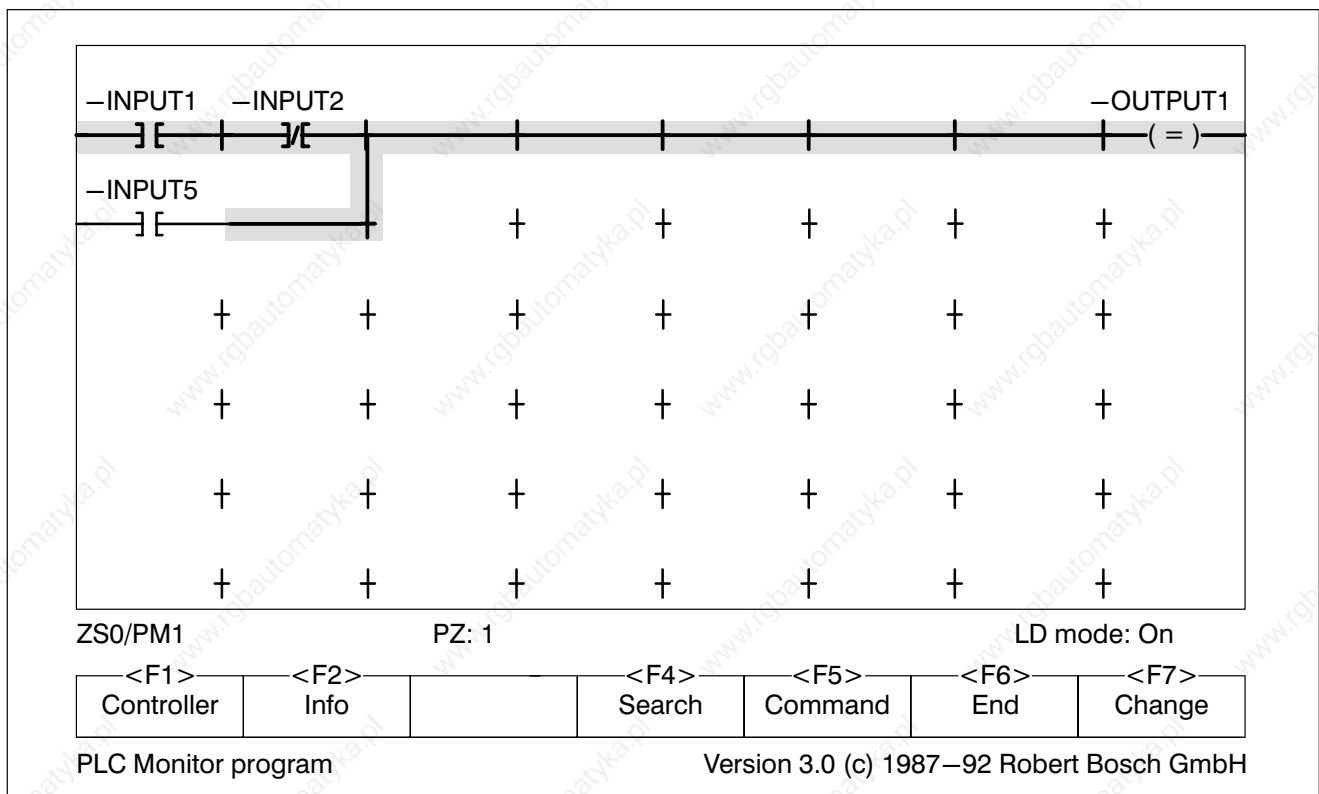


Fig. 2–24 LD monitor

- ★ F5 **Command**
- ★ F8 **Function diagram**

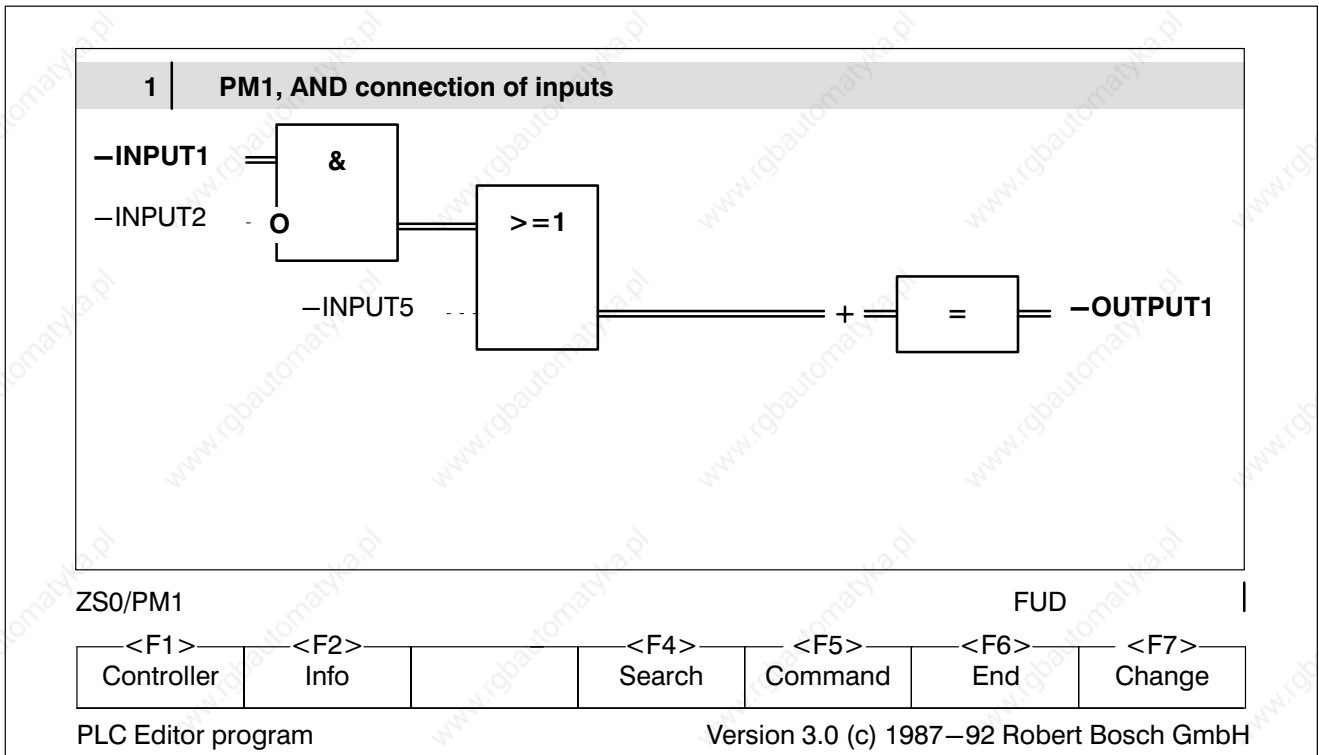


Fig. 2–25 FUD monitor

Reset the setting:

- ★ F7 **Change**
  - ★ F2 **Display / Load**
  - ★ F6 **Set**
  - ★ F1 **Bit**
- Bit operand: -INPUT1



- ★
- ★  **Reset setting**

**Data module**



**The contents of the data module can only be changed when the data module is in the RAM.**

The contents of the data word –DW10 in the data module –DMTEST is dependent on the input –INPUT2.

- If –INPUT2 = 1, then –DW10 = 0000000000001111
- If –INPUT2 = 0, then –DW10 = 0000000001010101

- ★  **Change**
- ★  **Display / Load**
- ★  **Symbol file**

file name: SYMBOL.S3S

- ★
- By setting –INPUT2 to 1, –DW10 is changed.

- ★  **Change**
- ★  **Display / Load**
- ★  **Set**
- ★  **Bit**

Bit operand:



-INPUT2



F1

**Set value 1**

The data word -DW10 now has the value 000000000001111.



F10

**Help.**



**Contents**

	Page
<b>3 Editor .....</b>	<b>3-1</b>
3.1 Network .....	3-5
3.2 Screen layout .....	3-7
3.3 Module file editor IL .....	3-9
3.3.1 Edit field .....	3-9
3.3.2 Information line .....	3-13
3.3.3 Block .....	3-14
3.3.4 Assign .....	3-17
3.3.5 Symbol .....	3-20
3.3.6 Search .....	3-21
3.3.7 Command .....	3-23
3.3.8 End .....	3-26
3.3.9 Change .....	3-28
3.4 Module file editor LD .....	3-29
3.4.1 Edit field .....	3-30
3.4.2 Normally open/closed circuits .....	3-32
3.4.3 Connection .....	3-32
3.4.4 Output commands .....	3-33
3.4.5 Command .....	3-35
3.4.6 Return .....	3-36
3.5 Module file editor FUD .....	3-37
3.5.1 Edit field .....	3-39
3.5.2 And element & .....	3-44
3.5.3 OR element $\geq 1$ .....	3-45
3.5.4 Input pin $- $ .....	3-46
3.5.5 Complex elements $=0-$ .....	3-46
3.5.6 Command .....	3-52
3.5.7 Optimise .....	3-53
3.5.8 Return .....	3-53

	Page
3.6 Network overview .....	3–54
3.6.1 Block .....	3–57
3.7 Parameter list/module file description .....	3–58
3.7.1 Parameter list .....	3–59
3.7.2 Module file description .....	3–61
3.8 Module library .....	3–62
3.9 Symbol file editor .....	3–64
3.9.1 Module list .....	3–65
3.9.2 Data module .....	3–66
3.9.3 Operand form .....	3–70
3.10 Text file editor .....	3–71
3.11 Batch file .....	3–74

**Illustrations**

Fig.		Page
3-1	Change menu .....	3-2
3-2	Special lines .....	3-4
3-3	Network .....	3-5
3-4	Network overview .....	3-6
3-5	Editor .....	3-7
3-6	Editor function blocks .....	3-8
3-7	Single operand instruction .....	3-10
3-8	Dual operand instruction .....	3-10
3-9	Time grid/code number .....	3-12
3-10	Information line .....	3-13
3-11	Copying .....	3-15
3-12	Search criteria .....	3-21
3-13	Ladder diagram .....	3-30
3-14	Forbidden bridge circuit .....	3-31
3-15	Network and function diagram .....	3-38
3-16	Function diagram with & element .....	3-39
3-17	Fields in the function diagram .....	3-40
3-18	AND element & .....	3-44
3-19	OR element $\geq 1$ .....	3-45
3-20	XOR element .....	3-47
3-21	Comparator .....	3-48
3-22	SR flip-flop .....	3-48
3-23	Time element .....	3-49
3-24	Counter element .....	3-50
3-25	Allocation .....	3-50
3-26	Branch .....	3-51
3-27	Module call .....	3-51
3-28	Optimisation .....	3-53
3-29	Network .....	3-54

Fig.		Page
3–30	Network overview .....	3–55
3–31	Parameter list .....	3–59
3–32	Module file description .....	3–61
3–33	BOSCH.BIB directory .....	3–62
3–34	Symbol file forms .....	3–64
3–35	Module list .....	3–65
3–36	Data module .....	3–66
3–37	Data module overview list .....	3–69
3–38	Operand form .....	3–70
3–39	Change menu .....	3–71
3–40	ASCII line set .....	3–73



### 3 Editor

The PLC editor program is used for editing

- the **module file**
  - in the **instruction list IL**, see section 3.3
  - in the **ladder diagram LD**, see section 3.4
  - in the **function diagram FUD**, see section 3.5
- the **network overview**, see section 3.6
- the **parameter list/module file description**, see section 3.7
- the **symbol file**, see section 3.9
- any desired **text file**, see section 3.10
- a **batch file** as a special text file, see section 3.11

The **module library** is described in section 3.8.

#### editor

The PLC editor program is called up from the **main menu** by pressing

F2

**editor**

twice.

With the first press of the function key the defaults are displayed on the screen.

Change defaults:



See section 1.7 **Entering defaults**.

By pressing the function key a second time the PLC editor program is started.

F10

**Help.**



## Change between module, symbol and text file editor

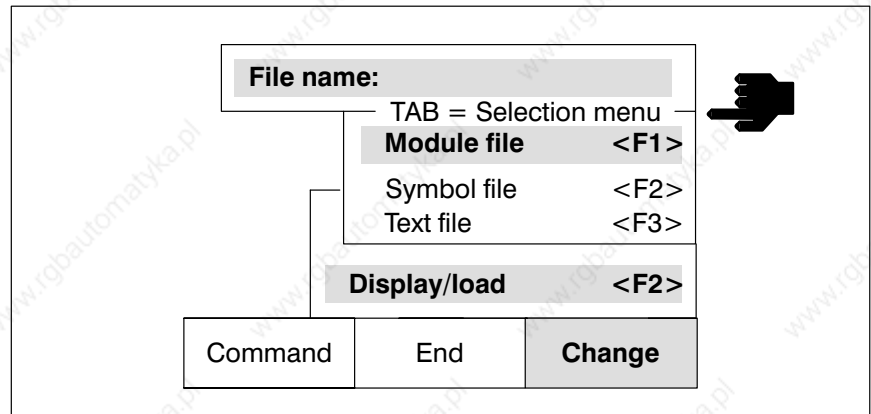
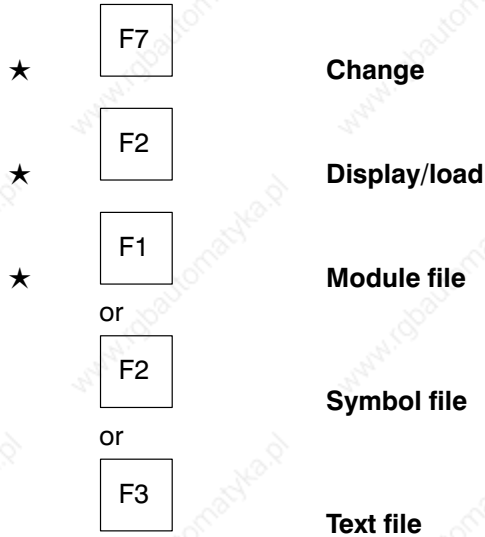


Fig. 3-1 Change menu



**Change between IL, LD and FUD**

In the module file editor a difference is made between the IL, LD and FUD display. The current setting is shown in the information line.

Toggle between IL, LD and FUD:

★	<input type="button" value="F5"/>	<b>Command</b>
★	<input type="button" value="F6"/>	<b>Instruction list</b>
	or	
	<input type="button" value="F7"/>	<b>Ladder diagram</b>
	or	
	<input type="button" value="F8"/>	<b>Function diagram</b>

**Edit / Command level**

The editor distinguishes between an **edit level** and a **command level**.

- A **flashing cursor** in the edit field displays the **edit level**.
- A **highlighted Command** in the function key bar displays the **command level**.

**Toggle between edit and command level**



**Call up a command**

**Language translation**

For example, an “AWL–Programm” written in the German language set–up is automatically translated as an IL (instruction list) program when the English set–up is called, and vice versa.

## Permitted lines

In the **symbol file** and **module file editor** the following lines are permitted:

- Letters from A to Z
- Numbers 0 to 9

ASCII lines below 20 H and above/equal to 80 H are **not** permitted.

In the **module lists** of the symbol file the MS DOS wild cards \* and ? are **not** permitted.

The following table shows which **special lines** are permitted.

Special line	Symbol file editor		Module file editor	
	Symbol column	Module list	Symbolic operand	Symbolic module call
"	yes	no	yes	no
/	yes	no	yes	no
\	yes	yes	yes	no
[	yes	no	yes	no
]	yes	no	yes	no
:	yes	yes	yes	no
!	yes	no	yes	no
<	yes	no	yes	no
>	yes	no	yes	no
+	yes	no	yes	no
-	yes	no	yes	no
=	yes	no	yes	no
;	no	yes	no	no
,	no	no	no	no
.	yes	yes	yes	no
Blank	no	no	no	no

Fig. 3-2 Special lines





### 3.1 Network

A module file can be divided into individual **networks**. When you are programming in the **IL** or **LD display**, this division into networks is unnecessary. The division of a module file into networks is only necessary for programming in the **FUD display**.

A network consists of several consecutive program lines. One network comprises several program branches, and its maximum size is that of a module file.

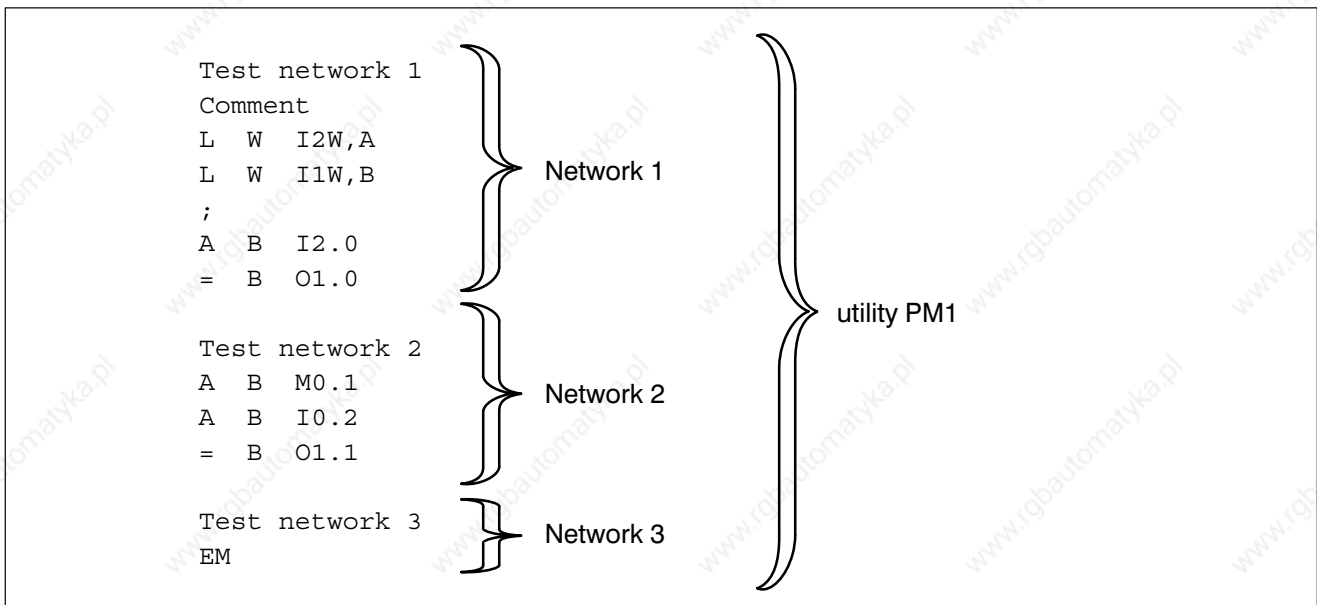
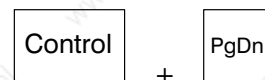


Fig. 3–3 Network

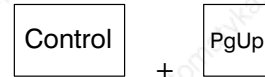
In the module editor only one network is displayed on the screen.

#### Scroll through networks

**branch to next network**



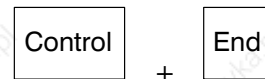
or



**branch to first network**

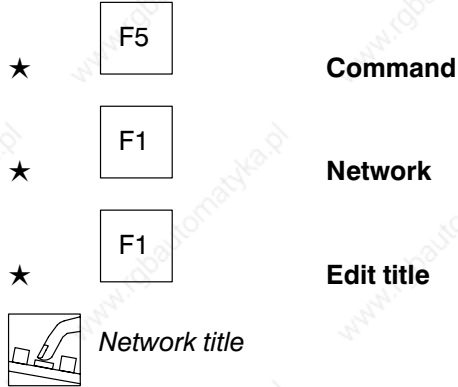


**branch to last network**

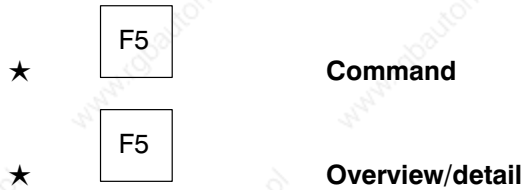


**Edit network title**

Each network receives a **network title** as a name.



**Network overview/detail**



**Network overview**

1	Network 1
2	<b>Network 2</b>

ZS0/OM1 Insert FUD

<b>&lt;F1&gt;</b> Block	<b>&lt;F2&gt;</b> Assign	<b>&lt;F3&gt;</b> Symbol	<b>&lt;F4&gt;</b> Search	<b>&lt;F5&gt;</b> Command	<b>&lt;F6&gt;</b> End	<b>&lt;F7&gt;</b> Change
----------------------------	-----------------------------	-----------------------------	-----------------------------	------------------------------	--------------------------	-----------------------------

PLC Editor program Version 3.0 (c) 1987–92 Robert Bosch GmbH

Fig. 3–4 Network overview



**See also section 3.6 Network overview.**



### 3.2 Screen layout

Fig. 3–5 shows the general screen layout in the PLC editor program.

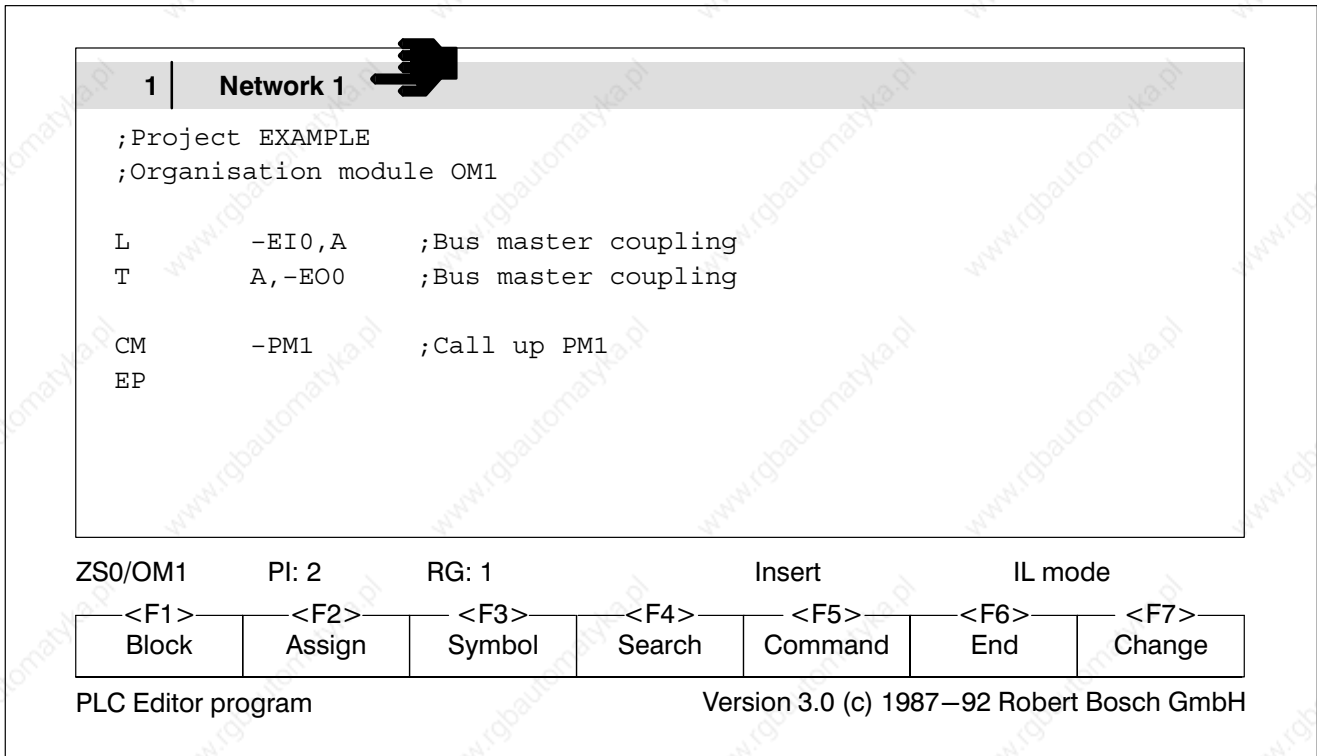


Fig. 3–5 Editor

The screen splits into 5 function blocks:

- **Network title**
- **Edit field**
- **Information line**
- **Function key bar**
- **Version/messages**

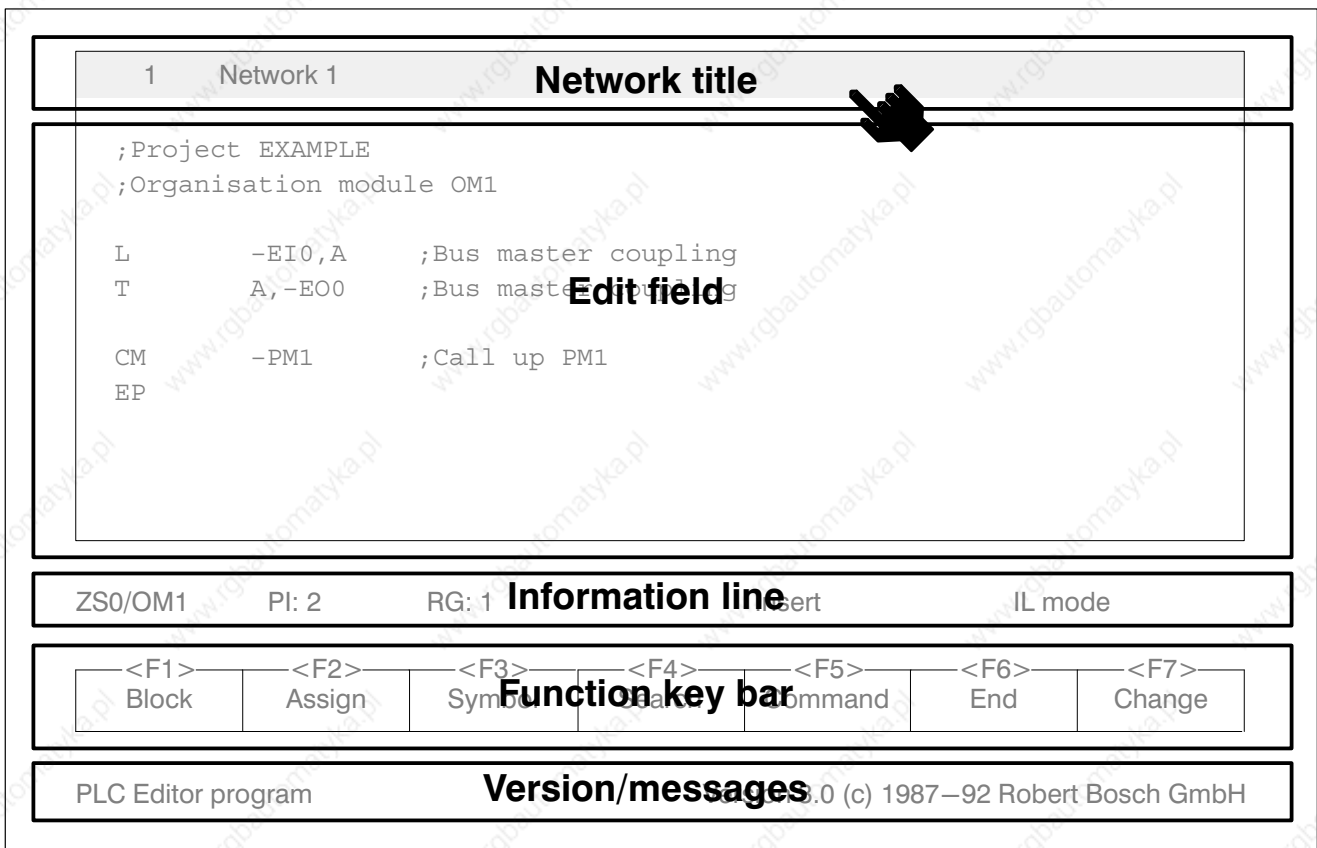


Fig. 3–6 Editor function blocks

The network title is only displayed in IL and FUD mode.

Information on the edit field and the information line can be found at the beginning of the individual sections of this chapter.



### 3.3 Module file editor IL

Calling up the module file editor IL:

★	<input type="button" value="F5"/>	<b>Command</b>
★	<input type="button" value="F6"/>	<b>Instruction list</b>

#### 3.3.1 Edit field

The edit field is subdivided into two areas.

- The first 4 columns of each line are reserved for possible error messages.
- In the remaining area you create the PLC program according to the rules governing the controller to be programmed.

#### Instruction line

A module file in IL mode consists of individual instruction lines. The instruction line comprises an **instruction part** and a **comment** (line comment), separated by a semicolon. Figs. 3–7 and 3–8 show the structure of an instruction line for **single** and **dual operand instructions**.

