B&R PROgramming SYStem
User's Manual

Version: **2.00** (February 1992)
Published By: Bernecker und Rainer Industrie-Elektronik GmbH
Model. Nr. MAPROSYS-0

This Description refers to the
B&R PROgramming SYStem Version 5.40

The information contained herein is believed to be accurate as of the date of publication, however, Bernecker und Rainer Industrie-Elektronik GmbH will not be liable for any damages, including indirect or consequential, or reliance on the accuracy or this documentation. The information contained herein is subject to change without notice.

TABLE OF CONTENTS

Table of Contents:

Chapter 1 General Information

Chapter 2 Software Installation

Chapter 3 Introduction

Kapitel 4 Ablaufdefinition / Kontaktplan Logikplan / Funktionsplan

Kapitel 5 Anweisungsliste

Kapitel 6 Klartext-Editor

Kapitel 7 Tabellen-Editor

Kapitel 8 Laden, Speichern, ...

Kapitel 9 Drucken

Kapitel 10 Erstellen von Funktionsblöcken

Kapitel 11 Online-Programmierung über Modem

Kapitel 12 Online Schnittstellen - Hardware

Anhang A Zusammenfassung - Bedienbefehle

Anhang B Übersicht - AWL Befehle

Anhang C Zusatzeinrichtungen

Anhang D Fehlermeldungen

Anhang E Stichwortverzeichnis

CHAPTER 1
GENERAL INFORMATION

Contents:

1. GENERAL

The efficiency of the PLC-System has been enhanced by the user-friendliness of the Programming System and the available Standard Software. With the PROgramming SYStem B&R offers a unique software package for MS-DOS that is efficient and comfortable to operate.

All B&R PLC's, from the MINICONTROL to the MULTICONTROL, can be programmed with the B&R PROgramming SYStem. The PROgramming SYStem is also multilingual, leaving the programming language used, completley up to the user. The B&R PROgramming SYStem has unlimited resources when it comes to solving problems with ladder diagrams (LAD), function blocks (FBK), logic diagrams (LD) or statement lists (STL). The PROgramming SYStem also has no problems mixing programming languages in a programm.

A voluminous selection of Standard-Function blocks supports programming application and shortens Software-Developement time considerably. Additionally a FBK-Editor makes life easy for the applicator with application specific Function Blocks.

The B&R PROgramming SYStem speaks many languages and can switch between German, English, Italian, Spanish and French with a single command. Naturally the PROgramming SYStem User's Handbook is also just as flexible when it comes to languages.

THE PC

The B&R PROgramming SYStem will run with any IBM AT/XT compatible Personal Computer (with MS-DOS Version 2.11 or above), including battery operated laptops and Portable ATs to the powerful System 2 Office computers. For speed efficiency in Online-operation, data is sent at 60 Kbaud through CENTRONICS interfaces and special cabel to the PLC. For Online work with more than one processor from the same programming unit, Two Port Interface Modules are available.

LADDER DIAGRAMS (LAD)

Ladder Diagrams are especially effective for logic programms. The LAD-Editor's menu technique facilitates the comfortable application of Ladder Diagrams. The LAD debugger permits the "online" observation and influencing of the program. When the contact is in a high state it is displayed inversley. Naturally visible comments and printing capablilities are also at the user's service.

FUNCTIONAL PLAN (FP)

Functional plan programming is a type of programming that approaches structured programming. Program units are represented as rectangles on the screen and the programmer only needs to connect the input parameters and the outputs of the module. A collection of over 100 standard function blocks - e.g., analog processing, mathematic functions, general data processing, communication, and hardware support - is available from B&R. In addition, the PROgramming SYStem permits the user to create his/her own application-specific function blocks.

LOGIC PLAN (LP)

Logic plan programming can be seen as a complement to the ladder diagram. AND/OR/XOR connections are drawn using logic blocks very similar to logic gates used in digital engineering. The size and number of the inputs can be determined arbitrarily. Logic plan symbols can be freely combined with LAD elements and function blocks.

STATEMENT LIST (STL)

Statement list programming facilitates the solution of complex problems either in original Motorola® mnemonics or in B&R-specific command abbreviations. STL is more than assembler programming. The command set includes elementary processor commands (6303, 6809) to complex floating point arithmetic routines that are integrated as standards in all B&R systems.

FBK-EDITOR

The FBK Editor enables the user to create his/her own application-specific function blocks. The first step establishes the form of the module, i.e., the number of inputs and outputs. The functionality of the FP is then programmed with an STL-program. Naturally the FBK Editor also permits the editing of standard modules and their modification to meet specific requirements.

SYMBOL ASSIGNMENT (SYMB)

Symbols are identifiers assigned to an input, output or storage. The user can either use the physical address of the storage or the symbolic name. The B&R PROgramming SYStem automaticlly and interactively complements the missing information. The SYMB list can be displayed or printed completely or selectively for a group of registers, flags, inputs or outputs.

DATA TABLES

Data tables are constant data records that the user can access. Input is handled by a comfortable Data table Editor. Decimal, binary, hexadecimal and ASCII number formats can be mixed within a table or even within a row of a table. Tables are components of the application program and are stored in the non-volatile protected application program storage.

PROGRAM DOCUMENTATION

Particular attention was devoted to the documentation capabilities of the B&R-PROgramming SYStem. Comments can be written at any location in a program, in STL as well as LAD/FP/LP. The program printout contains the complete program documentation, including cross references, symbol tables, function modules and symbol assignments.

DEBUGGING

A number of powerful tools are available to aid in locating errors. The STL debugger permits the stepwise execution of STL programs, the setting of break points, and the use of trace functions. The LAD debugger inversely displays set contacts and superimposes current values within the scope of function blocks. The utilization of windows gives the advantage of simultaneous monitoring of multiple processors.

2. B&R PROgramming SYStem and PG1000 compatibility

Neither **A** or **B** levels of programming software are completley compatible with the **PG1000**.

PROgramming SYStem level C differentiates between the **PG1000 and PG-PC Modes.**

Starting the software in PG1000 Mode provides complete compatibility between the PG1000 and the PC generated programs. This enables the exchange of programs or program sections between PC and PG1000 via serial interface.

Switching from PG1000 to PG-PC Mode

A	B	C	PG1000	CP80	PG-PC	PC80
		●	●	●		

The operating command ***PC*** **[↵]** gives access to functions additional to those in the PG1000.

The large number of additional functions enable much more efficient and comfortable programming with a personal computer (sequence charts, logic diagrams,...)

Programs or macros generated in PG-PC Mode can be loaded and used only in this working mode. They are not compatible with PG1000 Mode.

3. IMPORTANT NOTES ON THIS MANUAL

Prerequisites

It is necessary for the user to know the basic facts about his/her personal computer and the B&R control system (MULTI, MIDI or MINICONTROL) to utilize this manual properly.

For more detailed information about the operating system of the PC, refer to the corresponding user manual.

For more detailed information about the control system, refer to the relevant hardware manuals (distributed by B&R) or contact a B&R engineer.

How to use this manual

Chapter 1 - GENERAL INFORMATION and **Chapter 2** - INSTALLATION: We recommend that this section be read in any case.

Chapter 3 - INTRODUCTION TO USAGE: is a start with the basics and fundamentals and a good introduction to the B&R PROgramming SYStem .

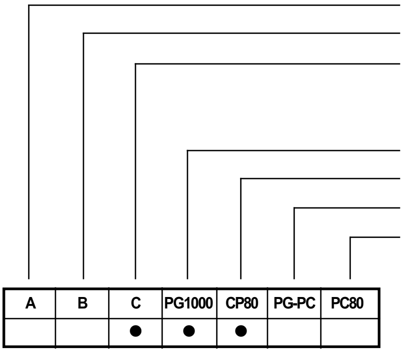
Chapter 4 - **Chapter 10** - is a reference guide for users with a certain amount of basic knowledge with regard to programming software. These chapters comprise detailed descriptions of all programming options and softkey functions.

Chapter 11 - ONLINE PROGRAMMING OVER A MODEM describes all necessary functions for programming a B&R-controller over a modem.

Chapter 12 - ONLINE INTERFACES lists the various hardware possibilities, and connections involving your PC and the B&R-controller.

Technical Appendix - This chapter is a reference for all commands and instructions necessary for programming.

This user's manual describes three distinct levels of the PROgramming SYStem in conjunction with their operation possibilities, and this table shows which operation function fits into which expansion level respectivley.



Expansion levels:

- Level A
- Level B
- Level C

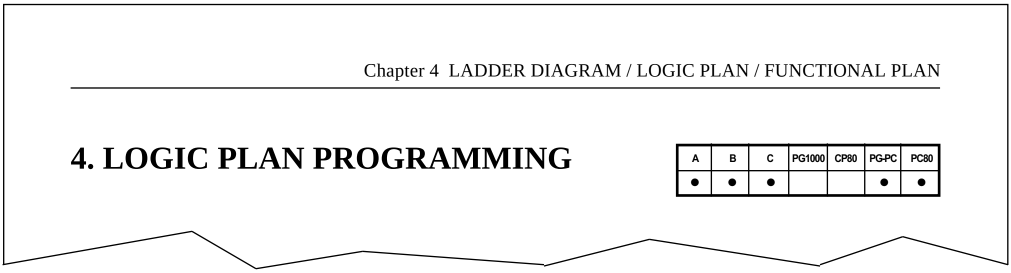
Operation:

- PG1000 Mode, CPU Type A
- PG1000 Mode, CPU Type B
- PC Mode, CPU Type A
- PC Mode, CPU Type B

A	B	C	PG1000	CP80	PG-PC	PC80
		●	●	●		

CPU Typ **A**	...	CP30, CP31, CP40, CP41 NTCP3#, PP40
CPU Typ **B**	...	CP60, CP80, NTCP6#, PP60

Example:



Chapter 4 LADDER DIAGRAM / LOGIC PLAN / FUNCTIONAL PLAN

4. LOGIC PLAN PROGRAMMING

A	B	C	PG1000	CP80	PG-PC	PC80
●	●	●			●	●

Logic plan programming is available in all three levels, **A**, **B** and **C,** and exists in the **PG-PC** and **PC80** operations**.** Logic plan programming is impossible in **PG1000** mode.

Important Note

Particularly important notes are framed to stand out on the page and attract attention.

Key Designation

All keys are shown in square brackets `[]`:

Key		Description
Engl.	Germ.	
[↑]	[↑]	Cursor up
[↓]	[↓]	Cursor down
[←]	[←]	Cursor left
[→]	[→]	Cursor right
[↲]	[↲]	Enter / Return
[PgUp]	[Bild↑]	Cursor one page up
[PgDn]	[Bild↓]	Cursor one page down
[Ins]	[Einfg]	Insert character before cursor
[Del]	[Entf]	Delete character at cursor position
[Esc]	[Esc]	Interrupt, exit from editors and/or menus
[Space]		Space bar
[Ctrl]	[Ctrl]	Control key
[⇑]	[⇑]	SHIFT key
[Alt]	[Alt]	Alternate key
[❖]	[❖]	One of the four cursor movement keys
[⇐]	[⇐]	Back Space

If two keys must be pressed simultaneously, then the two key symbols are connected by a "+" sign.

Example: `[Ctrl] + [Ins]`

Text which must be entered directly into the programming device is printed in **bold** face and ***italics***.

Example: ***CLEAR***

Input Commands and Descriptions

Certain requests are shown using the following symbols:

<NAME>	enter a name comprising of up to 4 characters (e.g. Functional plan name...)
<LAD_NAME>	enter a name comprising of up to 4 characters (e.g. Ladder diagram name...)
<TBL_NAME>	enter a data table name comprising of up to 4 chacters
<FNAME>	enter a file name comprising of up to 8 characters (when storing and loading)
<LINE>	enter a Line number
<LINE/LAB>	enter a Line number or Label
<NR>	enter a number (e.g.:number of lines to delete)

Abbreviations

The abbreviations used in this user's manuel are:

SEQC	Sequence Chart
STL	Statement List
FBK	Function Block
FP	Functional plan
LAD	Ladder diagram
SYMB	Symbol assignment
LP	Logic plan
TAB	Data table

Operating Commands

While the cursor is in the command line, all function key commands can be entered directly from the keyboard.

CHAPTER 2

INSTALLATION

Contents:

1. HARDWARE AND SOFTWARE PREREQUISITES

The **B&R PROgramming SYStem** software is available on two 5¼"diskettes and one 3½" disk.

Before installing the software, be sure you have the following requirements:

- **your personal computer must be IBM AT or IBM XT compatible (Processor 80386, 80286, 8088 or 8086)**
- **IBM compatible monochrome or color adapter**
- **Hard Disk (with 2 MB free space)**
- **Floppy Drive 5¼" (360 kByte or 1,2 MByte) or 3½" (720 kByte or 1,44 MByte)**
- **640 kBytes RAM; of which at least 553 kBytes (565.840 Bytes) must be available. If necessary programs like FDO, CHAD, Device Drivers (e.g.: Mouse driver), loaded character sets (DOS 3.X) must be deleted to achieve this free space.**
- **MS-DOS Version 2.11 and above or PC-DOS**
- **No expansion boards using the COM 2 Interrupt.**

If these prerequisites apply, the software can readily be installed on the hard disk.

To fully utilize *all* operational functions additional hardware is required:

- **Transfers from PC to PG1000**
  V24 serial interface COM 1
- **For printing:**
  V24 serial interface COM 1
  or parallel port
- **For Online-communication to the PLC (Chapter 11 - ONLINE-INTERFACES)**
  Centronics Online-interface on a parallel interface,
  B&R Online-interface and cable or
  Modem for Online-Programming over the telephone.

2. MODEL NUMBERS - DESCRIPTIONS

Expansion levels	**without Centronics Online Interface**	**with Centronics Online Interface**
Level A	PROgramming SYStem Programming software, Level A	PROgramming SYStem Programming package, Level A
Model No.	SWPIA-0	PGPROSLA-0
Level B	PROgramming SYStem Programming software, Level B	PROgramming SYStem Programming package, Level B
Model No.	SWPIB-0	PGPROSLB-0
Level C	PROgramming SYStem Programming software, Level C	PROgramming SYStem Programming package, Level C
Model No.	SWPIC-0	PGPROSLC-0

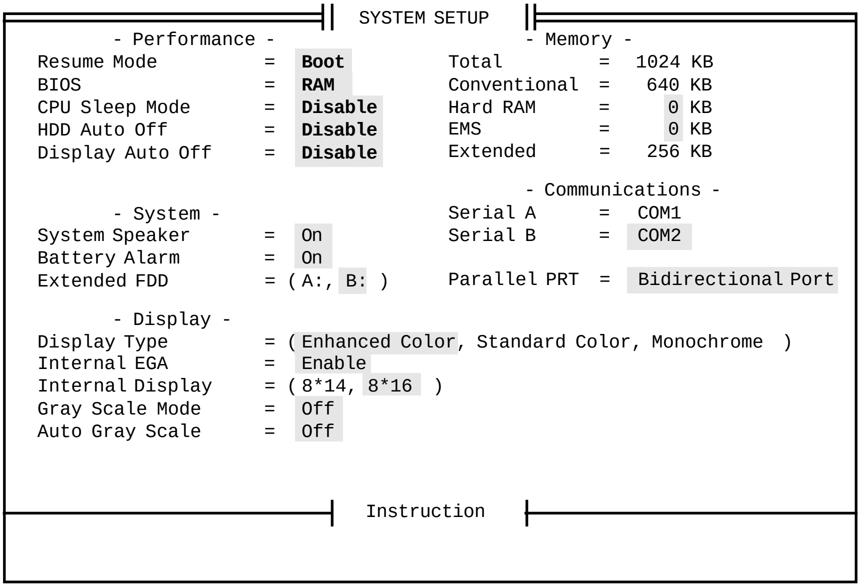
The software is always delivered with the “User's Manual B&R PROgramming SYStem”.

3. FUNCTIONS PROVIDED IN THE EXPANSION LEVELS

Function	**Level A**	**Level B**	**Level C**
LAD	YES	YES	YES
Standard-FBKs	YES	YES	YES
Comments, SYMB	YES	YES	YES
Additional FBKs	NO	YES	YES
Data tables	NO	YES	YES
STL	NO	NO	YES
FBK-Editor	NO	NO	YES

4. PORTABLE PC-CONFIGURATION

One very important step in having a problem free B&R PROgramming SYStem is the proper configuration in the PC's setup menu. With the wrong configuration, the B&R PROgramming SYStem will run only partially or not at all.



```
                              SYSTEM SETUP
       - Performance -                              - Memory -
Resume Mode          =  Boot          Total          =  1024 KB
BIOS                 =  RAM           Conventional   =   640 KB
CPU Sleep Mode       =  Disable       Hard RAM       =     0 KB
HDD Auto Off         =  Disable       EMS            =     0 KB
Display Auto Off     =  Disable       Extended       =   256 KB

                                             - Communications -
       - System -                     Serial A       =  COM1
System Speaker       =  On            Serial B       =  COM2
Battery Alarm        =  On
Extended FDD         = ( A:, B: )     Parallel PRT   =  Bidirectional Port

       - Display -
Display Type         = ( Enhanced Color, Standard Color, Monochrome  )
Internal EGA         =  Enable
Internal Display     = ( 8*14, 8*16  )
Gray Scale Mode      =  Off
Auto Gray Scale      =  Off


                              Instruction
```

The selections "HDD Auto Off" and "Display Auto Off" must stay set to **"Disable".**

5. INSTALLATION ON THE HARD DISK

5.1. NEW INSTALLATION

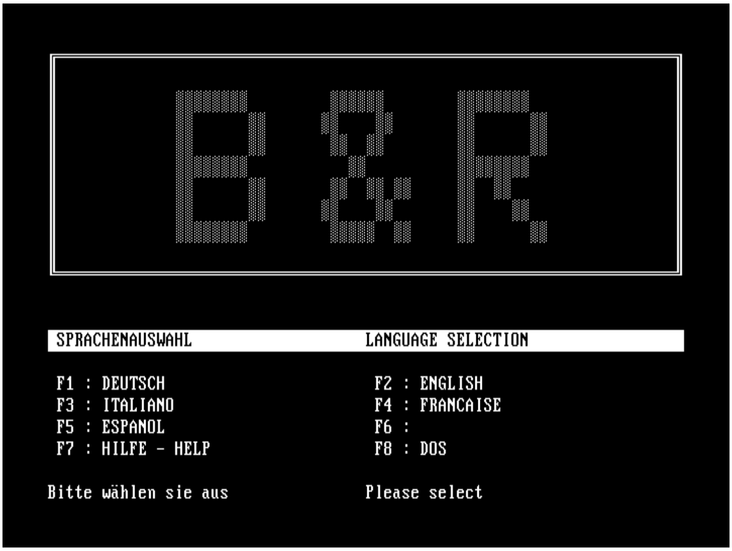
To start a new install, insert setup diskette number one into the source drive. The installation can be started from drive A by typing the MSDOS command as follows:

```
a:setup[↵]
```

The installation can also be started from drive B:

```
b:setup[↵]
```

From this point on you will be guided by menu's. Here the screen displays the language menu and from here the language choice for conversation to and from the PC is made.

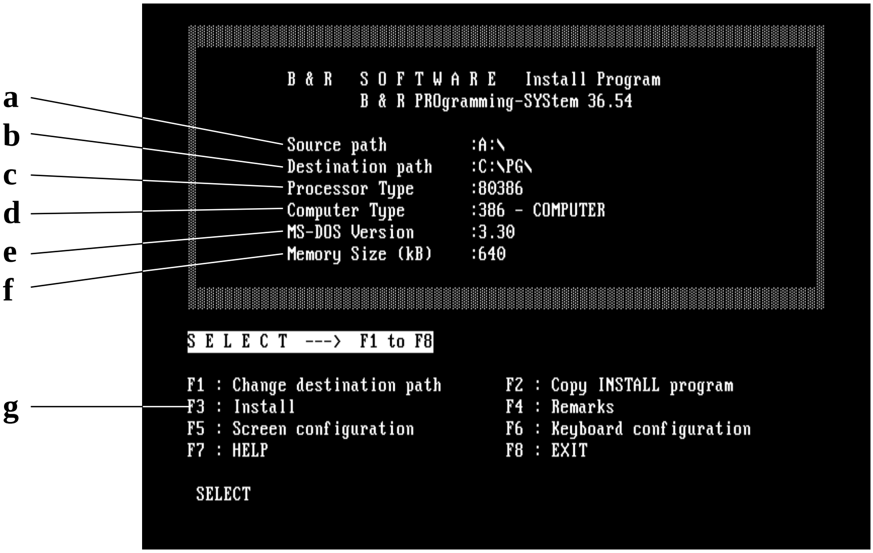


```
SPRACHENAUSWAHL                  LANGUAGE SELECTION

 F1 : DEUTSCH                      F2 : ENGLISH
 F3 : ITALIANO                     F4 : FRANCAISE
 F5 : ESPANOL                      F6 :
 F7 : HILFE - HELP                 F8 : DOS

Bitte wählen sie aus             Please select
```

The **[F7] HELP** key will bring down the help text for the language selection of the PROgramming SYStem.

The function key **[F8] DOS** cancels the installation process.