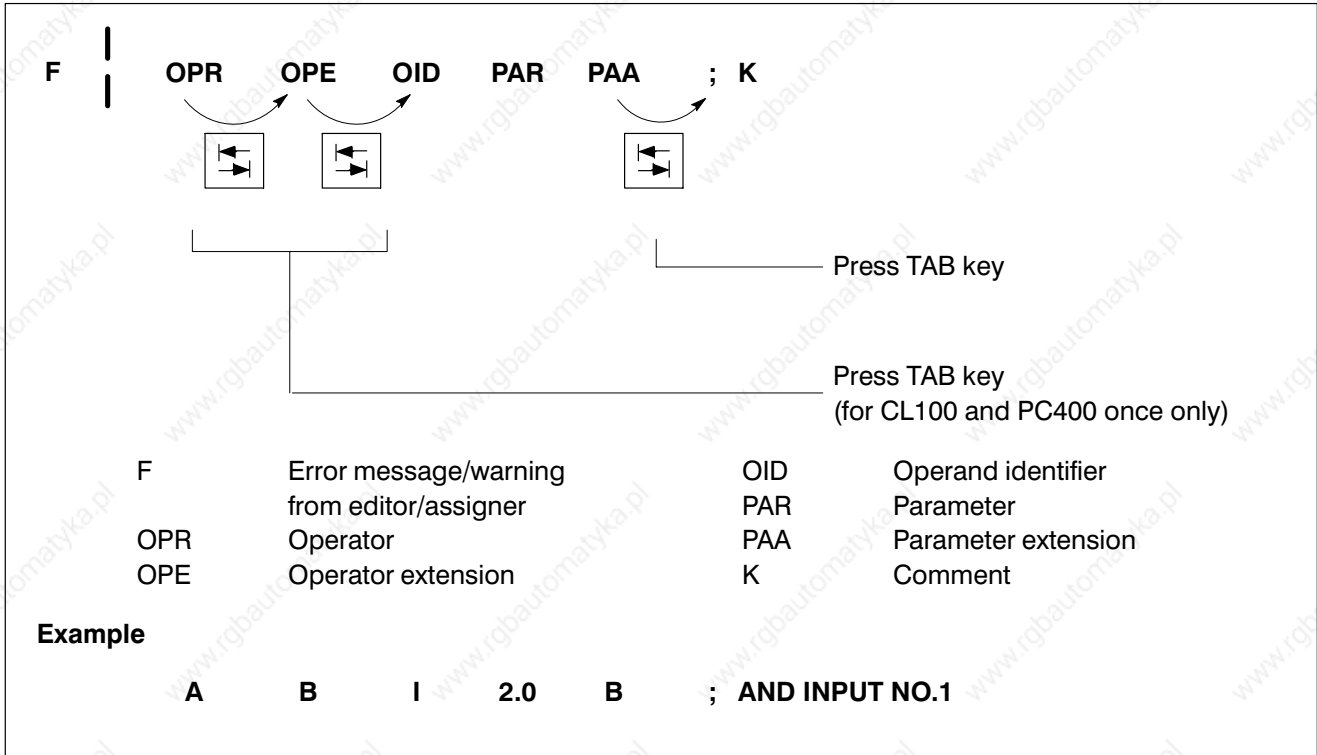


Fig. 3-7 Single operand instruction

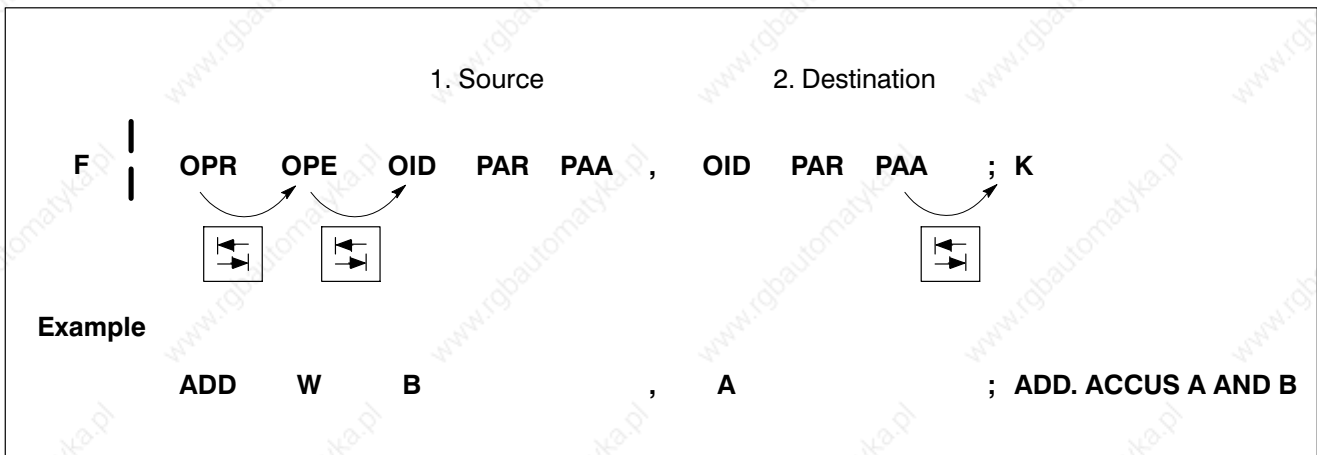


Fig. 3-8 Dual operand instruction

## Error message/warning

Instructions are checked as soon as the cursor moves on to a new line. If an error is discovered, it is shown by an error message at the beginning of the line. Errors are not recognized, however, if they result from contexts applying to several lines. Errors such as this are found and marked only when assigning.



A differentiation is made between a **warning** and an **error**.

A **warning** is a reference to possible errors. A program containing only warnings is assigned without any error.

**Warning >**

This pointed bracket indicates that no symbolic operand description for an absolute operand entered was found in this line during assignment.

**Error message**

There is an automatic search function for lines with error messages.

★  **Search**

★  **Error line**

The cursor branches to the next line containing errors.

The Help function provides additional indisplyation on an error message.

★  **Help**

**Comment**

A comment can be entered in every line. The start of a comment is marked by a **semicolon ;**.

; utility PM1  
A B I0.0 ;And input I0.0 key 1

**Page feed**

When you require a page feed at a position in the PLC program when printing, you must enter the line **\$P** preceded by a semicolon.

EM ;End of module  
; \$P

**Symbolic operand description**

In the symbolic notation, the operand is marked by a preceding **hyphen -**. The symbolic operand description, symbolic address, can consist of a maximum of 8 lines.

A B -INPUT1

**Time programming**

The times are programmed via time grids and code numbers.

Time grid	Code number
10 ms	0
100 ms	1
1 s	2
10 s	3

Fig. 3–9 Time grid/code number

The desired time is programmed by specifying an integer multiplier and the code number of the time grid.

**Examples**

## 1. Desired time 1 s

SP      K100.0                                        ;time grid 10 ms,  
   ;code number 0

or

SP      K1.2    ;time grid 1 s,  
   ;code number 2

Programming with the time grid 10 ms is more precise.

## 2. Desired time 250 ms

SP      K25.0                                        ;time grid 10 ms,  
   ;code number 0

The desired time of 250 ms **cannot** be programmed using the time grid 100 ms (code number 1).

**PC400**

For the PC400 the time value is entered on the display and control panel of the time/counter module.

**PC600**

For the PC600 the times are programmed directly.

SR      K80.3,T30                                        ;Time value 800 s

**CL300/CL500**

For the CL300 the times are loaded indirectly via battery A.

L      K20.1,A                                        ;Time value 2000 ms





### 3.3.2 Information line



Fig. 3–10 Information line

#### Control unit/module file

The description of the the control unit and the name of the current module file are shown at the beginning of the line.

#### PI

PI refers to the position of the cursor in the edit field and indicates the number of the **program instruction line**.

#### RG

RG refers to the position of the cursor in the edit field and indicates the number of the **program branch**.

#### Replace/Insert

The function of the editor is shown by **Replace** or **Insert**. Toggle:



#### IL mode

The end of the line indicates the operating mode in which the editor is working. The following operating modes are available, see also section 3.4 Module file editor LD and section 3.5 Module file editor FUD:

- **IL mode**
- **LD mode: On**
- **LD mode: Off**
- **FUD mode**

### 3.3.3 Block

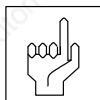
The commands are used to copy, move and delete file blocks.

#### Menu structure

- Block
  - Start
  - Delete marker
  - Store
  - Delete
  - Delete block? Yes / No
  - Store and delete
  - Copy

The size of the block may not exceed the entire module file.

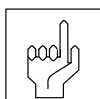
#### Store



The block marked by the block start marker and the current cursor position is stored in a **buffer**. This block can also be copied to another file.

**Only one block can be stored in a buffer at any one time. Further storing overwrites the contents of the buffer. The buffer is cleared when you exit the editor!**

#### Delete



The block marked by the block start marker and the current cursor position is **deleted** after an additional inquiry.

**The block is not stored in the buffer!**



**Store and delete**



The block marked by the block start marker and the current cursor position is deleted from the file **and stored in the buffer**.

**Only one block can be stored in the buffer at any one time. Further storing overwrites the contents of the buffer. The buffer is cleared when you exit the editor!**

**Copy**

The **contents of the buffer** are inserted at the current cursor position. The contents of the buffer **are thus retained**. The Copy command can therefore be used several times in succession.

It is also possible to copy the contents of one **file type** to a file of another file type. Fig. 3–11 shows the permitted copy operations.

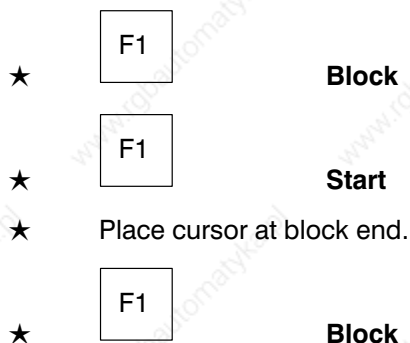
Copy possible from	to		
	Module file	Symbol file	Text file
Module file	yes	no	yes
Symbol file	no	yes	yes
Text file	yes	yes	yes






Fig. 3–11 Copying

**Copying between different files**

A block has to be copied from one file to another.

Store block:






- ★  **Store**
- Load another file:
- ★  **Change**
- ★  **Display/load**
- ★  **Module file**
- or
-  **Text file**

 file name:

 *Text.txt*

★ Place cursor at the desired position in the new file.

- ★  **Block**
- ★  **Copy**

 **Help.**





### 3.3.4 Assign

When assigning, the **symbolic** and **absolute** operands (addresses) from the module files and the symbol file are assigned to one another.

The module files are additionally searched for errors and the lines with errors are marked, see subsection **3.3.1 Edit field**.

The **file type** of the module file is **.PxT** before assignment.

After assignment the module file has the file type **.PxO**.

#### Menu structure

- Assign
  - Current module file
    - Priority symbols
    - Priority absolute values
    - Generate library module
  - According to symbol file (not PIC/PC400/CL100)
    - Priority symbols
    - Priority absolute values
  - According to batch file (not PIC/PC400/CL100)
    - Priority symbols
    - Priority absolute values

## Generate library module



In contrast to a normal module file .PxO, a library module consists solely of one compact, operable machine code. The **library module .PxL** must no longer be assigned and is transferred to the program file during linking.

### **Library modules lead to considerable time—saving when linking.**

Library modules can be protected against user intervention.

The library module **must be independent of a symbol file**; symbolic operands may only be used when they have been specified via DEF instructions in the library module. The input and output parameters are defined by the parameter list.



### **See also sections 3.7 Parameter list/Module file description and 3.8 Module library.**

Following a non—errored generating sequence, a library module .PxL is formed in addition to the .PxO module file. Both modules are interlinked with the same creation date to distinguish between versions.

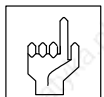
The library module .PxL can be copied into a special subdirectory **BOSCH.BIB**. In this way, the module can be set to **encompass more than one project**.



### **If the same file is present in the subdirectory BOSCH.BIB and in the project directory, the file in the project directory has priority.**

If a .PxL file and a .PxO file with the same date are present in the same subdirectory, the PxL file is always used.

By copying the .PxO file into another directory and then deleting the .PxO file from the current project directory, the .PxL file is **sealed**.



### **If the .PxO file is deleted the .PxL file can no longer be edited!**

**According to batch file**

If the assignment **according to batch file** is selected, all the modules entered in the batch file are assigned. Please see section **3.11 Batch file**.

**Priority symbols**

This is the standard case when assigning. The absolute operands from the symbol file are assigned to the symbolic operands in the module file.

**Priority absolute values**

The corresponding symbolic operands from the symbol file are assigned to the absolute operands in the module file.

## 3.3.5 Symbol

These commands relate specially to the symbolic operands.

### Menu structure

- Symbol
  - Insert (symbol file)
    - Address:
    - Symbol:
    - Comment:
  - Compact/expanded (symbol file)
  - Edit data module header (only in data module)
    - Data module name:
    - Comment:
    - EPROM RAM
    - DM length

### Insert symbol file

This command inserts symbolic operands in the symbol file. This takes place when editing the module file, **without** the symbol file being displayed on the screen.

Example

Address:	I2.0
Symbol:	Elevup01
Comment:	Limit switch elevating platform up station 1

### Compact/expanded

Toggle between the compact and expanded display of the symbol file forms. In the compact display, only the occupied lines of the forms are represented. In the expanded display, all possible absolute addresses for that controller type are represented.

### Edit data module header

See Data module header after Fig. 3–36.



F10

**Help.**





### 3.3.6 Search

#### Menu structure

- Search
  - Start of file
  - End of file
  - line string
    - line string:
    - Only whole words
    - Ignore UPPER/lower case
    - Backwards
  - Next (repeat)
  - Absolute address (only symbol file)
    - Address:
  - Error line
  - Program instruction number (PI)
    - No.:

#### line string

The system searches for the line string after the current cursor position. Additional search criteria can be selected in an additional menu.

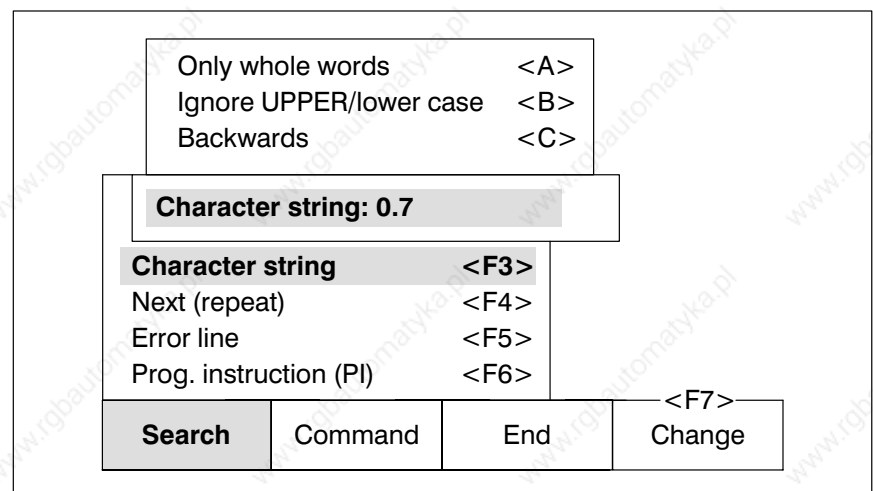
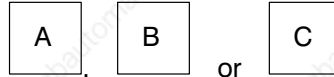


Fig. 3–12 Search criteria

Use the keys



to select the search criteria. The keys have a **second function**. In the default status the three search criteria are **not** active and the **contrary** applies:

- The system also searches for the string as **part** of a word, e.g.: line string: **Bus**  
the system finds: **Bus**master, **Bus**, **Bus**station, PROFIBUS
- The search function is also sensitive to **upper** and **lower case** in the string, e.g.:  
line string: **Bus**  
the system finds: **Bus**master, **Bus**, **Bus**station  
it does not find: **BUS**, Autobus, PROFIBUS
- The system searches for the string after the current cursor position **to the end of the file**.

When you press one of the three keys the search criterion is selected and identified by a bar.

#### Whole words only

Only strings which are preceded and followed by a **blank space** or a **tab** are found, e.g.:

line string: **Bus**

the system finds: **Bus**, **BUS**, **bus**

it does not find: **Bus**master, PROFIBUS

#### Ignore UPPER/lower case

The use of upper and lower case lines in the line string is **ignored** during the search, e.g.:

line string: **Bus**

the system finds: **Bus**master, PROFIBUS, Autobus

#### Backwards

The system searches for the line string from the current cursor position towards the **start of the file**.



### 3.3.7 Command

With these commands you can change the screen display.

You can choose between the following displays:

- **Instruction list,**
- **Ladder diagram** and
- **Function diagram.**

For

- **Networks** and
- **Data modules**

you can choose between

- **Overview** and
- **Detail**

With the command **Parameter list**

- the **parameter list** and
- the **module file description** are created,
- and a **module call** is generated.

The command **Screen mode** is used to toggle the screen display between

- **25 lines** and
- **43/50 lines.**

## Menu structure

- Command
  - Network
    - Edit title
    - Insert
      - Before current network
      - After current network
    - Delete
 

Network not empty! Delete network      Yes/No
    - Disconnect
    - Connect
      - To preceding network
      - To following network
  - Display display
 

Sym. operand / abs. operand
  - Parameter list
    - Edit
 

Version:
    - Call
 

with module description  
without module description

Module call:  
File name:
  - Overview/detail
  - Instruction list
  - Ladder diagram (not PIC)
  - Function diagram (not PIC)
  - Screen mode
 

50/43 lines    25 lines

**Network**

Module files can be divided into as many clear networks as desired.



**See also section 3.6 Network overview.**

**Edit title**

The **network title** is displayed in the **network overview** and in the module editor and monitor, and is printed in the PLC program documentation.

**Edit parameter list**

A parameter list is created for the current module file. The parameter list contains the **number of parameters**, the **symbolic operands** and their **type**.



**See also section 3.7 Parameter list/Module file description.**

**Call up parameter list**

With this command, a **module call** for a module in the line before the current cursor position is written. All the module call commands of the current controller are available for the module call.

**Overview/detail**

Toggle between

- **Overview** and
- **Detail**

for

- **Network display** in **IL**, **LD** or **FUD** and
- **Data modules**



**See also section 3.6 Network overview and section 3.9.2 Data module.**

**Screen mode**

Switches the screen display over to another operating mode. In the standard setting, **25 lines** are displayed on the screen. With this switchover, **43** or **50 lines** can be displayed depending on which video card your programming unit is equipped with.

## 3.3.8 End

Using the commands in this menu you can

- return to the defaults
- or
- change to another utility.

### Menu structure

- End
  - Exit
  - Editor
  - Monitor (not PIC)
  - Loader
  - Lister
  - Loader + Monitor (not PIC)

### Exit

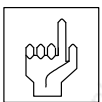
Use the command Exit to exit the utility and call up the **defaults**.

### Editor, Monitor, Loader, Lister

These commands switch you directly to another utility. The current defaults are retained.

### Loader + Monitor

When testing and changing a PLC program a special sequence results from the command **Loader + Monitor**. This function takes you quickly from the editor to the monitor.



**Only the lines which have been changed are reassigned, i.e. changes in the symbol file and DEF instructions are not effective!**

**The program file .PxA remains unchanged. The program file is only updated when it is linked!**



### Data modules cannot be reloaded.

### The controllers PC400 and CL100 must be stopped for re-loading.

The following sequence results from this command when changing a PLC program.

You are in **Monitor** and are testing a module. You discover an error and change to error elimination in the editor.

- ★ 

F6
----

**End**
- ★ 

F2
----

**Editor**
- ★ The cursor position within the module is retained. ■
- ★ Edit the line with the error.
- ★ 

F6
----

**End**
- ★ 

F6
----

**Loader + Monitor**

The module file is saved, assigned and loaded in the background.

## 3.3.9 Change

This menu offers commands for **storing** the file and changing between the **module file**, the **symbol file** and the **text file editor**.

### Menu structure

- Change
  - Save
  - Display/load
    - Module file  
file name:
    - Symbol file  
file name:
    - Text file  
file name:



F10

**Help.**





### 3.4 Module file editor LD



#### Does not apply to PIC.

Call up module file editor LD:

★  **Command**

★  **Ladder diagram**

In LD mode the editor differentiates between two operating modes

- **LD mode: On**
- **LD mode: Off**

To toggle between the two operating modes:

+

#### LD mode: On

In this operating mode the screen display is switched over to the **ladder diagram** display as soon as the cursor is positioned at a branch which can be represented in the ladder diagram.

At the same time, the **function key bar** also changes. In the function key bar special LD commands are displayed.

#### Return

★  **Return**

The display of the function key bar is thus switched over so that the commands which can be selected for the IL display (e.g. Assign) are also available in the LD display.

★

You thus switch back to the special LD function key bar.

#### LD mode: Off

In this operating mode the **instruction list** display is retained until the **ladder diagram** display is selected again. In this operating mode all those program entries which are not possible in the ladder diagram display (comment lines, word commands,...) can be made.

### 3.4.1 Edit field

The ladder diagram is divided into individual **display fields** via connection crosses.

In this auxiliary grid the cursor is moved with the cursor keys. If the cursor is positioned on a connection cross, you can change the direction from horizontal to vertical and vice versa. If the cursor is positioned vertically between two superimposed connection crosses only connections can be drawn.

A display field contains all the indisplayion pertaining to a contact or an output command. Inputs via the **function key bar** (normally closed contacts, normally open contacts, connections,...) are only permitted when the cursor is in a display field.

There are a maximum of 12 \* 8 display fields available for one branch:

- 7 \* 12 **contact fields** (normally closed contacts, normally open contacts and connections)
- 1 \* 12 **output fields** in the right-hand column

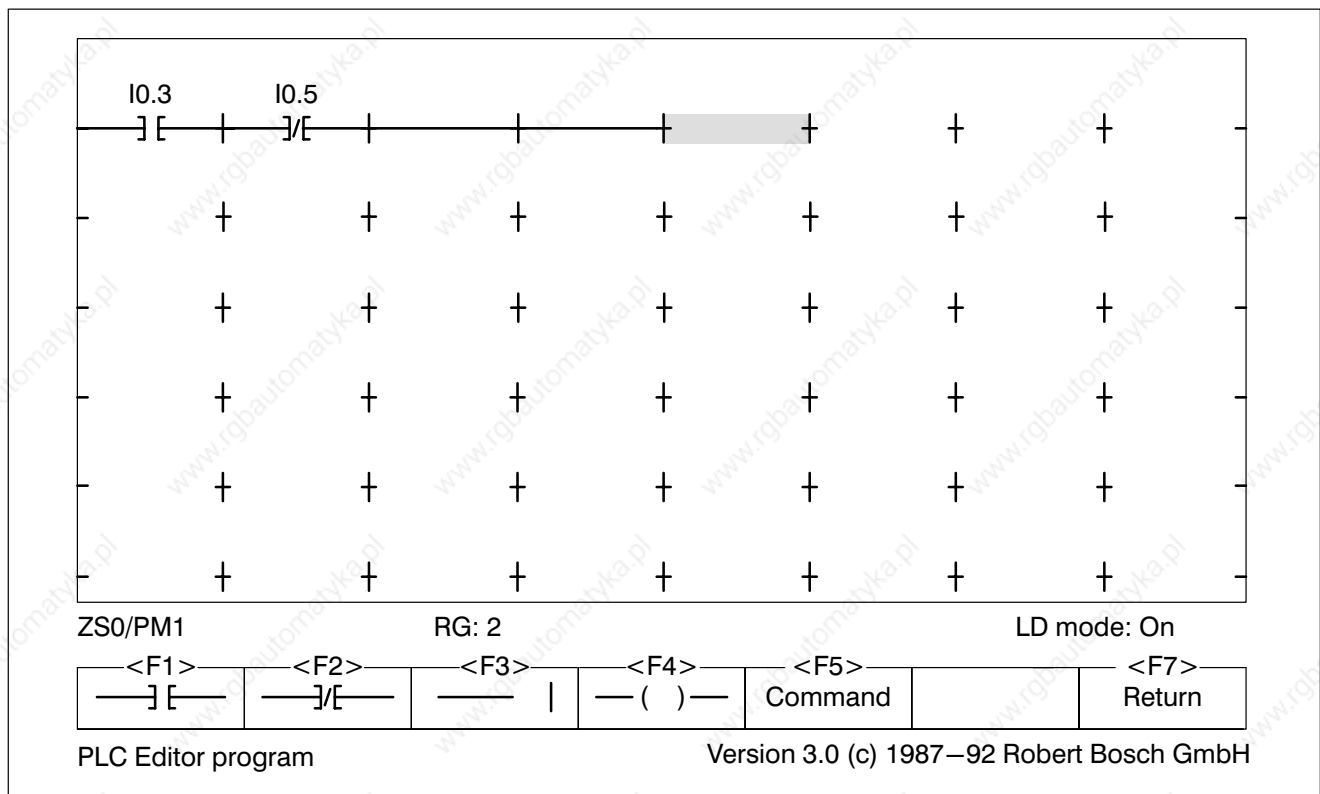


Fig. 3-13 Ladder diagram



### Error messages

Error messages during testing or translation of a branch are displayed as plain text in the bottom line of the screen.

Operand and assigner errors are displayed above the contact symbol for normally closed and normally open contacts, and to the right of the command abbreviation for outputs, see also subsection 3.3.1 **Edit field**.

### Rules for LD

- The **beginning of a branch** is in the top left corner of the ladder diagram field.
- The **end of the branch** is in the top right corner of the ladder diagram field.
- The first line must be drawn **continuously**.
- Output commands can only be programmed in the **right** column.
- The current flow through the contacts must always be **from left to right**.
- Short-circuited contacts are **not permitted**.
- Bridge circuits (Fig. 3–14) are **not permitted**.

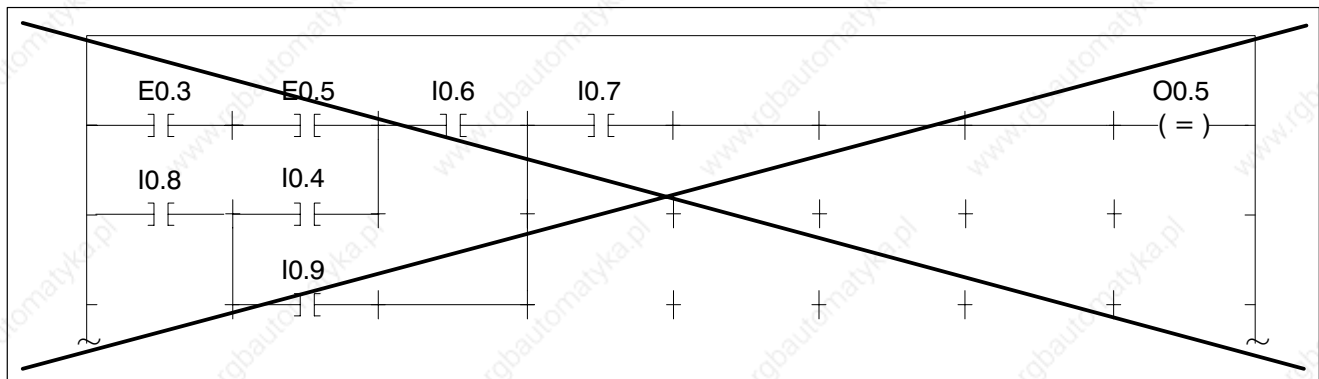


Fig. 3–14 Forbidden bridge circuit

- A **symbolic address** can be represented in the ladder diagram by a maximum of 8 lines plus a hyphen as a symbol identifier.
- A **function module with parameters** can only be programmed at the **end** of an LD branch. The number of parameters is specified after the module name and separated by a comma. The associated parameters belong to the beginning of the next branch and must be entered in IL mode.
- A branch **cannot** be exited if it could not be translated without error.

- During the translation process, there can be minor **differences** in the ladder diagram. E.g. additionally inserted redundant filler fields are no longer displayed. However, the differences only concern the display format, and the logical function of the ladder diagram network is maintained. Optimisation according to the rules of Boolean algebra does not take place.

### 3.4.2 Normally open/closed circuits

—] [— —] [—

These commands can only be selected if the cursor is in a display field. After entering the operand and

Enter

the contact appears in the display field.

#### Menu structure

- —] [—  
Operand:
- —] [—  
Operand:

### 3.4.3 Connection

— |

Connections are shown when the cursor is positioned vertically or horizontally between two connection crosses. Contacts can be overwritten with this command.



F10

**Help.**



### 3.4.4 Output commands

—( )—

The function key opens a pull-up menu for the output commands. These commands can only be selected if the cursor is in the right-hand column.

#### Menu structure

- —( )—
  - Equal to =  
Operand:
  - Set S  
Operand:
  - Reset R  
Operand:
  - Time
    - Pulse SP (not CL100)  
Timer loop name:  
Time value: (not PC400)
    - Extended pulse SPE  
Timer loop name:  
Time value: (not PC400)
    - Start-up delay time SR (not PC400/CL100)  
Timer loop name:  
Time value:
    - Storing SR SRE (not PC400/CL100)  
Timer loop name:  
Time value:
    - Turn-off delay time SF (not PC400/CL100)  
Timer loop name:  
Time value:
    - Stop TH (only CL300/CL500) ■  
Timer loop name:
    - Reset RT (not PC400/CL100)  
Timer loop name:

- Counter
  - Set counter SC (not PC400)  
Counter name:  
Counter value:
  - Count up CU (not PC400)  
Counter name:
  - Count down CD  
Counter name:
  - Reset counter RC (not PC400/CL100)  
Counter name:
- Branch
  - Branch on RLO = 1 JPC  
Branch destination:
  - Branch on RLO = 0 JPCI (only CL300/CL500)  
Branch destination:
- Module (not PC400/CL100)
  - Module call on RLO = 1 CMC  
Module name:
  - Module call on RLO = 0 CMCI (only CL300/CL500)  
Module name:
  - End of module on RLO = 1 EMC

## Timer CL300/CL500

Time commands in the ladder diagram create **two** IL commands for the CL300/CL500.

## Time/Counter PC400

In the PC400 the timer and counter values are entered via the display and **control panel** of the **time/counter module**.

## Counter CL100/CL300/CL500

Counter commands in the ladder diagram produce **two** IL commands for the CL100/CL300/CL500.



### 3.4.5 Command

By pressing this function key you receive special commands for editing the LD display.

#### Menu structure

- Command
  - Edit help
    - Line creator
    - Eraser
    - Shift right
    - Shift down
  - Branch
    - Delete
    - Check
    - Insert

#### Line creator

The **line creator** draws horizontal and vertical connecting lines. It overwrites contact symbols. The line creator is de-activated when reselected or with the **eraser** command.

#### Eraser

The **eraser** deletes connections and contact symbols by moving the cursor. The **eraser** function is de-activated when reselected or when the function **line creator** is called.

#### Shift right

The command **Shift right** moves the right-hand part of the branch one display field to the right, based on the current cursor position. The maximum number of 7 contact fields per row cannot be exceeded.

#### Shift down

Based on the current cursor position, the row in which the cursor is situated is shifted **down** one position.

## Delete



## Check

## Insert

The current branch is deleted. The respective **comment** is also deleted. If the last branch of a module is deleted an empty branch appears on the screen.

**The branch is deleted without further query!**

The current branch is examined for **errors**. Errors are displayed in plain text in the last line on the screen.

An empty branch is inserted next to the **current** branch.

### 3.4.6 Return

★

F7

#### Return

Here the display of the **function key bar** is switched over so that the other editor commands are also available in the LD display.

★

Esc

You can thus switch back to the special LD function key bar.





### 3.5 Module file editor FUD



#### Does not apply to PIC.

Call the module file editor FUD:

★  **Command**

★  **Function diagram**

When the module file editor FUD is called up the function key bar is changed. Special FUD commands are displayed in the function key bar.

#### Return

★  **Return**

Here, the display of the **function key bar** is switched over so that the other editor commands are also available in the FUD display.

★ 

You can thus switch back to the special FUD function key bar.

**Rules for FUD**

A network **cannot** be represented in the function diagram,

- if the network contains more than one **program branch**.
- if an IL command of the network cannot be represented in the function diagram, e.g. **EM, JP**,...
- if the display of the network in the function diagram encompasses more than 50 lines and 80 columns.
- if the network contains **line comments** or **scattered comment lines**. **Leading** comment lines are permitted.