After the language selection is made, the installation menu is displayed.



```
        B & R   S O F T W A R E   Install Program
                B & R PROgramming-SYStem 36.54

        Source path          :A:\
        Destination path     :C:\PG\
        Processor Type       :80386
        Computer Type        :386 - COMPUTER
        MS-DOS Version       :3.30
        Memory Size (kB)     :640

S E L E C T  ---> F1 to F8

F1 : Change destination path        F2 : Copy INSTALL program
F3 : Install                        F4 : Remarks
F5 : Screen configuration           F6 : Keyboard configuration
F7 : HELP                           F8 : EXIT

 SELECT
```

The installation menu contains the following configuration information:

- **a)** **Source Path** - Path where the software installation can be accessed
- **b)** **Destination Path** - Path name under which the software is installed
- **c)** **PC type** - (IBM AT, IBM XT, AT COMPATIBILITY, ...)
- **d)** **Processor Type** - (8088, 8086, 80286, 80386)
- **e)** **Main Memory** - Size of RAM memory in kBytes, allocated by DOS
- **f)** **MS-DOS Version**
- **g)** Function key commands

[F1] Choosing another destination path for the PROgramming SYStem:

The PROgramming SYStem sets a default path of `C:\PG` but with the **F1** option you can set up a path to suit your system.

[F2] INSTALL:

The INSTALL program is installed on the hard disk, and will configure the PROgramming SYStem to be utilized properly. The files necessary to configure the software, are at the same time copied to the chosen destination path, in the directory (C:\PG).

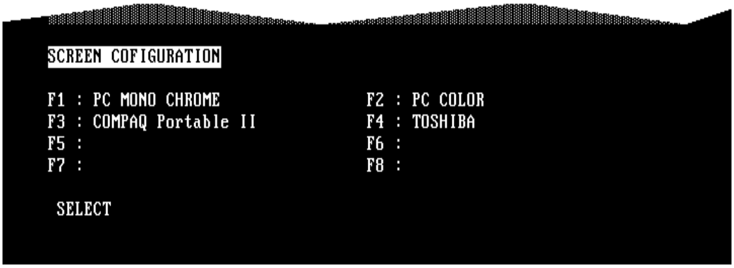
[F3] Software installation:

The directory listed in the destination path is set up automatically by the installation program.

To set up the screen and keyboard the menus **SCREEN CONFIGURATION** and **KEYBOARD CONFIGURATION** are chosen respectively.

Screen configuration:

The menu for the SCREEN CONFIGURATION looks like this. The function keys in this menu are allocated as follows:



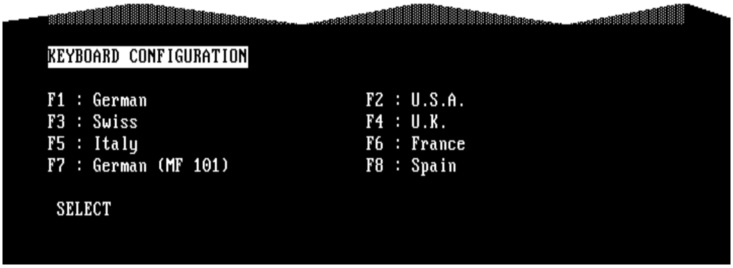
```
SCREEN COFIGURATION

F1 : PC MONO CHROME              F2 : PC COLOR
F3 : COMPAQ Portable II          F4 : TOSHIBA
F5 :                             F6 :
F7 :                             F8 :

 SELECT
```

The desired screen driver is selected by means of the function keys. The selected screen driver is displayed in inverse video, and the configuration is installed on the hard disk.

When the correct driver has been chosen the keyboard must be configured. The installation program automatically switches to the **KEYBOARD CONFIGURATION** menu:

Keyboard configuration:

configuration menu:

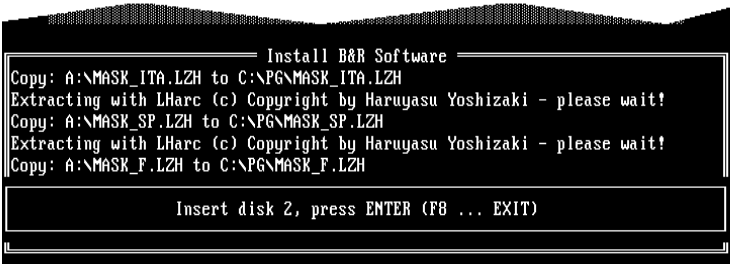


```
KEYBOARD CONFIGURATION

F1 : German                    F2 : U.S.A.
F3 : Swiss                     F4 : U.K.
F5 : Italy                     F6 : France
F7 : German (MF 101)           F8 : Spain

 SELECT
```

The keyboard language is chosen, by pressing the respective function key. The selected keyboard language is displayed in inverse video, and the configuration is installed to the hard disk.

Copying the software to the hard disk:

With the screen and keyboard configuration completed, the files from the setup disk are copied to the hard disk.and then the software will ask for the second diskette. The following message will blink on the screen

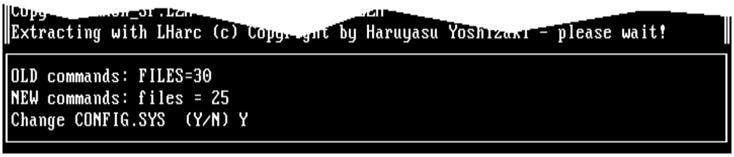


```
═══════════════════ Install B&R Software ═══
Copy: A:\MASK_ITA.LZH to C:\PG\MASK_ITA.LZH
Extracting with LHarc (c) Copyright by Haruyasu Yoshizaki - please wait!
Copy: A:\MASK_SP.LZH to C:\PG\MASK_SP.LZH
Extracting with LHarc (c) Copyright by Haruyasu Yoshizaki - please wait!
Copy: A:\MASK_F.LZH to C:\PG\MASK_F.LZH

              Insert disk 2, press ENTER (F8 ... EXIT)
```

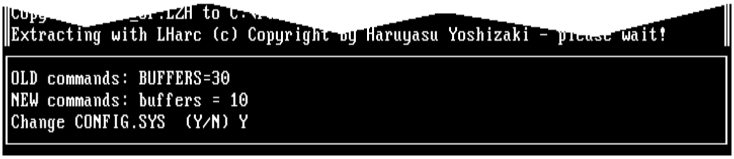
After the second diskette is inserted the installation can be continued by pressing the enter/return **[↵]** key.

When all files are copied to the hard disk a message is displayed on the screen regarding the PC-setup menu and the COM-2 interrupt (online interface). This message is the same as is displayed when the **[F4] REMARKS** function key is pressed from the main menu.

After pressing any key, the **CONFIG.SYS** file can be changed. Two modifications are made if to the file if needed.

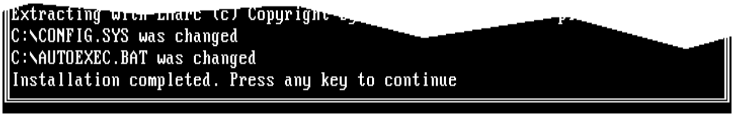


```
Extracting with LHarc (c) Copyright by Haruyasu Yoshizaki - please wait!

OLD commands: FILES=30
NEW commands: files = 25
Change CONFIG.SYS  (Y/N) Y
```



```
Extracting with LHarc (c) Copyright by Haruyasu Yoshizaki - please wait!

OLD commands: BUFFERS=30
NEW commands: buffers = 10
Change CONFIG.SYS  (Y/N) Y
```

If the two system files CONFIG.SYS and AUTOEXEC.BAT on your hard disk have been changed, the following message will appear on the screen.

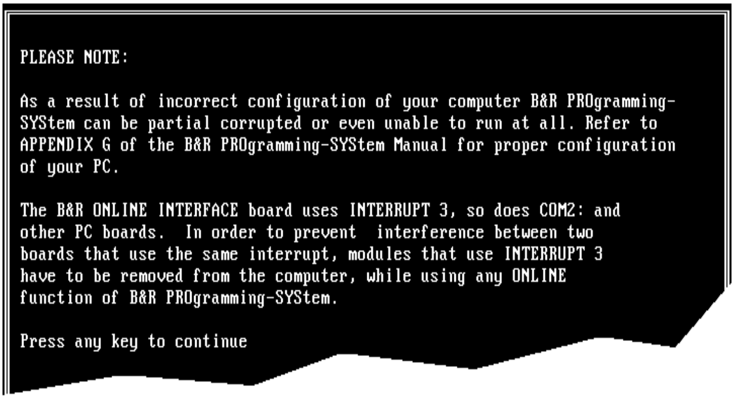


```
Extracting with LHarc (c) Copyright
C:\CONFIG.SYS was changed
C:\AUTOEXEC.BAT was changed
Installation completed. Press any key to continue
```

At this point, pressing any key will bring you back to the main menu.

[F4] Remarks:

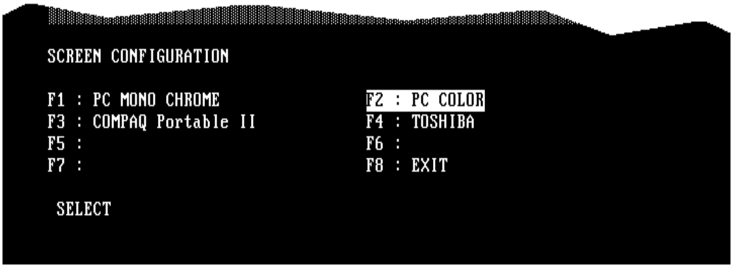
The key brings down a piece of text explaining a bit about running a failure free PROgramming SYStem and also shows some installation information (PC setup menu and COM 2 port interrupt).



```
PLEASE NOTE:

As a result of incorrect configuration of your computer B&R PROgramming-
SYStem can be partial corrupted or even unable to run at all. Refer to
APPENDIX G of the B&R PROgramming-SYStem Manual for proper configuration
of your PC.

The B&R ONLINE INTERFACE board uses INTERRUPT 3, so does COM2: and
other PC boards.  In order to prevent  interference between two
boards that use the same interrupt, modules that use INTERRUPT 3
have to be removed from the computer, while using any ONLINE
function of B&R PROgramming-SYStem.

Press any key to continue
```

[F5] New screen configuration:

The **[F5]** is only used to now to change the screen if a wrong key has been pressed before. If this is the case **F5** will bring you back to the screen configuration menu where another choice can be made.



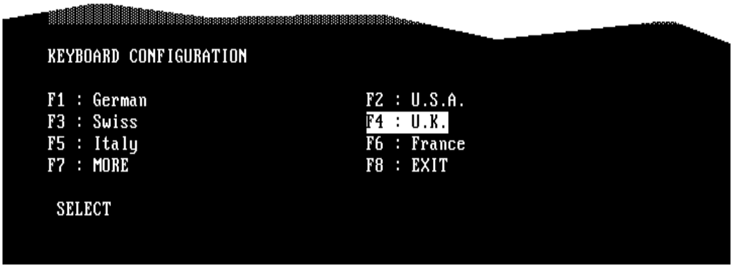
```
SCREEN CONFIGURATION

F1 : PC MONO CHROME                F2 : PC COLOR
F3 : COMPAQ Portable II            F4 : TOSHIBA
F5 :                               F6 :
F7 :                               F8 : EXIT

 SELECT
```

The selection made previously is shown in inverse video. By pressing the corresponding function key a new screen driver will be installed. The chosen item will be displayed in inverse video.

The **[F8]** function key will take you back to the main installation menu.

[F6] New keyboard configuration:

By pressing the **[F6]** function key, a new keyboard driver can be installed. This function should only have to be used if you have made an incorrect entry the first time around. If you must change the keyboard driver, press [F6] and the following menu will be displayed on the screen.



```
KEYBOARD CONFIGURATION

F1 : German                        F2 : U.S.A.
F3 : Swiss                         F4 : U.K.
F5 : Italy                         F6 : France
F7 : MORE                          F8 : EXIT

 SELECT
```

The previously selected keyboard is shown in inverse video. By pressing the function key that corresponds to the keyboard of your choice, the new driver will be installed and the selection will be displayed in inverse video.

The **[F7]** function key gives a larger selection of keyboard drivers to choose from.

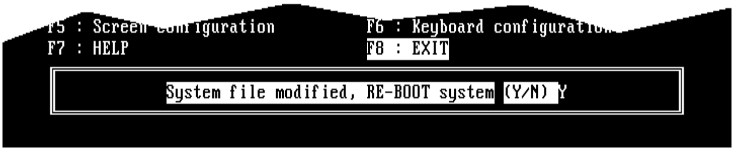
[F8] will take you back to the installation menu.

[F7] HELP

A help text will be displayed giving a short description of the **[F1] - [F8]** function keys and what they do.

[F8] EXIT INSTALL PROGRAM

Before starting the PROgramming SYStem for the first time, the PC must be rebooted to put into effect the modifications to the system files. When the **[F8] EXIT** function key is pressed you will be asked if the system should be restarted.



```
F5 : Screen configuration          F6 : Keyboard configuration
F7 : HELP                          F8 : EXIT

        System file modified, RE-BOOT system (Y/N) Y
```

[Y] Is a PC warmstart which restarts the system.
[N] Exits to the MS-DOS interpreter.

Note:

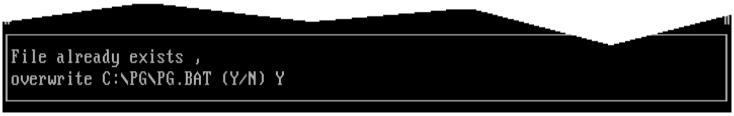
The installation has created the following directories:

Default Directory: (Working software Directory):	**PGDEF**
Standard Directory: (Directory for Standard function blocks):	**PGSTD**
User Directory: (Directory for your own function blocks):	**PGUSER**

5.2. SUBSTITUTING WITH A NEW SOFTWARE RELEASE

To replace your old software with a new release follow the exact same procedure as a new install. The new software will overwrite the old.

The setup program looks for the file path of the old B&R PROgramming SYStem. When the program is found you are asked if the PG.BAT file should be over written.



```
File already exists ,
overwrite C:\PG\PG.BAT (Y/N) Y
```

By pressing **[N]** you can give the new PG.BAT file another name.

With **[↵],** the PG.BAT file is over written. From this point on the process is exactly like a new installation.

6. THE INSTALL PROGRAM

6.1. CHANGING THE CONFIGURATION

There are two ways to go about changing the configuration of the B&R PROgramming SYStem.

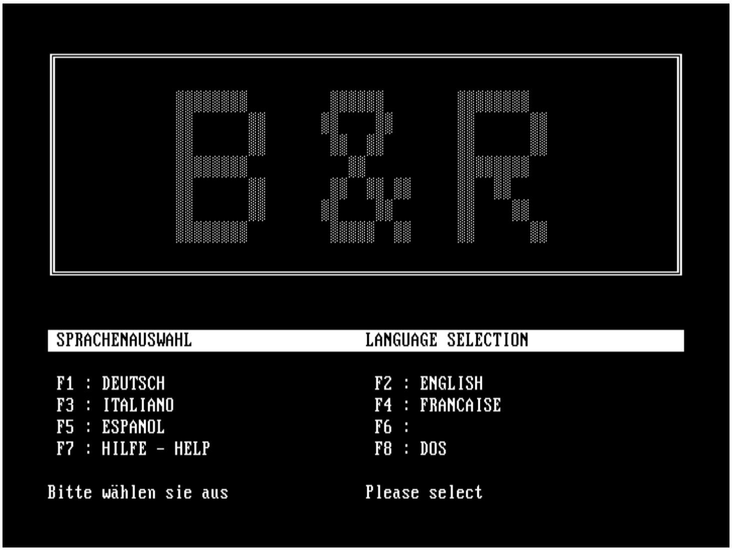
- with the **PROgramming SYStem setup disk. Reconfiguring** the B&R PROgramming SYStem with the setup disk is done the same as the installation. Information about the installation can be found under 5. INSTALLATION ON THE HARD DISK

- with the **INSTALL-Program** on the hard disk.

The INSTALL program is called from MS-DOS with the following command:

```
C:\PG> install [↵]
```

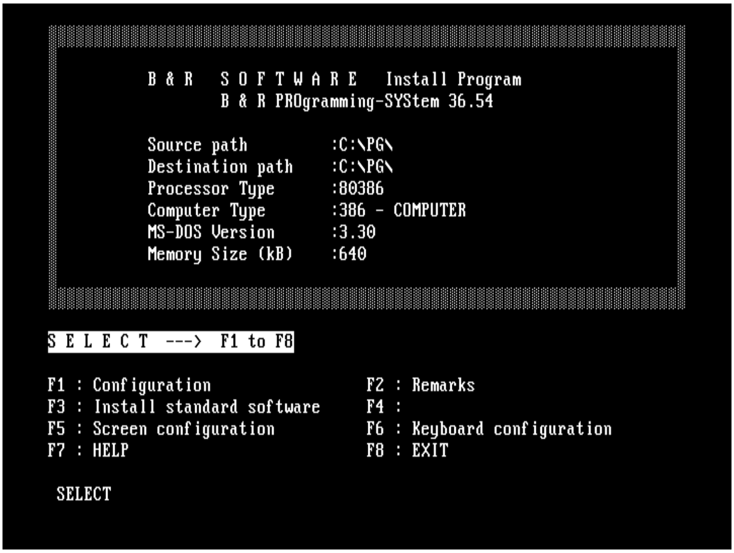
The configuration starts just like the installation with the language menu from the INSTALL program showing on the screen.



```
SPRACHENAUSWAHL                    LANGUAGE SELECTION

 F1 : DEUTSCH                        F2 : ENGLISH
 F3 : ITALIANO                       F4 : FRANCAISE
 F5 : ESPANOL                        F6 :
 F7 : HILFE - HELP                   F8 : DOS

Bitte wählen sie aus               Please select
```

From the menu, select the language you would like to communicate with the PC in.

After choosing the language, the installation menu from the INSTALL program is displayed:



```
          B & R   S O F T W A R E   Install Program
                  B & R PROgramming-SYStem 36.54

          Source path          :C:\PG\
          Destination path     :C:\PG\
          Processor Type       :80386
          Computer Type        :386 - COMPUTER
          MS-DOS Version       :3.30
          Memory Size (kB)     :640

S E L E C T  ---> F1 to F8

F1 : Configuration                  F2 : Remarks
F3 : Install standard software      F4 :
F5 : Screen configuration           F6 : Keyboard configuration
F7 : HELP                           F8 : EXIT

 SELECT
```

The following function keys are used for configuring :

[F1] Configuration

[F5] Screen configuration

[F6] Keyboard configuration

[F1] Configuration

The screen and keyboard can be reconfigured from the **SCREEN CONFIGURATION** and **KEYBOARD CONFIGURATION** menus. These menus are exactly the same as the software installation setup menus (Pg.8&9 of this chapter). When the proper drivers are selected, the program copies the files to the correct destination directory. When the copy process is finished, pressing any key will bring you back to the installation menu.

[F5] New screen configuration and [F6] new keyboard configuration

Both of these functions are exactly the same as in the setup program (Pg. 11).

6.2.ADDITIONAL FUNCTIONS OFTHE INSTALL PROGRAM

[F3] Standard software installation:

This function installs the B&R standard software. The installation is explained more precisely in the ***Users Manual / Standard software,*** (Volume 1, General)

[F4] Remarks:

This is a short text giving information on how to run a failure free B&R PROgramming SYStem and some operation requirements (PC setup and COM 2 interrupt).

[F7] HELP:

A help text giving the operation of function keys **[F1] - [F8].**

[F8] EXIT:

Quits the install program and exits to MS- DOS.

7. USABLE PC-VERSION

For a usable PC version you must have the following files:

Language dependent files:

CMDS.PG	....	Command names and function keys text
ERROR.PG	....	Error messages
WARN.PG	....	Other messages
STATMELD.PG	....	Status test messages
HEADER.PG	....	Function block FBK - Outline - Text
PROSYS.PG	....	Title screen with parameters

Normal data files:

KEYB.DEF	....	Keyboard definition
CPU1.DTA	....	OPCODE-Data table
EDITOR.PAR	....	EDITOR-Parameters, Field length and attributes
EDITOR.CTL	....	Control strings for B&R Mnemonics
EDITOR.CT0	....	Control strings for MOTOROLA Mnemonics
FKEY.BEL	....	Function key definition
PRV_AEGA.DEF	....	Output code definition for MAESTRO-Emulation (EGA-Graphics card)
PRV_AVGA.DEF	....	Output code definition for MAESTRO-Emulation (VGA-Graphics card)

Program:

RUN.EXE	....	Device drivers for ONLINE, SERIAL, CRT, KBD, ...
PG02.EXE	....	PG-Software

Modem:

Modem operation files:

CCITT53.CFG DIAL53.CFG HAYES53.CFG MODEM53.CFG
SEEPASS.DTA SEEPROM.DTA STAND53.CFG

Screen masks for Modem operation in the PG\MASK\PG directory:

BAUD.MAS CHAN.MAS FIND.MAS GOTO.MAS
KSTATUS.MAS MOABBR.MAS MOALARM.MAS MOCALLB.MAS
MOCPTYP.MAS MODEM1.MAS MODEM2.MAS MOMODEM.MAS
MOPARA.MAS MOPASS.MAS MOTIMEB.MAS MOTIMEI.MAS
PARI.MAS PASSWD.MAS SAFETY.MAS SELECT.MAS
SHOW.MAS STANDBY.MAS TELNUM.MAS

Files created by PC-PG (These files can be deleted without loss of data).