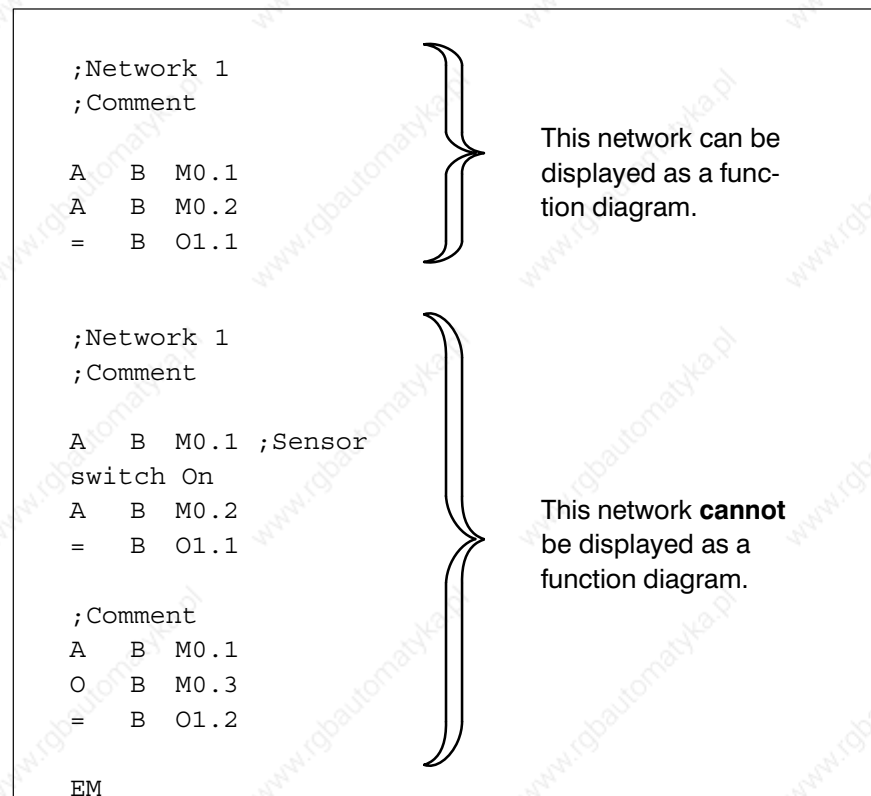


Fig. 3–15 Network and function diagram

If the current network cannot be displayed as a function diagram, it is displayed in the instruction list.



### 3.5.1 Edit field

The cursor can only be moved in an already edited area of the network. If no element of the network has been edited as yet, the cursor cannot be moved. First, an element must be created with

- F1                      &
- or
- F2                      >=1
- or
- F4                      =0-
- ★ F1                      &

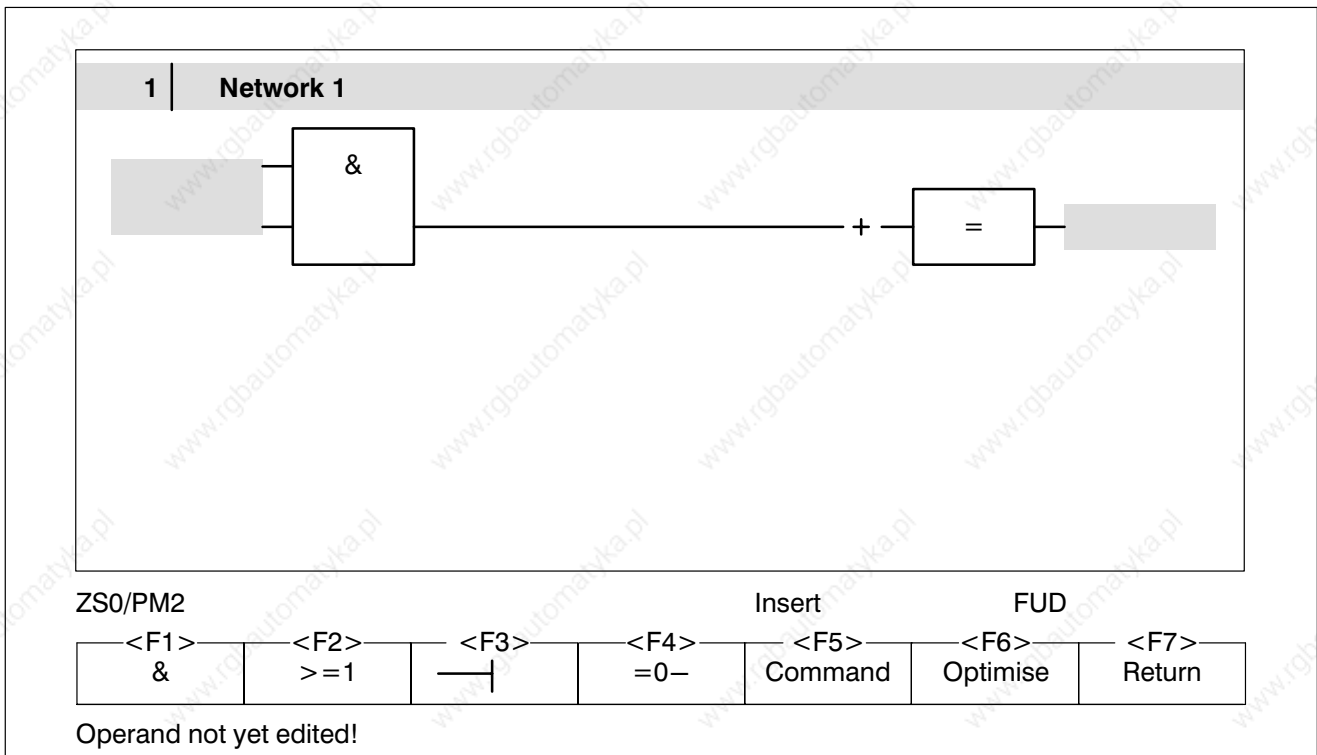


Fig. 3-16 Function diagram with & element

The first element is placed in the top left corner. The network is closed automatically with an **output element**.

In the function diagram there are several fields in which certain functions are executed. If the cursor is in one of these fields, the field will be displayed inversely. The following fields exist:

- **Operand field**
- **Input pin**
- **Element field**
- **Connection line**
- **Docking point**

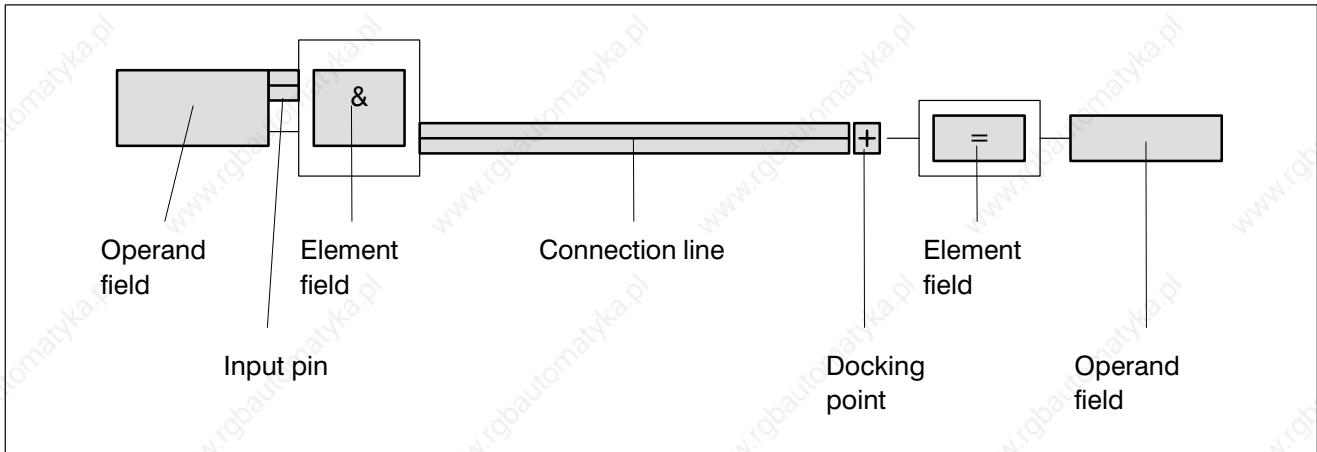


Fig. 3–17 Fields in the function diagram

### Operand field

In the operand field the **symbolic** or **absolute operand description** is entered, e.g.:



–Input1

If the operand description is **not** free from error, an error message is displayed in the message line and the cursor branches to the first line of the operand description. With a renewed attempt it is possible to exit the operand field.

Operand fields which have **not** been edited, or which have been **edited** incorrectly are displayed **inversely**. In the IL display these operands are identified by question marks.





A new element is displayed with all operand fields, including optional ones. **Optional operands** are inputs and outputs which are available for **time** and **counter elements** but which do not have to be connected. If an optional operand field is left empty the input or output pin is deleted. The inverse display of the operand field is cancelled.

If the cursor is set to an **empty optional operand field** or to the pertaining **input** or **output pin**, the operand field or the input/output pin appears once more.



With an optional operand which has not yet been edited but for which an input/output pin is available, the corresponding command sequence appears in the **IL display**. The operand is represented by question marks ????.

Branching in the **operand fields** with the cursor:

-  The cursor branches from one operand field to the next. Non—edited optional operand fields can also be branched in this way.
- and
-  The cursor only branches to the operand fields which have not been edited or which have been incorrectly edited.

### Input pin

Here, the following functions are possible, with a **bit input**:

- **Delete input pin**   
(only with AND and OR)  
An input pin can only be deleted when it is **not** connected to an element. If only two input pins remain present they can usually not be deleted.  
**Exception:**  
An AND element can also be operated with only one input pin if it is the only element on the input page.
- **Insert input pin**   
(only with AND and OR)



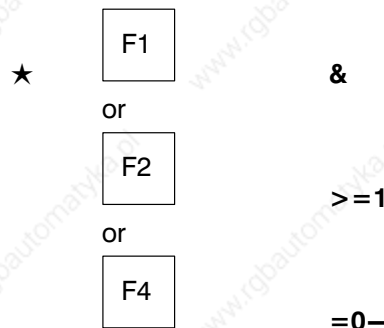
- **Negate input pin**

The **bit input** of an AND or OR element is negated. The input pin must be connected to an **operand**.

**Exception:**

With the **CL500** the input pin can also be connected to another element.

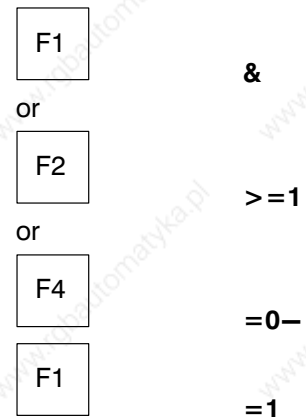
- **Insert element**



**Element field**

The following functions are possible:

- **Rename element**



An AND element can be renamed as an OR or XOR element and vice versa. An AND or OR element can only be renamed as an XOR element if it has 2 inputs which are **not** negated. An AND element with 1 input pin **cannot** be renamed.

A **comparator** can be renamed as any other comparator which is supported by the controller command set.

A **forwards counter** can be renamed as a **backwards counter** and vice versa. A **combined forwards/backwards counter** may **not** be renamed.



Each **time element** can be renamed as any other time element.

An **RS flip–flop** can be renamed as an **SR flip–flop** and vice versa.

An **output element cannot** be renamed.

A **branch or module call on RLO = 1** can be renamed as a branch or module call on RLO = 0 and vice versa.



- **Delete element**

This deletes a single element which is connected to no more than one element at an input pin.

After deletion, the output pin of the preceding element is connected to the input pin of the following element. The operands of the deleted element are also deleted. If an input pin of the element to be deleted is negated, the negation is also deleted.

The last **output element** may **not** be deleted.

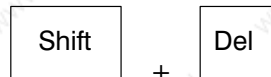
If the last input element is to be deleted, the following question is asked:



Delete whole network screen? Yes/No

If **Yes** is entered the last input element is deleted along with **all its output elements**.

- **Delete element and all preceding elements**



**Connection line**

The following functions are possible:

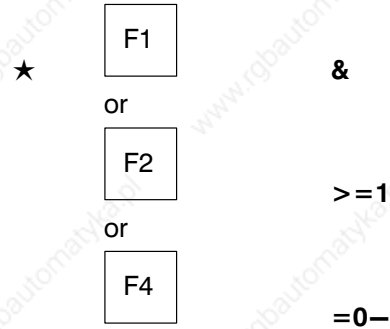
- **Negate input pin**  
(only CL500)



- **Insert another input pin**  
(only with AND and OR)



- **Insert element**



**Docking point**

In this position another **output element** can be inserted in the network.

### 3.5.2 And element &

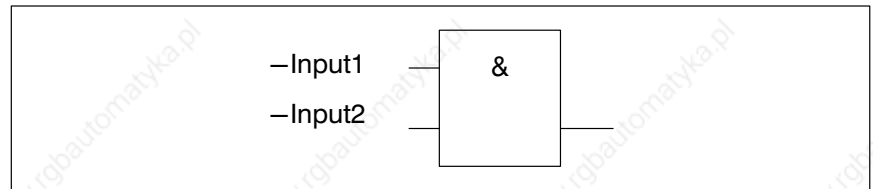


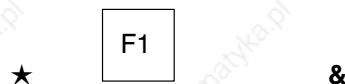
Fig. 3-18 AND element &

This function is dependent on the current cursor position:

- **Insert element**
- **Rename element**

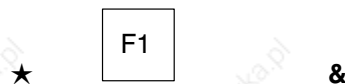
**Insert element**

Cursor is positioned at an **input pin** or at a **connection line**:



**Rename element**

Cursor is positioned in an **element field** of an OR or XOR element:







**Delete input pin**



An AND element normally has at least two inputs. The penultimate input pin can be deleted if the AND element is the only element on the input side.

**If an AND element with only one input pin is present, the insertion of additional input elements and renaming are blocked.**

Further input elements can only be inserted when a second input pin has been attached to the AND element.

**3.5.3 OR element  $\geq 1$**

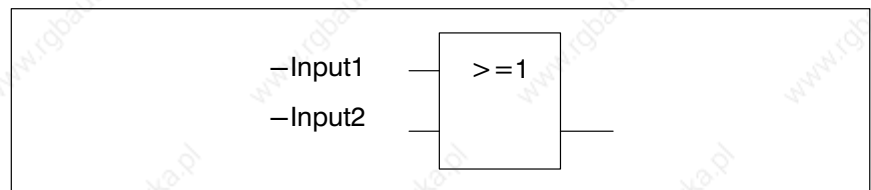


Fig. 3–19 OR element  $\geq 1$

This function is dependent on the current cursor position:

- **Insert element**
- **Rename element**

**Insert element**

Cursor is positioned at an **input pin** or at a **connection line**:



**Rename element**

Cursor is positioned in the **element field** of an AND or XOR element:



**An AND element to be renamed must have at least 2 inputs.**

## 3.5.4 Input pin –|

The cursor must be at an input pin.

Input pins are

- **inserted**  
(only with AND or OR elements)
- **deleted** or  
(only with AND or OR elements)
- **negated.**

### Menu structure

- –|
  - Insert
  - Delete
  - Negate

## 3.5.5 Complex elements =0–

### Menu structure

- =0–
  - XOR
  - Comparison
    - Equal
    - Unequal
    - Greater
    - Greater or equal to
    - Smaller
    - Smaller or equal to
  - Flip–flop
    - with dominating set input
    - with dominating reset input



- Time
  - Pulse SP (not CL100)
  - Extended pulse SPE
  - Start-up delay time SR (not PC400/CL100)
  - Storing SR SRE (not PC400/CL100)
  - Turn-off delay time SF (not PC400/CL100)
- Counter
  - Count up CU (not PC400)
  - Count down CD
  - Counter up/down CU&CD (not PC400)
- Equal to =
- Branch
  - Branch on RLO = 1 JPC
  - Branch on RLO = 0 JPCI (only CL300/CL500)
- Module (not PC400/CL100)
  - Module call on RLO = 1 CMC
  - Module call on RLO = 0 CMCI (only CL300/CL500)
  - End of module on RLO = 1 EMC

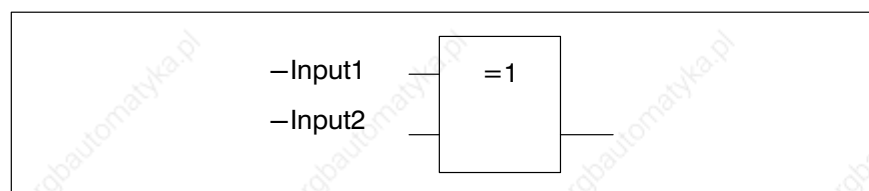
**XOR**

Fig. 3-20 XOR element

An **XOR element** can be inserted or, through renaming, can consist of an AND or OR element. An AND or OR element can only be renamed as an XOR element if it has exactly **2** inputs which are **not** negated and **not** connected to other elements. An XOR element **cannot** be inserted between two other elements. The XOR element must **not** be an output element.

**Comparison**

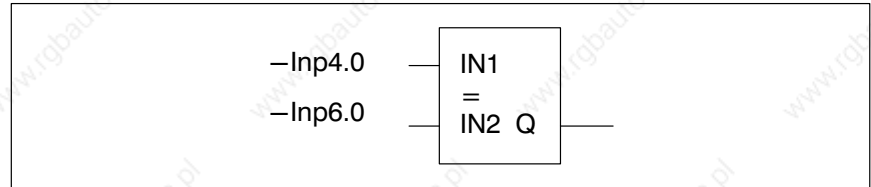


Fig. 3–21 Comparator

A **comparator** can be inserted or renamed. A comparator can only be inserted at an **input pin**. The inputs of the comparator are **word inputs** which must be occupied by an **operand**. If the comparison is fulfilled, the output bit **Q** is set.

**With some controllers not all comparators are available.**

A comparator must **not** be an output element.

**Flip–flop**

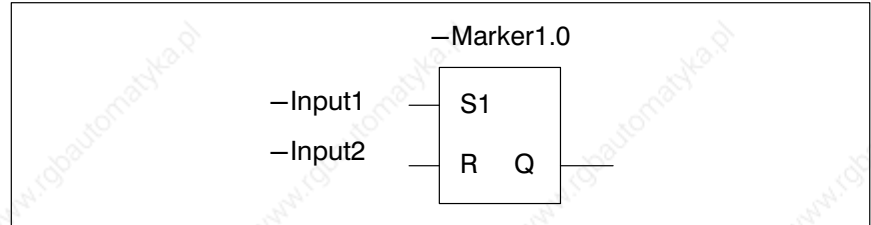


Fig. 3–22 SR flip–flop

A flip–flop element can be inserted or renamed. Two types are available for selection:

- **with dominating set input SR flip–flop**
- **with dominating reset input RS flip–flop**

If the flip–flop is used as an output element, then the priority input pin is always connected to the docking point.

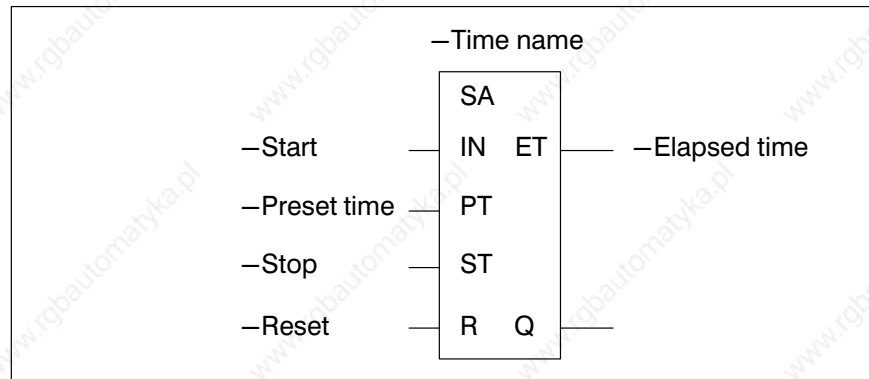
**Time**

Fig. 3–23 Time element

The following abbreviations are used:

- **SP** Start time as pulse
- **SPE** Start pulse extended
- **SR** Start time as raising delay
- **SRE** Start time as raising delay extended
- **SF** Start time as falling delay
- **IN** Input Bit
- **PT** Preset Time Word
- **ST** Stop Time Bit optional
- **R** Reset Bit optional
- **ET** Elapsed Time Word optional
- **Q** Output Bit

If the time element is used as an output element, then the upper input **IN** is connected to the docking point.



**With some controllers not all connections are available.**

**Counter**

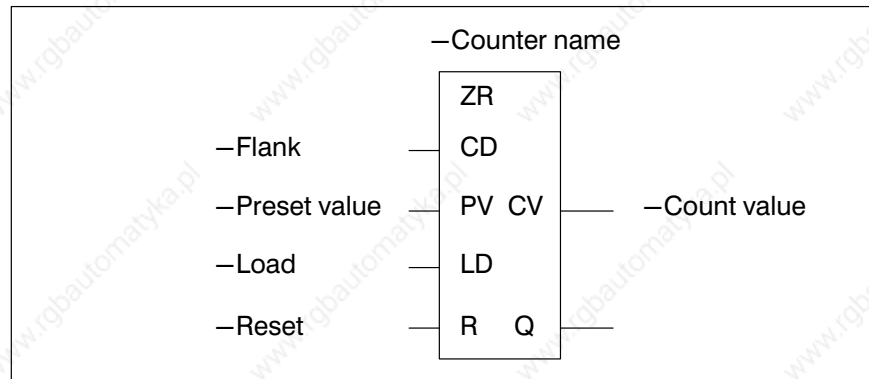


Fig. 3–24 Counter element

The following abbreviations are used:

- **CF** Counter forwards
- **CB** Counter backwards
- **CF&CB** Counter forwards/backwards
- **CU** Count Up Bit
- **CD** Count Down Bit
- **PV** Preset Value Word optional
- **LD** Load Bit optional
- **R** Reset Bit optional
- **CV** Count Value Word optional
- **Q** Output Bit

If the counter element is used as an output element, then the upper input **CU/CD** is connected to the docking point.

**With some controllers not all connections are available.**



**Allocation**

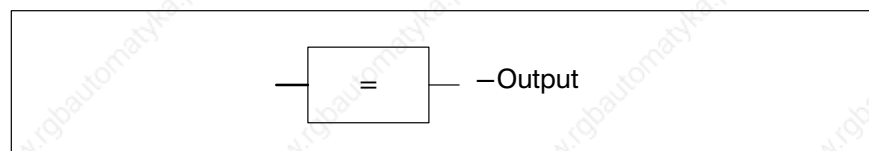
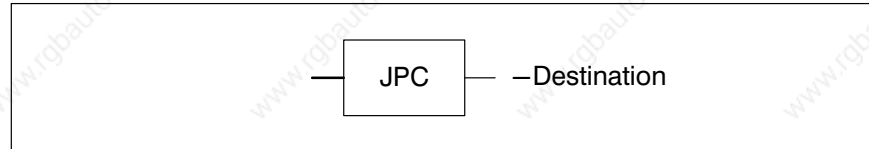


Fig. 3–25 Allocation

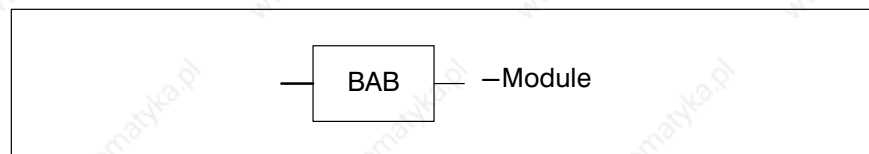
The allocation transfers the result of the operation to an operand. So long as the operations are fulfilled, the operand remains set.

**Branch***Fig. 3–26 Branch*

There are two branch commands:

- **JPC** Branch when input condition **is fulfilled**,  
RLO = 1
- **JPCI** Branch when input condition is **not fulfilled**,  
RLO = 0

The program line with the **branch destination** can only be entered in the IL editor.

**Module***Fig. 3–27 Module call*

Three module commands are available for selection:

- **CMC** Module call when input condition **is fulfilled**,  
RLO = 1
- **CMCI** Module call when input condition is **not fulfilled**, RLO = 0
- **EMC** End of module when input condition is **fulfilled**,  
RLO = 1

## 3.5.6 Command

See also subsection **3.3.7 Command**

### Menu structure

- Command
  - Network
    - Edit title
    - Insert
      - Before current network
      - After current network
    - Delete
 

Network not empty! Delete network? Yes/No
    - Disconnect
    - Connect
      - To preceding network
      - To following network
  - Delete screen
 

Delete whole network screen? Yes/No
  - Overview/detail
  - Instruction list
  - Ladder diagram
  - Function diagram

### Delete screen

The command deletes the entire **network contents**. The **network title** and the network are retained.





### 3.5.7 Optimise

With this command the network is optimised.

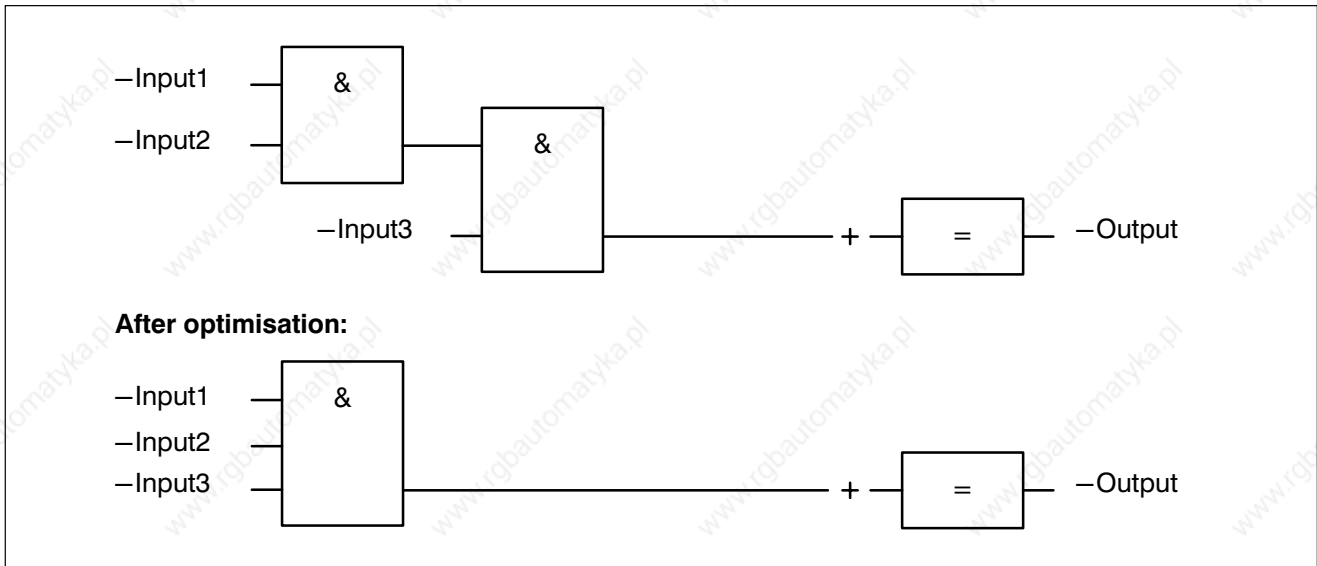


Fig. 3-28 Optimisation



**When exiting the network optimisation takes place automatically.**

### 3.5.8 Return

With

F7

**Return**

the display of the **function key bar** is switched over so that other editor commands are also available in the FUD display.

With

Esc

you switch back to the special FUD function key bar.

### 3.6 Network overview

A module file consists of several networks. A network consists of several consecutive program lines. A network can encompass several program branches and its maximum size equals that of a module file.

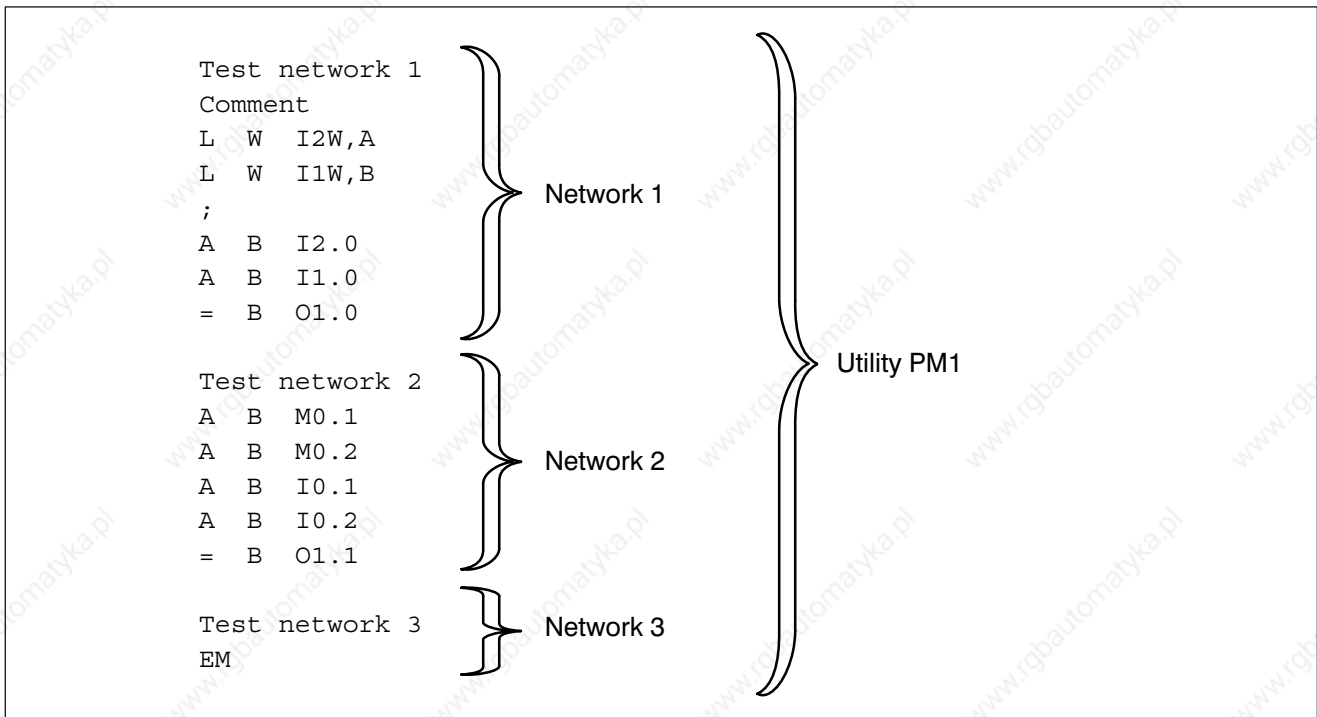
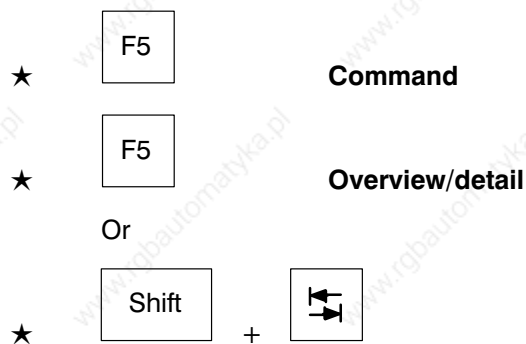


Fig. 3-29 Network

In the module editor only one network is ever shown on the screen.