V_MEM.TMN	....	Temporary memory for <UNDO>
V_MEM.TM1	....	Files in temporary memory
V_MEM.TM2	....	STL-Programs
PG.WDW	....	Window configuration

CHAPTER 3
INSTRUCTIONS

Contents:

1. STARTING THE SOFTWARE

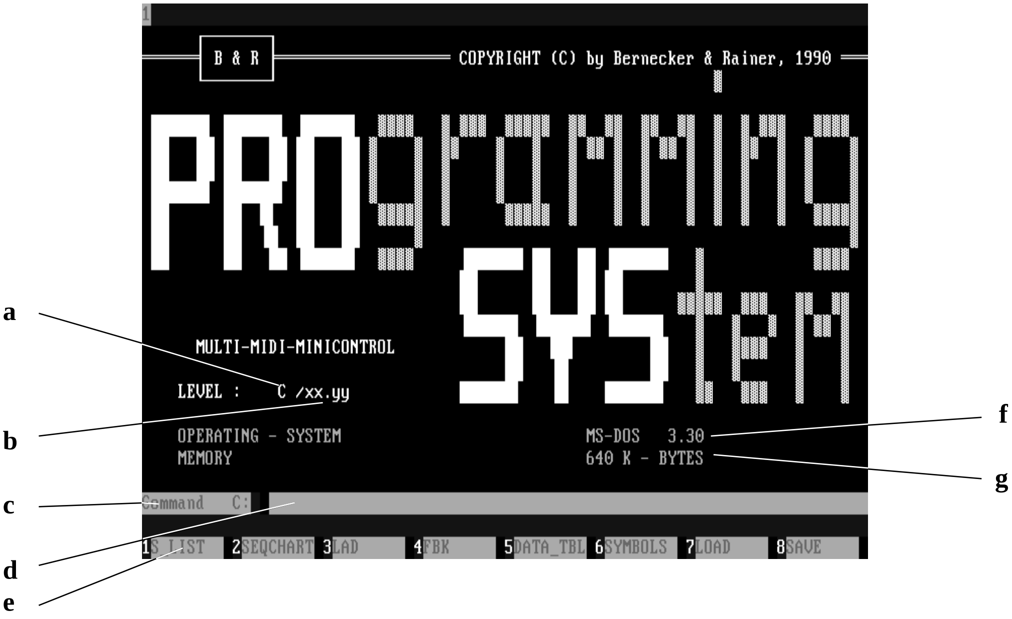
To start the PROgramming SYStem after the installation is finished, just:

1. Turn on the power to the PC
2. Type ***pg*** and press the [↵] key. This is possible from any directory.

The PROgramming SYStem is started.

At the top, left of your screen “`MEMSIZE: XXXX paragraphs`” is displayed. The number of paragraphs must be at least 3000 for the PROgramming SYStem to run. If not enough memory is available the program is ended with FATAL ERROR 9.

After a few seconds the opening menu screen is displayed:



```
1
B & R          COPYRIGHT (C) by Bernecker & Rainer, 1990

PROgramming
        SYStem

MULTI-MIDI-MINICONTROL

LEVEL :    C /xx.yy

OPERATING - SYSTEM                      MS-DOS   3.30
MEMORY                                  640 K - BYTES

Command   C:

1S_LIST  2SEQCHART 3LAD    4FBK    5DATA_TBL 6SYMBOLS 7LOAD   8SAVE
```

The opening menu screen gives the following information:

a) LEVEL Installation level (A, B or C)

b) VERSION xx.yy Version of the system module and PROgramming SYStem

Example: 36.54—Version number of PROgramming SYStem: 5.4
└——Version number of CP60/80 System module: 3.6

c) Command line: The command line is where all operating commands for the PROgramming SYStem are given.

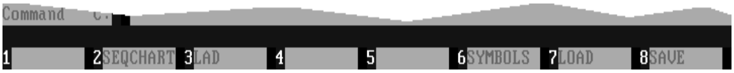
d) Blinking cursor: Shows the point, that the next character can be placed on the screen.

e) Softkeys: At the bottom of your screen eight function keys are shown. These keys are for the most important operating commands that should have to call while working in the PROgramming SYStem.

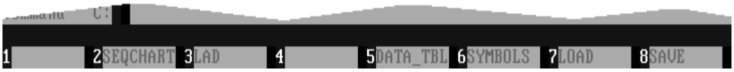
A, **B** and **C** levels are different in their softkey functions.

For example in the main menu:

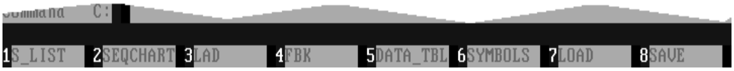
Level A:



Level B:



Level C:



f) Operating system: Version of operating system in your PC

g) Memory: Amount of base memory in your PC

2. STARTUP PARAMETERS OF THE PROGRAMMING SYSTEM

Upon startup of the B&R PROgramming SYStem, certain parameters can be set.

The following parameters can be assigned:

- PROgramming SYStem language:	**LANG = <Language>**
- Mnemonics used for instructions:	**MNEM = <Mnemonic>**
- Operating mode:	**PGMODE = <Mode>**
- Default-Directories [1]:	**DDIR = \xxxxxxxx\...**
- User-Directories [1]:	**UDIR = \xxxxxxxx\...**
- Standard-Directories [1]:	**SDIR = \xxxxxxxx\...**
- Standard symbols	**STDSYM = <FNAME>**
- LEARN-File , is invoked directly after starting the software.	**EXE = <FNAME>**
- set parameters in one file:	**PAR = <FNAME>**

2.1. START PARAMETER DETAILS

The B&R PROgramming SYStem can start up with any parameters that you would like to set up by typeing them into the MS-DOS command line:

```
C:\> cd\pg [↵]
C:\PG\> run pg02 <Parameter1> <Parameter2> ... [↵]
```

More parameters can be entered, with a space between each.

[1] These parameters can only be used on the same drive that the PROgramming SYStem is installed.

2.2. LANGUAGES IN THE PROGRAMMING SYSTEM

The desired communicating language can be set or reset using this parameter. With it you can choose one of five different languages.

Syntax: `C:\PG\> run pg02 lang=<Language> [↵]`

Parameters:	`<Language>`	`GR`	(German)
		`USA`	(English)
		`F`	(French)
		`ITA`	(Italian)
		`SP`	(Spanish)

Example: `C:\PG\> run pg02 lang=usa [↵]`

Language is now English

2.3. STATEMENT LIST MNEMONICS

This parameter sets the input mnemonics for the PROgramming SYStem and it also gives a choice of which mnemonics you would like displayed. Four choices are given.

Syntax: `C:\PG\> run pg02 mnem=<Mnemonic> [↵]`

Parameters:	`<Mnemonic>`	`B&R`	B&R Mnemonic
		`MOTOROLA`	MOTOROLA® Mnemonic
		`MIXM`	Displayed in MOTOROLA® Mnemonic Entered in B&R Mnemonic
		`MIXB`	Displayed in B&R Mnemonic Entered in MOTOROLA® Mnemonic

Example: `C:\PG\> run pg02 mnem=b&r [↵]`

The STL commands are now entered in B&R mnemonics.

2.4. OPERATING MODE

The desired mode of the PROgramming SYStem is chosen with this parameter. The functions you would like to use depend on the selection of operating mode that you make.

Syntax:

```
C:\PG\> run pg02 pgmode=<Mode> [↵]
```

Parameters:	<Mode>	PC	PG-PC Mode (not compatible with PG1000-Mode)
		PG1000	PG1000 Modus[1] (not compatible with PC-Mode)
		CP80	CP80 Mode[1]
		CP80PC	PC80 Mode

Example:

```
C:\PG\> run pg02 pgmode=cp80 [↵]
```

The program is started in CP80 mode.

2.5. DEFAULT-DIRECTORY

This parameter sets the desired Default Directory. When a program or program part is loaded or saved this is the directory that is used. If no parameter is set then the program assumes c:\pgdef to be the Default Directory.

Syntax:

```
C:\PG\> run pg02 ddir=<Directory> [↵]
```

parameters:	<Directory>	\xxxxxxxx	...Directory name, up to eight chacters Longer path names can also be given.

Example:

```
C:\PG\> run pg02 ddir=\test [↵]
```

c:\test is now defined as the default directory.

[1] Only possible in Level C.

2.6. USER-DIRECTORY

The User Directory that you would like to use is selected with this parameter. This directory is used when a program or program part is loaded or saved.If this start parameter is not set up, the program assumes c:\pguser to be the User Directory.

Syntax: `C:\PG\> run pg02 udir=<Directory> [↵]`

Parameter: `<Directory>` `\xxxxxxxx` ..Directory name up to eight characters A longer path name can also be given.

Example: `C:\PG\>run pg02 udir=\test [↵]`

c:\test is now the defined User Directory.

2.7. STANDARD-DIRECTORY

This parameter sets the Standard Directory that you wish to use.All Standard FBK's are automatically loaded to this directory. If no parameter is set, the program assumes c:pgstd to be the Standard Directory.

Syntax: `C:\PG\> run pg02 sdir=<Directory> [↵]`

Parameter: `<Directory>` `\xxxxxxxx` ...Directory name up to eight characters. A longer path name can also be given.

Example: `C:\PG\>run pg02 sdir=\test`

c:\test is now the defined Standard Directory.

2.8. STANDARD SYMBOLS

To define the start parameters with **STDSYM=<FNAME>**, puts all Standard Symbols in **<FNAME>** upon startup and after every NEW.

Syntax: `C:\PG>run pg02 stdsym=<FNAME>[↵]`

Parameter: <FNAME> ...Drive + Path + Filename without extension; if no path is given the file is looked for in \PG

Example: `C:\PG\> run pg02 STDSYM=c:\prosys\usa [↵]`

In the directory of `c:\prosys` the `usa` file must have the extension .sym. The SYM file usa.sym will be loaded.

If the file holding the STDSYM parameters does not exist, the PROgramming SYStem will start up nevertheless and no error message will be shown.

2.9. CALLING UP A LEARN FILE

By entering the start parameters **EXE=<FNAME>**, a LEARN-File is loaded and output. (LEARN see appendix C)

Syntax: `C:\PG> run pg02 exe=<FNAME> [↵]`

Parameter: <FNAME> ...Drive + Path+ File name; If no path is given the directory \PG\ is searched

Example: `C:\PG> run pg02 exe=init.lrn [↵]`

The file `init.lrn` is loaded to the directory `c:\pg` and called from it as well.

2.10. STARTUP PARAMETER FILE

With this parameter a file can be created which will contain any other parameters you wish to start up with.

Syntax: `C:\PG\> run pg02 par=<FNAME> [↵]`

Parameter: `<FNAME>` ...Up to an eight character file name. A longer path name can also be given.

Example: The following are held in the`c:\pg\start` file:

```
lang=gr
mnem=b&r
ddir=\test
udir=\fub
```

The following calls have identical functions:

```
C:\PG\> run pg02 par=start [↵]
```

```
C:\PG\> run pg02 lang=gr mnem=b&r ddir=\test udir=\fub
[↵]
```

Note:

During the Programming SYStem installation a separate parameter file is created, which is activated by starting the program with ***pg*** **[↵]**. The PG.BAT File looks like this:

```
C:
cd C:\PG
run pg02 PAR=startup.pg
cd ..
```

The parameter file `startup.pg` contains these Parameters:

```
mnem=b&r
stdsym=gr
```

2.11. CHANGING PARAMETERS AFTER STARTUP

Start parameters can also be set, after the startup of the software, in the command line.

Command	**Function**
***MOTOROLA* [↵]**	Switch to MOTOROLA® Mnemonics
***B&R* [↵]**	Switch to B&R Mnemonics
***MIXM* [↵]**	Displayed in MOTOROLA® Mnemonics and Input in B&R Mnemonics
***MIXB* [↵]**	Displayed in B&R Mnemonics and Input in MOTOROLA® Mnemonics
***LANG* <LANGUAGE> [↵]**	Language options
***PC* [↵]**	Switch from PG1000 to PG-PC Mode Switch from CP80 to PC80 Mode
***CP80* [↵]**	Switch from PG1000 to CP80 Mode Switch from PG-PC to PC80 Mode
***CP40* [↵]**	Switch from CP80 to PG1000 Mode Switch from PC80 to PG-PC Mode
***DDIR* <\PATH> [↵]**	Default-Directorie options
***SDIR* <\PATH> [↵]**	Standard-Directorie options
***UDIR* <\PATH> [↵]**	User-Directorie options

After switching from PC mode to PG1000 mode you can not switch back. Working in PG1000 mode is only possible if the PROgramming SYStem has been started in this mode or with the command NEW. (PG 1000 only operates if the PROgramming SYStem has started with it.)

Example: Switching from B&R Mnemonics to MOTOROLA® Mnemonics.

Enter in the command line: ***MOTOROLA* [↵]**

The command automatically loads the MOTOROLA® Mnemonics to the program and uses them from this point.

3. KEYBOARD

3.1. ESC-KEY

This key is used quite often. Pressing the [Esc] puts you in the nest higher menu or breaks out af a currently running command.

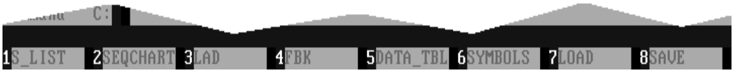
> If the keyboard that you have does not have an [Esc] key the [5] key in the number block replaces it.

3.2. FUNCTION KEYS

The function keys are used to call up the corresponding commands seen at the bottom of the screen.

The function keys [F9] and [F10] bring up alternate softkey functions that are used by holding the [F9] or [F10] key down and pressing the function key corresponding to the desired command.

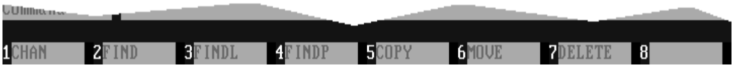
Example: The start of the PROgramming SYStem shows the following softkey functions:



```
1S_LIST  2SEQCHART 3LAD      4FBK      5DATA_TBL 6SYMBOLS  7LOAD     8SAVE
```

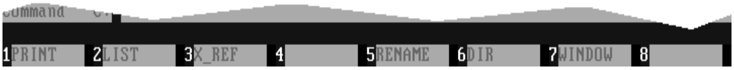
Pressing [F9] or [F10] changes them:

By pressing [F9]



```
1CHAN    2FIND     3FINDL    4FINDP    5COPY     6MOVE     7DELETE   8
```

By pressing [F10]



```
1PRINT   2LIST     3X_REF    4         5RENAME   6DIR      7WINDOW   8
```

3.3. RESET-KEY

If you need to momentarily break into a currently executing program, a RESET key is available on the programming device keyboard. To take a look at where the keys are situated on your keyboard the illustrations on page 3-14 and 3-15 should provide an answere.

If your keyboard has no number block, the RESET keys are [F11] nad [F12]. When the keyboard has both a number block and function keys either can be used. If you are running the B&R PROgramming SYStem on a TOSHIBA laptop computer the RESET keys are [Sys Req] and [Prt Sc] keys, situated in the top right corner of the keyboard.

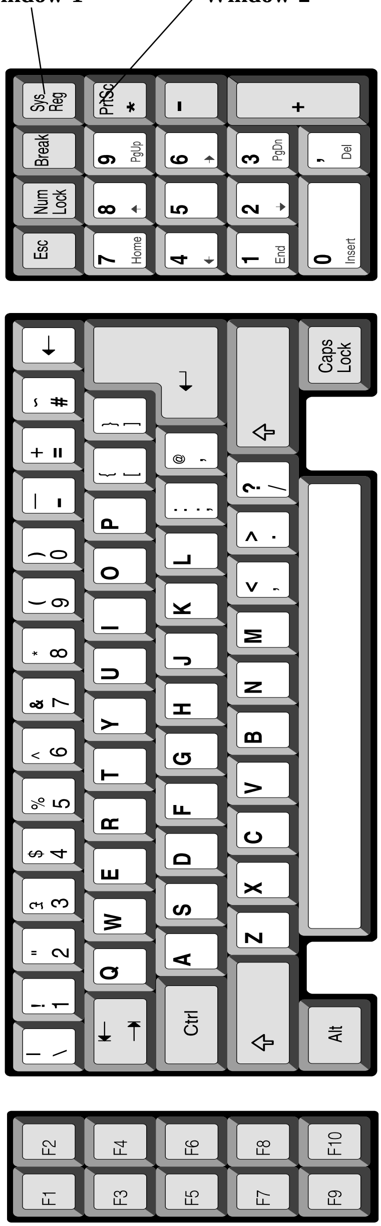
The RESET key functions differently depending on how the Online Interface is utilized:

> If a **B&R Online-Interface**,has been incorporated for use in your computer, then you have the possibility of working in two windows simultaneously. So, there are also two RESET keys (one for each window).
>
> If using a **Centronics Online-Interface** on a parallel interface, then both RESET keys are for one window.

AT-KEYBOARD

Reset key for Window 1

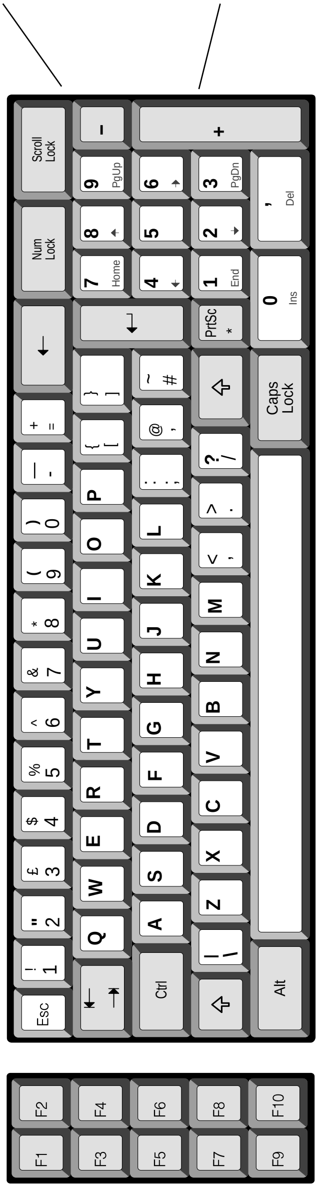
Reset key for Window 2



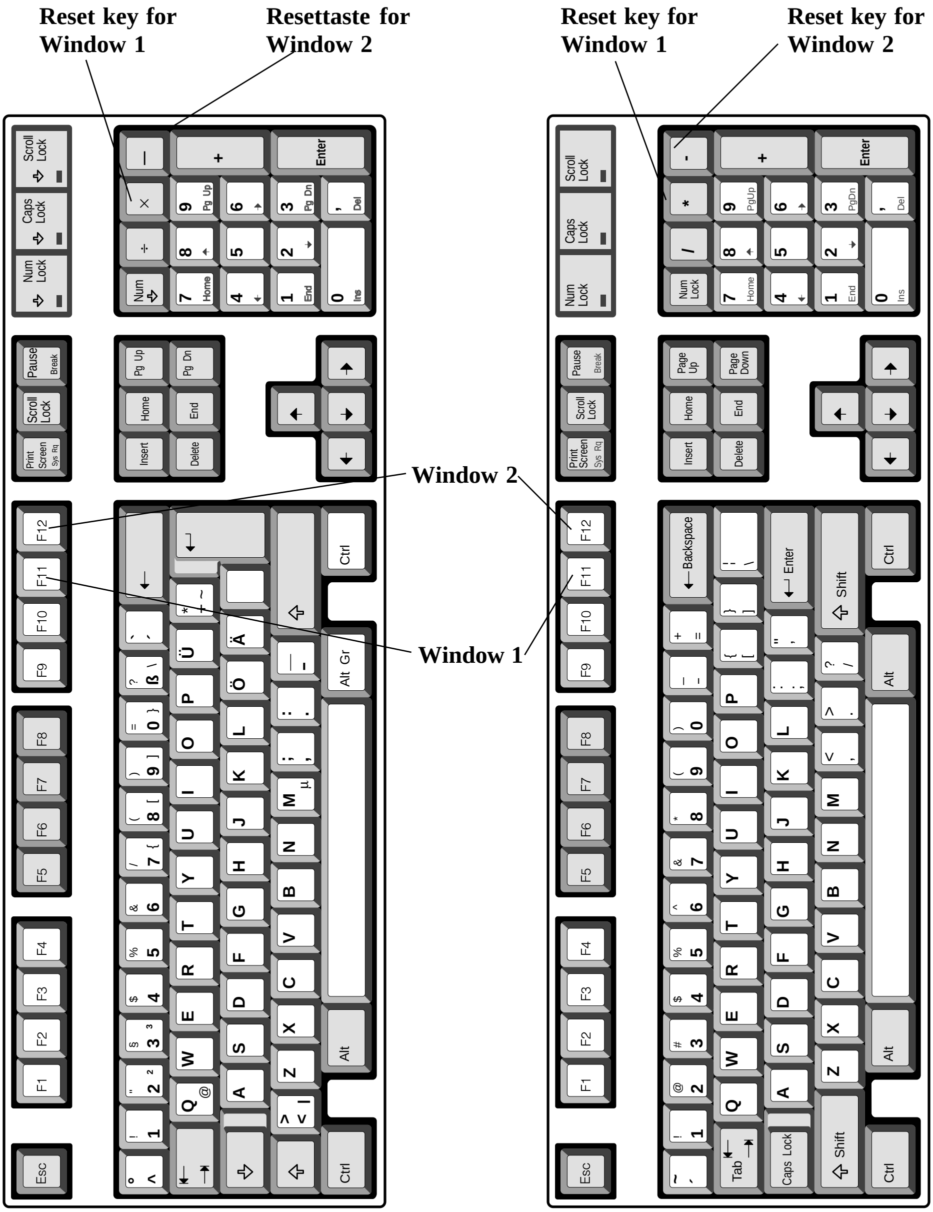
XT-TASTATUR

Reset key for Window 1

Reset key for Window 2



EUROPIAN-KEYBOARD **US-KEYBOARD**

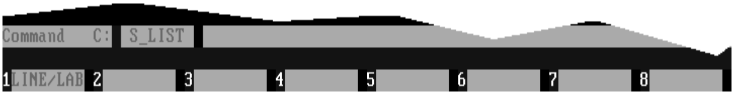


4. THE COMMAND LINE

This is where all operational commands are entered.After the initial call up of the PROgramming SYStem, the cursor will flash in the command line. The command line also shows the actual drive where all loading and saving will be done.

All commands controlled by the function keys are also entered in the command line.When a command is entered in the command line from the keyboard, the function key allocation is changed

Example: By entering ***STL*** in the command line the function key commands change:



Command C: S_LIST

Functions, from the command line:

[↑] or [↓]	SEQC [1], STL, SYMB or TAB-Editor, moves one line up or one line down. (depends on which editor the commands are entered in).
[PgUp]	SEQC [1], STL, SYMB or TAB-Editor, moves to the next page up.
[PgDn]	SEQC [1], STL, SYMB or TAB-Editor, moves to the next page down.
[Ctrl] + [↑] or [↓]	Input line (The line that is displayed inversley in the editor)moves up or down one line.
[Ctrl] + [Del]	Command line is deleted
[Ctrl] +[Alt] + [↑] or [↓]	Repeats the previous command (up to 10 commands are stored).
[Alt] + [TAB]	Tabulator line turned off.
[↵]	Depending on the editor the cursor will go to the chosen input line.

[1] Dies ist nur in den Ausbaustufen A und B möglich.

5. TABULATORS

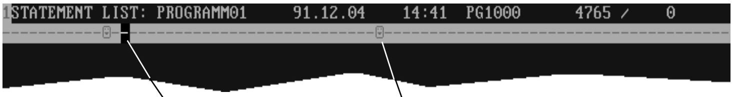
Tabulators are utilized in the following editors: STL, SYMB and TAB-Editor.

To set the tabs place the cursor in the desired position and with the following key combinations ...

...tab is set: `[Ctrl] + [TAB]`

...tab is removed: `[Ctrl] + [TAB]`

...the tabulator line is switched on: `[Alt] + [TAB]`



```
1STATEMENT LIST: PROGRAMM01     91.12.04    14:41  PG1000     4765 /    0
```

tabulator set

current curtsor position

...the tabulator line is switched off: `[Alt] + [TAB]`

The tabulator line can be switched on directly from the command line. However it is impossible to set or remove tabs from it.

6. WINDOW TECHNIQUES

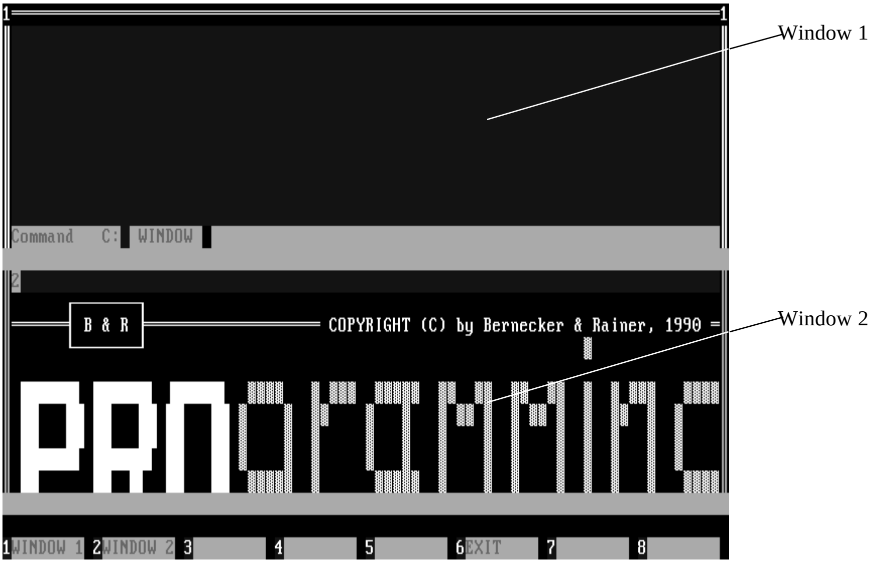
6.1. WINDOW DEFINITION

The B&R **PROgramming SYStem** makes working on two programs at the same time, possible. You can alternate between the two windows or work in both simultaneously. With a B&R two port Online interface you can connect two controllers (or 2 Processors) and work with them both.

With the ***"WINDOW"*** command you can position and size the windows

Command: **[F10] + [F7 WINDOW] [↵]**

Both window are displayed, according to the last set position and size.



```
Command   C: WINDOW
```

```
B & R    COPYRIGHT (C) by Bernecker & Rainer, 1990
```

Softkey allocation:

[F1 WINDOW 1] When the cursor is in the command line of window 1, the size of the window can be changed.

[F2 WINDOW 2] When the cursor is in the command line of window 2, the size of the window can be changed.

[F6 EXIT] Switches back to the previous menu.

Window defining functions :

[F10] + [F7 WINDOW] [↵]	Start editing windows
[F1 WINDOW 1]	Edit window 1
[F2 WINDOW 2]	Edit window 2
[❖]	Entire window moves
[PgUp]	Window moves diagnally up and right
[PgDn]	Window moves diagnally down and right
[Home]	Window moves up and left
[End]	Window moves down and left
[Ctrl] + [❖]	Changes window size
[Alt] + [←] or [→]	Moves visable window interior horizontally
[Esc] or [F6 EXIT]	Exits to the previous menu

Note: The B&R PROgramming SYStem stores all window size and position changes automatically in a file called PG.WDW, so that the last changes made are the start up parameters for your windows.

6.2. PROGRAMMING IN WINDOWS

With the key combination [Ctrl] + [F1] or [Ctrl] + [F2], you can switch between the two windows as they have been previously set up. The cursor changes to the window that you have chosen:

Window 1	[Ctrl] + [F1]
Window 2	[Ctrl] + [F2]

Once the windows have been organized and you are again working with a program, the windows cannot be changed. Window sizing and positioning can only be done while in the window edit operation.

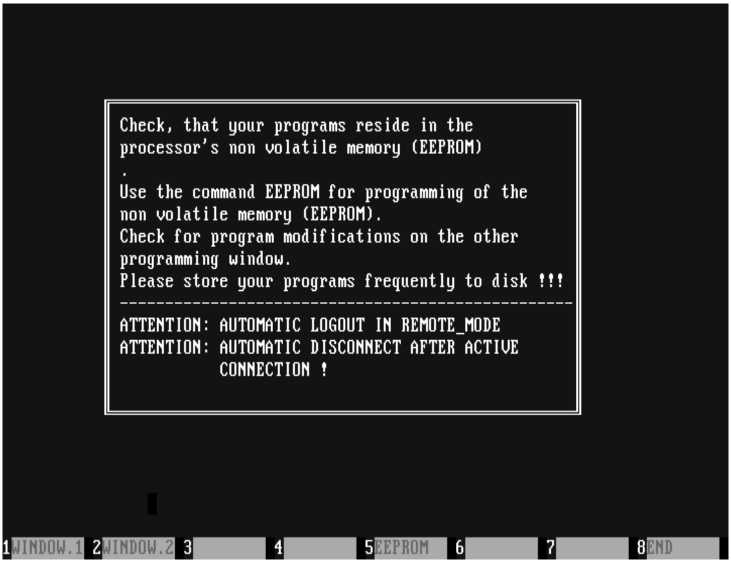
When the softkey line displays the standard operating commands, the window chosen with the previous command is the one that you can program in.

This window can be switched to the full size of the screen:

Switch window 1 to full size	[⇑] + [F1]
Switch window 2 to full size	[⇑] + [F2]

7. QUITTING THE PROGRAMMING SYSTEM

When your PROgramming SYStem work is done, enter in the command line the operating command ***END.*** By choosing this command the following information will be displayed on the screen:



```
Check, that your programs reside in the
processor's non volatile memory (EEPROM)
.
Use the command EEPROM for programming of the
non volatile memory (EEPROM).
Check for program modifications on the other
programming window.
Please store your programs frequently to disk !!!
--------------------------------------------------
ATTENTION: AUTOMATIC LOGOUT IN REMOTE_MODE
ATTENTION: AUTOMATIC DISCONNECT AFTER ACTIVE
           CONNECTION !
```

In the softkey line the following functions are available:

[F8 END] Ends the program

[F1 WINDOW 1] Return to window 1.

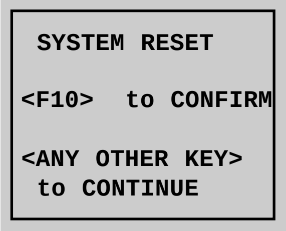
[F2 WINDOW 2] Return to window 2.

[F5 EEPROM] If a program hasn't yet been saved to the PROM-user memory, it can still be done without going back into the STL. By pressing [**F5 EEPROM**], the EEPROM menu wil be displayed.

8. SYSTEM-RESET

A warm start (reboot) can be done by pressing the [CTRL], ALT] and [DEL] keys at the same time. This causes the MS-DOS® operating system to start again.

To avoid a SYSTEM RESET occuring by accident the operating system asks the following question:



```
 SYSTEM RESET

<F10>  to CONFIRM

<ANY OTHER KEY>
 to CONTINUE
```

To carry out the SYSTEM RESET, press function key [F10].

If the operating system does not restart, you must press any other key to carry out the procedure.

9. PROGRAMMING TRANSFERS BETWEEN PG1000 AND PC

It is possible in Level C to transfer programs from the PG1000 to the PC in PG1000 mode or programs in the PC (in PG1000 mode) can be transfered to the PG1000.

9.1. SWITCHING FROM PG1000 TO PC

A	B	C	PG1000	CP80	PG-PC	PC80
		●	●			

1) Make connection from PG1000 (serial interface) to PC (serial interface COM 1) (cable BRKAPC-0)

2) Enter **[F7 LOAD] [F6 PG1000] [↵]** command to PC.
The PROgramming SYStem switches to the PG1000 mode. After a short time the following message is displayed in the message line of the PC:

```
W042  QUIT WITH <ESC> KEY
```

When the message appears, continue with the next step.

3) Enter ***PRIBM*** **[↵]** command to the PG1000.

If a CP80 program is in the PG1000, this is recognized by the PROgramming SYStem in the PC and the following message is displayed:

```
W236  Convert to 3 byte mnemonics ( Y / N ) ?
```

Every command in the PG1000 has a 3 byte length - even with CP80 programs. Contrary to the PC, where CP80-program commands aren't always 3 bytes long. The user can now decide if during the program transmission he/she wants a conversion to 3 byte mnemonics to occur or not:

[Y] Generates 3 byte commands, i.e. every command that is shorter than 3 bytes is expanded to a 3 byte length..
This function is applied if step-by-step switching devices, which only function with a command length of 3 byte, are used in the program to be transfered.

[N] No conversion to 3 byte mnemonics.

The program loaded by the PG1000 can be processed. Also the additional functions of the PC-mode can be used, when switching to the PC mode when entering the command ***PC*** .It is very important to remember, that it can't be switched back to the PG1000 mode.

Note: The STL instructions of ladder diagrams are inserted to the PC after the last existing program END . If t he field is filled with NOP's, there might not be enough program memory.
Remedy: Delete unnecessary ENDs in the PG1000 before the transfer.

9.2. SWITCHING FROM PC TO PG1000

A	B	C	PG1000	CP80	PG-PC	PC80
		●	●		●	

Programming can only be switched to PG1000, if the PC has been started in PG1000 or PG-PC mode.

1) Make connection from PG1000 (serial Interface) to PC (serial Interface COM 1) (Cable BRKAPC-0)

2) Enter ***LIBM*** **[↵]** command to PG1000.

3) Enter **[F8 SAVE]** ***PG1000*** to PC.

4) Enter **[F1 INTER] [↵]**, if the program should switch in MOTOROLA® Mnemonics to the PG1000.

   Enter **[F2 GERMAN] [↵]**, if the program should switch in B&R Mnemonic to the PG1000.

If a program written in the PG-PC mode is found, the following question is displayed in the messageline:

```
W228  SAVE PG1000 in Mode PG-PC ? ( Y / N )
```

With [Y] or [N] the user can decide, if he/she should quit the current transfer or go ahead to the PG1000 mode.

The program can be edited again in PG1000 mode.

10. BASIC OPERATION COMMANDS

Following are some important commands used in the PROgramming SYStem:

• Change the program name	***NAME***
• Change program date	***DATE***
• Delete entire program	***NEW***
• Select online interface	***ONL***
• Define program memory	***SMALL*** / ***LARGE***
• Call MS-DOS command interpreter	***DOS***
• Text insert mode in the editor	**[Alt] + [Ctrl] + [Ins]**
• Use “command” name as label name	

10.1. CHANGE PROGRAM NAME

Syntax: ***NAME*** **<Program name> [↵]**

Parameter: <Program name> text with max. 10 ASCII characters.

Function: The input of program names.

This is the name given by the user that is indicated in the command line when you call up the PROgramming SYStem. This name is set as a default for every new start or every time the command ***NEW*** is used.It also defaults when you use "PROGRAM 01"for window 1 and "PROGRAM02"for window2.

The program name is ...

...saved together with a program (SAVE <FNAME> PROG)

...loaded with a program (LOAD PROG <FNAME>)

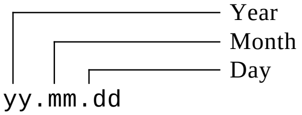
...printed on documentation

...transfered in the control with the program

10.2. CHANGE PROGRAM DATE

Syntax: ***DATE* <Date> [↵]**

Parameter: For <Date>, text can be entered that is 8 characters long. No letters may be used. The default setting after the start up of the PROgramming SYStem, e.g. after the command ***NEW*** the date is shown in the following format



Example: `91.04.25`

Function: The program date is set with <date>.

The program date is, just as the program name, saved with the program, transfered in the PLC and printed.

10.3. DELETING A PROGRAM

Syntax: ***NEW* [↵]**

Function: The entire program of the current window is deleted. The following default values are set:

- STL-Mnemonic: B&R or the selection defined on start up (MNEM=...).
- Operation: PG1000 or the choice made on start up (PGMODE=...).
- Symbols: SYMBs are deleted or the SYMB files that were defined on start up are loaded.

To prevent an accidential program deletion, the following question is displayed in the command line:

```
W010  ARE YOU SURE  ? ( Y / N )
```

Input: [Y] The entire program is deleted.
[N] The procedure is interrupted.

10.4. SETTING THE ONLINE INTERFACE

Syntax: ***ONL*<interface> [↵]**

Parameter: After the input of ***ONL*** different <interface> possibilities are offered in the softkey line:

[F1 B&R0] Selection of the installed B&R online interface.
[F2 LPT1] Selection of the centronics online-interface on the parallel interface LPT 1 of the PC's
[F3 COM1] Selection of the modem on the serial interface COM 1 of the PC's

Function: Selection and definition, of the online interface that should be applied, for programming a PLC.

10.5. DEFINITION OF THE PROGRAM MEMORY

A	B	C	PG1000	CP80	PG-PC	PC80
●	●	●	●		●	

Syntax: ***LARGE* [↵]**
***SMALL* [↵]**

Function: Program memory definition for different CPU-memory modules of type **A**.

Available Program Lines			
	Mode		
Command	**PG1000**	**PG-PC**	**Memory Module**
LARGE	4766	4766	EP05 (16 KByte EPROM)
SMALL	2719	2719	RA02 (8 KByte RAM)

Note: LARGE is the default setting.

10.6. MS-DOS COMMAND INTERPRETER CALL

Syntax: ***DOS* [↵]**
***DOS* "command" [↵]**

Parameter: "Command" can be any MS-DOS command.

Function: Call up and execution of the MS-DOS command interpreter. The PROgramming SYStem is not terminated, and remains in the PC work memory.

1) ***DOS* [↵]**

Accessing the MS-DOS interpreter. The MS-DOS input prompt appears. All MS-DOS commands and other application programs can be executed on this level, if the remaining PC work memory is sufficient.

The MS-DOS command interpreter can be exited with the ***EXIT*** command.

2) ***DOS* "command" [↵]**

MS-DOS "command" and automatically returns to the PROgramming SYStem after command termination.

Example: Copying all programs of directory c:\project to a disk in drive A:

DOS "COPY C:\project\*.prg a:" [↵]

10.7. TEXT INSERT MODE IN EDITORS

Syntax: **[Alt] + [Ctrl] + [Ins]**

Function: If the cursor is located in one of the editors, the text insert mode is either turned on or off with this key combination.

Text insert mode on: When entering text with the keyboard, the existing text is not overwritten, but moved to the right.. The newly entered text is inserted.

Text in sert mode off: When entering text with the keyboard the existing text is overwritten beginning at the cursor position.

10.8. USE "COMMAND" NAME AS LABEL NAME

It is possible to use all operating commands like LIST, RUN, COPY, MOVE etc., as label names. These names can be enteredwithout any difficulties as label names to the STL editor. But to enter these labels in the command line, they must be put in quotation marks.

Example: A program part, beginning at label LIST should be printed to the line number 1000 . The followimg line is to be entered:

```
[F10] + [F1 PRINT] [F1 PROG] [F1 FROM] "LIST" [F2 TO] 1000 [↵]
```

If the STL program line should be accessed to edit the STL, ***"LIST"*** has to be entered in the command line.

CHAPTER 4

SEQUENCE CHART / LADDER DIAGRAM LOGIC PLAN / FUNCTION PLAN

Contents:

1. GENERAL INFORMATION

The B&R PROgramming SYStem offers the possibility of writing in a graphic programming language. This programming mode has the advantage, of visualizing "abstract" STL commands making them clearer and easier to look at. This programming mode is comparable with the design of a circuit diagram. The user doesn't necessarily need to know the functions of STL commands.

This chapter explains four important PROgramming SYStem functions, that are part of the graphic programming:

- Sequence Chart
- Ladder Diagram Programming
- Logic Plan Programming (only in the PC mode)
- Function Plan Programming

Advantages:

- This type of programming is easier to understand than STL.
- Complex tasks can be completed in a clearly arranged format.
- Ladder diagrams are easier to read than STL programs. Especially, when a program should be changed and processed after a lengthy period of use.
- In the ladder diagram status test, contact status and internal memory locations are visualized. This makes control of a running program possible and is a great support for error detection.

Disadvantages:

- Since the graphic programming language of PROgramming SYStem must be translated into a STL program, the final result will be longer than a program written in STL.

2. SEQUENCE CHART

A	B	C	PG1000	CP80	PG-PC	PC80
●	●	●			●	●

2.1. GENERAL INFORMATION

The Sequence Chart (SEQC) serves to establish a programming procedure. The sequence for ladder diagrams that must be processed, STL subprograms and other SEQCs is defined. The call can only be performed in series, i.e. in the SEQC no options influencing the processing sequence are possible.

2.2. STL EDITOR CALL

Command: **[F2 SEQC] <NAME> [↵]**

With definition of a name An existing or a newly defined SEQC with the name entered is called and displayed on screen.

Without name definition The last processed SEQC is called.

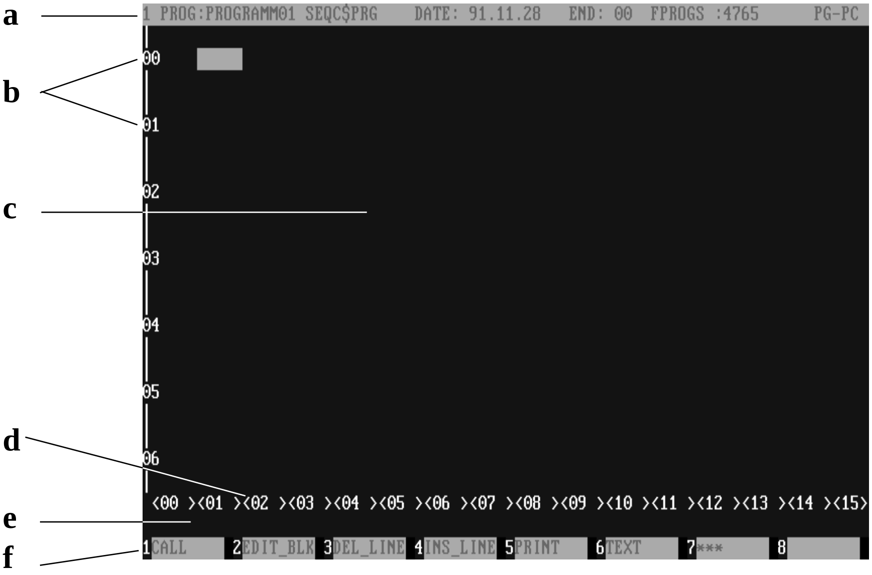
If no SEQC is existing, an SEQC is called with the name $PRG.

A	B	C	PG1000	CP80	PG-PC	PC80
●	●				●	●

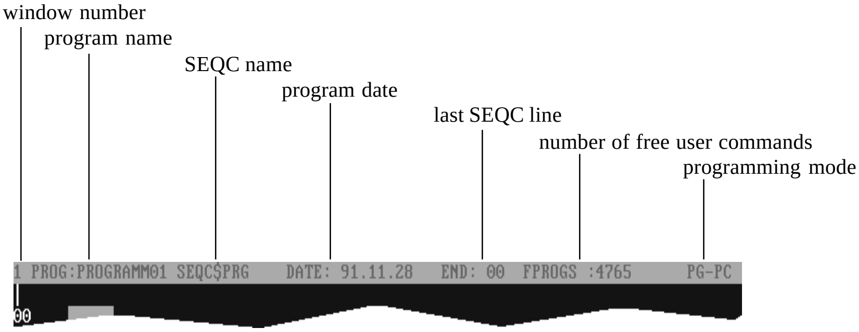
Important for expansion levels A and B:

A sequence chart with the name **$PRG** is **absolutely** necessary in expansion levels A and B. This sequence chart **$PRG** corresponds with the main program, from which all ladder diagrams and additional sequence charts are called.