#### Network overview/detail



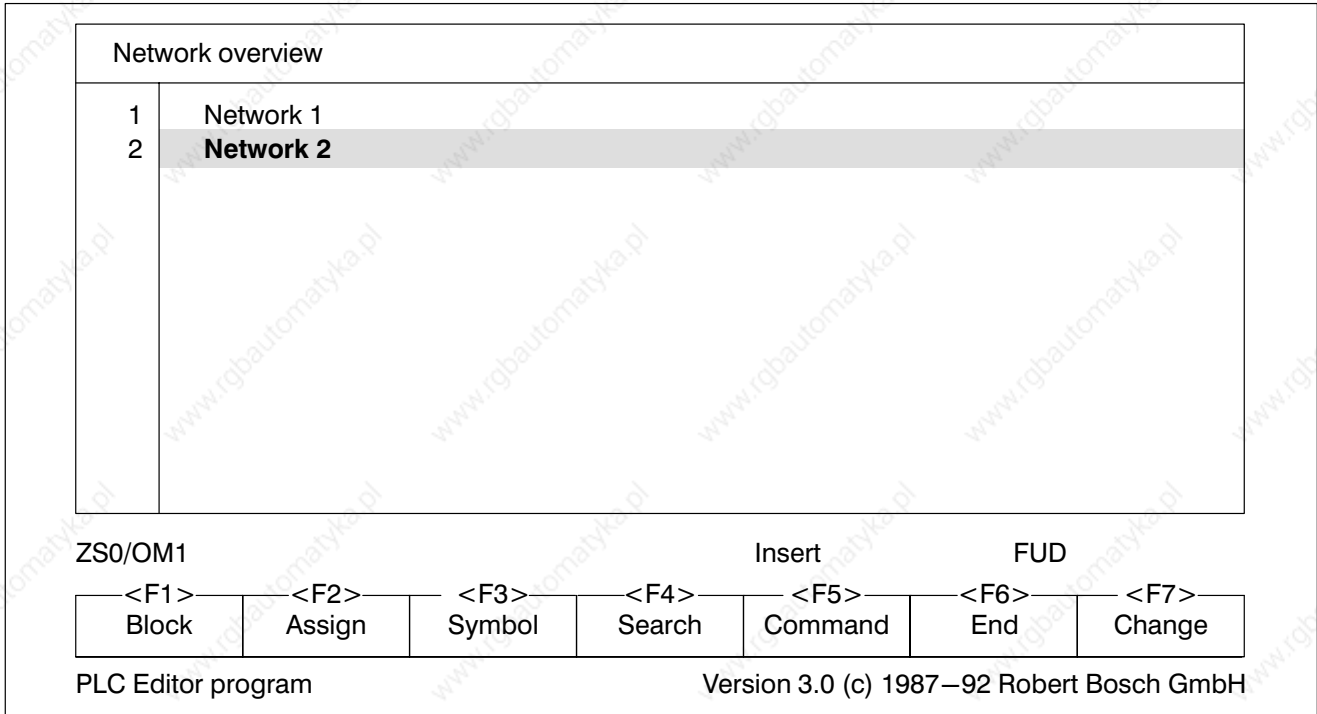


Fig. 3-30 Network overview

The network overview serves as a contents list of the module file. In the network overview you can:

- **edit the network title,**
- **insert a network,**
- **delete a network,**
- **copy a network and**
- **call up a network.**

**Edit network title**

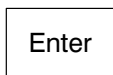
- ★ Move the bar cursor to the desired line using



or



or



- ★ Enter the network title.

## Insert network

If the cursor is positioned at the **first** line of a network title, a new network is inserted **before** the current network with

Enter

If the cursor is **not** positioned on the first line of a network title, a new network is inserted **after** the current network with

Enter

## Delete network

★ Select the network to be deleted with

↓

or

↑

Shift

+

Del

★



**The last network cannot be deleted.**

If, however, you wish to delete the last network, first delete the contents of the network and then connect the empty last network to the penultimate network.

## Copy network



Block commands are available for copying networks. The block commands always relate to the network as a whole.

**See subsection 3.6.1 Block.**



**Call up network**

★ Select the network with



or



★



**Command**

★



**Overview/detail**

**3.6.1 Block**



**See also subsection 3.3.3 Block**

**Only one block at a time can be stored in the buffer. Further storing overwrites the contents of the buffer. The buffer is deleted when the editor is exited!**

Block commands are available for copying networks. The block commands always relate to the network as a whole.

A block consists of all the networks located between the block start marker and the currently selected network.

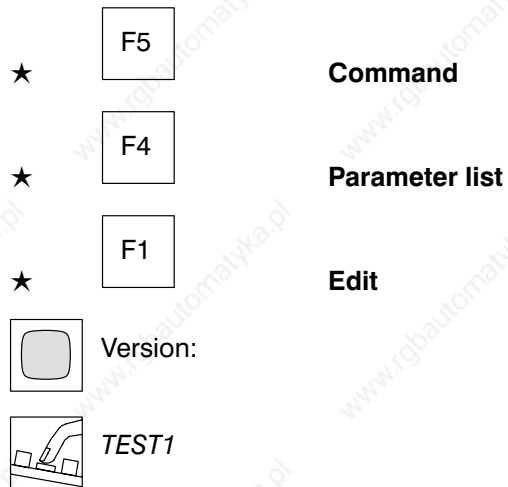
**The last network in the network overview cannot be stored or deleted.**

If, however, you wish to store or delete the last network, first create an additional empty network below this one. Then you can copy or delete the desired network. Finally, connect the empty last network to the penultimate network.

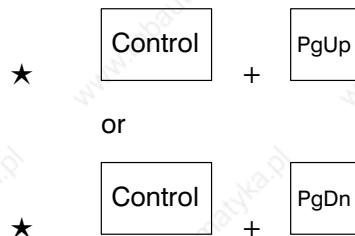
## 3.7 Parameter list/module file description

The parameter list simplifies the parameterising of module files. In the module file description the programmer can create any desired text to describe the parameter and the module file.

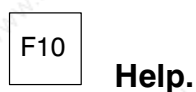
### Call up parameter list



### Switch to parameter list/module file description



### Exit module file description/parameter list





**3.7.1 Parameter list**

Para.	Ext.	Ind	Symbol	< >	Comment	Version: TEST1		
P0	<b>B</b>		<b>Input</b>	<b>&lt;</b>	<b>ON-switch</b>			
Parameter list						Ctrl-PgUp	Ctrl-PgDn	Esc
ZS0/OM1				Insert				
Block	Assign	Symbol	Search	Command	End	Change		
PLC Editor program				Version 3.0 (c) 1987-92 Robert Bosch GmbH				

Fig. 3-31 Parameter list

With the parameter list it is possible to program parameter modules with **symbolic operands**. These can be used instead of the absolute operands P0 to Pn.

**Para. Parameter number**

This column cannot be edited. The maximum number of parameters is dependent on the controller type:

- **PC600** 32 parameters
- **CL300** 32 parameters
- **CL500** 63 parameters

**Ext. Extension**

The **parameter extension** must be entered in this column:

- **PC600** B, BL, BR, W
- **CL300** B, BY, W
- **CL500** B, BY, W

**Ind** Indirect address

If you enter **&** in this field, the symbolic operand is interpreted as an **indirect address**.

**Symbol**

In this column you enter the **symbolic operand**. This offers you the possibility of also programming symbolically in the parameter module. When assigning, the symbolic operands are regarded as DEF instructions and so they only apply locally in the parameter module.

&lt; &gt;

Identification of parameters as input or output parameters:

- <      **Input parameter**
- >      **Output parameter**

**Comment**

You may enter a comment for each parameter.





### 3.7.2 Module file description

#### Call up module file description

The module file description is called from the parameter list.



<div style="background-color: #cccccc; width: 100%; height: 20px; margin-bottom: 5px;"></div>						
Module description			Ctrl–PgUp	Ctrl–PgDn	Esc	
ZS0/OM1			Insert			
Block	Assign	Symbol	Search	Command	End	Change
PLC Editor program			Version 3.0 (c) 1987–92 Robert Bosch GmbH			

Fig. 3–32 Module file description

In the module file description any desired text can be entered for the description of the parameter module. The text can be max. 75 lines wide and 80 lines long.

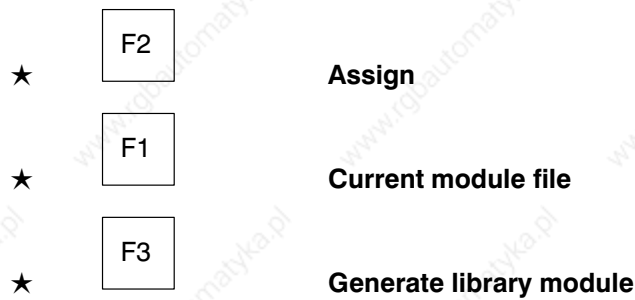
### 3.8 Module library

Most module files are dependent on a **symbol file**. Therefore, these module files can only ever be used in one project.

In order for a module file to apply to several projects, it must be created in such a way that it is independent from a symbol file.

- Only **absolute** operands may be used in the module file.
- Symbolic operands may only be used when they are defined within the module file via **DEF instructions**.
- For the transfer of input and output parameters, a **parameter list** is created, see section **3.7 Parameter list/module file description**.

From the module file a **library module .PxL** is generated.



#### Library modules lead to considerable time—saving when linking.

Moreover, these library modules must be placed in a directory which is accessible to all projects. This generally accessible directory is the **module library BOSCH.BIB**. The directory BOSCH.BIB is situated parallel to the project directories.

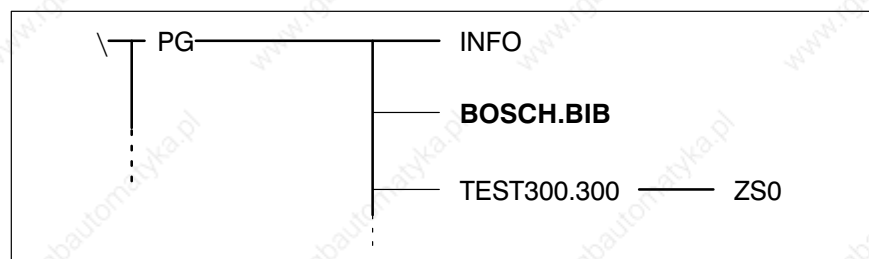









Fig. 3–33 BOSCH.BIB directory



The generated library module .PxL must be copied to the BOSCH.BIB directory.

★		<b>End</b>
★		<b>Exit</b>
★		<b>Command</b>
★		<b>Command</b>
	 C>	
	 copy <i>PM1.P3L C:\PG\BOSCH.BIB</i>	
★		

If only the .PxL file in the BOSCH.BIB directory must be accessed for further PLC programming, the PxO and .PxL files must be removed from the current project directory and stored safely in another subdirectory or on a floppy disk.

No module files can be edited in the module library .BIB. In order to edit a library module

- the library module .PxL must be copied in the corresponding project directory **.ZSx**, and
- the pertaining **.PxO file** must also be present in the directory.

It is thus possible to deny user access to the library modules by removing the corresponding .PxO file.

**If the .PxO file is deleted, the .PxL file can no longer be edited!**

**During assigning and linking, the current directory .ZSx is always searched before the BOSCH.BIB directory. If a .PxL file and a .PxO file with the same date are present in the same directory, the .PxL file is always used.**



## 3.9 Symbol file editor

The symbol file contains

- **module lists,**
- **operand forms** and
- **data modules.**

The symbol files are structured differently according to the controller selected.

Form	PIC	PC400	PC600	CL100	CL300	CL500
<b>Module lists</b>						
OM form		•	•	•	•	•
PM form			•		•	•
FM form			•			
EM form			•			
<b>Operand forms</b>						
I form	•	•	•	•	•	•
O form	•	•	•	•	•	•
M form	•	•	•	•	•	•
SM form					•	•
C form		•	•	•	•	•
T form		•	•	•	•	•
DM form			•		•	•
XI form		•				
XO form		•				
EI form			•		•	•
II form			•		•	•
EO form			•		•	•
IO form			•		•	•
DB form					•	•
F form		•			•	
S form					•	•

Fig. 3-34 Symbol file forms



F10

**Help.**



**3.9.1 Module list**

The **symbolic** module file names are entered in the forms of the module list. Fig. 3–35 shows a module list for utilities.

Type	Module name ;Comment	R/E
PM 1	OPMO;PM Operating mode	R
PM 2	PARTFLO;PM Parts flow	R
PM 3	;PARTREJ;PM Reject parts	R
PM 4	PARTTRAN;PM Parts transport	R
PM 5		R
PM 6		R
PM 7		R
PM 8		R
PM 9		R
PM 10		R
PM 11		R
PM 12		R
PM 13		R
PM 14		R
PM 15		R
PM 16		R

ZS0/SYMBOL Replace

<F1> Block	<F2> Assign	<F3> Symbol	<F4> Search	<F5> Command	<F6> End	<F7> Change
---------------	----------------	----------------	----------------	-----------------	-------------	----------------

PLC Editor program Version 3.0 (c) 1987–92 Robert Bosch GmbH

Fig. 3–35 Module list

In the module lists, individual modules can be masked by a leading **semi-cololon**. These modules are then ignored during the **assigning** and **linking** processes.

**Comment**

After the module name is a **comment**. The comment must be separated from the module name by a **semicolon ;**.

**R/E**

The controller memory in which the module is to be stored is entered.

R

**RAM**

or

E

**EPROM**

In the CL100 only RAM is possible.


**PC400/CL100**

With the controllers PC400/CL100 the name for an **organisation module** must be entered in the module list.

**3.9.2 Data module**

Data modules (not in PC400/CL100) consist of the **data module header** and the **data module form**, see Fig. 3–36.

DM	1	Name: ProdM1	Comment: Production on machine 1	RAM/EPROM: R	
No.	Symbol	Type	S	Data field	F
D 0	Set speed	Word	N	120	D
D 2	Set ctrl	Word	N	76	D
D 4	Wait at	Word	N	2	D
D 6	Set prod Total	Word	N	1256	D
D 8		Word	N		D
D 10		Word	N	; Comment line for D6: Production target no.	D
D 12		Word	N		D
D 14		Word	N		D
D 16		Word	N		D
D 18		Word	N		D
D 20		Word	N		D
D 22		Word	N		D
D 24		Word	N		D

ZS0/SYMBOL Replace 

<b>&lt;F1&gt;</b> Block	<b>&lt;F2&gt;</b> Assign	<b>&lt;F3&gt;</b> Symbol	<b>&lt;F4&gt;</b> Search	<b>&lt;F5&gt;</b> Command	<b>&lt;F6&gt;</b> End	<b>&lt;F7&gt;</b> Change
----------------------------	-----------------------------	-----------------------------	-----------------------------	------------------------------	--------------------------	-----------------------------



PLC Editor program Version 3.0 (c) 1987–92 Robert Bosch GmbH

Fig. 3–36 Data module



**Data module header**

First the data module header must be filled in.

- ★  **Symbol**
- ★  **Edit data module header**
-  *DMTest*
-  *Comment on DMTest*
- ★  **RAM/EPROM**

If the data module is stored in the **RAM area** of the controller it can be **edited** during execution of the program. If, on the other hand, the data module is in the **(E)EPROM area**, it is only possible to read the contents.



The permissible **data module length** is between 0 and 512 bytes. The input 0 means that the data module is not stored in the controller. This enables a data module which has been created to be simply masked out in the controller.

The number of data words which can be entered in the data module is not dependent on the length of the data module. Thus part of the data module can also be masked out.

The data module length is visible in the **No.** column. The associated data words are highlighted, see Fig. 3–36 data words 0 to 8.

**Data module form**

Use the following key to proceed from column to column:



**Symbol**

The symbolic name of the data word is entered in the **Symbol** column.

Symbolic names may not be used more than once. They are checked in the whole symbol file.

## Type

In the **Type** column the **data type** is specified. You can specify **ASCII** for ASCII lines or **word** for numeric data.

## S

In the **S** (sign) column, the entry of **Y** (yes) or **N** (No) specifies whether the data in the data field have a sign or not.

## Data field

The **data field** is used for entering

- **data** and
- **comments**.

Please note the following when entering data.

- For word entries, one word of data per line can be entered.
- For ASCII entries the entire data field column can be used for entries, i.e. up to a maximum of 44 bytes.

If ASCII lines beyond the range 20 H to 7F H are to be entered, the following rules apply:

- Range below 20 H:  
Either a hexadecimal value, a decimal value or the ASCII abbreviation of the control line in question can be entered. The entry must be enclosed by pointed brackets.  
Example: <0DH> or <13D> or <CR>
- Range above 7F H and below FF H:  
Either a hexadecimal or a decimal value can be entered.  
Example: <129D> or <81H> or <ü>  
The line < itself is entered by doubling <<.

Using

Enter

you can create **comment lines** before an occupied data word.

## F

In the **F** column the **data display** for the numeric data is entered. The following data displays are available.

- B binary
- D decimal
- H hexadecimal
- O octal





**Data module overview list**

★ F5 **Command**

★ F5 **Overview/detail**

The data module overview list summarizes the contents of the data module headers for all data modules. You are thus provided with a rapid overview of all available data modules.

In the data module overview list you can

- **edit** the contents of the data module headers,
- **file** new data modules and
- **call up** a data module.

DM No.	Name	Comment	R/E	Length
DM 0	<b>DMTEST0</b>	<b>Test of machine</b>	<b>R</b>	<b>20</b>
DM 1	<b>DMTEST1</b>	<b>Production data</b>	<b>R</b>	<b>0</b>
DM 5	<b>DMTEST5</b>	<b>Site data component 1</b>	<b>R</b>	<b>150</b>
DM 6	<b>DMTEST6</b>	<b>Site data component 2</b>	<b>R</b>	<b>150</b>

ZS0/SYMBOL

Insert

	<F2>	<F3>	<F5>	<F6>	<F7>
Block	Assign	Symbol	Search	Command	End
			Command	End	Change

PLC Editor program

Version 3.0 (c) 1987–92 Robert Bosch GmbH

Fig. 3–37 Data module overview list

When you exit the data module overview list, the data module marked by the bar cursor is called.

F10

**Help.**



### 3.9.3 Operand form

Fig. 3–38 shows an example input form.

Address	Symbol	Comment	Type
I 2.0	Sensor1	Sensor 1, machine On	
I 2.1	Sensor2	Rapid mode	
I 2.2			
I 2.3			
I 2.4	Key3	Manual operation	
I 2.5			
I 2.6			
I 2.7			
I 3.0			
I 3.1			
I 3.2			
I 3.3			
I 3.4			
I 3.5			
I 3.6			
I 3.7			

ZS0/SYMBOL Replace

<F1>	<F2>	<F3>	<F4>	<F6>	<F7>
Block	Assign	Symbol	Search	Command	End
				Change	

PLC Editor program Version 3.0 (c) 1987–92 Robert Bosch GmbH

Fig. 3–38 Operand form

#### Symbol

Symbolic address with max. 8 lines.

#### Type

The I, O, M, SM, EI, II, EO and IO forms have a “type” column. The type (8, 16, 32) of module present in the controller can be specified in this “type” column. The entry is made in the lines 0.0, 1.0, 2.0, etc. The address set at the module is entered here.

This specification is not mandatory. The type column may also be left empty. The entry is not checked for correctness but serves as an additional comment.



### 3.10 Text file editor

The text file editor is used to

- create or edit text files,
- create or edit batch files, or
- display and edit print files on the screen.

#### Call up text file editor

- ★ 


F7
----

**Change**
  - ★ 

F2
----

**Display/Load**
  - ★ 

F3
----

**Text file**
-  *file name.File type*

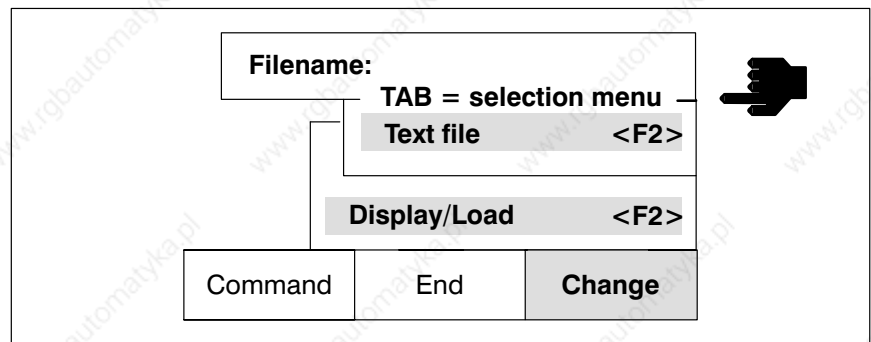


Fig. 3–39 Change menu

#### Edit text

All ASCII lines from **decimal 32** to **decimal 126** are available. ASCII lines which are not directly available on the keyboard of the programming unit are to be entered according to the following example.

The line { has to be entered.

- ★ Refer to table, Fig. 3–40 for the decimal code of the character { (123).



Keep

Alt

**depressed** and simultaneously enter the

1

2

3

decimal code **123** with the numeric keyboard.



Decimal code	ASCII line	Decimal code	ASCII line	Decimal code	ASCII line
32	SP	64	@	96	'
33	!	65	A	97	a
34	"	66	B	98	b
35	#	67	C	99	c
36	\$	68	D	100	d
37	%	69	E	101	e
38	&	70	F	102	f
39	'	71	G	103	g
40	(	72	H	104	h
41	)	73	I	105	i
42	*	74	J	106	j
43	+	75	K	107	k
44	,	76	L	108	l
45	-	77	M	109	m
46	.	78	N	110	n
47	/	79	O	111	o
48	0	80	P	112	p
49	1	81	Q	113	q
50	2	82	R	114	r
51	3	83	S	115	s
52	4	84	T	116	t
53	5	85	U	117	u
54	6	86	V	118	v
55	7	87	W	119	w
56	8	88	X	120	x
57	9	89	Y	121	y
58	:	90	Z	122	z
59	;	91	[	123	{
60	<	92	\	124	
61	=	93	]	125	}
62	>	94	^	126	~
63	?	95	_		

Fig. 3-40 ASCII line set

### 3.11 Batch file

Batch files offer many advantages for processing several files. Batch files are used for listing and assigning.

The batch file is created using the **text file editor**. The batch file contains a consecutive list of the files to be processed. A separate line is used for each file. This line must contain the file name:

file name.File type

A minimal batch file for printing or assigning can have the following forms, for example:

PM1

OM1

With this abbreviated form of the batch file, the files must be contained in the current directory.

If the data type is entered, then only the appropriate file is used. If no data type is entered and both files \*.PxT and \*.PxO are present, the file \*.PxO is used.

If a syntax error occurs when the batch file is being edited, this error is displayed in the edit log together with the number of the line containing the error.

**If a syntax error has occurred in a batch file, the file must be saved following correction before it can be reused for editing.**

**Contents**

	Page
<b>4 Monitor .....</b>	<b>4-1</b>
4.1 Monitor commands .....	4-3
4.1.1 Control .....	4-3
4.1.2 Info .....	4-6
4.1.3 Search .....	4-7
4.1.4 Command .....	4-7
4.1.5 End .....	4-8
4.1.6 Change .....	4-9
4.2 IL Monitor .....	4-11
4.3 LD Monitor .....	4-13
4.4 FUD Monitor .....	4-14
4.5 Data module .....	4-15
4.6 Operand field .....	4-17
4.6.1 Edit .....	4-20
4.6.2 Control .....	4-20
4.6.3 Display .....	4-20
4.6.4 File .....	4-21
4.6.5 Return .....	4-21

## Illustrations

Fig.		Page
4-1	Monitor commands .....	4-3
4-2	Trace line .....	4-5
4-3	Monitor line .....	4-11
4-4	LD Monitor .....	4-13
4-5	FUD Monitor .....	4-14
4-6	Data module .....	4-15
4-7	Operand field .....	4-17
4-8	Operand types .....	4-18
4-9	Operand extension .....	4-18





## 4 Monitor

The PLC Monitor program offers the following possibilities:

- Program test
- Status displays
- Program manipulation
- Setting
- Controlling
- Error status

To operate the PLC Monitor program the **EP/AG module** and the connection to the controller are necessary, see section **5.1 Connection of programming unit ↔ controller**.

### Monitor operation is not possible for PIC.



#### Monitor

The PLC monitor program is called up from the main menu by pressing

F3

#### Monitor

twice.

When the function key is pressed the first time the defaults are displayed on the screen.

Change the defaults:



See section **1.7 Entering defaults**.

The second press of the function key starts the PLC Monitor program.

The monitor differentiates between 5 displays: ■

- Module file in IL, see section **4.2 IL Monitor**
- Module file in LD, see section **4.3 LD Monitor**
- Module file in FUD, see section **4.4 FUD Monitor** ■
- Data module, see section **4.5 Data module**
- Operand field, see section **4.6 Operand field**

## Toggle between IL, LD and FUD



F5

**Command**



F6

**Instruction list**

or

F7

**Ladder diagram**

or

F8

**Function diagram**

## Display/command level

The monitor differentiates between a display level and a command level. A highlighted command in the function key bar indicates the command level.

## Toggle between display and command levels

Esc

## Call up command

F1

...

F7

## Communication with the controller

A **rotating bar** in the right-hand margin of the information line indicates whether communication with the controller is taking place and whether the states are being refreshed. If communication is interrupted because, e.g. a command is called up, the bar remains stationary. See Fig. 4-2.



F10

**Help.**



## 4.1 Monitor commands

The commands are called up with the function keys. The pull-up menus show the meaning of the function keys. In the operand field display there is a special function key bar, see section 4.6 Operand field.

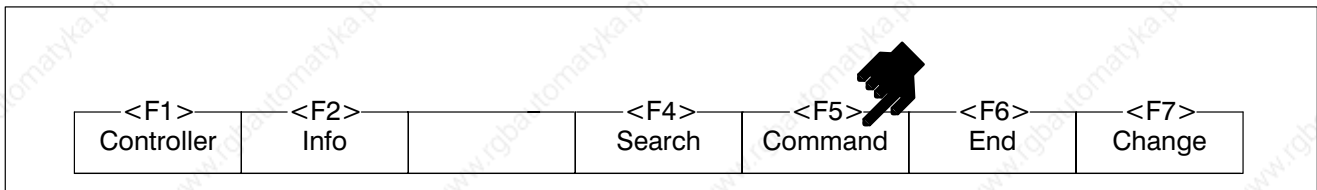


Fig. 4-1 Monitor commands



**Not all commands are available in the FUD format.**



**Help.**

### 4.1.1 Control

This function key displays control-specific commands.

#### Menu structure

- Control
  - Replace
  - : ...
  - Run
    - Current control unit (only CL500)
    - Complete system (only CL500)
  - Stop
    - Current control unit (only CL500)
    - Complete system (only CL500)
  - Trace (not PC400)
    - Switch off trace

## PC600:

- Registers A, B, C, D
- Registers AB, C, D
- Registers A, BC, D
- Registers A, B, CD
- Registers AB, CD

## CL100:

- Registers A, H

## CL300:

- Registers A,B,C
- Registers AB, C
- Registers A, BC

## CL500:

- Registers A, B, C, D
- Registers AB, C, D
- Registers A, BC, D
- Registers A, B, CD
- Registers AB, CD
- Module tracing (not PC400/CL100)
  - Activate
  - Deactivate
  - Display

Source: *OM1*

PI: 3

Destination: *PM1*

## Replace



The command changes

- **individual program lines** while the program is running, or
- the value of **single data words** within the current data module.

Enter

**writes the new instruction in the subsequent I/O state to the RAM or EEPROM of the PLC!**

**Replace is not possible with EPROM.**



All changes are only possible with the **absolute** address.

Single word commands **cannot** be replaced by dual word commands.

Esc

aborts the **Replace** command immediately and the last command becomes effective again.



**Changes which are carried out with the Replace command are only available in the controller memory. The .PxA program file remains unchanged. The program file is only updated by linking!**

When exiting the module, the changes can be transferred to the .PxO module file.

**Trace**

The **register display** appears above the function key bar and the contents are updated after each I/O state depending on the cursor position.

1		Network 1			
; PROGRAM MODULE PM1					
A	B	-INPUT1	0	0	
AN	B	-INPUT2	0	0	
O	B	-MARKER0	0	0	
=	B	-OUTPUT	0	0	
CM		-TEST	0		
L	W	-DWa,A	0		A= 18D
L	W	-DWb,B	0		B= 21D
<b>ADD</b>	<b>W</b>	<b>B,A</b>	<b>0</b>		<b>A= 39D</b>
T	W	A,-DWc	0		A= 39D
A= 12D		B= 27D	C= 0D	D= 0D	
ZS0/PM1		PI: 8	RG:	PAA: 57D	IL mode
<F1>	<F2>		<F4>	<F5>	<F6>
Controller	Info		Search	Command	End
					<F7>
					Change
PLC Monitor program			Version 3.0 (c) 1987-92 Robert Bosch GmbH		

Fig. 4-2 Trace line

## CL100

In addition to the actual register **A** a second **help register H** is displayed. This register cannot be altered.

## Trace module

The trace module command facilitates the bringing into service of utilities which have **different parameters** but are otherwise the same.

Register and status displays always relate to the current state of the selected module.

Before branching off to the corresponding program module the starting point is saved with the trace module command.

The flashing module name in the information line indicates that the trace module has been activated.

## 4.1.2 Info

### Menu structure

- Info (not PC400)
  - Info status
  - Assignment plan (not CL100/CL500)
    - Output assignment
      - Overview (only CL300)
      - Set/actual value comparison (only CL300)
    - Input assignment
      - Overview (only CL300)
      - Set/actual value comparison (only CL300)
    - Extended outputs (only CL300)
      - Overview (only CL300)
      - Set/actual value comparison (only CL300)
    - Extended inputs (only CL300)
      - Overview (only CL300)
      - Set/actual value comparison (only CL300)
    - Memory assignment



- Reference list (not CL100)
  - All modules
  - Organisation modules
  - Utilities
  - Function modules (only PC600)
  - Data modules
  - Extended modules (only PC600)

### 4.1.3 Search

See subsection **3.3.6 Search**

### 4.1.4 Command

With these commands you can change the screen format.