2.3. SEQC MENU STRUCTURE


a
b
c
d
e
f
1 PROG:PROGRAMM01 SEQC$PRG DATE: 91.11.28 END: 00 FPROGS :4765 PG-PC
00
01
02
03
04
05
06
<00 ><01 ><02 ><03 ><04 ><05 ><06 ><07 ><08 ><09 ><10 ><11 ><12 ><13 ><14 ><15>
1CALL 2EDIT_BLK 3DEL_LINE 4INS_LINE 5PRINT 6TEXT 7*** 8

a) Status line: Various information is displayed in the status line:


window number
program name
SEQC name
program date
last SEQC line
number of free user commands
programming mode
1 PROG:PROGRAMM01 SEQC$PRG DATE: 91.11.28 END: 00 FPROGS :4765 PG-PC
00

b) Line numbering:

The SEQC editor is numbered from 0 to 49 on the left side of the screen. In a SEQC a maximum of 50 LAD calls can be located. If this is not sufficient, an SEQC can be called up instead of an LAD.

c) SEQC input field

All calls are entered and graphically represented in the input field. Comments can be entered at any location.

d) Column numbering

The SEQC editor input field is divided into 16 columns (0 - 15). LADs or a different sequence chart can always be called in column 1. The graphic illustration of a call occupies column 1 to 3. A comment can be entered in any location that is not occupied with a call.

A	B	C	PG1000	CP80	PG-PC	PC80
●	●				●	●

In expansion levels A and B after pressing [Esc] in a numbered line, a command line in the SEQC editor is displayed. Here all PROgramming SYStem operator commands can be entered.

e) Message line

In the message line all messages, that apply to the operator, are displayed. These are error messages that are caused by the user because of inorrect operation. Messages that require an input of parameters, can also be displayed.

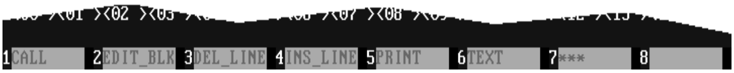
f) Softley line

In the softkey line the functions [F1] - [F7] of the SEQC editor are offered.

If the cursor is located in the command line of the expansion levels A or B, the SEQC $PRG or the last processed SEQC is displayed on screen. In this case the function keys are provided with the most important PROgramming SYStem operating commands.

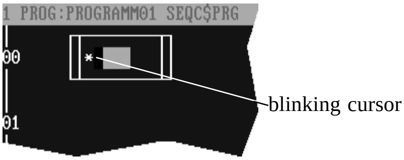
2.4. SEQC EDITOR FUNCTIONS

All SEQC editor functions can be called up with function keys [F1] to [F7]. If the cursor is located in the SEQC input field, the function keys are allocated as follows:



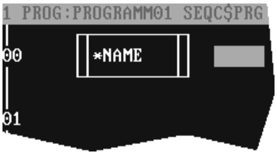
[F1 CALLS]

With this function ladder diagrams, sequence charts or statement list subprograms (only in expansion level C) can be accessed. Only if the cursor is positioned in **column 1**, can a call be made. When this key is pressed, the following figure is displayed in the current input field (the linethat the cursor is in):



The blinking cursor is in a four-character input field. The user can now enter a name with a maximum length of four characters. The name must start with a letter.

If a name is entered, to which no ladder diagram, SEQC program, or sequence chart exists, the call is marked with a "*".



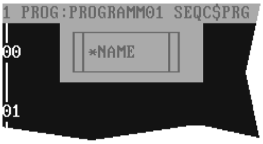
[F2 EDIT_BLK]

In order to edit an existing call, it must be first selected. This is possible in two different ways:

[F2 EDIT_BLK] + [❖] or

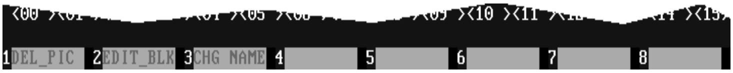
[Alt] + [❖]

A selected call can be displayed in inverse video on the screen:



```
1 PROG:PROGRAMM01 SEQC$PRG
00      *NAME
01
```

If a call is selected, the allocation of the function keys changes:



```
<00 ><01 ...  ><07 ><05 ><08 ... ><09 ><10 ><11 ><1... ...14 ><15>
1DEL_PIC 2EDIT_BLK 3CHG NAME 4  5  6  7  8
```

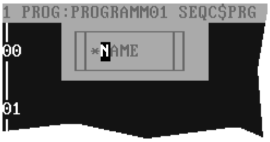
- [F1 DEL_PIC]

  Deletes the call from the SEQC. The softkey line is reset again.

- [F2 EDIT_BLK]

  With [F2 EDIT_BLK] + [❖] or [Alt] + [❖] a different call can be selected.

- [F3 CHG NAME]

  ```
  1 PROG:PROGRAMM01 SEQC$PRG
  00      *NAME
  01
  ```

  After pressing this function key, the name of the selected call is displayed in inverse video. The name can be changed. The change is confirmed with [↵].

  [❖]

- By pressing one of the four cursor keys, the selected call is exited and the softkey line is switched back again.

[F3 DEL_LINE] or **[Ctrl] + [Del]**

A line in the SEQC input field is deleted with this function. If a call or comment has been inserted in this line, it can't be deleted.

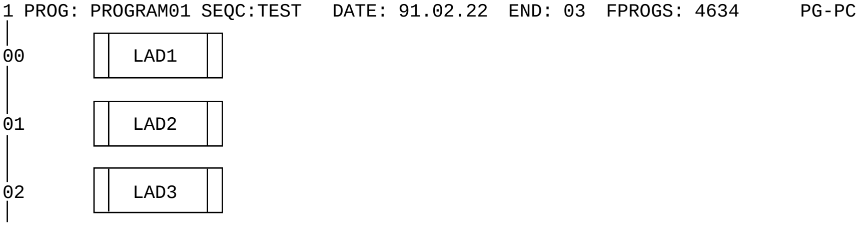
[F4 INS_LINE] or **[Ctrl] + [Ins]**

With this function a line in inserted in the SEQC input field. As soon as a call or comment is in line 49, the following error is displayed in the message line:

```
E106  CANNOT INSERT LINE !
```

[F5 PRINT]

The current sequence chart is displayed.



```
1 PROG: PROGRAM01 SEQC:TEST   DATE: 91.02.22  END: 03  FPROGS: 4634     PG-PC
|
00        LAD1
|
01        LAD2
|
02        LAD3
|
```

[F6 TEXT]

In order to make an operation or program editing possible later on, it is important to create clear and comprehensive documentation for the program. For this purpose comments can be entered with the function [F6 TEXT]. After pressing the function key, text can be entered beginning at the current cursor position. The text input is terminated with [↵].

[F7 *]**

With this function a call is marked, so that it is not considered during the program processing. This is necessary for example, if certain program parts should not be executed during the program test.

Before pressing [F7 ***] the cursor must be located to the left of the respective call. The call is marked with "*******" and is deactivated. Repeating this procedure activates the call and it is included in the program execution again.

2.5. STARTING A PROGRAMM

The program can be transfered to the PLC and started in two different ways:

- Pressing the function key **[F1 RUN]** of the ladder diagram
- Typing ***RUN*** **[↵]** in the command line.

a) Expansion Level A and B:

A sequence chart with the name $PRG must be available. If not, an error message is displayed after the program start:

```
E051  INVALID BRANCH
```

This error message is also displayed, when an invalid call is entered.

b) Expansion Level C:

If sequence charts are used in expansion level C, they must be called up like subprograms in the SEQC editor.

Example: **JSR** **xxxx** xxxx.... SEQC name

"Calls", that are entered in a sequence chart, are automatically transformed when exiting the sequence chart. A transformed sequence chart can look like this:

$PRG... (Sequence chart name)

```
$PRG      JSR      LAD1
          JSR      LAD2
          JSR      ....
            .
            .
          RTS
```

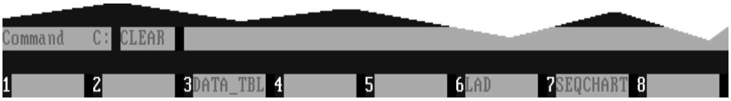
The main program, from which the program is called, is quit with the command "END".

```
0000      JSR   $PRG
           .
           .
          END
```

2.6. *CLEAR* - DELETING A SEQUENCE CHART

COMMAND: ***CLEAR*** [F7 SEQC] <name> [↵]

With the operating command ***CLEAR*** a SEQC can be deleted. If this command is entered to the command line, the function key allocation changes:



```
Command   C: CLEAR
```

With the selection of [F7 SEQC], input of a SEQC name and confirming with [↵] the respective SEQC is deleted.

2.7. SEQC EDITOR EXIT

[Esc] Exits the SEQC editor to the command line. The last edited sequence chart is translated to the STL. This STL translation is only available in level C.

[Ctrl] + [Esc] Exitsfrom the SEQC editor to the command line, and changes after the last access are not considered. The SEQC is not translated.

3. LADDER DIAGRAM PROGRAMMING

A	B	C	PG1000	CP80	PG-PC	PC80
●	●	●	●	●	●	●

3.1. GENERAL INFORMATION

Ladder diagram programming is a graphic programming language, that is comparable to the representation of a circuit diagram.

For ladder diagram programming all conventional contact symbols (NC and NO contacts,...) can optionally be connected.

A 9 character name (symbol) can be assigned to all contacts that are used in the ladder diagram, making it much easier to read.

B&R PROgramming SYStem gives the possibility to mix programming of ladder diagrams, logic plans and function plans. For this reason it is possible to resolve the most complex tasks in a clearly arranged representation.

3.2. LAD EDITOR CALL

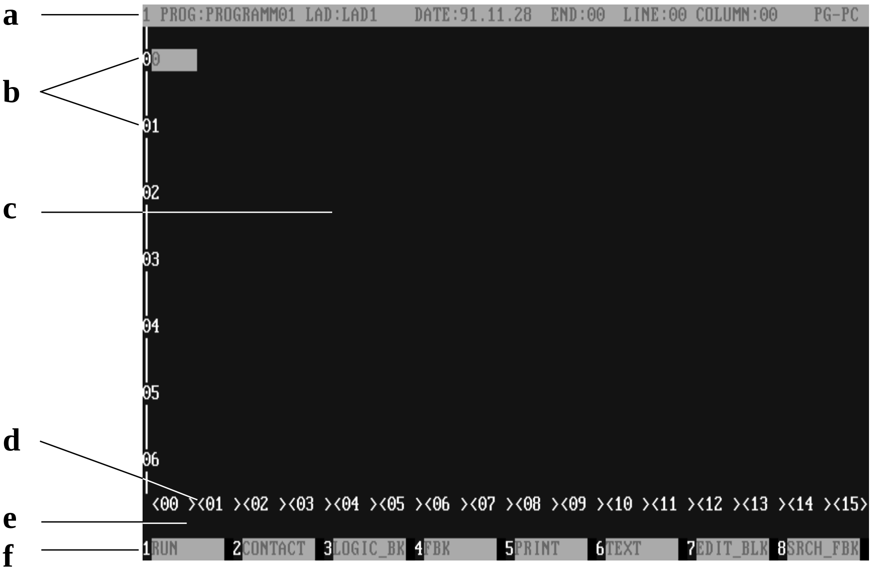
Command line input:	***LAD*** **<NAME> [↵]**
Pressing the function key:	**[F3 LAD] <NAME> [↵]**

A LAD with the name entered is called or created.

Without the definition of a name, the LAD, that was processed last, is called and displayed. If no LAD exists, the following error message is displayed:

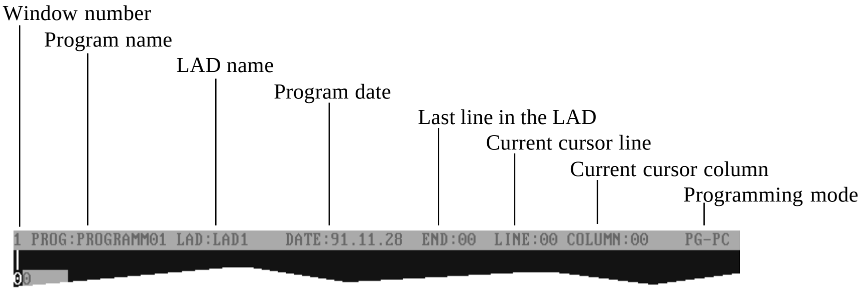
```
E174  DEFAULT NAME NOT DEFINED OR WRONG !
```

3.3. THE LAD MENU STRUCTURE



```
1 PROG:PROGRAMM01 LAD:LAD1    DATE:91.11.28  END:00  LINE:00 COLUMN:00    PG-PC
00
01
02
03
04
05
06
 <00 ><01 ><02 ><03 ><04 ><05 ><06 ><07 ><08 ><09 ><10 ><11 ><12 ><13 ><14 ><15>
1RUN  2CONTACT  3LOGIC_BK 4FBK  5PRINT  6TEXT  7EDIT_BLK 8SRCH_FBK
```

a) Status line: Various information is given in the status line:



```
1 PROG:PROGRAMM01 LAD:LAD1    DATE:91.11.28  END:00  LINE:00 COLUMN:00    PG-PC
00
```

b) Line numbering

In every LAD created, a maximum of 50 lines are available. The LAD editor is numbered from 0 to 49 on the left side of the screen.

c) LAD Editing field

In the LAD editing field all connections and contacts are inserted and represented graphically. For LAD documentation, comments can be entered as well.

After accessing a LAD, the cursor is located in line 0 / column 0. Within the editing field the cursor can be moved by means of [❖].

d) Column numbering

PG1000 mode The LAD is divided into 8 columns (00 to 07) .
PG-PC mode The LAD is divided into 16 columns (00 to 15) .

e) Message line

In the message line, messages concerning user errors or messages requiring the input of parameters are displayed.

f) Softkey line

In the softkey line, the most important commands of the LAD editor are offered on function keys [F1] to [F8]. [F9] serves as a shift key for further functions.

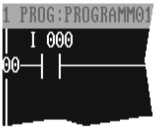
3.4. LAD PROGRAMMING BASICS

A	B	C	PG1000	CP80	PG-PC	PC80
●	●	●	●	●	●	●

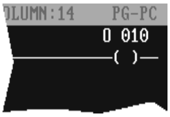
Allocations

All paths are started in the **column 0** with input allocations (load commands) and ended with output allocations in **column 15**. Input and output allocations are presented differently.

Input allocation



Output allocation



The remaining columns are used for the creation of connections and the installation of logic and function blocks.

Path

In the ladder diagram the horizontal lines are called PATHS. These paths are provided with an 8 bit value contrary to “digital transmission”.

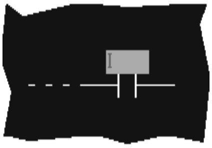
Address preselections

The following address preselections are possible in an LAD:

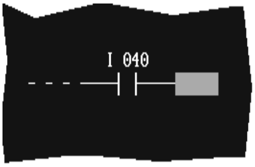
I	Digital Input	**S**	Start Timer
O	Digital Output	**T**	1 bit Signal elapsed time
F	Flag (1 bit memory)	**P**	Peripheral module
R	Register (8 bit memory)	**G**	Global RAM in the PP60[1]
#	Immediate constant value	**B**	Block memory in the PP60[1]

The input of these address preselections makes the call of a contact possible.

Example: Call of an Input allocation. The I 040 should be loaded . The cursor is located in column 0. After pressing the key **[I]** the following contact appears:



Entering ***040* [↵]** completes the input. The cursor is now located to the right of entered input allocation.

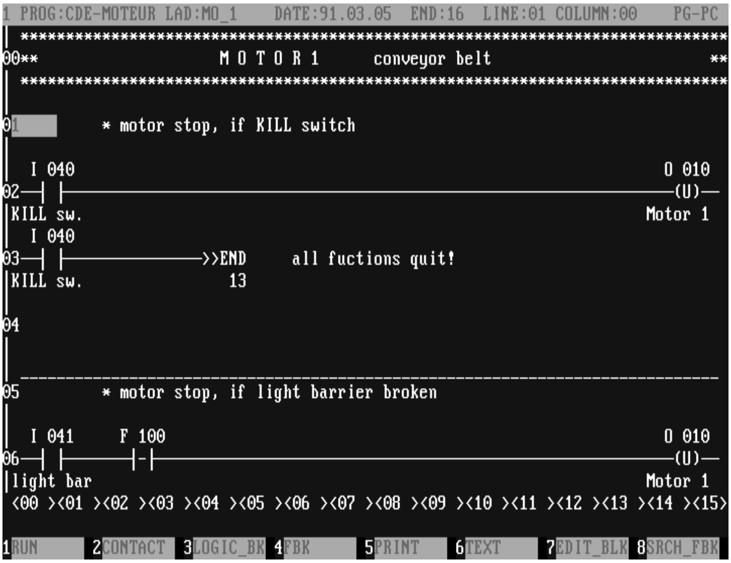


[1] Only possible in the PC80 mode

3.5. SYMBOLS / SYMB MODE

A	B	C	PG1000	CP80	PG-PC	PC80
●	●	●	●	●	●	●

In the SYMB editor (chapter 6 SYMBOL EDITOR) a 9 character symbol can be allocated to every address used in the LAD editor, displayed under the contact in the LAD. Reading and understanding a LAD is made easier, like the following example shows:



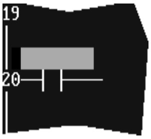
```
1 PROG:CDE-MOTEUR LAD:MO_1    DATE:91.03.05  END:16  LINE:01 COLUMN:00    PG-PC
| ********************************************************************************
00**                     M O T O R 1      conveyor belt                         **
| ********************************************************************************
|
01          * motor stop, if KILL switch
|
|  I 040                                                               O 010
02─┤ ├─────────────────────────────────────────────────────────────────(U)─
|KILL sw.                                                             Motor 1
|  I 040
03─┤ ├─────────────>>END     all fuctions quit!
|KILL sw.             13
|
04
|
| ---------------------------------------------------------------------------
05          * motor stop, if light barrier broken
|
|  I 041     F 100                                                     O 010
06─┤ ├──────┤-├────────────────────────────────────────────────────────(U)─
|light bar                                                            Motor 1
 <00 ><01 ><02 ><03 ><04 ><05 ><06 ><07 ><08 ><09 ><10 ><11 ><12 ><13 ><14 ><15>
```

With **[F9] + [F1 SYMB MODE]** it is switched to the symbol mode. By again pressing this key combination, the display mode is exited. So it can be selected between two address representations:

1) Symbol allocation display above the contact
(Only addresses, that are not provided with SYMB, are displayed)

2) Address display above the contact and SYMB display under the contact

A contact can also be called up with the direct input of a symbol. The advantage is, that the user doesn't have to remember a lot of addresses. A contact is simply called up with the key **[K]**, after which a 9 character input field for the SYMB is displayed above the contact:



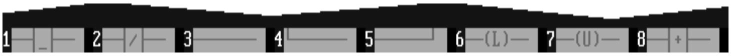
```
19
|
|
20─┤ ├──
|
```

The user can now enter an address with the input of an symbol allocation, that is already distributed in the SYMB-editor.
E.g.: **NOT OUT** (see example above)

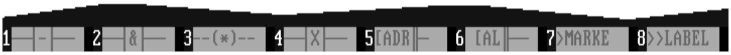
3.6. CONTACTS AND CONNECTIONS

A	B	C	PG1000	CP80	PG-PC	PC80
●	●	●	●	●	●	●

All contacts are offered by the function keys. Pressing [F2 CONTACT] opens the contact input menu; the occupancy of the softkey line changes:



With the function key [F9] the occupancy of the softkey line is switched and additional contacts are offered:



With the function keys any contacts and connections, that can be used in the LAD editor, are accessible.

With the [Esc] key the contact input menu can be exited. The softkey line is switched back and the most important LAD functions are displayed again.

CONTACT OVERVIEW

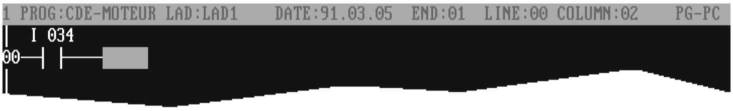
[F1...F8]		**[F9] + [F1...F8]**	
F1 —| _ |—	Contact (NO)	F1 —| - |—	Negative Edge Generation
F2 —| / |—	Negation (NC)	F2 —| & |—	Positive and Negative Edge Generation
F3 ———	Horizontal Connection	F3 —(*)—	Temporary Storage in a Flag or Counter
F4 └———	Vertical Connection left of the cursor to top	F4 —| X |—	Exclusive-Or Connection
F5 ———┘	Vertical Connection right of the cursor to top	F5 [ADR ||—	Address contact to the Connection to a function block
F6 —(L)—	Latch a 1 bit address	F6 [AL ||—	Connecting a Label to a Function Block
F7 —(U)—	Unlatch a 1 bit address	F7 >MARKE	Temporary Storage without Address Definition
F8 —| + |—	Positive Edge Generation	F8 >>LABEL	Conditional Branch

F1 ─┤_├─

A	B	C	PG1000	CP80	PG-PC	PC80
●	●	●	●	●	●	●

Contact

Representation:



```
1 PROG:CDE-MOTEUR LAD:LAD1    DATE:91.03.05  END:01  LINE:00 COLUMN:02    PG-PC
|  I 034
00─┤ ├──
```

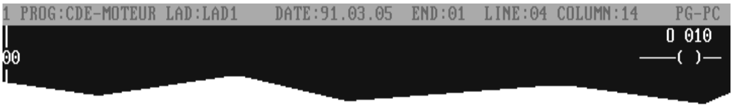
Call:

- <address preselection> <address> [↵]
- [F2 CONTACT] [F1] <address preselection> <address> [↵]
- [F2 CONTACT] [F1] [F3 SYMB_INP] <Symbol allocation> [↵]
- [K] <Symbol allocation> [↵]

Function: The contents of the specified address is loaded (input allocation).

If the contact is not located in the first or last column, it is loaded in the defined memory location and is connected in AND relation with the status of the path.

If the contact is in the **last** column, the status of the path is saved under the defined address (Output allocation). The contact has a different appearance in the right column:



```
1 PROG:CDE-MOTEUR LAD:LAD1    DATE:91.03.05  END:01  LINE:04 COLUMN:14    PG-PC
|                                                                    O 010
00                                                                 ───( )─
```

In order to call up an output allocation, there is another possibility in the LAD editor main menu:

Call:

- Exiting the contact input menu with [Esc], to get into the main menu.
- [F9] + [F2] <address preselection> <address> [↵]
- [F9] + [F2] [F3 SYMB_INP] <symbol assignment> [↵]

[F9] + F2 →( )──

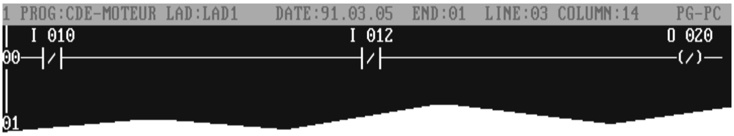
Function: With this function a path is made from the current cursor position to the last column and in the last column a contact is automatically assigned for an output.

F2 —┤/├—

A	B	C	PG1000	CP80	PG-PC	PC80
●	●	●	●	●	●	●

Negation (Normal Closed Contact)

Representation:



```
1 PROG:CDE-MOTEUR LAD:LAD1    DATE:91.03.05  END:01  LINE:03 COLUMN:14    PG-PC
|  I 010                                 I 012                             O 020
00─┤/├───────────────────────────────────┤/├───────────────────────────────(/)─
|
|
01
```

Call:
- [F2 CONTACT] [F2] <address preselection> <address> [↵]
- [F2 CONTACT] [F2] [F3 SYM_INP] <symbol allocation> [↵]
- [N] <address preselection> <address> [↵]
- [N] [F3 SYM_INP] <symbol allocation> [↵]
- [/] <address preselection> <address> [↵]
- [/] [F3 SYM_INP] <symbol allocation> [↵]

Restrictions: only **1 bit** addresses (I, O, F, S, T)

Function: To load the contents of the defined memory location.

If this contact is in the **last** column, the value of the path is negated and stored in the defined memory location.

If the contact is located in a **different** column, the contents of the defined memory location are inverted and connected in AND relation with the path.

If existing contacts should be negated or the negation should be interrupted, the cursor must be positioned in the contact and the key [N] or [/] must be pressed.

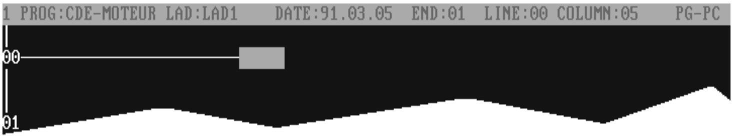
Note: The negation for the addresses G to N of extended output/input modules (24 channel cards) is not possible.

F3 ———

A	B	C	PG1000	CP80	PG-PC	PC80
●	●	●	●	●	●	●

Horizontal Connections

Representation:



```
1 PROG:CDE-MOTEUR LAD:LAD1    DATE:91.03.05  END:01  LINE:00 COLUMN:05    PG-PC
```

Call:

- [F2 CONTACT] [F3]
- [Space]

Function: This function is used, to draw a horizontal path or to delete an existing path.

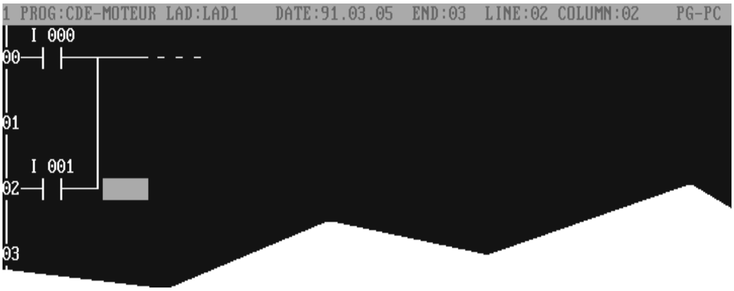
If the cursor is located on an existing path, it is deleted by pressing [Space] or [F2 CONTACT] [F3].

F4 └────── F5 ──────┘

A	B	C	PG1000	CP80	PG-PC	PC80
●	●	●	●	●	●	●

Vertical Connection

Representation:



```
1 PROG:CDE-MOTEUR LAD:LAD1    DATE:91.03.05  END:03  LINE:02 COLUMN:02    PG-PC
|  I 000
00─┤ ├─────── - - -
|
01
|  I 001
02─┤ ├──┘
|
03
```

Call:

- [F2 CONTACT] [F4] [V] } to the left of the cursor
- [F2 CONTACT] [F5] [Ctrl] [V] } to the right of the cursor

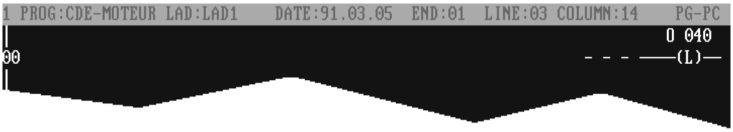
Function: The connection is made to the left/right of the cursor up to the next path. With the same function an existing connection to the next horizontal path is deleted.

F6 —(L)—

A	B	C	PG1000	CP80	PG-PC	PC80
●	●	●	●	●	●	●

Latch a 1 bit address

Representation



```
1 PROG:CDE-MOTEUR LAD:LAD1    DATE:91.03.05  END:01  LINE:03 COLUMN:14    PG-PC
|                                                                        O 040
00                                                               - - - ——(L)—
|
```

Call:

- [L] <address preselection> <address> [↵]
- [F2 CONTACT] [F6] <address preselection> <address> [↵]
- [F2 CONTACT] [F6] [F3 SYM_INP] <symbol allocation> [↵]
- [L] [F3 SYM_INP] <symbol allocation> [↵]

Restrictions: This contact can be called up in the last column of the LAD in connection with **1 bit** addresses (F, O, S).

Function: The terminated memory location is set to “1”, if the low value bit of a path is “1”. The application of [L] is only recommended in connection with [U].

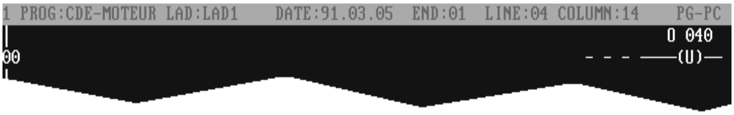
Note: Presetting the addresses G to N with extended input/output modules (24 channel cards) is not possible.

F7 —(U)—

A	B	C	PG1000	CP80	PG-PC	PC80
●	●	●	●	●	●	●

Unlatch a 1 bit address

Representation:



```
1 PROG:CDE-MOTEUR LAD:LAD1    DATE:91.03.05  END:01  LINE:04 COLUMN:14    PG-PC
                                                                          0 040
00                                                                 - - - ——(U)—
```

Call:
- [F2 CONTACT] [F7] <address preselection> <address> [↵]
- [F2 CONTACT] [F7] [F3 SYM_INP] <symbol allocations> [↵]
- [U] <address preselection> <address> [↵]
- [U] [F3 SYM_INP] <symbol allocations> [↵]

Restrictions: This contact can be called up in the last column of the LAD in connection with **1 bit** addresses (F, O, S).

Function: The connected memory location is set "0", if the low value bit of the path is "1". The application of [U] is only recommended in connection with [L].

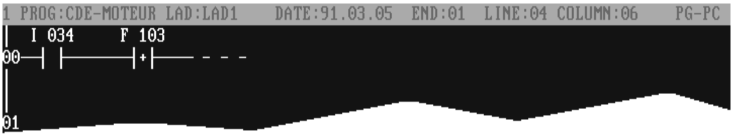
Note: Resetting addresses G to N of extended input/output modules (24 channel cards) is impossible.

F8 ─┤+├─

A	B	C	PG1000	CP80	PG-PC	PC80
●	●	●	●	●	●	●

Positive Edge Generation

Representation:



```
1 PROG:CDE-MOTEUR LAD:LAD1    DATE:91.03.05  END:01  LINE:04 COLUMN:06    PG-PC
|   I 034     F 103
00─┤ ├──────┤+├── - - -
|
|
01
```

Call:

- [F2 CONTACT] [F8] [F] <address> [↵]
- [F2 CONTACT] [F8] [F3 SYM_INP] <symbol allocations> [↵]
- [+] [F] <address> [↵]
- [+] [F3 SYM_INP] <symbol allocations> [↵]

Function : If a least significant bit of a path changes status from “0” to “1”, this contact gives a pulse with the length of exactly one program cycle “1”.

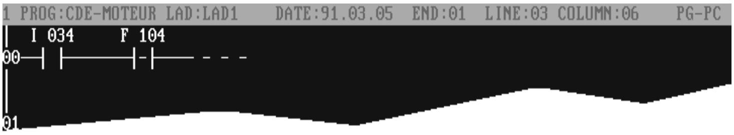
Note: An edge-flag must be defined for this function, that can't be changed by the program in another location. If positive edge generation is used more often, a new flag must be distributed every time.

[F9] + F1 ─┤-├─

A	B	C	PG1000	CP80	PG-PC	PC80
●	●	●	●	●	●	●

Negative Edge Generation

Representation:



```
1 PROG:CDE-MOTEUR LAD:LAD1    DATE:91.03.05  END:01  LINE:03 COLUMN:06    PG-PC
|  I 034     F 104
00─┤ ├──────┤-├── - - -
|
01
```

Call:

- [F2 CONTACT] [F9] + [F1] [F] <address> [↵]
- [F2 CONTACT] [F9] + [F1] [F3 SYM_INP] <symbol allocation> [↵]
- [-] [F] <address> [↵]
- [-] [F3 SYM_INP] <symbol allocation> [↵]

Function: If the least significant bit of the path changes its status from “1” to “0”, this contact gives a pulse that is exactly one program cycle long “1”.

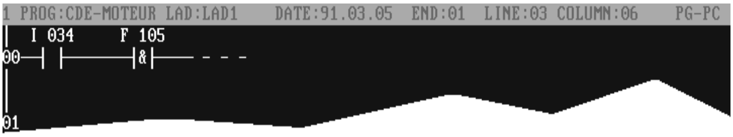
Note: For this function an edge flag must be defined, that may not be changed by the program in a different location. If negative edge generation is used more often, a different flag must be defined every time.

[F9] + F2 —| & |—

A	B	C	PG1000	CP80	PG-PC	PC80
●	●	●	●	●	●	●

Positive and Negative Edge Generation

Representation:



```
1 PROG:CDE-MOTEUR LAD:LAD1    DATE:91.03.05  END:01  LINE:03 COLUMN:06    PG-PC
|  I 034     F 105
00—| |———|&|—— - - -
|
|
01
```

Call:

- [F2 CONTACT] [F9] + [F2] [F] <address> [↵]
- [F2 CONTACT] [F9] + [F2] [F3 SYM_INP] <symbol allocation> [↵]
- [&] [F] <address> [↵]
- [&] [F3 SYM_INP] <Symbol allocation> [↵]

Function: If the least significant bit changes its status, this contact gives a pulse, that is exactly one program cycle long “1”.

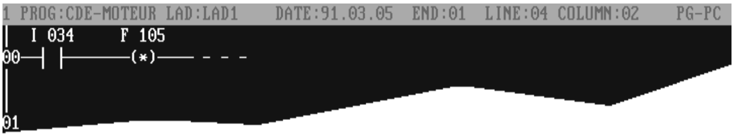
Note: For this function an edge-flag must be defined, that may not be changed by the program in a different location. If the edge generation is used more often, a different flag must be distributed every time.

[F9] + F3 —(*)—

A	B	C	PG1000	CP80	PG-PC	PC80
●	●	●	●	●	●	●

Temporary Storage

Representation:



```
1 PROG:CDE-MOTEUR LAD:LAD1    DATE:91.03.05  END:01  LINE:04 COLUMN:02    PG-PC
|  I 034     F 105
00—| |————(*)——— - - -
|
01
```

Call:

- [F2 CONTACT] [F9] + [F3] <address preselection> <address> [↵]
- [F2 CONTACT] [F9] + [F3] [F3 SYM_INP] <symbol allocation> [↵]
- [*] <address preselection> <address> [↵]
- [*] [F3 SYM_INP] <symbol allocation> [↵]

Restrictions: Only F, R, O or S can be entered as <address preselection >.

This function can not be called up in the first or last column.

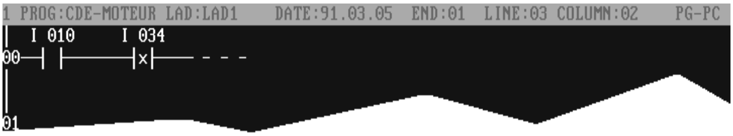
Function: The path connections are stored in the specified address. The status of the path does not change.

[F9] + F4 ─┤x├─

A	B	C	PG1000	CP80	PG-PC	PC80
●	●	●	●	●	●	●

Exclusive-OR Relation

Representation



```
1 PROG:CDE-MOTEUR LAD:LAD1    DATE:91.03.05  END:01  LINE:03 COLUMN:02    PG-PC
|  I 010     I 034
00─┤ ├──────┤x├── - - -
|
|
01
```

Call:

- [F2 CONTACT] [F9] + [F4] <address preselection> <address> [↵]
- [F2 CONTACT] [F9] + [F4] [F3 SYMB_INP] <symbol allocation> [↵]
- [X] <address preselection> <address> [↵]
- [X] [F3 SYMB_INP] <symbol allocation> [↵]

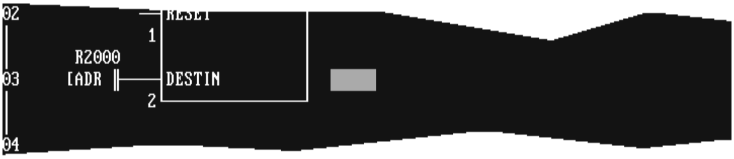
Function: The existing result of a path is connected in exclusive Or relation to the contents of the defined memory location.

[F9] + F5 [ADR ||—

A	B	C	PG1000	CP80	PG-PC	PC80
●	●	●	●	●	●	●

Address Contact

Representation:



Call:

- [F2 CONTACT] [F9] + [F5] <address preselection> <address> [↵]
- [F2 CONTACT] [F9] + [F5] [F3 SYM_INP] <symbol allocation> [↵]

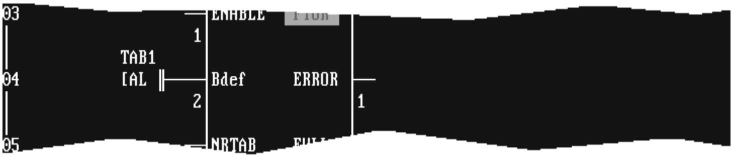
Function: This contact is used, to connect the address of a memory location to a function block. I.e.: The address is submitted to the FBK, not the contents of the memory location.

[F9] + F6 [AL |⊢—

A	B	C	PG1000	CP80	PG-PC	PC80
	●	●	●	●	●	●

Label contact

Representation:



Call: - [F2 CONTACT] [F9] + [F6] <LABEL> [↵]

Function: This contact is used, to connect the address of a label to a function block. It is used for example for the connection of tables to the FBKs.

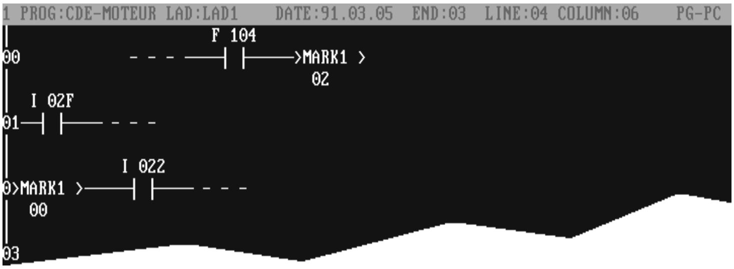
Note: A label may have a maximum length of four characters and must start with a letter.

[F9] + F7 >MARKE

A	B	C	PG1000	CP80	PG-PC	PC80
●	●	●			●	●

Temporary storage without address definition

Representation:



```
1 PROG:CDE-MOTEUR LAD:LAD1    DATE:91.03.05  END:03  LINE:04 COLUMN:06    PG-PC
|                         F 104
00          - - - ───┤ ├────>MARK1 >
|                              02
|  I 02F
01─┤ ├─── - - -
|
|            I 022
0>MARK1 >───┤ ├─── - - -
|  00
|
03
```

Call: - [F2 CONTACT] [F9] + [F7] <Marke> [↵]

Restrictions: <Marke> may have a maximum of 6 characters.

Function: The connection result of a 1 byte path is temporarily stored in an internal memory location of the PLC, that can't be accessed by the user. A defined marke may be present only twice in the ladder diagram; once as the source and once as the destination.

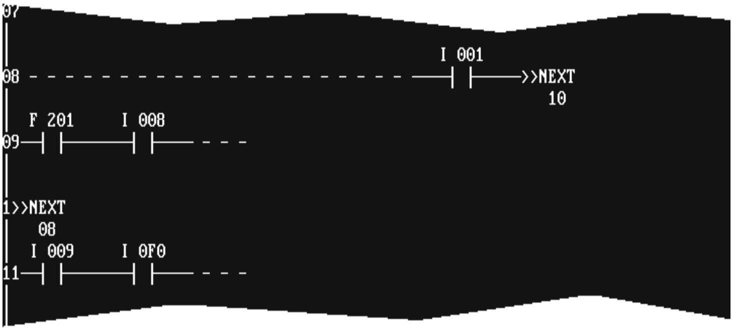
The path (line number) which is to continue or to be interrupted is displayed below the marke.

[F9] + F8 >>LABEL

A	B	C	PG1000	CP80	PG-PC	PC80
●	●	●			●	●

Conditional Branch

Representation :



```
07
|
|                                                  I 001
08 - - - - - - - - - - - - - - - - - - - - - ───┤ ├────>>NEXT
                                                            10
|  F 201      I 008
09─┤ ├───────┤ ├─── - - -
|
|
1>>NEXT
|   08
|  I 009      I 0F0
11─┤ ├───────┤ ├─── - - -
|
```

Call: - [F2 CONTACT] [F9] + [F8] <LABEL> [↵]

Note: <LABEL> may have a maximum length of four characters and must start with a letter.

In the line, where the second LABEL is entered, no additional contact may be entered.

Function: A conditional branch is allowed with the input of a label in a path. The branch is executed, if the path is not zero. (In the above example, line 9 is not executed any more.)

Under the label is displayed, in which line the processing is to be continued or interrupted.

Contact Input Interruption

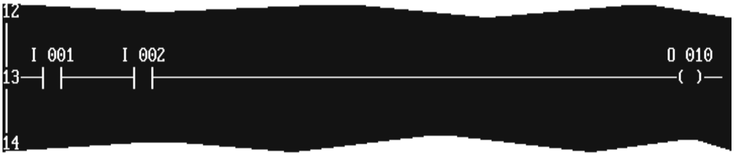
If a contact has been called with one of the described function keys, the PROgramming SYStem waits for the input of an address, label or marke. This input can be interrupted with the **[Esc]** key. The contact is deleted and the cursor appears again in the LAD editor.

AND Relation

A	B	C	PG1000	CP80	PG-PC	PC80
●	●	●	●	●	●	●

An AND relation corresponds with the **serial switch** of contacts.

Example:



The input I 001 is connected with the input I002 in AND relation and the result is assigned to the output O 010. The corresponding STL program is as follows:

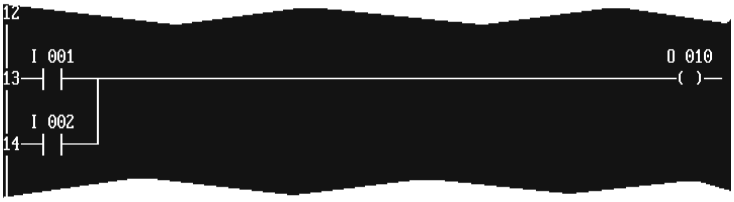
```
LDAA  I 001
ANDA  I 002
STAA  O 010
```

OR Relation

A	B	C	PG1000	CP80	PG-PC	PC80
●	●	●	●	●	●	●

An OR relation is equivalent to the **parallel connection** of contacts.

Example:



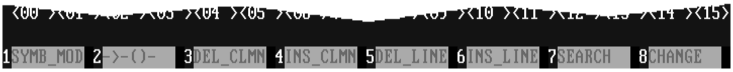
The input I 001 is connected to the input I 002 in OR relation and the result is assigned to the output O 010. The respective STL program is as follows:

```
LDAA  I 001
ORAA  I 002
STAA  O 010
```

3.7. EDIT FUNCTIONS

A	B	C	PG1000	CP80	PG-PC	PC80
●	●	●	●	●	●	●

In order to subsequently change and process ladder diagrams several editing functions are available. When shifting the softkey line with the [F9] key the following functions are offered:



```
<00 ><01 ><02 ><03 ><04 ><05 ><06          ><09 ><10 ><11 ><12 ><13 ><14 ><15>

1SYMB_MOD 2->-()-   3DEL_CLMN 4INS_CLMN 5DEL_LINE 6INS_LINE 7SEARCH   8CHANGE
```

Delete Column

Command: - **[F9] + [F3 DEL_CLMN]**
- **[Del]**

Function: The contact or path segment in which the cursor is located, is deleted. The cursor is moved to the left by one position.