You can change between the following formats:

- **Instruction list,**
- **Ladder diagram** and
- **Function diagram.**

For

- **Networks** and
- **Data modules**

you can choose between

- **Overview** and
- **Detail**

The command **Screen mode** is used to toggle the screen format between

- **25 lines** and
- **43/50 lines.**

## Menu structure



- Command
  - Display format  
Sym. operand / abs. operand  
Register display / Line comment
  - Number format
    - Decimal
    - Sign decimal
    - Hexadecimal
    - ASCII
    - Binary
    - Octal
  - Overview/detail
  - Instruction list
  - Ladder diagram
  - Function diagram
  - Screen mode

## Overview/detail



## Screen mode

**See also section 3.6 Network overview and section 3.9.2 Data module.**

Switch the screen display to another operating mode. In the standard setting 25 lines are displayed on the screen. With this switchover either 50 or 43 lines can also be displayed, depending on the video card used by your programming unit.

## 4.1.5 End

See subsection **3.3.8 End**





## 4.1.6 Change

Change between:

- **Module file**
- **Data module** (symbol file), see section 4.5 **Data module**
- **Operand field**, see section 4.6 **Operand field**
- **Setting**

### Menu structure

- Change
  - Save
  - Display/load
    - Module file  
File name:
    - Symbol file (not PC400/CL100)  
File name:
    - Operand field  
File name:
    - Set (not CL100)
      - Bit  
Bit operand:
        - Set value 1
        - Set value 0
        - Reset setting
      - Byte (not PC400)  
Byte operand:  
Set value:
      - Word (not PC400)  
Word operand:  
Set value:
    - List
      - Bit overview
      - Byte overview (not PC400)
      - Word overview (not PC400)

## Setting



The setting function specifies the logic state of the inputs and outputs.

**During program processing it is the set status of the transmitters and actuators connected which is processed, not the actual status!**



F10

**Help.**



**4.2 IL Monitor**

**Monitor field**

The program line is presented with the

- **symbolic** or
- **absolute**

**operand** in the left half of the line.

The right half of the line contains

- the **line comment** or
- the **monitor display**.

The monitor display consists of

- the **bit combinations** in bit commands,
- the **flags** set for word commands (not PC400/CL100)
- and the modified **registers**.

The set flags for identified by their first letter.

The contents of the registers A, B, C or D are displayed according to the defined numeric format.

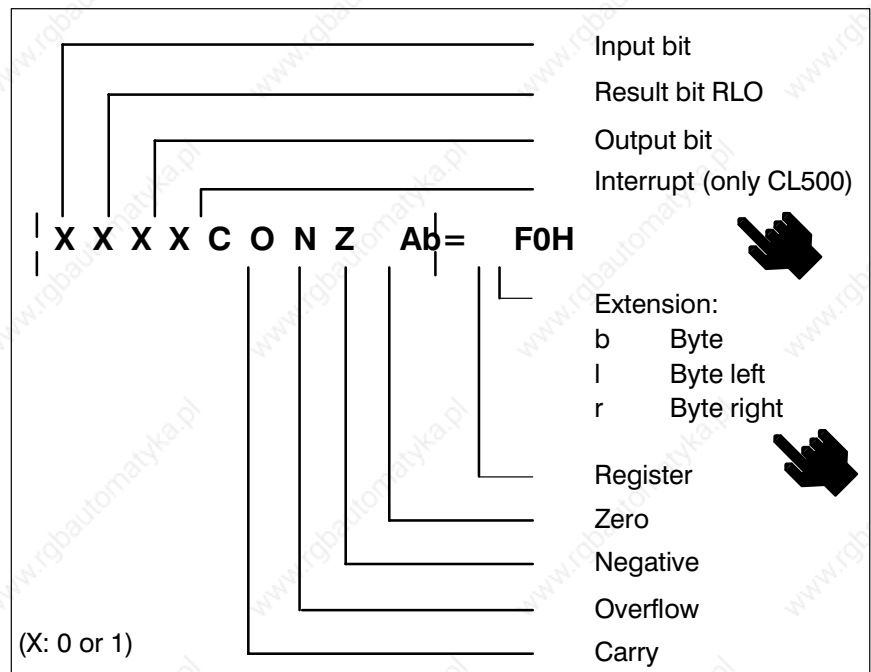


Abb. 4-3 Monitor line

For commands influencing time/counter the time or counter status is displayed.

If the interrupt is changed the new mask will be displayed in binary.

## Scrolling the screen



**The synchronisation between the display and the controller always takes place in the 1st line on the screen.**

In order to display a certain program branch on the screen it may be necessary to scroll the screen so that a certain program line is positioned in the first screen line. The position of the cursor bar remains unchanged.

★  Scroll  
Lock

★  ↑

or

↓

By pressing

Scroll  
Lock

the function of the cursor keys is toggled.

## PAA

Absolute **program address** in the PLC memory.



F10

**Help.**



**4.3 LD Monitor**

The monitor differentiates between two operating modes in the LD mode:

- **LD mode: On**
- **LD mode: Off**

To toggle between the two operating modes press:



Also see section 3.4 **Module file editor LD.**

**Monitor field**

In Monitor operation the current branch status is displayed. If one of the inputs or outputs shown has been set, the respective contact is displayed **inversely**. The LD connection lines are always displayed inversely.

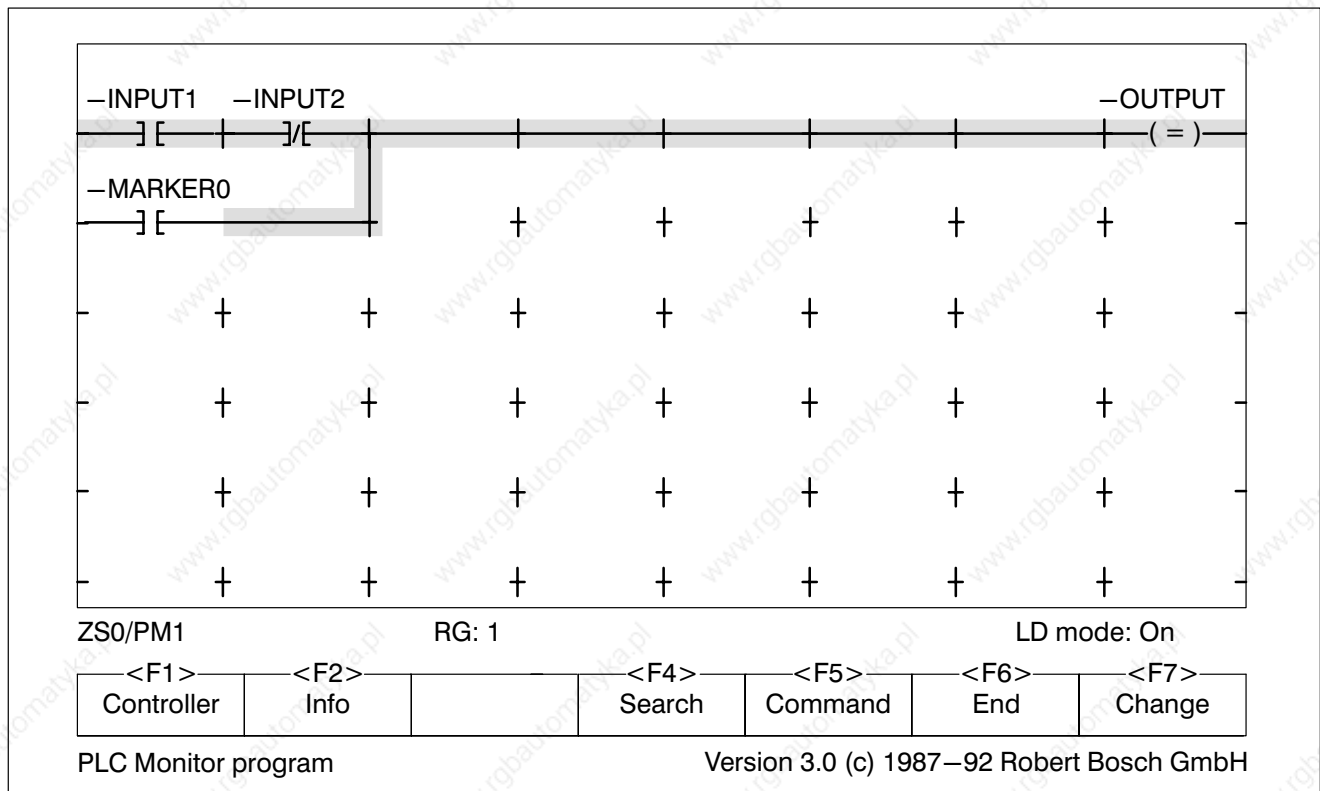


Fig. 4-4 LD Monitor

## 4.4 FUD Monitor

In the monitor operation the current network state is displayed. Elements and connections which produce single signals are highlighted (red or brighter).

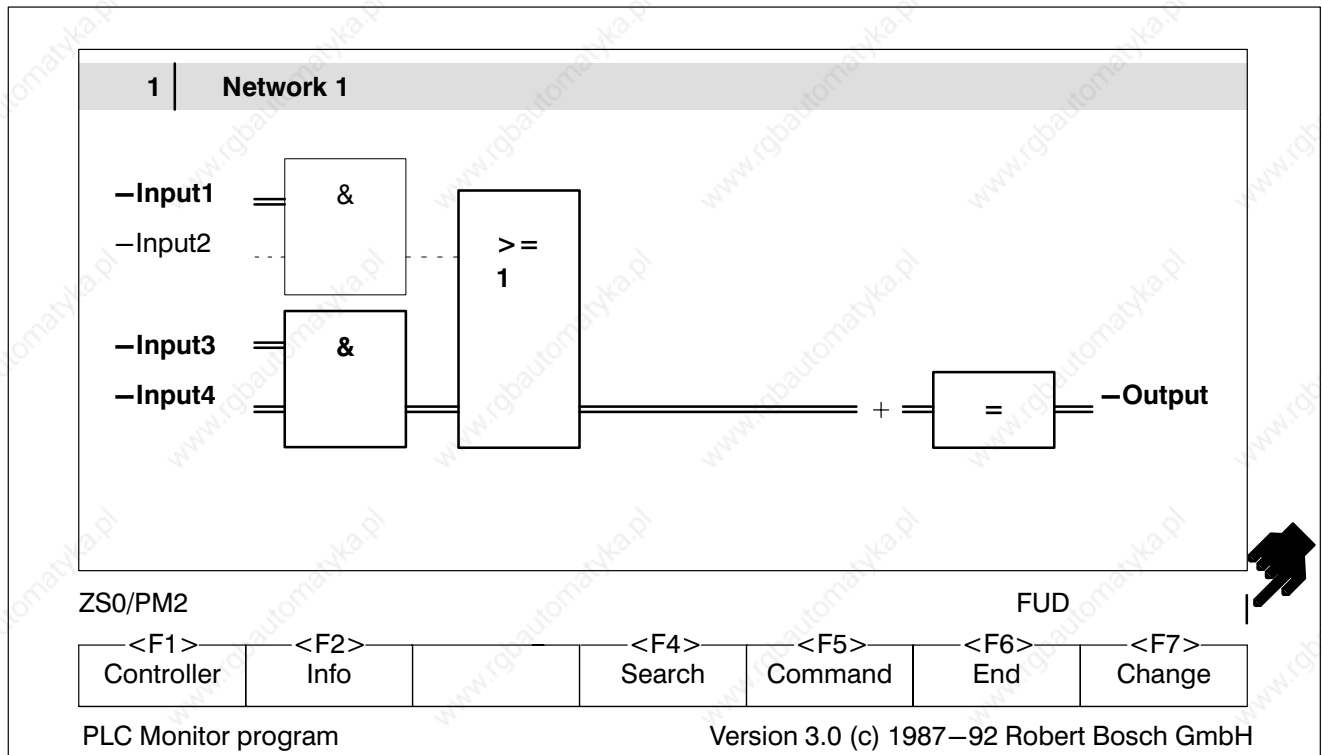
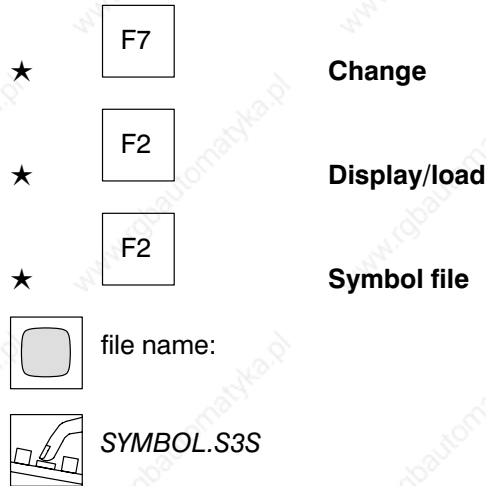


Fig. 4–5 FUD Monitor



**4.5 Data module**

Display of data modules or data module overview list:



The data modules are displayed as they appear in the controller. The length of the data module in the controller depends on the length of the **data module** which was specified in the **data module header**, see under Fig. 3—36.

DM	1	Name: ProdM1	Comment: Production on machine 1	RAM/EPROM: R	
No.	Symbol	Type	S	Data field	F
D 0	Set speed	Word	N	120	D
D 2	Set ctrl	Word	N	76	D
D 4	Wait at	Word	N	2	D
D 6	Set prod	Word	N	1256	D
D 8	Total	Word	N	1013	D
D 10		Word	N	0000	D
D 12		Word	N	0000	D
D 14		Word	N	0000	D
D 16		Word	N	0000	D
D 18		Word	N	0000	D
D 20		Word	N	0000	D
D 22		Word	N	0000	D
D 24		Word	N	0000	D
D 26		Word	N	0000	D

ZS0/SYMBOL

<F1> Controller	<F2> Info	<F4> Search	<F5> Command	<F6> End	<F7> Change
--------------------	--------------	----------------	-----------------	-------------	----------------

PLC Monitor program Version 3.0 (c) 1987—92 Robert Bosch GmbH

Fig. 4—6 Data module

## Data module overview list



F5

**Command**



F5

**Overview/detail**

The data module overview list in the Monitor acts as a directory of all the data modules present in the controller. Using the data module overview list you can change rapidly to another data module.



F10

**Help.**





**4.6 Operand field**

The operand field display enables any operands desired:

- inputs,
- outputs,
- markers,
- data words, ...

to be

- **represented,**
- **controlled** and
- **displayed**

on the screen.

In this way, many different operands can be simultaneously displayed on one screen page which cannot be simultaneously displayed in the IL Monitor. The operand field is stored in a file with the file type **.OxD**.

Operand	Ext	Value	F	C	O
-INPUT1	B	0	B		
-INPUT2	B	0	B		
-OUTP20	BY	11110000	B		
-OUTP40	BY	11110000	B		
-DMTEST -DW10	W	85	D		

ZS0/OPEFIELD.O3D Operand display

<F1> Edit	<F2> Control	<F3> Display	<F4> File	<F7> Return
--------------	-----------------	-----------------	--------------	----------------

PLC Monitor program Version 3.0 (c) 1987-92 Robert Bosch GmbH

Fig. 4-7 Operand field

The following operating modes are available:

- **Edit**
- **Control**
- **Display**

The operand field format comprises a maximum of 64 lines.

The operand field format contains a special function key bar.

## Operand



The absolute or symbolic operand is entered in the **operand** column.

A **comment line** is identified by a semicolon ; at the start of the line.

Operand	PC400	PC600	CL100	CL300	CL500
I Input	●	●	●	●	●
O Output	●	●	●	●	●
M Marker		●	●	●	●
SM Special marker				●	●
T Time		●		●	●
C Counter		●		●	●
DM DW Data word		●		●	●
DB Data buffer				●	●
E Error				●	
S System range				●	●

Fig. 4-8 Operand types

## Ext. Extension

Operand extension	PC400	PC600	CL100	CL300	CL500
B Bit	●	●	●	●	●
BY Byte			●	●	●
BR Byte right		●			
BL Byte left		●			
W Word		●	●	●	●

Fig. 4-9 Operand extension

**Value**

Data can be entered in this column so that they can be transferred to the controller in the **Control** mode. In addition, the controller data are displayed in this column in the **Display** mode.

As soon as data are entered or changed in this column, the line is automatically marked for control.

- For word and byte entries one word or one byte of data can be entered per line.
- For ASCII entries in the data modules the complete column can be used for input, i.e. maximum 48 bytes.

When entering ASCII lines beyond the range from 20 H to 7F H the following rules apply:

- Range below 20 H:  
Either the hexadecimal value, the decimal value or the mnemonic code of the control character can be entered. When input the character is written in pointed brackets <>  
Example: <0DH> or <13D> or <CR>
- Range above 7F H and below/equal to FF H:  
Either the hexadecimal or the decimal value can be entered. The character < must be doubled <<.  
Example: <129D> or <81H>

**F**

**Data format** of the data specified in the **value** column.

D decimal  
V sign decimal  
B binary  
O octal  
H hexadecimal  
A ASCII

**C**

**Control marker** (not PC400)

Entering \* or **C** identifies the operands which are loaded into the controller in the **Control** mode. For this to be possible, a data value has to have been entered in the **Value** column.

**X**

The **X** column indicates whether a specified input or output has been **set**. This column cannot be edited.

## 4.6.1 Edit

In the **Edit** operating mode the operands are entered in the operand field.

## 4.6.2 Control



### **Not PC400.**

The operands marked in the **C** column are loaded into the controller.



**During program processing, it is the changed status of the transmitters and actuators connected which is processed in the operand field, not the actual status!**



**The operands are loaded into the controller in succession, i.e. not in a PLC cycle!**

## 4.6.3 Display

The operands are displayed in dynamic format.



**The displayed operands are read from the controller in succession, i.e. they do not necessarily originate from a PLC cycle!**



#### 4.6.4 File

Save and load the operand field file with the file type **.OxD**. 

##### Menu structure

- File
    - Save file
    - Load file
- file name:

#### 4.6.5 Return

 F7

##### Return

The format of the function key bar is altered so that the other Monitor commands are also available for the operand field display.

 Esc

Return to the special operand field function key bar.

 F10

##### Help.

For your notes:



**Contents**

	Page
<b>5 Loader .....</b>	<b>5-1</b>
5.1 Connection of programming unit ↔ controller .....	5-4
5.2 Load a program file into the controller .....	5-10
5.3 Loader commands .....	5-11
5.3.1 Linking .....	5-12
5.3.2 Loading .....	5-13
5.3.3 (E)EPROM .....	5-15
5.3.4 Info .....	5-16
5.3.5 Configuration .....	5-17
5.3.6 End .....	5-18

## Illustrations

Fig.		Page
5-1	Functions of the Loader .....	5-2
5-2	Loader .....	5-3
5-3	Connection cable K7 .....	5-4
5-4	Connection cable K8 .....	5-5
5-5	Connection of programming unit ↔ PC400/PC600 .....	5-6
5-6	CL100 DIL switches 7 and 8 .....	5-7
5-7	Connection of programming unit ↔ CL100 .....	5-7
5-8	Connection of programming unit ↔ CL300 .....	5-8
5-9	Connection of programming unit ↔ CL500 .....	5-9
5-10	Logging procedure during loading .....	5-10
5-11	Loader commands .....	5-11
5-12	Linking .....	5-12
5-13	EP/AG module, (E)EPROM connection socket .....	5-15





## 5 Loader

The Loader offers the following functions which can be selected with function keys:

- Set memory configuration.
- Edit and load the SK table for the CL500.
- Link and unlink modules.
- Additive load and unload of modules into and out of the controller.
- Load and unload programs into and out of the controller.
- Load and unload PIC programs.
- Load transmission log into the computer coupling module, CL500 only.
- Program, duplicate and compare (E)EPROM.
- Info status.
- Compare programs or modules on floppy disk/hard disk with programs or modules in the controller.
- Reference list.

In the **Loader** observe the following operating sequence when loading a program into the controller.

- Specify the program file name for the program to be linked in the defaults.
- Set memory configuration.
- For the CL500 also load and edit the SK table.
- Link program.
- Connect controller.
- Load program into the controller.

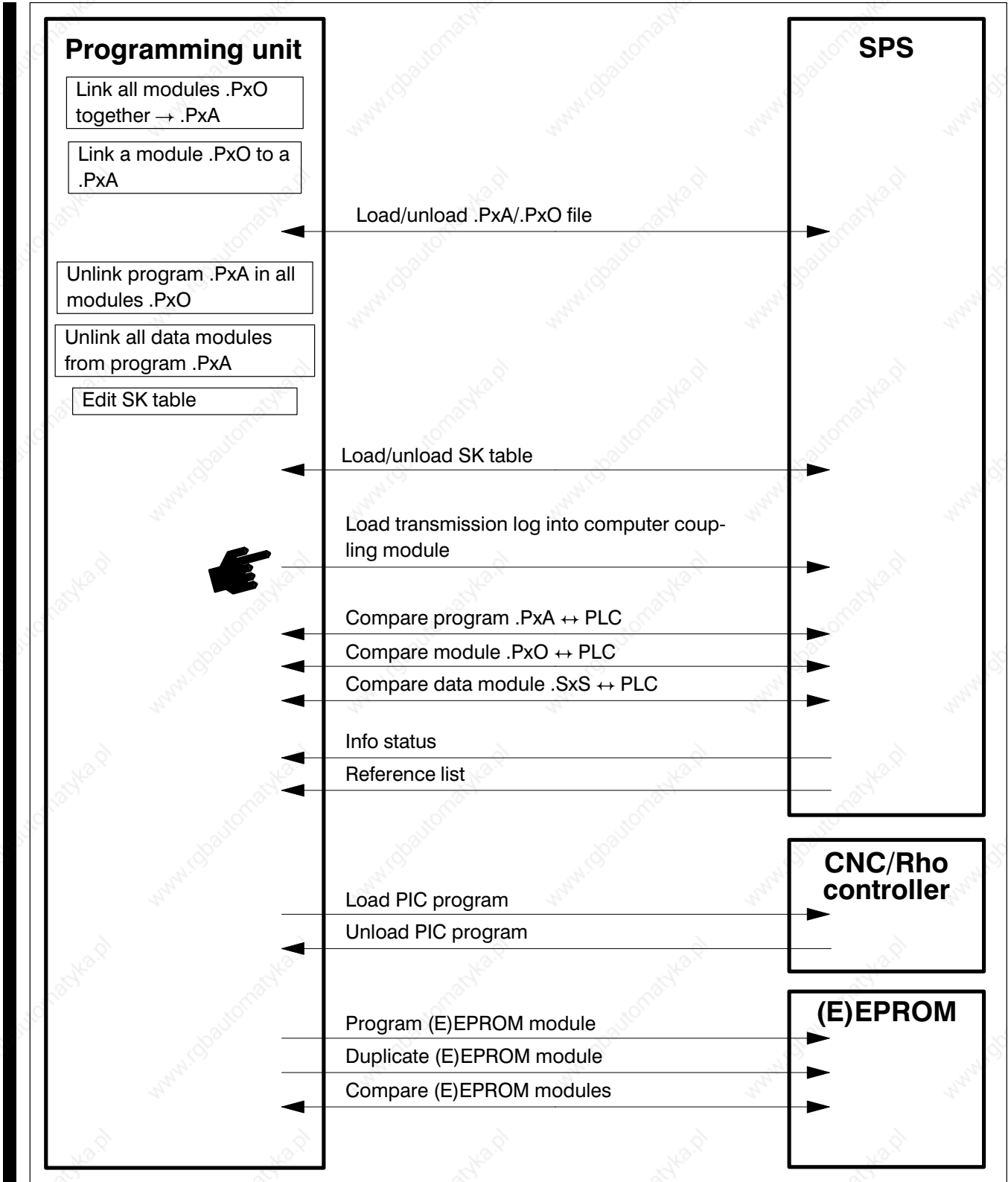


Fig. 5-1 Functions of the Loader

**Loader**

The PLC Loader program is called up from the main menu by pressing

F4
----

**Loader**

twice.

With the first press of the function key the defaults are displayed on the screen.

To edit the defaults press:



See section **1.7 Entering defaults**.

The second press of the key starts the PLC Loader program.

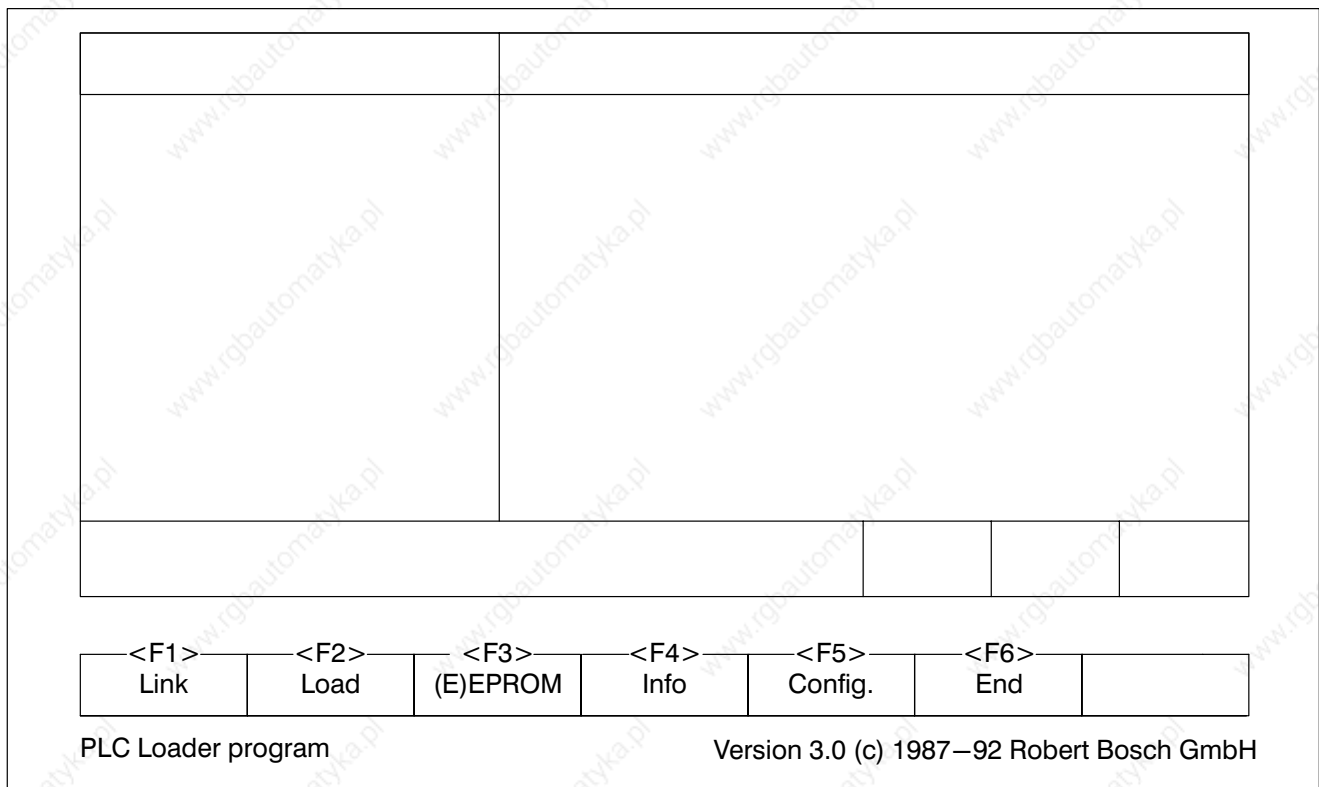


Fig. 5-2 Loader

The screen splits into two areas:

- **Log**
- **Function key bar**

The **loading** and **comparing** are logged on the screen.

## 5.1 Connection of programming unit ↔ controller

The execution of the commands of the PLC Loader program requires

- the **EP/AG module** and
- the **connection** between the programming unit and the controller.