Insert Column

Command: - **[F9] + [F4 INS_CLMN]**
- **[Ins]**

Function: A line draw (one column wide) is inserted in the path. The contacts to the right of the cursor are moved to the right. The cursor must be located on a contact, or directly to the left of a contact, to be able to perform this function.

Delete Line

Command: - **[F9] + [F5 DEL-LINE]**
- **[Ctrl] + [Del]**

Function: The line in which the cursor is positioned, is deleted.

Insert line

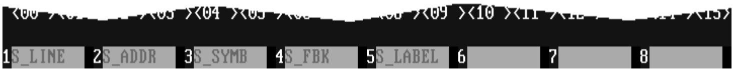
Command: - **[F9] + [F6 INS_LINE]**
- **[Ctrl] + [Ins]**

Function: The line in which the cursor is positioned, is moved down. A line is inserted in the current cursor position.

SRCH_FUNCTION

Command : - **[F9] + [F7 SEARCH]**

Function: With this function the softkey line allocation is changed. Different possibilities are offered, for which criteria the user can search:



```
1S_LINE  2S_ADDR  3S_SYMB  4S_FBK  5S_LABEL 6        7        8
```

[F1 S_LINE] - Search for line:

In an input field a line number between 0 and 49 can be entered. The cursor is set to the specified line.

[F2 S_ADDR] - Search for address:

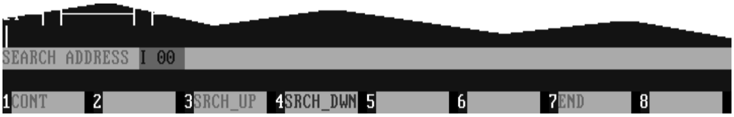
A search mask is displayed, where the desired addresses or address parts are to be entered. After quitting with [↵] the LAD is searched for the specified address from the current cursor position to the end.

If no contact, which is identical to the defined search mask exists, the following message is displayed:

```
E129  VARIABLE NOT FOUND
```

When a contact with the corresponding address is located, the LAD cursor is positioned on the contact.

In both cases the allocation of the softkey line is changed :



```
SEARCH ADDRESS I 00
1CONT    2        3SRCH_UP 4SRCH_DWN5        6        7END     8
```

[F1 CONT] After locating a contact, the search for other addresses can be continued by pressing this key.

[F3 SRCH_UP] The search goes from the current cursor position to the top.

[F4 SRCH_DWN] The search goes from the current cursor position to the bottom.

To continue the search procedure in a new direction, the function key [F1 CONT] must be pressed. The current search direction is marked by the text in the soft key line that is displayed in inverse video.

[F7 END] The search is ended with this function key.

[F3 S_SYMB] - Search for Symbols:

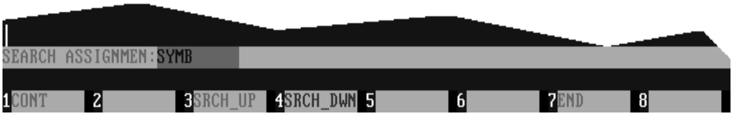
A search mask appears, in which the desired symbol must be entered. After quitting with [↵] the LAD is searched for the defined SYMB from the current cursor position to the end.

If no contact exists, that is identical to the specified search mask, the following message is displayed:

```
E129  VARIABLE NOT FOUND
```

When a contact with the corresponding SYMB is located, the LAD cursor is positioned on the contact.

In both cases the softkey line allocation is changed:



SEARCH ASSIGNMEN:SYMB

The rest of the operation is the same as the address search.

[F4 S_FBK] - Search for function blocks:

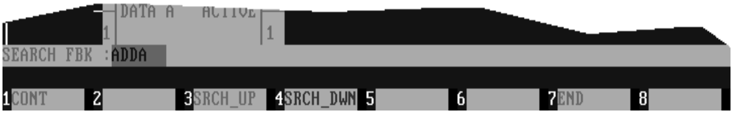
A search mask appears, where the desired FBK name must be entered. After quitting with [↵] the LAD is searched for the defined FBK from the current cursor position to the end.

If no FBK exists, that is identical with the search mask, the following message is displayed:

```
E129  VARIABLE NOT FOUND
```

When locating a FBK, the LAD cursor is positioned on it.

In both cases the softkey line allocation is changed:



SEARCH FBK :ADDA

The rest of the operation is the same as the address search.

If a FBK is found, it is displayed in inverse video. With [F7 END] the user gets back to the LAD editor. The FBK is still displayed in inverse video, and the LAD editor is positioned in the FBK editor mode (more information is given in this chapter, section 6.5. Editing FBKs).

[F5 S-LABEL] - Search for Label:

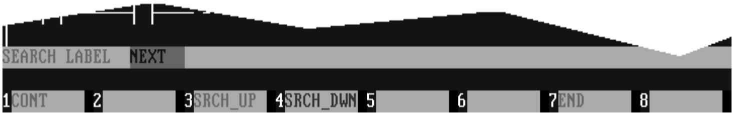
A search mask is defined, in which the desired label must be entered. After quitting with [↵] the LAD is searched for the specified label from the current cursor position to the end.

If no label exists, that is identical with the defined search mask, the following message is displayed:

```
E129  VARIABLE NOT FOUND
```

When locating a label, the LAD cursor is positioned on it.

In both cases the softkey line allocation is changed:



SEARCH LABEL NEXT

The rest of the operation is the same as the address search.

Exchange-Function

Command: - **[F9] + [F8 CHANGE]**

Function: Searches for an address (address part), to be exchanged with another address (address part) of your choice.

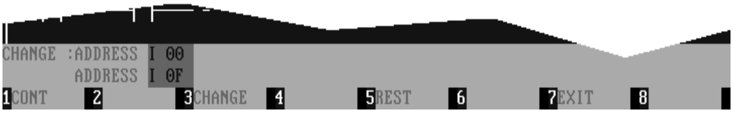
Two masks must be defined for this function; the search and exchange mask:

Search mask: Input of the address (address part), that should be exchanged
Exchange mask: Input of the address (address part), that should be inserted.

After the input of both masks is ended with [↵], the address that is identical with the search mask is searched for. The search procedure begins from the cursor position, downwards, to the end of the LAD.

Be sure to include spaces in the address. If no address exists, that is identical with the specified search mask, the exchange procedure is interrupted.

When locating an address, the LAD cursor is positioned there, and the allocation of the softkey line is changed:



```
CHANGE :ADDRESS I 00
        ADDRESS I 0F
```

[F1 CONT] The located address is not exchanged by the exchange mask and the search for the next address is continued.

[F3 EXCHANGE] The located address is exchanged by the exchange mask, and the search for the next address is continued.

[F3 REST] All addresses, that are identical with the search mask, are replaced by the exchange mask without any questions.

[F7 END] With this function key the exchange procedure is interrupted.

Note: To guarantee that an exchange procedure is executed in the entire LAD, the cursor must first be positioned in line 0 / column 0.

3.8. DOCUMENTATION

In order to make subsequent editing and processing of the program possible, it is important, to create clear and comprehensive documentation for the program. The clearer and the more definite a document is, the more understandable and efficient the program will be, even after several years.

The program readability is influenced by several elements:

- **Structured Programming**
- useful allocation of **Label, LAD, SEQC** and **FBK name**
- definite and clear **symbol allocation**
- **comments**, that are easy to understand

[F6 TEXT] - Comment allocation

Any comment can be allocated with this function on every LAD position, where **no** contact, path, FBK or logic block is located.

The entry of comments is terminated with [↵], and the input mode for the LAD is displayed again.

[F5 PRINT] - Print of the called LAD

With this function, the last selected LAD is printed. This print out can be used for correction, since a copy is easier to read, than the display on screen. Also, the user has complete documentation in hand.

Example:

```
1 PROG:MOTOR-CTRL  LAD:MO-1   DATE:92.02.19 END:16 LINE:01 COLUMN:00    PG PC
 ****************************************************************************
00**                   M O T O R   1    assembly line supply                **
│****************************************************************************
│
01         * Motor off, if NOT OUT is pressed!
│
│  I 000                                                                O 010
02──┤ ├─────────────────────────────────────────────────────────────────(U)──
│NOT OUT                                                              Motor
│
│  I 000                      no additional function
03──┤ ├───────────────>>END    processing!
│NOT OUT                13

│----------------------------------------------------------------------------
05         * Motor off, if object passed the light barrier!
│
│  I 001     F 100                                                      O 010
06──┤ ├───────┤-├───────────────────────────────────────────────────────(U)──
│Light barr.                                                            Motor

│----------------------------------------------------------------------------
08         * Motor on, if supply is required!
│          * Motor is not turned on, if an object is located in the
│            range of the light barrier.
09
│
│  I 002     F 101     I 001                                            O 010
10──┤ ├───────┤+├───────┤/├─────────────────────────────────────────────(L)──
│Supply               Light barr.                                      Motor

│----------------------------------------------------------------------------
12
│
│
│
13>>END
│   03
│
│  O 010                                                                O 011
14──┤ ├─────────────────────────────────────────────────────────────────( )──
│Motor                                                                Lamp M1
```

3.9. LAD CALL

Sequence Chart Call

A	B	C	PG1000	CP80	PG-PC	PC80
●	●	●			●	●

In all expansion levels of the PROgramming SYStem a LAD can be called from a SEQC. Detailed information on the LAD call from a SEQC is given in this chapter in 2. SEQUENCE CHARTS (SEQC-Editor)”.

Statement List Call

A	B	C	PG1000	CP80	PG-PC	PC80
		●	●	●	●	●

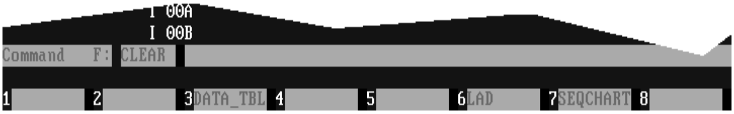
In expansion level C calling a LAD from a STL is also possible. The following command must be inserted in the desired location of the STL program:

`JSR xxxx`   xxxx .... name of the LAD to be called

3.10. DELETING A LAD

Command: ***CLEAR*** **`[F6 LAD] <Name> [↵]`**

With the operating command ***CLEAR*** a LAD can be deleted. If this command is entered in the command line, the softkey allocation line is deleted:



```
I 00A
I 00B
Command   F: CLEAR
1     2     3DATA_TBL 4     5     6LAD     7SEQCHART 8
```

With the selection of [F6 LAD], input of a LAD name and confirmation with [↵] the respective LAD is deleted.

3.11. EXITING THE LAD EDITOR

[Esc] Exits the LAD editor into the command line. The last processed LAD is translated to the processed STL. This STL translation is only accessible in expansion level C.

[Ctrl] + [Esc] Exits the LAD editor into the command line. The changes since the last access are not considered. The LAD is not translated new.

4. LAD STATUS TEST

A	B	C	PG1000	CP80	PG-PC	PC80
●	●	●	●	●	●	●

4.1. GENERAL INFORMATION

The LAD status test is an efficient device for an ONLINE watch of the PLC and for visualizing of the contact status and the internal memory locations.

The contacts and connections in the LAD are displayed at “logic 1” in inverse video or as double line. Active values are displayed in the function blocks.

The LAD status test is an important device for the installation of programs and for error detection.

4.2. LAD STATUS TEST CALL

4.2.1. [F1 RUN]

Command: **`[F1 RUN]`**

Function: 1) The program is translated and transferred to the PLC.

2) Contrary to a RUN from the command line **no** RESET is triggered in the PLC CPU.

3) The menu of the LAD status test is called.

4) If a status test is running in the other window, the following error message is displayed:

```
E188  LDMO NOT POSSIBLE
```

4.2.2. *LDMO* <NAME> [↵]

Command: ***LDMO* <NAME> [↵]**

Function: 1) The menu of the LAD status test is called. The program is not transferred to the PLC CPU.

2) If the program has been changed since the last status test, it is realized by the PROgramming SYStem. The following warning is displayed in the message line:

```
W233  MAKE PROGRAM INFO ( Y / N ) ?
```

[Y]	program in PLC and PG are compared
[N]	command is interrupted

If [N] is entered or the program in the PG is not identical with the one in the PLC, an error message is displayed:

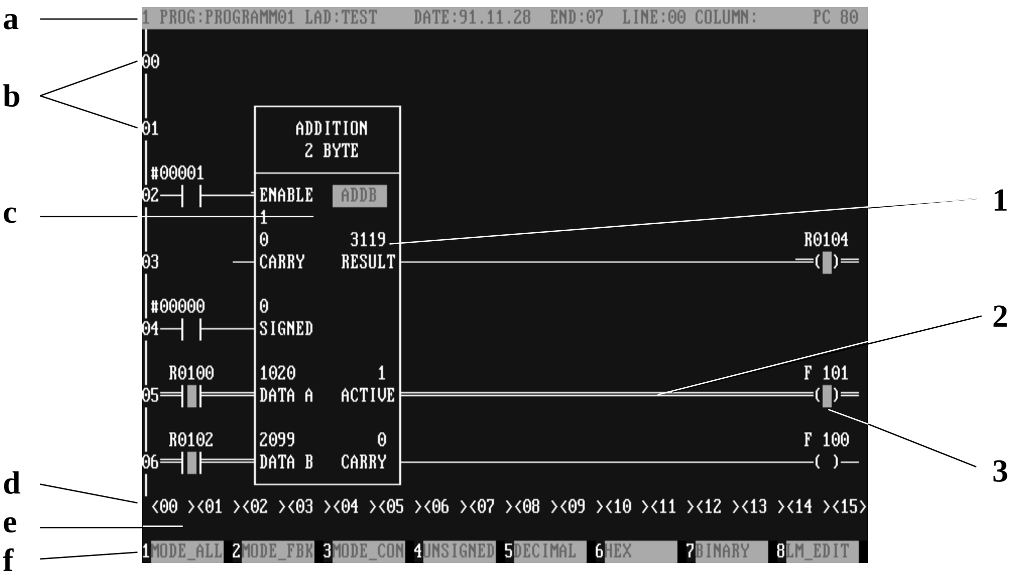
```
E179  PROGRAM IN CPU AND PG DIFFERENT !
```

3) If a STL status test is running in the other window, the following error message is displayed:

```
E188  LDMO NOT POSSIBLE
```

4) When calling ***LDMO*** without entering a name, the last used LAD status test is called.

4.3. MENU STRUCTURE



```
1 PROG:PROGRAMM01 LAD:TEST    DATE:91.11.28  END:07  LINE:00 COLUMN:      PC 80
00
01                ADDITION
                   2 BYTE
 #00001
02─┤ ├──────ENABLE  ADDB
            1
            0      3119                              R0104
03         ─CARRY  RESULT────────────────────────────═( )═
 #00000
            0
04─┤ ├──────SIGNED
   R0100    1020      1                              F 101
05═┤ ├══════DATA A  ACTIVE═══════════════════════════( )═
   R0102    2099      0                              F 100
06═┤ ├══════DATA B  CARRY────────────────────────────( )─
 <00 ><01 ><02 ><03 ><04 ><05 ><06 ><07 ><08 ><09 ><10 ><11 ><12 ><13 ><14 ><15>
1MODE_ALL 2MODE_FBK 3MODE_CON 4UNSIGNED 5DECIMAL 6HEX     7BINARY  8LM_EDIT
```

a) Status line is identical to the LAD editor status line.

b) Line numbering

In every LAD created, a maximum of 50 lines are available. The left part on the screen is numbered from 0 to 49.

c) LAD-Status Display Field

In the LAD status display field the logic states of the path and the contacts are graphically displayed. Active values are displayed in function blocks:

1) Active values in FBKs (different display formats)
2) Path status (double line => 8-Bit value of the path is higher than zero)
3) Contact status (inverse => value higher than zero)

If the LAD is larger than the screen, the displayed section can be moved with the [↑], [↓], [PgUp], [PgDn], [Ctrl] + [PgUp] and [Ctrl] + [PgDn] keys.

d) Column numbering

PG1000-mode	The LAD is divided into 8 columns (00 to 07).
PG-PC-mode	The LAD is divided into 16 columns (00 to 15).

e) Message line

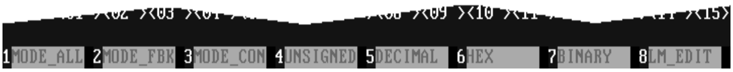
In the message, line messages concerning user errors or messages requiring the input of parameters are displayed.

f) Softkey line

In the softkey line, the most important functions of the LAD status test are offered on the function keys [F1] to [F8].

4.4. LAD STATUS TEST FUNCTIONS

All functions of the LAD status test are accessible with the function keys [F1] to [F8]. The softkey line is allocated as follows:



```
1MODE_ALL 2MODE_FBK 3MODE_CON 4UNSIGNED 5DECIMAL  6HEX      7BINARY   8LM_EDIT
```

[F1 MODE_ALL]

All contacts and paths used in the ladder diagram are represented in inverse video or as a double line according to their logic states .

All other active values (in function blocks) are displayed as numbers (the display format can be selected).

[F2 MODE_FBK]

Only the active values, that are used in the function blocks, are displayed. The logic states of the contacts and the path are not displayed.

With this representation , no time for the graphic illustration is required. This display mode is used, if active values, that rapidly change, should be displayed by FBKs.

[F3 MODE_CON]

Only the logic states of the contacts are displayed. Active values and status of the path are not considered for the representation.

With the function keys [F4] to [F7] different **display formats** for active values can be selected:

[F4 UNSIGNED] The display of active values is decimal without sign.

[F5 DECIMAL] The display of active values is decimal with sign.

[F6 HEX] The display of active values is hexadecimal.

[F7 BINARY] The display of active values is binary. It is only possible to display a value with a maximum size of one byte in the binary format. If the value is higher than one byte it is displayed in the hexadecimal format.

REPRESENTATION LIMITS OF ACTIVE VALUES

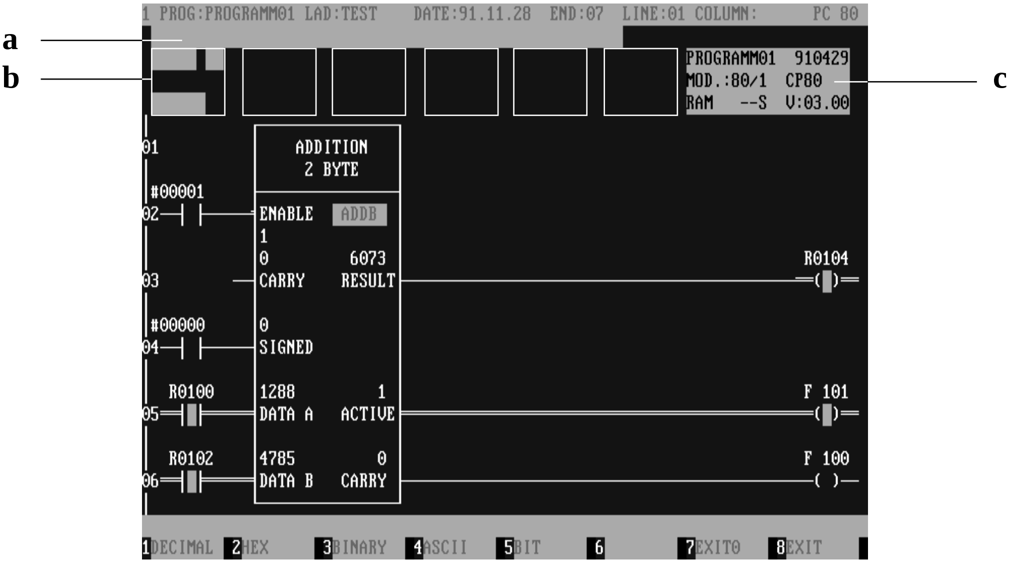
	F1 MODE_ALL		**F2 MODE_FBK**	
	Min.	Max.	Min.	Max.
UNSIGNED	0	9 999 999	0	16 777 215 (=$FFFFFF)
DECIMAL	-999 999	9 999 999	-9 999 999	16 777 215 (=$FFFFFF)
HEX	$0	$FFFFFF	$0	$FFFFFF
BINARY	% 00 0000	% 11 1111	% 0000 0000	% 1111 1111
overflow-display	>>>>>>>		>>>>>>>>	

[F8 LM_EDIT]

With this function, memory locations can be displayed and changed in the LAD status test.

The user also gets information about the CPU, operating system and the currently running program.