EP/AG module Part number 054 613

Connection cable K7  
(for CL100/CL300/CL500) Part number 054 334

Connection cable K8  
(for PC400/PC600) Part number 054 337

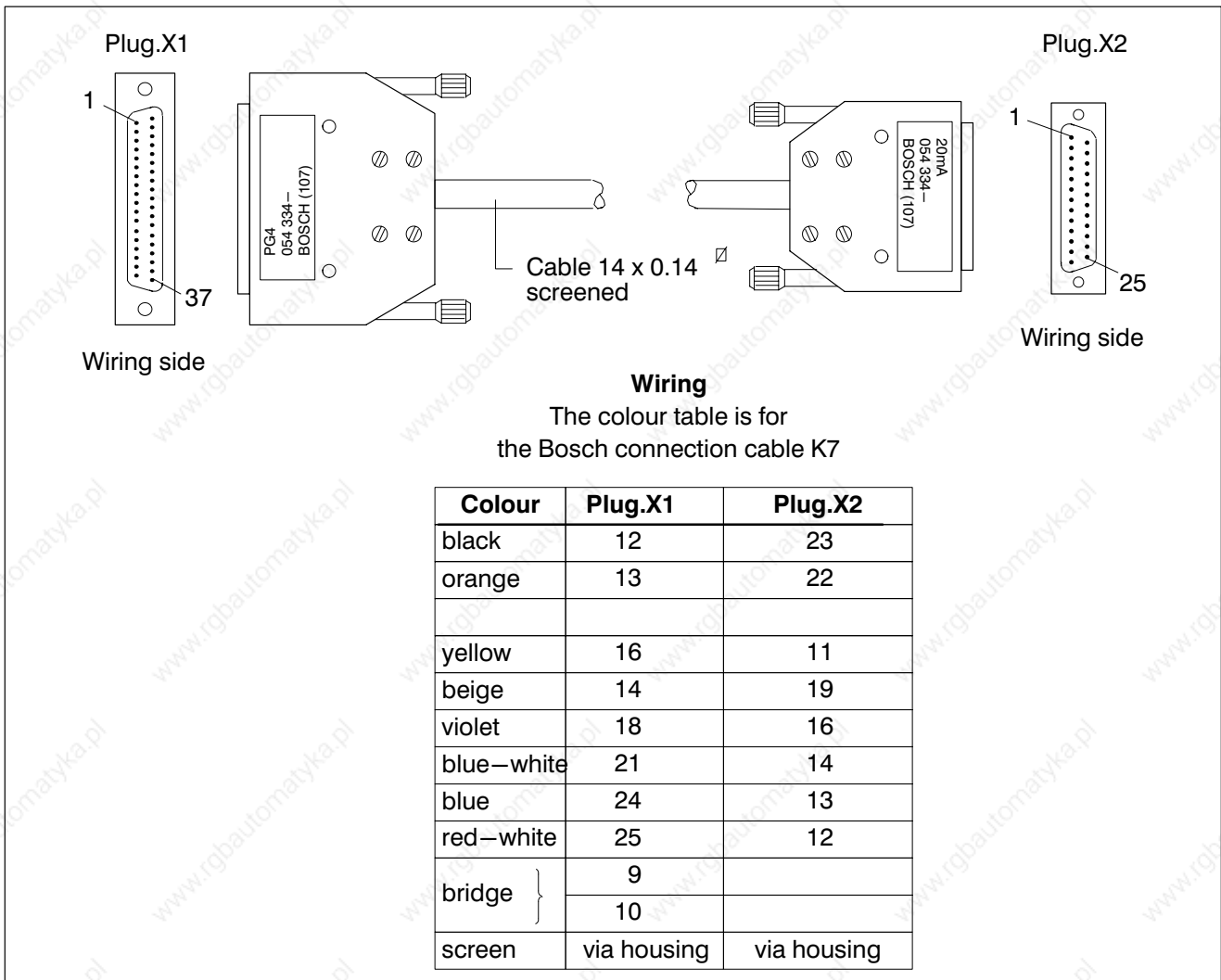


Fig. 5–3 Connection cable K7

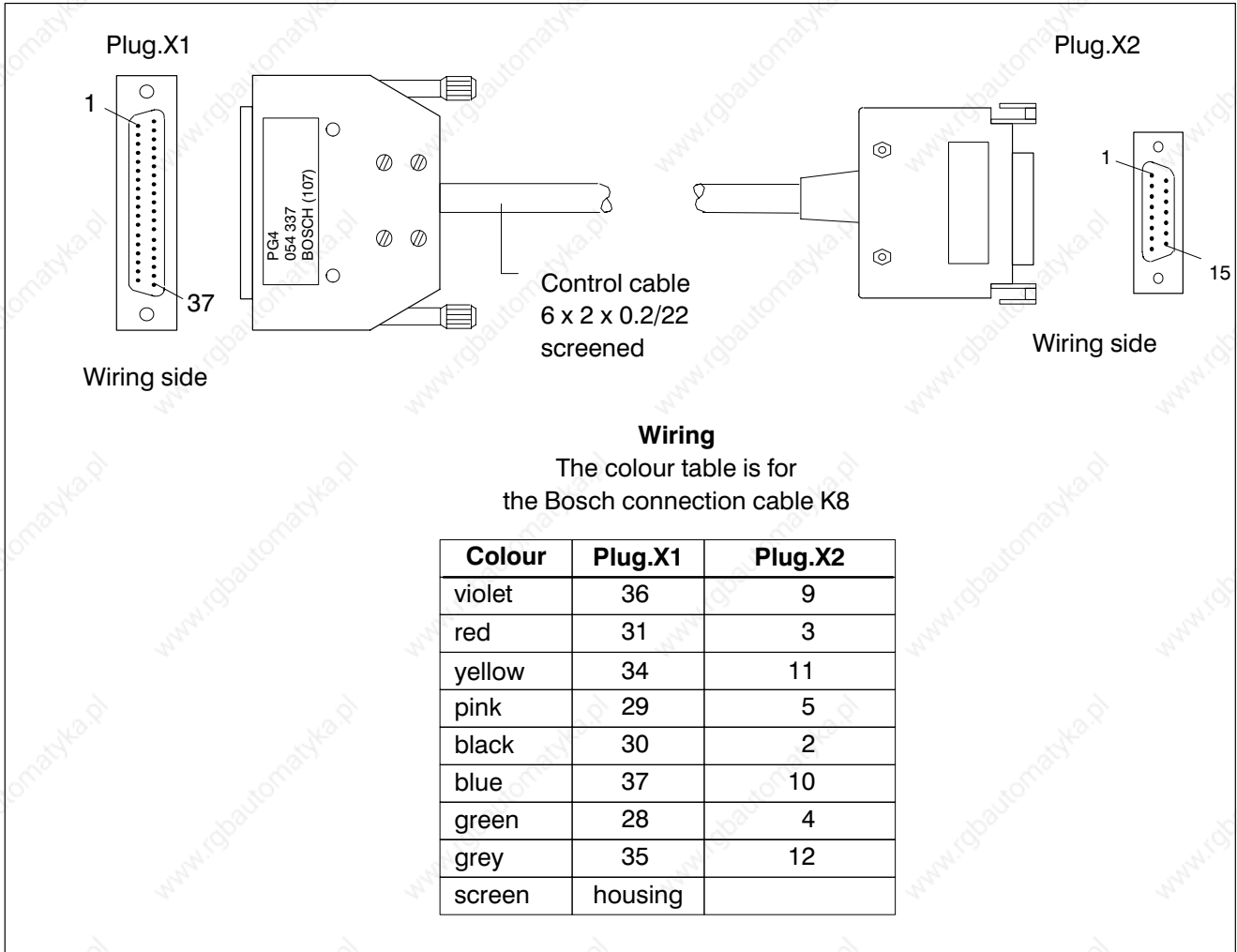


Fig. 5-4 Connection cable K8

## PIC

To load a PIC program into a Bosch CNC or Bosch Rho controller, see Technical Documentation:

**PIC200/PG4**  
**mit CC–Baureihe**  
**P.–Nr. 4119**

## PC400/PC600

Connect the programming unit to the PC400/PC600 controller:

- PC400 consisting of: ZE401 and P401
- PC600 consisting of: ZE611, ZE612, ZE613, M601 and P600.

★ Connect the **connection cable K8** to the PLC programming interface of the EP/AG module and to the **connection module P401/P600**.

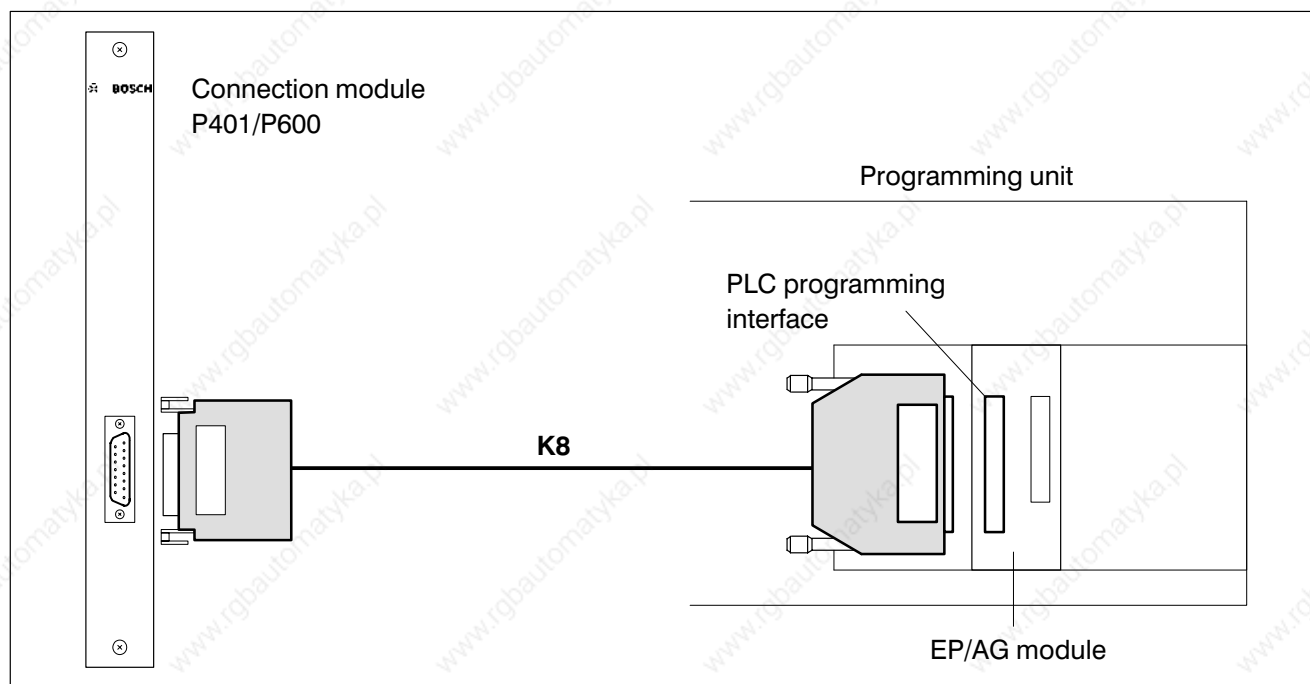


Fig. 5–5 Connection of programming unit ↔ PC400/PC600

★ Switch on controller.



## CL100

Connect the programming unit to the CL100 controller:

- ★ Remove the cover of the CL100 and switch the DIL switches 7 and 8 to **ON**. Replace the cover.

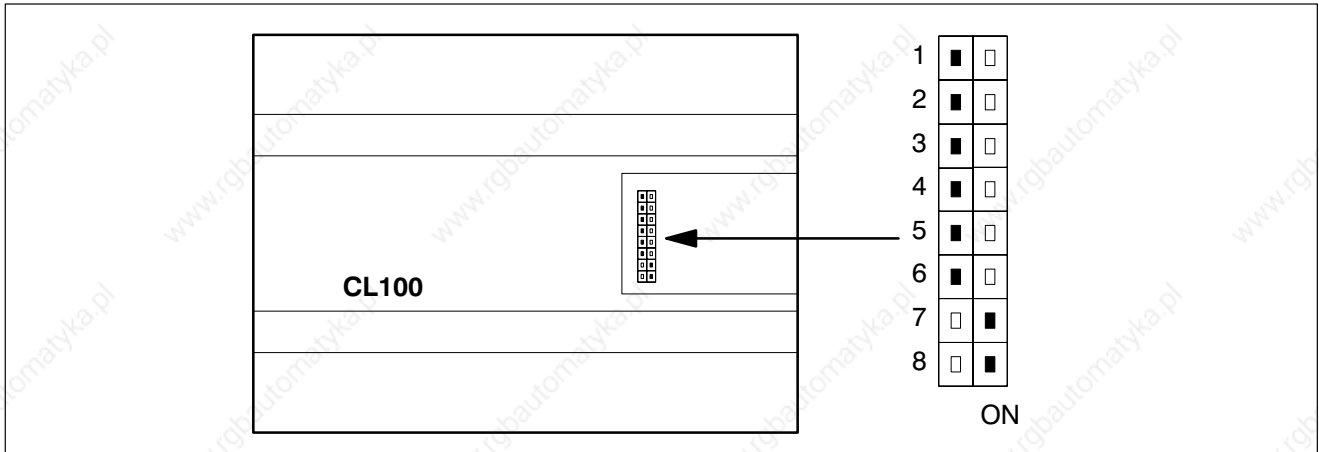


Fig. 5-6 CL100 DIL switches 7 and 8

- ★ Connect **connection cable K7** to the **PLC programming interface** of the EP/AG module.
- ★ Plug **adapter 050 551** into the **PG connection socket** of the CL100.
- ★ Plug **connection cable K7** into the **printer** socket of the adapter.

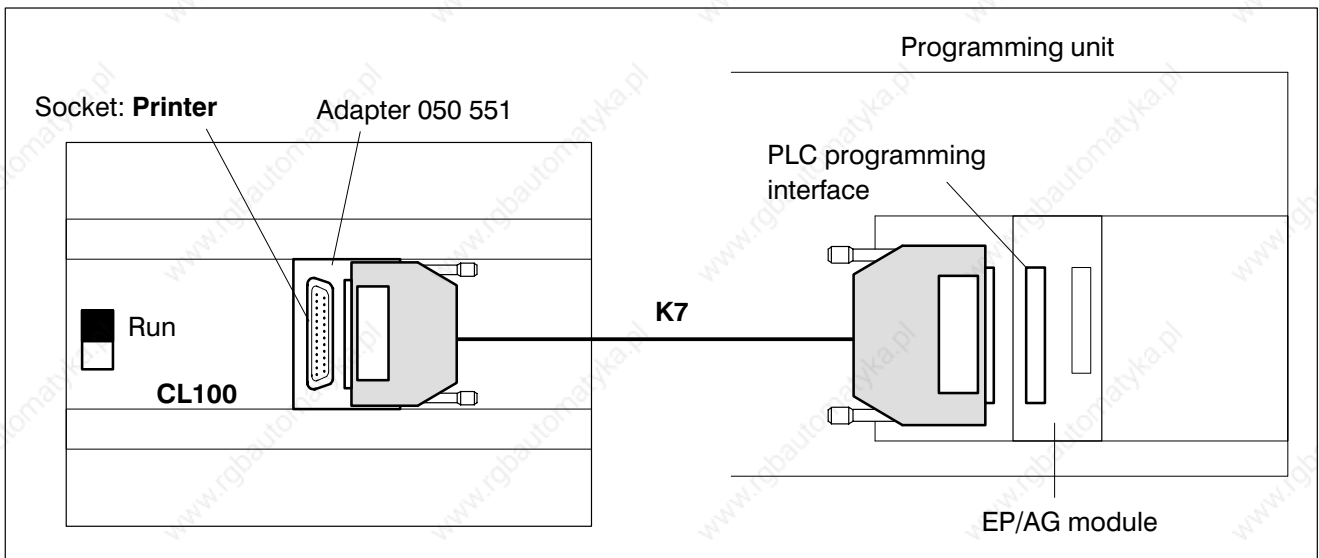


Fig. 5-7 Connection of programming unit ↔ CL100

- ★ Switch on controller.
- ★ Set the operating mode switch of the CL100 to **RUN**.

## CL300

Connect the programming unit to the CL300 controller:

- ★ Connect **connection cable K7** to the **PLC programming interface** of the **EP/AG module** and to the **central processing unit (CPU) ZE300/ZE301**.

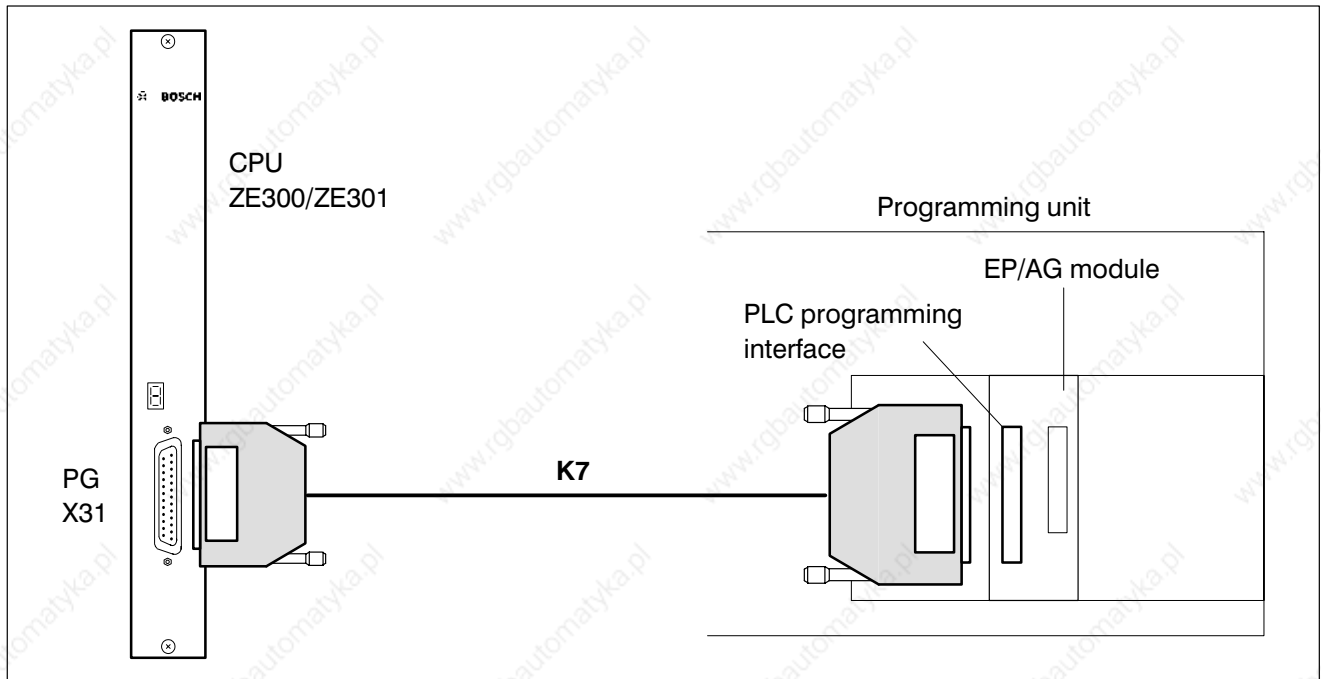


Fig. 5–8 Connection of programming unit ↔ CL300

- ★ Switch on controller.





## CL500

Connect the programming unit to the CL500 controller:

- ★ Connect **connection cable K7** to the **PLC programming interface** of the EP/AG module and to the interface **PG X31** of the **SK500 system coordinator** module.

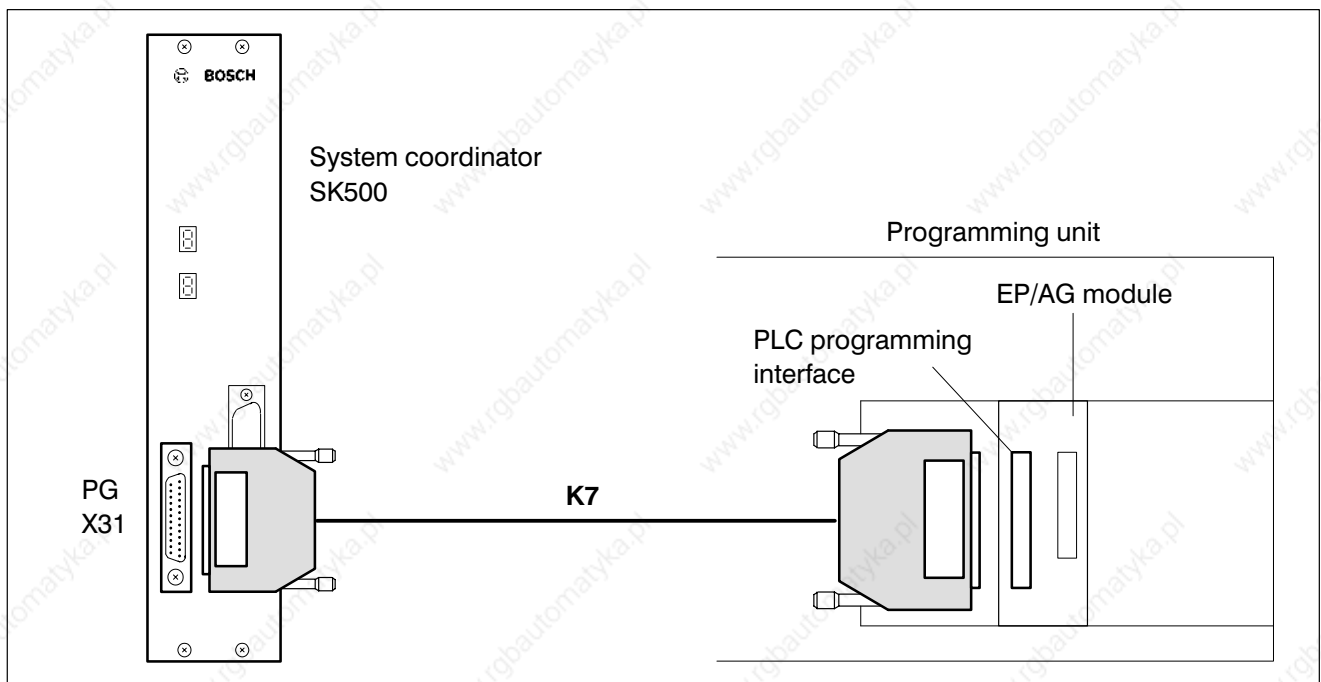


Fig. 5-9 Connection of programming unit ↔ CL500

- ★ Switch on controller.

## 5.2 Load a program file into the controller

The most important function of the Loader is loading a program file into the controller.

To load a program file the following steps must be carried out:

- **Connect** the programming unit and the controller, see subsection **5.1 Connection of programming unit ↔ controller**.
- Specify the name of the **program file** to be linked in the defaults.
- **Set memory configuration**.
- Edit and load **SK table** for the CL500.
- **Link** program.
- **Load** program file.

The linking and loading of the program file is logged on the screen.

Edited modules:		Editing result:			
Module:	OM1	free of errors			
Module:	PM1	free of errors			
Intermediate result: Edited modules free of errors		2	2	Esc	
<F1> Link	<F2> Load	<F3> (E)EPROM	<F4> Info	<F5> Config.	<F6> End
Controller is in STOP (Edit) mode! Switch back to RUN (Monitor) mode?					Yes/No

Fig. 5–10 Logging procedure during loading



### 5.3 Loader commands

The commands are called up with the function keys. The pull-up menus indicate the meaning of the function keys.

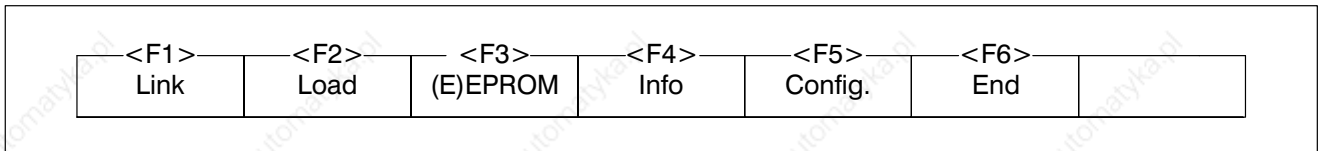


Fig. 5–11 Loader commands

**F10****Help.**

## 5.3.1 Linking



### The memory configuration must be set before linking.

The linking process links the individual module files (e.g. OM1.P6O, PMANL.P6O, PMAUTO.P6L) and the data from the symbol file to form a complete loadable program file with the **file type** .PxA. The modules used are entered in the **reference list**.

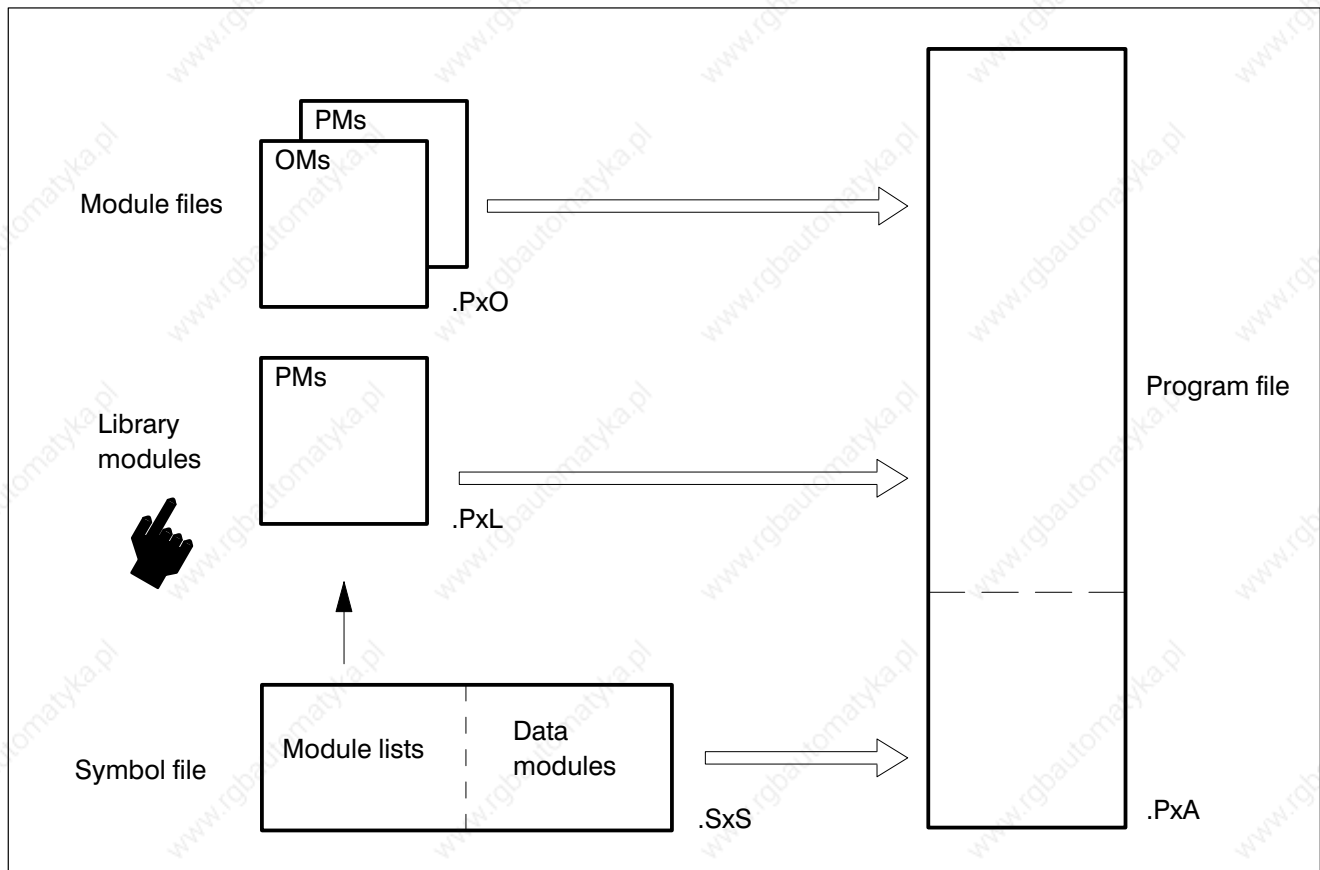


Fig. 5–12 Linking

### Menu structure

- Linking (not PIC)
  - Link all modules
  - Unlink one module (not PC400/CL100)
  - Unlink all modules (not PC400/CL100)
  - Unlink all data modules (not PC400/CL100)



### 5.3.2 Loading

The loading process is logged on the screen.

#### Menu structure

- Load
  - Load program into the controller
    - With reset retentive markers/operands (not PC400/CL100)
    - Without reset retentive markers/operands (not PC400/CL100)
  - only CL100:
    - .P1A file → controller
    - E<sup>2</sup>PROM → RAM
    - RAM → E<sup>2</sup>PROM
  - Additive load of a module into the controller (not PC400/CL100)
  - Unload program from the controller
  - Unload module from the controller
    - OM, PM, ZM, ... (not CL100)
  - Module name:
    - OM (not PC400/CL100)
  - DM no.:
- only PIC:
  - CC100
    - Load PG → controller
    - Unload PG ← controller
  - CC200
    - Load PG → controller
    - Unload PG ← controller
  - CC300
    - Load PG → controller
    - Unload PG ← controller
  - rho1
    - Load PG → controller

- Unload PG ← controller
- rho2
- Load PG → controller
- Unload PG ← controller

## Additive load of a module into the controller

For **additive loading** a link is only made in the memory of the controller. The newly linked file does **not** exist on the hard disk or floppy disk of the programming unit.

## PC400

When **unloading** programs from the controller, absolute jumps are automatically provided with a jump destination.



### 5.3.3 (E)EPROM



The **EP/AG module** is also used for programming, duplicating and comparing (E)EPROMs.

**When handling EPROM modules all ESD safety measures must be observed! Avoid electrostatic discharges!**

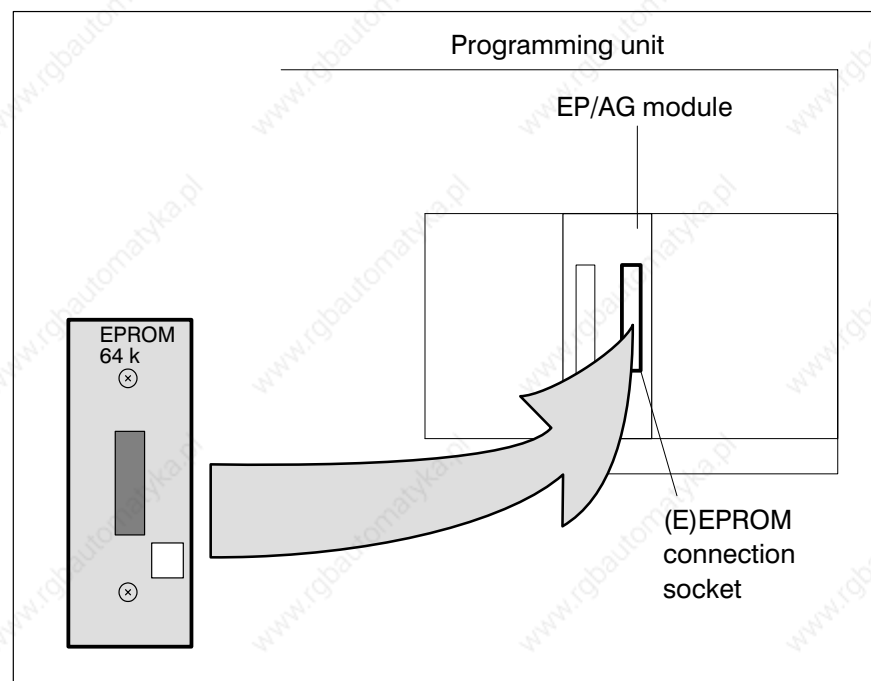


Fig. 5–13 EP/AG module, (E)EPROM connection socket

#### Menu structure

- (E)EPROM (not PIC/CL100)
  - utilities
  - Duplicate modules
  - Compare modules
    - Module ↔ Program file
    - Module ↔ Module
    - Module deleted?



**The deletion time for the EPROM modules must be at least 30 mins! With a shorter deletion time individual memory cells can regenerate themselves!**

## 5.3.4 Info

These commands provide information about the controller connected.

### Menu structure

- Info (not PIC)
  - Info status (not PC400)
  - Compare
    - Program: PxA ↔ PLC
    - Module: PxO ↔ PLC
    - Module name:
    - Data module: SxS ↔ PLC (not PC400/CL100)
    - Data module number:
    - Program files: PxA ↔ PxA
    - file name:
    - Ref. file:
    - Module: PxO ↔ PxO
    - file name:
    - Ref. file:
  - Reference list (not PC400/CL100)
    - All modules
    - Organisation modules
    - Utilities
    - Function modules (only PC600)
    - Data modules
    - Extended modules (only PC600)

### Info status

Provides information about the system status of the CPU or the control unit.

### Compare

**.PxA**, **.PxO** and data modules are compared with each other. This comparison is performed in both directions.

### Reference list

The **reference list** provides information on the modules present in the controller. The information is independent of the symbol file set in the defaults.





The module length and the start address are specified as decimal **D** or hexadecimal **H** in the **Word** unit.

In the **Attr.** column the entry **Bosch** refers to **Bosch standard function modules**.

### 5.3.5 Configuration

The hardware configuration of the controller is set. These details are required to link the program file.

#### Menu structure

- Configuration (not CL100)
  - Memory configuration
  - System configuration (only CL500)
    - Edit table  
file name:
    - Load table into the controller  
file name:
    - Unload table from the controller  
file name:
  - Log loader (only CL500)
    - Display computer module
    - Display log file
    - Load log  
Module:
      - Both channels  
file name:
      - Only channel 0  
file name:
      - Only channel 1  
file name:
    - Delete log memory  
Module:

## Memory configuration



### The memory configuration must be set before linking.

One of the two memory areas (RAM or EPROM) must start at the address 0000 H, because the **reference list** is stored from address 0000 H onwards.

## System configuration

The **SK table** must contain the following information:

- **Module number.**
- Module-specific **system parameters.**

The file receives the **file type .S5K**. The SK table must be edited for every project before a PLC program is loaded, and it must be loaded into the **SK500 system coordinator**.

You can find further information on the SK table in the technical documentation:

**CL500  
Manual  
Part 1  
P.–Nr. 4090**

A subdirectory for the SK file is created in the project directory.

## Log loader

The transmission logs are stored in the **RBG** subdirectory.

## 5.3.6 End

See subsection **3.3.8 End**.



**Contents**

	Page
<b>6 Lister .....</b>	<b>6–1</b>
6.1 Printer connection .....	6–3
6.2 Printing a module or symbol file .....	6–5
6.3 Lister commands .....	6–9
6.3.1 Module .....	6–9
6.3.2 Symbol .....	6–11
6.3.3 Cross–reference .....	6–12
6.3.4 Print .....	6–13
6.3.5 End .....	6–14

## Illustrations

Fig.		Page
6-1	Connection cable K6 .....	6-3
6-2	Printer connection to LPT1 .....	6-4
6-3	Lister commands .....	6-9



## 6 Lister

The PLC Lister program offers the following functions:

- Documentation of the module file in IL, LD or FUD
- Documentation of the symbol file
- Documentation of the cross references
- Documentation of the text file
- List parameters
- Printer settings

It is possible to display and edit a print file as a **text file** in the **PLC Editor program** using PLC utilities, see section **3.10 Text file editor**.

You set the **type of printer**, **character set**, number of **lines/pages**, **start sequence** and **end sequence** in the **configuration program**, see technical documentation:

**PLC/DESI Utilities**

**Professional Integrator**

**Installation instructions**

**P.–Nr. 4308**

### Lister

The PLC Lister program is called up from the main menu by pressing

 F5

**Lister**

twice.

The first press of the key displays the defaults on the screen.

To edit the defaults, press:



See section **1.7 Setting the defaults**.

By pressing the function key a second time the PLC Lister program is started.



**When you first call up the PLC Lister program you must first use the commands**

**F5**

**Print**

**and**

**F4**

**Select printer type**

**to select one of the installed printers.**

Connect the printer to the **parallel interface LPT1** or to the **serial interface COM1** of the programming unit.



**If you intend printing via the serial interface COM1 you must configure it using the MS DOS command MODE, see MS DOS documentation.**



### 6.1 Printer connection

#### LPT1

To connect the Centronics interface of the programming unit to the printer you will need **connection cable K6**, see Fig. 6–1.

Connection cable K6

Order number

054 097

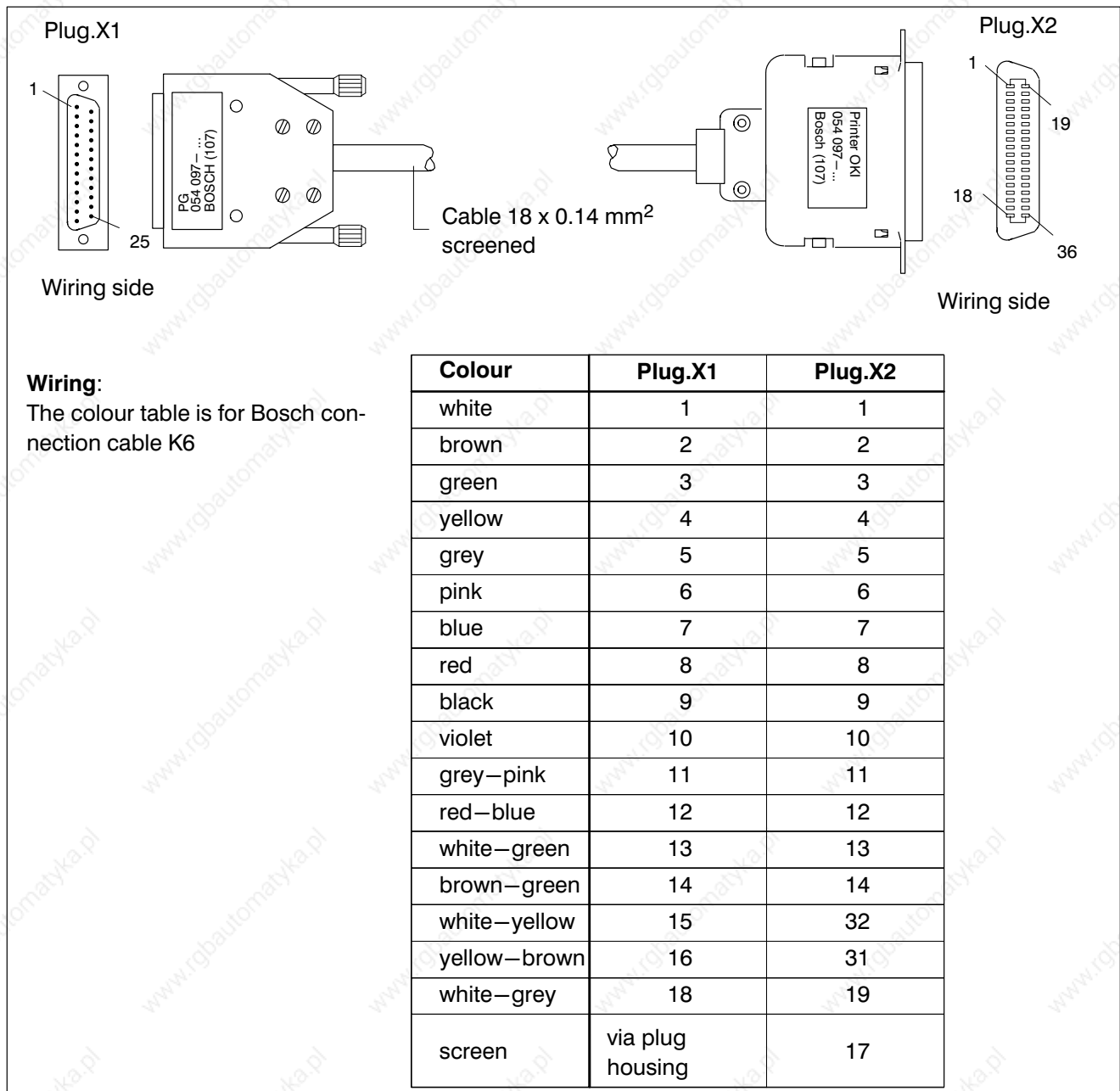


Fig. 6–1 Connection cable K6

- ★ Connect the **connection cable K6** to the output of the **software dongle**.

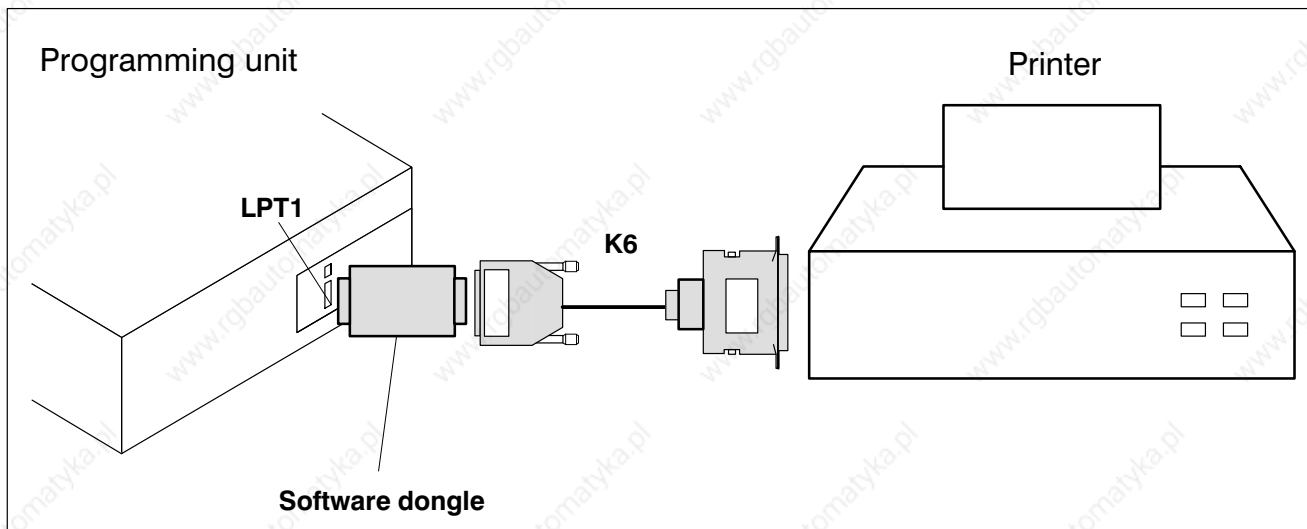


Fig. 6-2 Printer connection to LPT1

- ★ **Switch on** the printer and set to **On-line**.



**If a printer is connected to the software dongle, the printer must always be kept switched on to prevent faults.**





## 6.2 Printing a module or symbol file

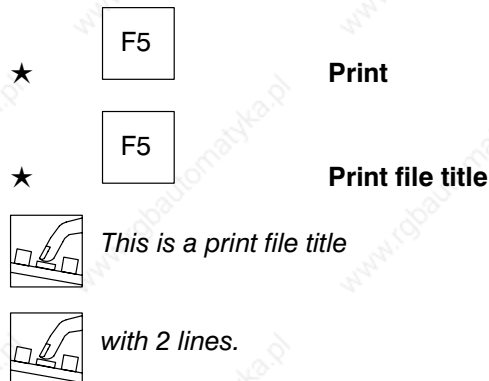
Printing a module/symbol file is divided into 5 steps.

- **Print file title**
- **Header file**
- **Set list parameters**
- **Select and create print file**
- **Activate printer**

This section describes the printing of a module file. The first two steps **Print file title** and **Header file** can be dropped. However, the stipulated sequence must be observed.

### Print file title

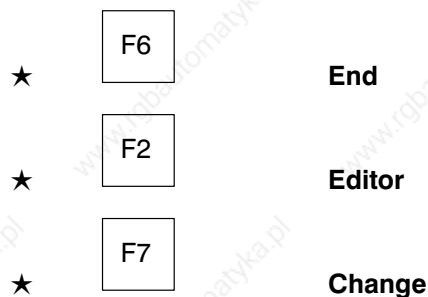
The print file title consisting of a maximum of **2** lines is printed out on every page in the print header.



### Header file

The header file is placed at the start of the print file and is printed with the file. The header file contains additional comments about the project.

The header file is created in the PLC Editor program as a text file *Header.TXT*.



★  **Display/Load**

★  **Text file**

 file name:

 *Header.TXT*

★ Create header file.


★  **End**

★  **Lister**

★  **Print**

★  **Header file**

 Header file:

 *Header.TXT*

## Set list parameters

The list parameters determine the print image.

★  **Module**

★  **List parameters**

★ Select the list parameters with the keys

to

★



- ★  **Instruction list**
- or  **Ladder diagram**
- or  **Function diagram**

**Select and create print file**

- ★  **Module**
- ★  **Current module file**
- or  **According to symbol file**
- or  **According to batch file**



Name of print file:



*Print.PRN*



**Select .PRN as file type.**

The print file is created as a text file and the process is logged on the screen.

**Activate printer**

- ★  **Print**
- ★  **Print list file**



Name of print file:



*Print.PRN*



Name of list device [PRN]:

To which interface (LPT1 or COM1) is the printer connected?



LPT1

or



COM1

Printing begins.



Esc



F10

**Help.**



### 6.3 Lister commands

The commands are called up with the function keys. The pull-up menus show the meaning of the function keys.

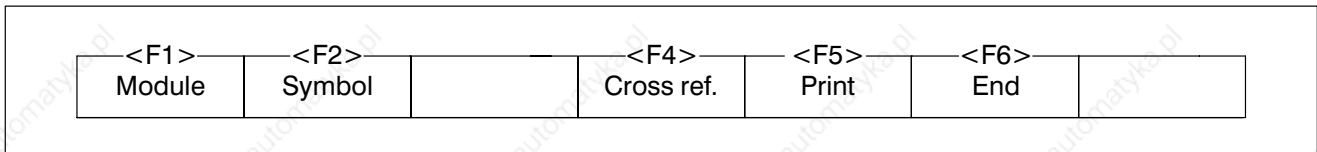


Fig. 6-3 Lister commands

#### 6.3.1 Module

Creating the print file of a module file.

##### Menu structure

- Module
  - Current module file  
Name of print file:
  - According to symbol file (not PIC)  
Name of print file:
  - According to batch file  
Name of print file:
  - List parameters
    - List RG number  
with symbol comment  
symbol and line comment  
sym. com. byte/word add. on  
80 characters per line  
operands symbolic
    - Instruction list
    - Ladder diagram
    - Function diagram

no RG number  
without symbol comment  
symbol comment in line  
sym. com. byte/word add.  
off 132 characters per line  
operand absolute

## List parameters

The print image is set using **control sequences**. The control sequences can also be written to the module file. They must be positioned at the start of a line and preceded by a semicolon.



```
A      B  I10.0
=      B  O23.2
```

```
EM                                ;End of module
;$P                               ;Page feed
```

The following control sequences are available:

\$AWL	Display type instruction list
\$FUP	Display type function diagram
\$KPL	Display type ladder diagram
\$PZ+	Output RG number
\$PZ-	Output no RG number
\$SK+	List with symbol comment
\$SK-	List without symbol comment
\$SUZ+	Symbol and line comment
\$SUZ-	Symbol comment in line
\$SBW+	Symbol comment byte/word address on
\$SBW-	Symbol comment byte/word address off
\$SYM	Symbolic operands
\$ABS	Absolute operands
\$80	80 characters per line
\$132	132 characters per line
\$P	Page feed
\$NOLIST	Interrupt print-out for current module
\$LIST	Continue print-out for current module

F10

**Help.**



### 6.3.2 Symbol

Creating the print file of a symbol file or individual forms.

#### Menu structure

- Symbol
  - Module form(s) (OM, PM, ...) (not PIC)
    - PC400:  
OM  
Name of print file:
    - PC600:  
OM, PM, FM, EM  
Name of print file:
    - CL100:  
OM  
Name of print file:
    - CL300:  
OM, PM  
Name of print file:
    - CL500:  
OM, PM  
Name of print file:
  - Operand forms (I, O, M, ...)
    - PIC:  
I, O, M  
Name of print file:
    - PC400:  
I, O, M, C, T, XI, XO, F  
Name of print file:
    - PC600:  
I, O, M, C, T, EI, II, EO, IO  
Name of print file:
    - CL100:  
I, O, M, C, T  
Name of print file:
    - CL300:  
I, O, M, SM, C, T, EI, II, EO, IO, F, DB, S  
Name of print file:
    - CL500:  
I, O, M, SM, C, T, EI, II, EO, IO, DB, S  
Name of print file:

- Data modules (not PC400/CL100)  
Name of print file:
- List all forms  
Name of print file:
- List parameters  
Several forms per page                      One form per page  
Expanded    Compact

**6.3.3 Cross-reference**

The cross-reference provides information about which operand was programmed in which module and in which program line.

**Menu structure**

- Cross-reference
  - Current module file
    - Symbolic operand
      - Cross-ref. for one operand  
Operand:  
Name of print file:
      - Cross-ref. for all operands  
Name of print file:
      - Cross-ref. for several operand types  
PIC:  
I, O, M  
Name of print file:  
PC400:  
I, O, M, C, T, XI, XO, F  
Name of print file:  
PC600:  
PM, FM, EM, I, O, M, C, T, DM, DW, EI, II, EO, IO  
Name of print file:  
CL100:  
I, O, M, C, T  
Name of print file:





CL300:

PM, I, O, M, SM, C, T, DM, DW, EI, II, EO, IO,  
F, DB, S

Name of print file:

CL500:

PM, I, O, M, SM, C, T, DM, DW, EI, II, EO, IO,  
DB, S

Name of print file:

- Absolute operand

See: **Symbolic operand**

- According to symbol file (not PIC)

See: **Current module file**

- According to batch file

See: **Current module file**

**R/W**

The flag **R** or **W** specifies whether the operand is read R or written W.

### **6.3.4 Print**

Use these commands to initiate the print—out of the **print file** at the printer.

#### **Menu structure**

- Print
  - Print list file  
Name of print file:
  - Status interrogation
  - Abort
  - Select printer type  
Printer type:
  - Print file title
  - Header file

**Print list file**

The **print file** is added to the printer queue. Printing is carried out in the background. After acknowledging with

Esc
-----

the programming unit is free for other tasks.

**Abort**

The active printing operation is aborted. The aborted printing operation **cannot** be resumed and will have to be restarted if required.

**Select printer type**

This command calls up a menu of all the installed printers. The printers are installed in the **configuration program**, see technical documentation:

**PLC/DESI Utilities**  
**Professional Integrator**  
**Installation instructions**  
**P.–Nr. 4308**

When you call up the PLC Lister program for the first time, you must select a printer. Your choice of printer is displayed in the information line.

**Print file title**

The print file title consists of a maximum of **2** lines. These 2 lines are printed on every page in the **print header**.

**Header file**

The header file is placed at the top of the print file and is printed with the file. An additional comment on the project can be included in the header file.

The header file is created with the PLC Editor program as a text file *Header.TXT*.

**6.3.5 End**

See subsection **3.3.8 End**.



**Contents**

	Page
<b>7      Key functions .....</b>	<b>7-1</b>
















## Illustrations

Fig.		Page
7-1	Key functions in the Contents and in the Defaults .....	7-6
7-2	Key functions in utilities .....	7-6
7-3	Key functions for the Help function .....	7-7
7-4	Key functions for the pull-up menus .....	7-7



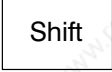


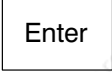
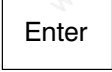
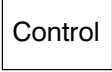



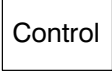






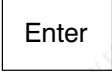





## 7 Key functions



The key functions are mostly identical in all utilities. The functions of the function keys are displayed in menus. The function keys always refer to the pull-up menu most recently opened.

Function	Key
Print out screen copy	 + 
Scroll in pages/branches	 or 
Call up module file description from parameter list	 + 
Cursor left	
Cursor up	
Cursor right	
Cursor down	
Take file name from the contents	
Detail/overview	  or  + 

Function	Key
Insert input pin at AND or OR element in FUD	Ins
Delete input pin at AND or OR element in FUD	Del
Negate input pin in FUD	N
Delete element in FUD	Del
Delete element including all preceding elements in FUD	Shift + Del
Replace/Insert	Ins
Function keys	F1 to F9
Toggle function key bar	F7 or Esc
Call up/exit Help function	F10
Command/edit levels	Esc
Start command	Enter
Insert comment line in data module	Enter
Switch to LD mode	Alt + F9
Deletes current character	Del




Function	Key
Deletes whole line/column/ network	 + 
Deletes character left of cursor	 Backspace
Start MS DOS command	
Insert network in network overview	
Network, first	 + 
Network, last	 + 
Network, next	 + 
Network, preceding	 + 
Restart	 +  + 
Jump to operand fields in FUD	
Jump to operand fields, which are empty or contain errors, in FUD	
Call up parameter list from mod- ule description	 + 
Abort pull-up menu	
Change column	

Function	Key
Jump to file/network/form/ overview list/start of directory	Home
Jump to file/network/form/ overview list/end of directory	End
Jump to start of line	Shift + ←
Jump to end of line	Shift + →
Jump to selected directory	+
TAB key	
Overview/detail	F5 F5 or Shift + 
Exit: Module description, Screen logs of loader and lister, Help function, Info status, Command, Parameter list, Memory configuration	Esc
Adopt directory/file name	+





<b>Function</b>	<b>Key</b>
Open/Close Defaults	
Close line	Enter

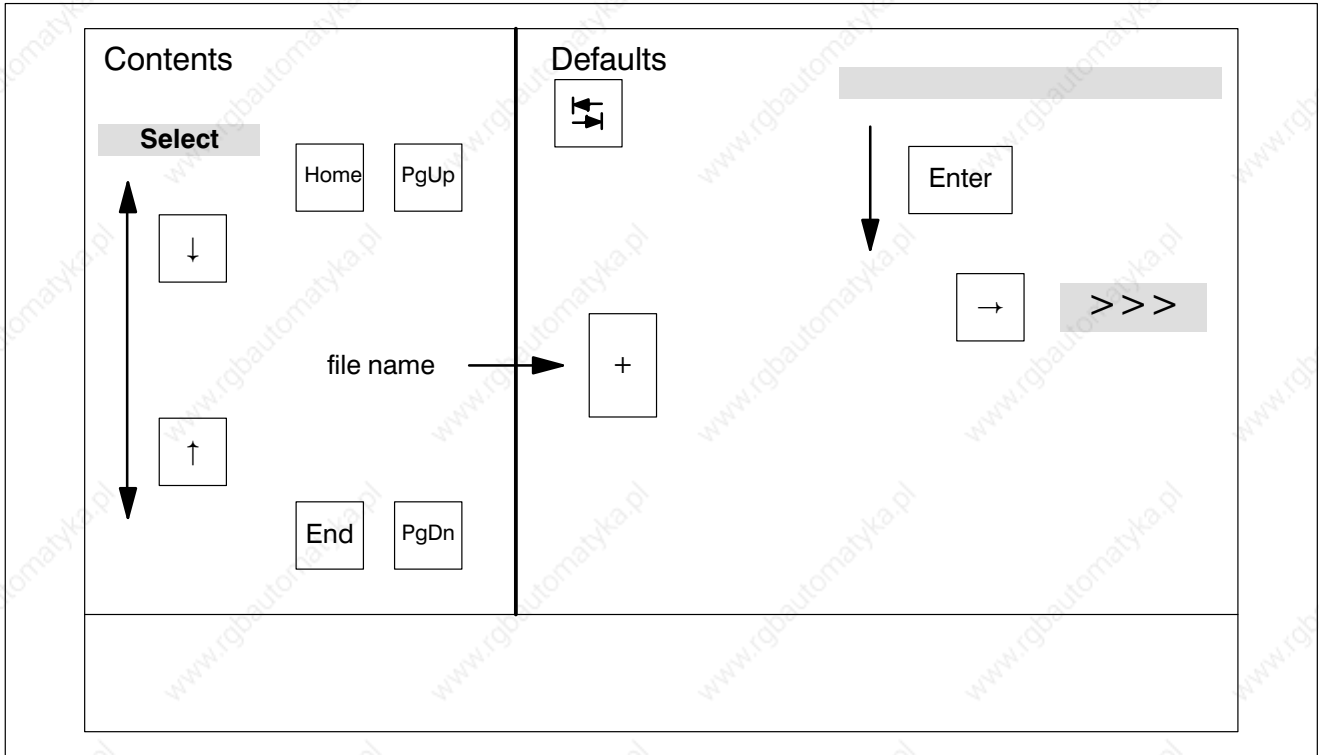


Fig. 7-1 Key functions in the Contents and in the Defaults

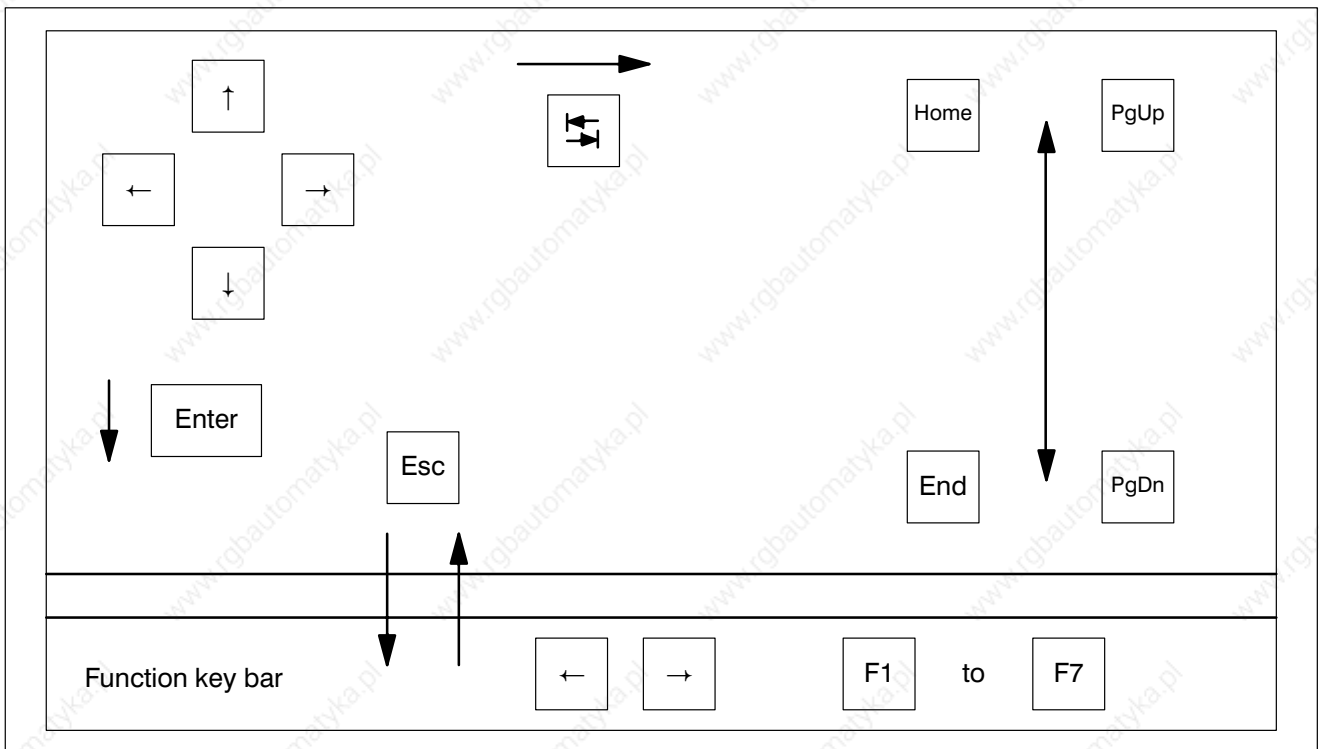


Fig. 7-2 Key functions in utilities

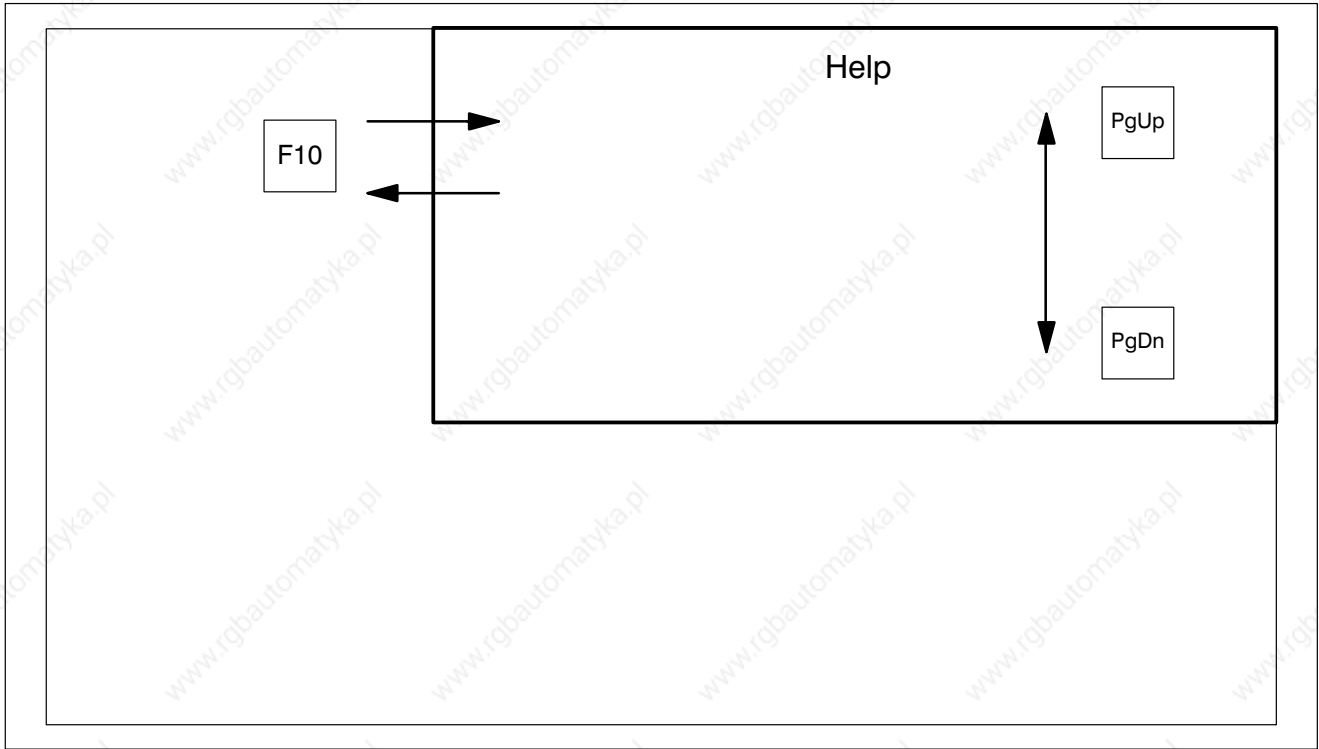


Fig. 7-3 Key functions for the Help function

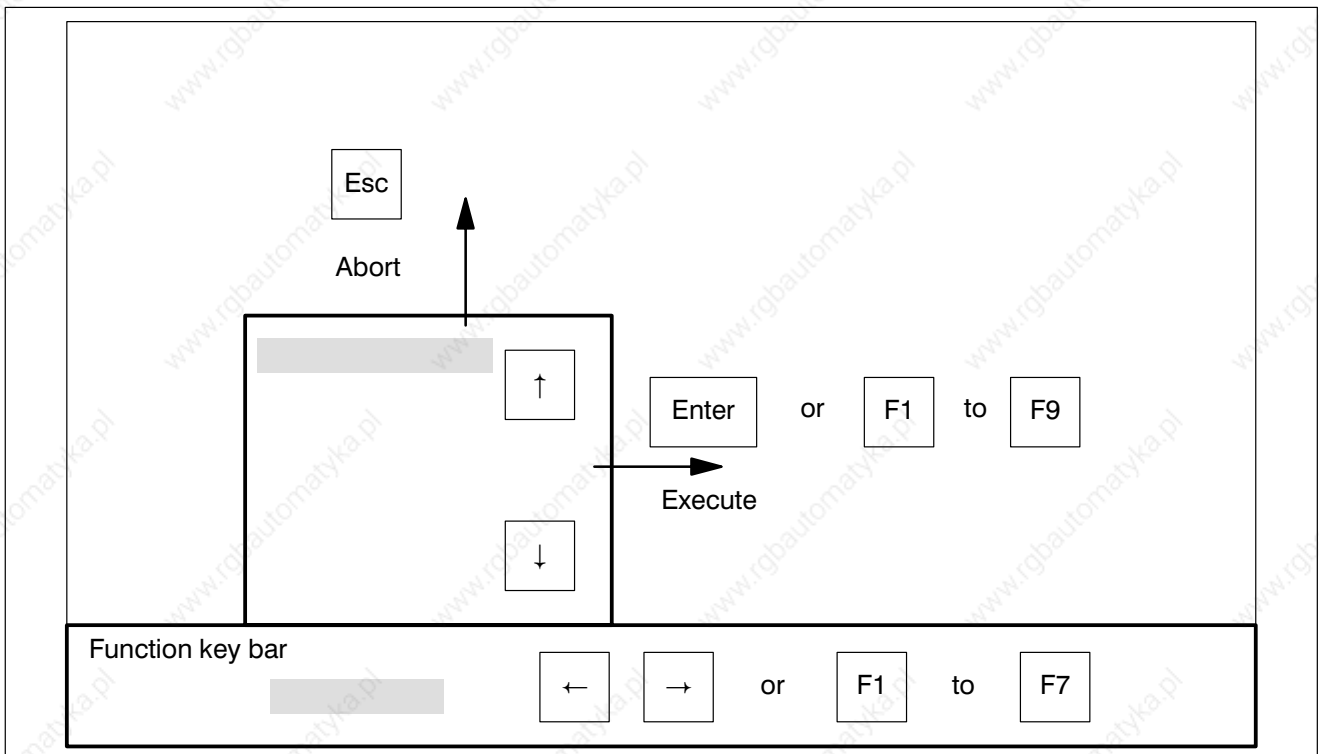


Fig. 7-4 Key functions for the pull-up menus

---

# Key functions

---



For your notes:



**Contents**

	Page
<b>A Appendix .....</b>	<b>A-1</b>
A.1 Abbreviations .....	A-1
A.2 Index .....	A-4
A.3 Alterations .....	A-12







## **A Appendix**

### **A.1 Abbreviations**

\	See: Backslash
—	Identifier of a symbolic operand description (address)
>	Prompt character, marks the input line
>	Warning after assigning
.100	Subdirectory for a project with the CL100
.200	Subdirectory for a project with PIC
.300	Subdirectory for a project with the CL300
.400	Subdirectory for a project with the PC400
.500	Subdirectory for a project with the CL500
.600	Subdirectory for a project with the PC600
.BAT	File type of a batch file
.BIB	Subdirectory for the module library
.CFG	File type for the system configuration
.COM	File type of a command to be executed
.EXE	File type of a program to be executed
.OxD	File type of the operand field file
.PRN	File type of a print file
.PxA	File type of the program file (x: see controller type)
.PxO	File type of an assigned module file (x: see controller type)
.PxT	File type of a non—assigned module file (x: see controller type)
.PxL	File type of a library module
.R5P	Log file of the transmission logs of the CL500
.S5K	File type of the SK table (only CL500)
.SPS	File type for the project status
.SxA	File type for the assignment of absolute addresses (x: see controller type)

.SxB	File type for the assignment of symbolic addresses (x: see controller type)
.SxS	File type for the contents of the symbol file (x: see control type)
.TXT	File type of a text file
	
	
Attr.	Attribute of a module in the reference list
Backslash	Character for separating directories in a path
C	Control marker
C	Counter
DB	Data buffer
Directory	Contents of floppy disk, hard disk or RAM disk
DW	Data word
DM	Data module
E	Error
EI	Extended input
EM	Extended module
EO	Extended output
ESD	Electrostatic discharge, standard abbreviation for all descriptions relating to electrostatic discharges, e.g. ESD protection
Ext.	Operand extension/Parameter extension
F	Data format
FM	Function module
FUD	Function diagram
I	Input
II	Interface input
IL	Instruction list
Ind	Indirect address in the parameter list
IO	Interface output
LD	Ladder diagram
M	Marker





O	Output
OM	Organisation module
Para.	Parameter number
Path	Path through consecutive directories
PB	Program branch
PI	Program instruction line
PIC	Programmable Integrated Control
PM	Program module
R/E	RAM/EPROM
RLO	Result of logic operation
S	Sign
S	System range
SM	Special marker
T	Time
X	Marks setting in operand field
x	Controller type
	x: 1 CL100
	2 PIC
	3 CL300
	4 PC400
	5 CL500
	6 PC600
XI	Extended input (only PC400)
XO	Extended output (only PC400)
ZS	Control unit

## A.2 Index

### Symbols

.100, A-1  
 .200, A-1  
 .300, A-1  
 .400, A-1  
 .500, A-1  
 .600, A-1  
 .BAT, A-1  
 .BIB, 3-18, 3-62, A-1  
 .CFG, A-1  
 .COM, A-1  
 .EXE, A-1  
 .OxD, 4-17, 4-21, A-1  
 .PLC, 1-23, 1-26, A-1  
 .PRN, 6-7, A-1  
 .PxA, 1-25, 3-26, 5-2, 5-12, 5-16, A-1  
 .PxL, 3-18, 3-62, 5-12, A-1  
 .PxO, 1-24, 3-17, 3-63, 5-2, 5-12, 5-16, A-1  
 .PxT, 1-24, 3-17, A-1  
 .R5P, A-1  
 .S5K, 5-18, A-1  
 .SxA, 1-25, A-1  
 .SxB, 1-25, A-1  
 .SxS, 1-25, 5-12, A-2  
 .TXT, A-2  
 .ZSx, 3-63  
 ;, 3-11  
 &, 3-42, 3-44, 3-60  
 \$132, 6-10  
 \$80, 6-10  
 \$ABS, 6-10  
 \$AWL, 6-10  
 \$FUP, 6-10  
 \$KPL, 6-10  
 \$LIST, 6-10  
 \$NOLIST, 6-10  
 \$P, 3-11, 6-10  
 \$PZ+, 6-10  
 \$PZ-, 6-10  
 \$SBW+, 6-10  
 \$SBW-, 6-10  
 \$SK+, 6-10  
 \$SK-, 6-10  
 \$SUZ+, 6-10  
 \$SUZ-, 6-10

\$SYM, 6-10  
 -, 3-11, A-1  
 -- I, 3-32  
 -- () --, 3-33  
 -- ] [ --, 3-32  
 -- ]/[ --, 3-32  
 -I, 3-46  
 ★, 0-VI, 4-19  
 =0-, 3-42, 3-46  
 =1, 3-42  
 <, 3-60  
 >, A-1  
 >>, 3-11, 3-60  
 >>=1, 3-42, 3-45  
 \, A-1

### Numbers

25, - lines, 3-23, 3-25  
 43/50, - lines, 3-23, 3-25

### A

Abort, 6-14, 7-7  
 - pull-up menu, 7-3  
 Absolute  
 - address, 1-25, 2-24, 3-17, 4-5, A-1  
 - operand, 3-17, 3-40, 3-62, 4-11, 4-18, 6-10  
 According to batch file, 3-19  
 Activate printer, 6-7  
 Adapter, 5-7  
 Additive, - load of a module into the controller, 5-14  
 Additive loading, 5-14  
 Adopt  
 - directory name, 7-4  
 - file name, 7-4  
 Allocation, 3-50  
 AND element, 3-41, 3-42, 3-44, 3-45, 3-47  
 ASCII, 3-68  
 - character, 3-4, 3-68, 3-71, 3-73  
 Assign, 1-24, 1-25, 2-2, 2-24, 3-17, 3-65, 3-74, A-1  
 Attr., 5-17, A-2  
 Attribute, A-2  
 AUTOEXEC.BAT file, 1-3, 1-26

**B**

Backslash, A-2  
Backspace, 1-29, A-2  
Backwards, 3-22  
Backwards counter, 3-42  
Batch file, 1-26, 3-1, 3-74, A-1  
Beginning of a branch, 3-31  
Beginning of line, 1-35  
Bit, 4-18  
Bit combination, 4-11  
Bit input, 3-42  
Block, 3-14  
Block start marker, 3-14, 3-15  
Bosch, 5-17  
BOSCH.BIB, 2-7, 3-18, 3-62  
Branch, 3-43, 3-51  
Buffer, 3-14, 3-15, 3-57  
Byte, 4-18  
  - left, 4-18  
  - right, 4-18  
Byte address, 6-10

**C**

C, 4-19, 4-20, A-2  
  - form, 3-64  
Call up  
  - help function, 1-32, 7-2  
  - module file description, 3-61, 7-1  
  - network, 3-57  
  - parameter list, 3-25, 3-58, 7-3  
Carry, 4-11  
CD, 3-50  
Central processing unit (CPU) ZE300/ZE301, 5-8  
Change, 3-28, 4-9  
Change column, 7-3  
Change to another program module, 3-26  
Changing  
  - between IL, LD and FUD, 3-3  
  - between module, symbol and text file editor,  
    3-2  
Character set, 6-1  
Character string, 3-21  
Characters, - per line, 6-10  
Check, 3-36  
CL100, 3-34, 3-66, 4-6, 4-18, 5-7  
CL300, 3-12, 3-34, 3-59, 4-18, 5-8  
CL500, 1-23, 3-12, 3-34, 3-42, 3-59, 4-18,  
  5-9

Close, 1-20  
  - defaults, 7-5  
Close line, 7-5  
Closed circuits, 3-32  
CMC, 3-51  
CMCI, 3-51  
Code number, 3-12  
COM1, 6-2  
Command, 1-9, 1-17, 1-27, 3-23, 3-35, 4-7  
Command level, 3-3, 4-2, 7-2  
Comment, 3-9, 3-11, 3-60, 3-65  
Comment line, 2-23, 3-38, 3-68, 4-18  
Comments, 3-68  
Communication with the controller, 4-2  
Compact/expanded, 3-20  
Comparator, 3-42, 3-48  
Compare, 5-16  
Comparison. See Comparator  
Complex elements, 3-46  
CONFIG.SYS file, 1-3, 1-26  
Configuration, 2-25, 5-17  
Configuration program, 6-1, 6-14  
Connection, 3-32  
  - of programming unit --> controller, 5-4  
Connection cable  
  - K6, 6-3, 6-4  
  - K7, 2-3, 2-27, 5-4, 5-7, 5-8, 5-9  
  - K8, 5-4, 5-6  
Connection line, 3-40, 3-43, 3-44, 3-45  
Connection module, 5-6  
  - P401/P600, 5-6  
Contact field, 3-30  
Contents, 1-11, 1-12, 1-13, 1-17, 1-18, 1-22,  
  1-23, 1-24, 1-25, 7-6, A-2  
Continue, - print-out, 6-10  
Control, 4-3  
Control marker, 4-19, A-2  
Control sequence, 6-10  
Control unit, 1-23, 2-7, 3-13, A-3  
Controller type, 1-22, 1-24, 1-25, 2-6, A-3  
Copy, 1-13, 3-15  
  - between different files, 3-15  
  - network, 3-56  
Count  
  - down, 3-50  
  - up, 3-50  
  - value, 3-50

- Counter, 3–50, 4–18, A–2
  - backwards, 3–50
  - forwards, 3–50
  - forwards/backwards, 3–50
- Counter element, 3–41
- Counter output, 3–50
- Cross–reference, 6–12
- CU, 3–50
- CU&CD, 3–50
- Cursor, 1–35
- CV, 3–50
- D**
- Data, 3–68
- Data buffer, 4–18, A–2
- Data field, 3–68
- Data format, 3–68, 4–19, A–2
- Data module, 1–24, 2–31, 3–25, 3–64, 3–66,  
4–15, 5–2, 5–12, A–2
  - overview list, 3–69, 4–16
- Data module form, 3–66, 3–67
- Data module header, 2–22, 3–66, 3–67, 4–15
- Data module length, 2–22, 3–67
- Data type, 3–68, 3–74, 5–18
- Data word, 4–18, A–2
- DB, A–2
  - form, 3–64
- Decimal code, 3–71, 3–73
- DEF instruction, 3–18, 3–26, 3–62
- Defaults, 1–11, 1–13, 1–19, 1–20, 2–2, 2–5, 2–6,  
3–26, 7–6
- Delete, 1–35, 3–14, 3–36
  - character left, 7–3
  - current character, 7–2
  - element, 3–43, 7–2
  - input pin, 3–41, 3–45, 7–2
  - network, 3–56
  - whole line/column/network, 7–3
- Delete screen, 3–52
- Detail, 3–23, 3–25, 3–69, 4–7, 4–8, 7–1, 7–4
- Directory, A–2
- Directory name, 1–13, 1–18
- Disk Info, 1–14
- Disk name, 1–12
- Display, 4–20
- Display level, 4–2
- DM, A–2
  - form. See Data module form
- Docking point, 3–40, 3–44
- Drive, 1–12
  - Info, 1–14
- Drive name, 1–17
- Dual operand instruction, 3–9
- DW, A–2
- E**
- E, A–2
- (E)EPROM, 5–15
  - connection socket, 5–15
  - module, 5–2
- Edit, 2–2, 4–20
  - data module header, 2–22, 3–20, 3–67
  - module file, 2–8
  - network title, 3–6, 3–55
  - parameter list, 3–25
  - symbol file, 2–19
- Edit field, 3–3, 3–8, 3–9, 3–30
- Edit level, 3–3, 7–2
- Edit title, 3–25
- Editing functions, 1–35
- Editor, 2–2, 3–1, 3–26
- EI, A–2
  - form, 3–64
- Elapsed time, 3–49
- Element field, 3–40, 3–42, 3–44, 3–45
- EM, A–2
  - form, 3–64
- EMC, 3–51
- End, 3–26
- End of line, 1–35
- End of module, 3–51
- End of the branch, 3–31
- End sequence, 6–1
- Enter, – defaults, 1–19, 2–6
- Entries, 0–VI
- EO, A–2
  - form, 3–64
- EP/AG module, 2–3, 2–27, 5–4, 5–6, 5–7, 5–8,  
5–9, 5–15
- Eraser, 3–35
- Error, 4–18
- Error message, 1–16, 3–10, 3–11, 3–31
- Error text, A–2
- ESD, A–2
  - safety measures, 5–15
- ET, 3–49



EXAMPLE.P3A, 2–2

Execute, 7–7

Exit

- help function, 1–32
- PLC, – utilities, 1–6

Exit Help function, log, ..., 7–4

Exit utilities, 3–26

- help function, 7–2
- module file description, 3–58
- parameter list, 3–58

Ext., 3–59, 4–18, A–2

Extended input, A–2, A–3

Extended module, A–2

Extended output, A–2, A–3

Extension, 3–59, 4–11, 4–18

## F

F, 3–68, 4–19, A–2

- form, 3–64

F10, 1–15

File, 1–14

- information, 1–14

File management, 1–12, 1–16

File name, 1–13, 1–18, 3–74, 7–6

File type, 1–23, 1–24, 1–25, 1–26, 3–15, 3–17,  
5–12, A–1

Filename, 1–26

Flag, 4–11

Flip–flop, 3–48

Floppy disk drive, 0–VII

FM, A–2

- form, 3–64

Forwards counter, 3–42

Forwards/Backwards counter, 3–42

FUD, 3–1, A–2

- display, 2–17, 3–5
- mode, 3–13
- monitor, 4–14

Function block, 1–11

Function diagram, 3–1, 3–23, 6–10, A–2

See also FUD

Function key, 1–27, 7–2

Function key bar, 1–6, 1–9, 1–11, 1–16, 3–8,  
3–29, 3–30, 3–36, 3–37, 3–53, 7–6, 7–7

Function module, A–2

## G

Generate, – library module, 3–18

## H

H, 4–6

Hard disk, 0–VII

Header file, 1–26, 6–5, 6–14

Help, 7–7

See also help function

Help function, 1–15, 1–32

Help register, 4–6

Hyphen, 3–11

## I

I, A–2

- form, 3–64

Ignore UPPER/lower case, 3–22

II, A–2

- form, 3–64

IL, 3–1, A–2

- display, 3–5, 3–41
- mode, 3–13
- monitor, 4–11

IN, 3–49

Ind, 3–60, A–2

Indirect address, 3–60, A–2

Info, 4–6, 5–16

Info status, 5–2, 5–16

Information, 1–14

Information line, 3–8, 3–13

Input, 3–49, 4–18, A–2

Input bit, 4–11

Input form, 2–20

Input line, 1–13, 1–17, 1–18

Input parameter, 3–60

Input pin, 3–40, 3–41, 3–44, 3–45, 3–46, 3–48

Insert, 1–35, 3–13, 3–36, 7–2

- comment line, 7–2
- element, 3–42, 3–44, 3–45
- file name, 7–1
- input pin, 3–41, 3–43, 7–2
- network, 3–56, 7–3
- symbol file, 3–20

Instruction line, 3–9

Instruction list, 3–1, 3–23, 3–29, 4–7, 6–10, A–2

Instruction part, 3–9

Interface

- input, A–2
- output, A–2

Interface X31, 2–3, 5–8, 5–9

Interrupt, – print–out, 6–10

IO, A–2

- form, 3–64

## J

JPC, 3–51

JPCI, 3–51

Jump

- to directory, 7–4
- to end of line, 7–4
- to file/network/end..., 7–4
- to file/network/start..., 7–4
- to first network, 3–5
- to last network, 3–5
- to next network, 3–5
- to start of line, 7–4

Jump destination, 3–51

Jump to, – operand field, 7–3

## K

K6. See Connection cable K6

K7. See Connection cable K7

K8. See Connection cable K8

Keyboard, 0–VI

Keys, 0–VI

## L

Ladder diagram, 3–1, 3–23, 3–29, 4–7, 6–10

See also LD

Language translation, 3–3

LD, 3–1, 3–50, A–2

- display, 2–17, 3–5
- mode:
  - off, 3–13
  - on, 3–13
  - monitor, 4–13

LD mode:

- off, 3–29
- on, 3–29

Library module, 1–21, 3–18, 3–62, 5–12, A–1

Line comment, 3–9, 3–38, 4–11, 6–10

Line creator, 3–35

Lines, – per page, 6–1

Link, 2–2, 2–25, 3–62

Linking, 3–65, 5–12

List parameters, 6–5, 6–9, 6–10

Lister, 3–26, 6–1

- commands, 6–9

Load, 2–2, 2–27, 3–50

- PIC program, 5–2

- PLC program into the controller, 2–25, 5–10

Loader, 2–2, 2–25, 3–26, 5–1, 5–3

- + monitor, 3–26
- commands, 5–11

Loading, 5–13

Log file, A–1

Log loader, 5–18

LPT1, 6–2, 6–3, 6–4

## M

M, A–2

- form, 3–64

Main directory, 1–3, 1–12

Main memory info, 1–14

Main menu, 1–2, 1–4, 1–5, 1–6, 1–9, 1–11, 1–19, 2–2, 2–5, 3–1, 4–1, 5–3, 6–1

Marker, 4–18, A–2

Memory configuration, 2–2, 5–18

Messages, 1–11, 1–16

Module, 3–51, 6–9

Module call, 3–4, 3–23, 3–25, 3–43, 3–51

Module file, 1–21, 1–24, 2–2, 2–24, 3–1, 3–13, 5–12, A–1

- editor, 3–4
- editor FUD, 3–37
- editor IL, 3–9
- editor LD, 3–29

Module file description, 3–1, 3–23, 3–58, 3–61

Module file name, 1–24, 2–7

Module library, 3–62, A–1

Module list, 3–4, 3–64, 3–65, 5–12

Module number, 5–18

Monitor, 2–2, 3–26, 4–1

Monitor display, 4–11

Monitor field, 4–11, 4–13

MS DOS

- command, 1–17
- utilities, 1–7
- version, 0–VII

## N

Negate, – input pin, 3–42, 3–43, 7–2

Negated, 3–47

Negative, 4–11

Network, 2–10, 3–5, 3–25, 3–52, 7–3

- overview/detail, 3–6, 3–54

Network contents, 3–52

Network display, 3–25

Network header, 3–8



Network overview, 2–10, 3–1, 3–25  
Network title, 2–10, 3–6, 3–25, 3–52  
Normally closed contact, 3–32  
Normally open contact, 3–32

**O**

O, A–2  
– form, 3–64  
OM, A–2  
– form, 3–64  
OM1.P3O, 2–2  
OM1.P3T, 2–2  
Open, – defaults, 1–20, 7–5  
Operand, 3–42, 3–48, 4–18  
– See *also* Address  
– optional, 3–49, 3–50  
Operand description. See Operand  
Operand extension, 4–18, A–2  
Operand field, 3–40, 3–41, 4–17  
– file, 4–21, A–1  
Operand form, 3–64, 3–70  
Optimise, 3–53  
Optional, – operand, 3–41  
OR element, 3–41, 3–42, 3–44, 3–45, 3–47  
Organisation module, 1–21, 2–8, 2–19, A–2  
Output, 3–49, 3–50, 4–18, A–2  
Output bit, 3–48, 4–11  
Output command, 3–31, 3–33  
Output element, 3–40, 3–43, 3–44, 3–47, 3–48  
Output field, 3–30  
Output form, 2–21  
Output parameter, 3–60  
Output pin, 3–41  
Overflow, 4–11  
Overview, 3–23, 3–25, 3–69, 4–7, 4–8, 7–1, 7–4

**P**

PAA, 4–12  
Page  
– down, 1–32  
– up, 1–32  
Page feed, 3–11, 6–10  
Para., 3–59, A–3  
Parallel interface, 1–3, 6–2  
Parameter, 3–25, 3–59  
Parameter extension, 3–59, A–2  
Parameter list, 3–1, 3–23, 3–58, 3–62, A–2  
Parameter number, 3–59, A–3  
PARENT, 1–13, 1–18

Parent directory, 1–12, 1–13  
Park, – hard disk, 1–7  
Path, 1–11, 1–12, A–3  
PB, A–3  
PC400, 3–12, 3–34, 3–66, 4–18, 5–6, 5–14  
PC600, 3–12, 3–59, 4–18, 5–6  
Permitted, – characters, 3–4  
PG, 1–14  
– information, 1–14  
– X31. See Interface X31  
PI, 3–13, A–3  
PIC, 3–37, 5–6, A–3  
PLC  
– editor program, 1–9, 1–13  
– lister program, 1–9, 1–13  
– loader program, 1–9, 1–13  
– monitor program, 1–9, 1–13, 2–28  
– programming example, 0–VI  
– programming interface, 5–6, 5–7, 5–8, 5–9  
– utilities, 1–2, 1–3, 1–4  
– structure, 1–8  
PM, A–3  
– form, 3–64  
PM1.P3O, 2–2  
PM1.P3T, 2–2  
Preset  
– time, 3–49  
– value, 3–50  
Press, – keys simultaneously, 0–VI  
Print, 3–74, 6–13  
Print file, 1–26, 6–13  
Print file title, 6–5, 6–14  
Print header, 6–5, 6–14  
Print list file, 6–14  
Print out screen copy, 7–1  
Printer connection, 6–3  
Printer type, 6–1  
Priority  
– absolute values, 3–19  
– symbols, 3–19  
PRODIR, 1–13, 1–18, 2–7  
Professional integrator, 1–3, 1–6  
PROFI. See Professional integrator  
Program address, 4–12  
Program branch, 3–13, 3–38, A–3  
Program file, 1–21, 5–12, A–1  
Program file name, 1–25, 2–7  
Program instruction line, 3–13, A–3

Program module, 1–21, 2–8, 2–10, 2–19, 3–65, A–3

Programmable Integrated Control. See PIC

Programming unit, 1–14

Project directory, 1–13, 2–7

Project name, 1–21, 1–22, 2–6

Project status, 1–23, 2–7, A–1

Prompt character, A–1

Prompt symbol, 1–17

PT, 3–49

Pull-up menu, 1–27

PV, 3–50

## Q

Q, 3–48, 3–49, 3–50

## R

R, 3–49, 3–50

R/E, 3–65, A–3

R/W, 6–13

Reference list, 5–2, 5–12, 5–17, 5–18, A–2

Register, 4–11

Register display, 4–5

Rename, – element, 3–42, 3–44, 3–45

Replace, 1–35, 3–13, 4–4, 7–2

Reset, 3–49, 3–50

Restart, 7–3

Result, A–3

Result bit, 4–11

Return, 3–29, 3–36, 3–37, 3–53, 4–21

– to MS DOS, 1–7

RG, 3–13

– number, 6–10

RLO, 3–43, 4–11, A–3

Root directory, 1–12

Rotating bar, 4–2

RS flip-flop, 3–43, 3–48

Rules for

– FUD, 3–38

– LD, 3–31

## S

S, 3–68, A–3

– form, 3–64

Screen, 0–VI

Screen display, 1–11

Screen layout, 1–8, 3–7

Screen mode, 3–23, 3–25, 4–7, 4–8

Scroll, – contents, 1–13

Scroll in pages/branches, 7–1

Scroll through networks, 3–5

Scrolling, – the screen, 4–12

Search, 3–21

Select, – printer type, 6–14

Select and create, – print file, 6–7

Select info, 1–14

Selection bar, 1–13, 1–18

Semicolon, 3–11, 3–65

Serial interface, 6–2

Set, 4–19

– list parameters, 6–6

Setting, 2–28, 4–10

SF, 3–49

Shift

– down, 3–35

– right, 3–35

Sign, 3–68, A–3

Single operand instruction, 3–9

SK table, 5–2, 5–18, A–1

SK500 system coordinator, 5–9, 5–18

SM, A–3

– form, 3–64

Software dongle, 1–3, 2–5, 6–4

SP, 3–49

SPE, 3–49

Special character, 3–4

Special marker, 4–18, A–3

SR, 3–49

SR flip-flop, 3–43, 3–48

SRE, 3–49

ST, 3–49

Start

– command, 7–2

– MS DOS command, 7–3

Start pulse, – extended, 3–49

Start sequence, 6–1

Start time as

– falling delay, 3–49

– pulse, 3–49

– raising delay, 3–49

– raising delay extended, 3–49

Stop, – time, 3–49

Stop time, 3–49

Store, 3–14

– and delete, 3–15

Storing defaults, 1–26

SUBDIR, 1–13, 1–18





- Subdirectory, 1–12, 1–22, 1–23, A–1
- Switch to, 7–2
  - module file description, 3–58
  - parameter list, 3–58
- Symbol, 0–VI, 3–20, 3–60, 3–67, 3–70, 6–11
- Symbol column, 3–4
- Symbol comment, 6–10
- Symbol file, 1–21, 1–24, 1–25, 2–2, 2–19, 2–24, 3–1, 3–18, 3–62, 5–12
  - editor, 3–4, 3–64
- Symbol file name, 1–24, 2–7
- SYMBOL.S3S, 2–2
- Symbolic, 6–10
  - address, 1–24, 1–25, 2–24, 3–11, 3–17, A–1
  - module file name, 3–65
  - operand, 1–24, 3–4, 3–17, 3–25, 3–40, 3–59, 3–60, 4–11, 4–18, A–1
- System, – messages, 1–16
- System configuration, 5–18, A–1
- System parameter, 5–18
- System range, 4–18, A–3
- T**
- T, A–3
  - form, 3–64
- TAB key, 7–4
- TAB menu, 1–30
- Take over
  - directory name, 1–20
  - file name, 1–20
- Test, 2–2
  - PLC program, 2–28
- Text file, 1–21, 3–1, A–2
  - editor, 3–71
- Text file name, 1–26
- Time, 3–49, 4–18, A–3
- Time element, 3–41, 3–43
- Time grid, 3–12
- Time programming, 3–12
- Time/counter module, 3–34
- Toggle, 7–2
  - between IL, LD and FUD, 4–2
- Toggle between display and command levels, 4–2
- Toggle function, 1–29
- Trace, 4–5
- Trace module, 4–6
- Transmission log, A–1
- Type, 3–68, 3–70
- U**
- Unload, – PIC program, 5–2
- Unloading, 5–14
- User guide, 1–2
- V**
- Value, 4–19
- Version, 1–1, 1–11, 1–16
- Version/messages, 3–8
- Volume, 1–12
- W**
- Warning, 3–10, 3–11
- Whole words only, 3–22
- Word, 3–68, 4–18
- Word address, 6–10
- Word input, 3–48
- X**
- X, 4–19, A–3
- x, A–3
- X31. See interface X31
- XI, A–3
  - form, 3–64
- XO, A–3
  - form, 3–64
- XOR. See XOR element
- XOR element, 3–42, 3–44, 3–45, 3–47
- Z**
- Zero, 4–11
- ZS, 1–23, A–3

## A.3 Alterations

In this revised edition E6, alterations have been made to the former edition E5 on the following pages.

- 0–VII  
1–1 to 1–3, 1–5, 1–9, 1–11 to 1–15, 1–17, 1–19, 1–21, 1–26,  
1–27, 1–32, 1–35  
2–5, 2–7 to 2–18, 2–23, 2–24, 2–26, 2–28 to 2–30  
3–1 to 3–3, 3–5 to 3–10, 3–13, 3–17, 3–18, 3–20, 3–23, 3–24,  
3–25, 3–27, 3–29, 3–31, 3–33, 3–34, 3–37 to 3–63, 3–65 to  
3–69, 3–71, 3–74  
4–1 to 4–3, 4–5 to 4–8, 4–11, 4–14 to 4–17, 4–19, 4–21  
5–1, 5–2, 5–12, 5–16 to 5–18  
6–1, 6–7, 6–9, 6–10  
7–1 to 7–4

All altered paragraphs or diagrams are marked by a correction bar. Changes to diagrams are additionally identified by the following symbol.