The screen format is changed as follows:



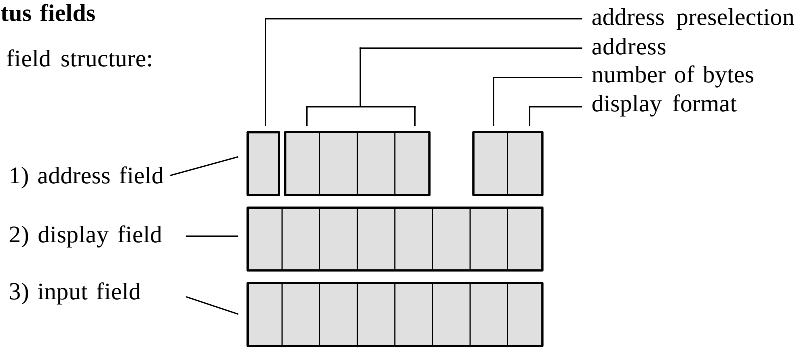
```
1 PROG:PROGRAMM01 LAD:TEST    DATE:91.11.28  END:07  LINE:01 COLUMN:      PC 80
                                                            PROGRAMM01  910429
                                                            MOD.:80/1   CP80
                                                            RAM   --S   V:03.00
01               ADDITION
                  2 BYTE
 #00001
02─┤ ├───ENABLE   ADDB
         1
         0          6073                                          R0104
03      ─CARRY    RESULT──────────────────────────────────────────═( )═

 #00000  0
04─┤ ├───SIGNED

  R0100  1288          1                                          F 101
05═┤ ├═══DATA A   ACTIVE══════════════════════════════════════════( )═

  R0102  4785          0                                          F 100
06═┤ ├═══DATA B   CARRY───────────────────────────────────────────( )─

1DECIMAL 2HEX     3BINARY 4ASCII  5BIT    6       7EXIT0  8EXIT
```

a) Symbol display

In this field the symbol allocation of the address, in which the cursor is positioned, is displayed.

b) Status fields

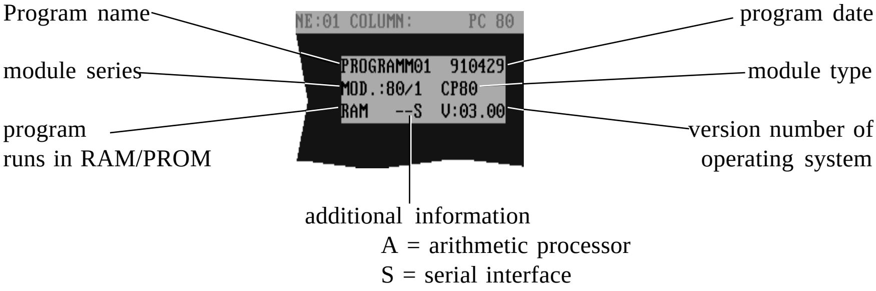
Status field structure:



A full description of the usage of status fields is given in chapter 5 STATEMENT LIST section “7. STL Status Test”.

c) Module information

This three-line display field contains the following information:



```
NE:01 COLUMN:      PC 80
PROGRAMM01  910429
MOD.:80/1  CP80
RAM   --S  V:03.00
```

SOFTKEY LINE ALLOCATION IN THE LM-EDIT MODE

[F1] to [F5] display formats

These function keys allow the change of the display format in the status field. The use of [F1] to [F5] is only possible, if the cursor is located in the field “display formats”.

[F1 DECIMAL]	Memory location contents are represented in decimal format.
[F2 HEX]	Memory location contents are represented in hexadecimal format.
[F3 BINARY]	Memory location contents are represented in binary-format.
[F4 ASCII]	Memory location contents are represented in ASCII-format.
[F5 BIT]	Only the least significant bit with a maximum of 8 memory locations is displayed. This display format is used more efficiently in connection with 1 bit memory locations.

With direct input via keyboard, additional display formats are possible. Detailed information about it and the handling of status fields is given in chapter 5 STATEMENT LIST section “7. STL status test”.

[F7 EXIT 0]

Exits the LM-EDIT-mode. The defined addresses (status fields) and the information about the module are still displayed.

[F8 EXIT]

Exits the LM-EDIT mode but status fields and module information are not displayed any more.

[Esc]

Pressing this key exits the LAD status test back into the LAD editor.

5. LOGIC PLAN PROGRAMMING

A	B	C	PG1000	CP80	PG-PC	PC80
●	●	●			●	●

5.1. GENERAL INFORMATION

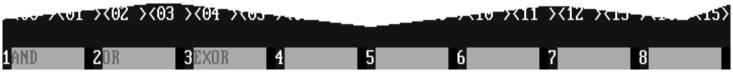
LP programming is to be considered as completion of the ladder diagram. The three basic types of logic connections are available in form of blocks, that are very similar to gates in the digital technique.

These logic blocks are included in the LAD and can be combined optionally with all LAD elements.

5.2. LOGIC BLOCK

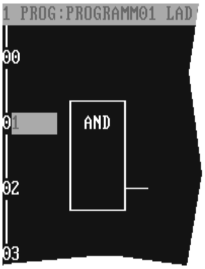
Before a logic block is called up, the cursor must be positioned in the LAD editor in the line/ column, where a logic block should be inserted.

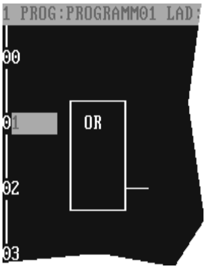
For the call of a logic block, the softkey line is switched by pressing the function key **[F3 LOGIC_BK]**, which is offered in the LAD editor. The softkey line allocation is changed:

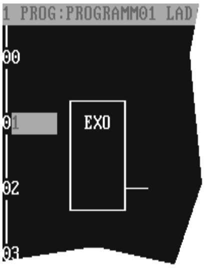


With the function keys [F1 AND], [F2 OR] and [F3 EXOR] the desired logic block can be selected.

If the cursor is for example in line 1/column 1, one of the following logic blocks will be displayed on screen:



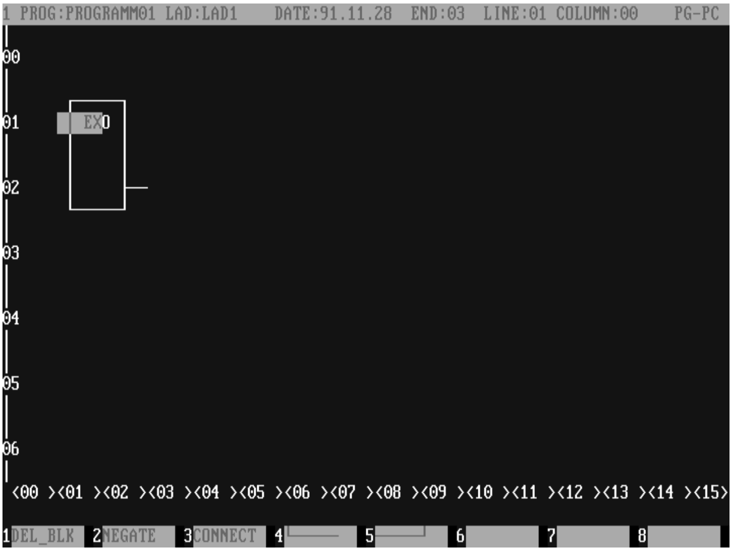




With the [Esc] key it can be switched back to the softkey line. The usual functions for the LAD editor appear.

5.3. LOGIC BLOCK EDITING

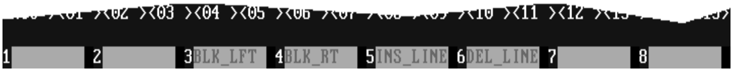
To change a logic block, it must be selected with [F7 EDIT_BLK] + [❖] or [Alt] + [❖]. The cursor is positioned within a logic block after its selection . The cursor can be moved within the logic block with the [❖] keys. If a logic block is selected, the softkey allocation changes:



```
1 PROG:PROGRAMM01 LAD:LAD1    DATE:91.11.28  END:03  LINE:01 COLUMN:00    PG-PC
```

Key **Key combination**	**FUNCTION**
[F1 DEL_BLK] [Alt] + [Del]	The selected logic block is deleted.
[F2 NEGATE] [N] [/]	The input/output, that is located on the cursor, is inverted.
[F3 CONNECT] [Space]	With this function, an input is deleted or set.
[F4] [V]	If the cursor is in the logic block, a vertical connection up to the next path is created directly left of the logic block.
[F5] [Ctrl] + [V]	If the cursor is in the initial page, a vertical connection up to the next path is created directly right to the logic block.

Additional functions are available with the function key **[F9]**:


><02 ><03 ><04 ><05 ><06 ><07 ><10 ><11 ><12
1
2
3BLK_LFT
4BLK_RT
5INS_LINE
6DEL_LINE
7
8

Key Key combination	FUNCTION
[F9] + [F3 BLK_LFT] [Del]	The selected logic block is moved to the left by one column.
[F9] + [F4 BLK_RT] [Ins]	The selected logic block is moved to the right by one column.
[F9] + [F5 INS_LINE] [Ctrl] + [Ins]	The selected logic block is expanded by one line. The cursor must be positioned on the input page for this function.
[F9] + [F6 DEL-LINE] [Ctrl] + [Del]	The selected logic block is reduced by one line. The cursor must be positioned on the input page for this function.

When exiting the block with the [❖] keys the LP editing mode is exited. The usual allocation of the softkey line in the LAD editor appears.

6. FUNCTION PLAN PROGRAMMING

A	B	C	PG1000	CP80	PG-PC	PC80
●	●	●	●	●	●	●

6.1. GENERAL INFORMATION

The FP programming is similar to structured programming. Program parts are displayed as rectangles on the screen. The user only connects input parameters and the outputs of the block. An extensive collection of standard function blocks is available at B&R, which makes specific programming easier. Standard FBKs exist for:

- analogue value processing
- mathematics function
- general data processing
- communication
- software for support of the hardware available at B&R.

These FBKs are included in the LAD and can be combined with all LAD elements.

Note

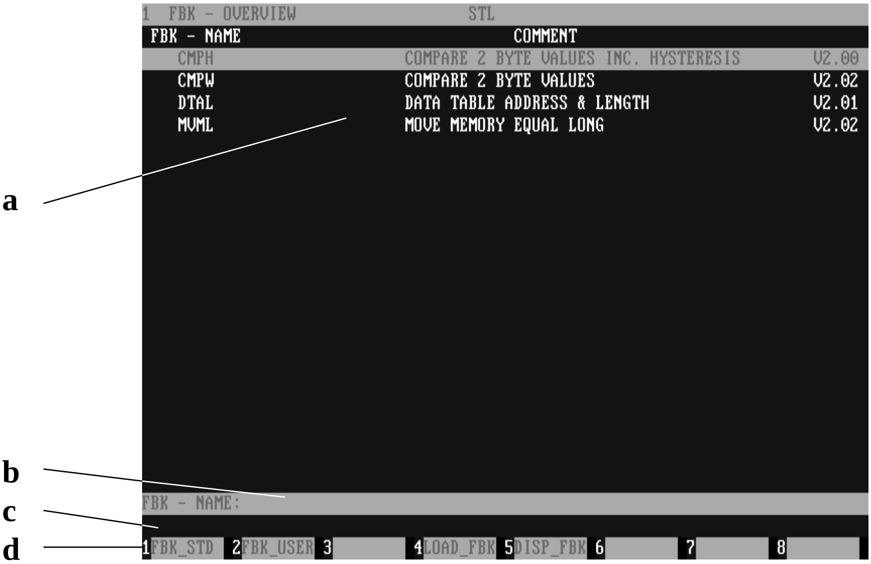
A	B	C	PG1000	CP80	PG-PC	PC80
●	●	●	●	●	●	●

In the expansion level A only the **standard** function blocks of B&R are loaded and included in the LAD.

6.2. FBK OVERVIEW MENU

Before an FBK can be called, the cursor must be positioned in the LAD editor in the line, where the FBK should be inserted. If the cursor is in column 0, no FBK can be positioned/loaded.

Pressing the function key **[F4 FBK]** opens the FBK overview menu for a FBK call. This overview displays the FBKs, that have already been loaded in the application program.



```
1  FBK - OVERVIEW                    STL
FBK - NAME                                   COMMENT
    CMPH                      COMPARE 2 BYTE VALUES INC. HYSTERESIS       V2.00
    CMPW                      COMPARE 2 BYTE VALUES                       V2.02
    DTAL                      DATA TABLE ADDRESS & LENGTH                 V2.01
    MVML                      MOVE MEMORY EQUAL LONG                      V2.02

FBK - NAME:

1FBK_STD 2FBK_USER 3        4LOAD_FBK 5DISP_FBK 6        7        8
```

a) FBK Overview

All FBKs, that are used in the program, are listed in this overview. If the FBK list is longer than one screen page, it can be leafed through with [↑], [↓], [PgDn] or [PgUp].

For easier recognition of an FBK, a short information line about the respective FBK is displayed. This short information is defined by the creation of a FBK (only possible in expansion level C, chapter 10 CREATING FUNCTION BLOCKS).

b) Input line

With the input of an FBK name, a desired FBK can be loaded without selecting [❖]. The input is terminated with [↵].

c) Message line

In the message line messages concerning user errors or messages requiring the input of parameters are displayed.

f) Softkey line

In the softkey line the most important functions are offered on keys [F1] to [F5].

6.3. CALLING EXISTING FBK'S

FBKs, that have been loaded once in the program, do not need to be called a second time from the hard disk or floppy, even if a FBK is used more often in a LAD or in several LAD's, i.e. storage space is saved.

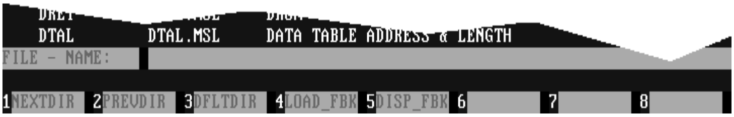
KEY	FUNCTION
[↑] **[↓]**	Selection of a FBK from the FBK overview. '
[F4 LOAD_FBK]	The selected FBK is included in the LAD. If the inserted FBK uses private memory locations, these must be entered. Private memory locations are register or flags, that are used by the program of the FBK (e.g.: for the storage of results). The FBK is displayed in inverse video in the LAD editor. In the softkey line all functions are offered, that are necessary for the FBK editing.
[F5 DISP_FBK]	The selected FBK is displayed, to check if the desired FBK corresponds. [Esc] leads back to the FBK overview menu.
[Esc]	Exiting to the LAD editor.

6.4. LOADING FBKS FROM HARD DISK OR DISKETTE

FBKs can only be read from the “current” drive. This drive can be changed only in the command line with the operating command ***DRIVE*** **<drive name> [↵]**.

KEY	**FUNCTION**
[F1 FBK_STD]	Loading a standard FBK for the standard library. After pressing this function key, the standard FBKs, that are located in the directory `\PGSTD` (standard directory) are displayed. FBKs, that are not marked as standard FBKs by B&R, are not listed and can't, for this reason, be read.
[F2 FBK_USER]	Loading an FBK from the user library. After pressing this function key, all FBKs are displayed, that are located in the directory `\PGUSER` (user directory).

In both cases the softkey line changes as follows:



```
DTAL          DTAL.MSL      DATA TABLE ADDRESS & LENGTH
FILE - NAME:
```

Key	**FUNCTION**
[F1 NEXTDIR]	If the cursor of the FBK overview menu is located in a directory, this directory is opened. The contents of this directory are displayed. This function corresponds with the DOS-command “`cd <directory name>`”.
[F2 PREVDIR]	The previous directory is changed and its contents are displayed. [F2 PREVDIR] corresponds with the DOS command “cd `..`”.
[F3 DFLTDIR]	The current directory is defined as the “standard-directory” or as the “user-directory”, depending if **[F1 FBK_STD]** or **[F2 FBK_USER]** was previously used.
[Esc]	The FBK overview menu of existing FBKs is displayed again.

With [↑], [↓], [F4 LOAD_FBK] and [F5 DISP_FBK] a FBK can be selected and loaded from the current drive the same as calling existing FBKs.

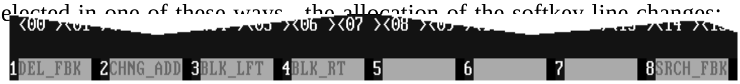
6.5. FBK EDITING

To change an FBK, it first must be selected. This is possible in two different ways:

- With either of the key combinations **[F7 EDIT_BLK] + [❖]** or **[Alt] + [❖]** the LAD cursor is moved into a FBK. If a FBK is selected in this way, it is displayed in inverse video.
- With the function **[F8 SRCH_FBK] + [❖]** or **[Ctrl] + [❖]** an FBK is searched for in the direction entered beginning from the current cursor position. The first FBK found is selected and displayed in inverse video. If no FBK is found, an error message is displayed:

```
E117  NO FBK IN SEARCH PATH
```

If a FBK is selected in one of these ways, the allocation of the softkey line changes:



```
<00 ><01 ... ><05 ><06 ><07 ><08 ><09 ... ><13 ><14 ><15
1DEL_FBK 2CHNG_ADD 3BLK_LFT 4BLK_RT 5        6        7        8SRCH_FBK
```

Key **Key combination**	**FUNCTION**
[F1 DEL_FBK] **[Alt] + [Del]**	The selected FBK is deleted of the LAD.
[F2 CHNG_ADD]	The addresses of private memory locations can be changed. Every change must be terminated with [↵]. If an FBK has more private memory locations, these can be selected with [❖]. This mode is exited with [Esc].
[F3 BLK_LFT] **[Del]**	The FBK is moved left one column.
[F4 BLK_RT] **[Ins]**	The FBK is moved right one column.
[F8 SRCH_FBK] + [❖] **[Ctrl] + [❖]**	According to which of the [v] keys is pressed, the next FBK is searched for in the respective direction. The located FBK is displayed in inverse video, and can be edited.
[❖]	The FBK editing mode is exited and the FBK is displayed normally again.

6.6. DELETING A FBK

A	B	C	PG1000	CP80	PG-PC	PC80
		●	●	●	●	●

STL listings of FBKs that can't be used in the LAD, can be located in the working memory. The FBK is also listed in the FBK overview menu and could be included again in a LAD. If such FBKs are not required any more, valuable memory is occupied.

The STL listing of a FBK can only be deleted from the STL in the expansion level C .

DELETING A FBK FROM THE STL:

In order to delete a STL listing of a FBK from the STL, the start address and the end address must be known. With the operating command ***DELETE BLOCK*** or **[F9] + [F7 DELETE] [F7 BLOCK]** the specified range is deleted from the STL. The description of this command is given in chapter 5 STATEMENT LIST.

CHAPTER 5
STATEMENT LIST PROGRAMMING

Contents:

1. GENERAL INFORMATION

A	B	C	PG1000	CP80	PG-PC	PC80
		●	●	●	●	●

The most complicated programming problems can be solved with status list programming - either with B&R specific, German command abbreviations or with MOTOROLA® -mnemonics. STL is more than assembler programming. The command block includes elementary processing commands as well as complex floating point calculation routines, that are integrated as default in all B&R systems.

Advantages:

- Creating memory and time optimization programs is possible with the help of STL-programming.
- Almost all problems are soluble in STL.

Disadvantages:

- Good knowledge of STL-commands are required.
- An STL is not arranged as clearly as an LAD.
- An STL program is not as flexible as an LAD. It is not changed easily.
- STL programs must be documented more extensively than the LAD, because the graphic representation of the LADs is already part of the documentation.

Requirements for this chapter:

In this chapter basic knowledge of STL commands are required because only the STL editor is described. A summary of the STL commands is given in Appendix B.

2. STL EDITOR ACCESS

A	B	C	PG1000	CP80	PG-PC	PC80
		●	●	●	●	●

Command: **`[F1 STL] <LN/LB> [↵]`**

After confirming the command input with [↵], the STL editor is displayed. The input cursor (blinking cursor) is located in the command line.

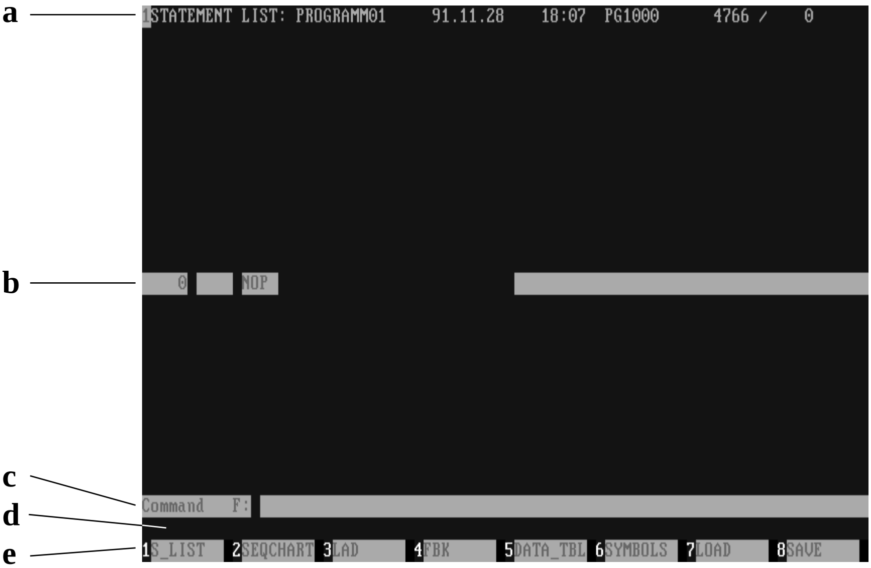
With the input of a line number (LN) or a label (LB), the respective line of the STL editor is performed inverse (=input line of the STL editor). If no parameter follows the ***STL*** command, the line that was previously processed in the STL is described inverse.

By pressing **[↵]** in the command line the cursor is positioned to the input line of the STL editor. If something needs to be changed or inserted in a certain STL line, the input cursor can be set to the respective line with the input of **<LN/LB> [↵]**.

The cursor is set back to the command line with [Esc].

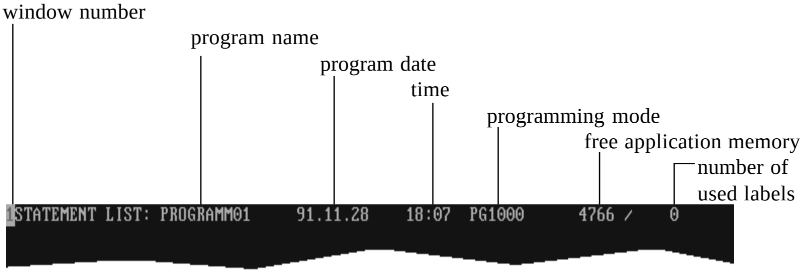
3. STL EDITOR STRUCTURE

After accessing the STL editor with **[F1 STL] [↵]** the following figure is displayed:



```
1STATEMENT LIST: PROGRAMM01     91.11.28    18:07  PG1000      4766 /    0

         0        NOP

Command    F:
```

a) Status line: contains information such as:



```
1STATEMENT LIST: PROGRAMM01     91.11.28     18:07  PG1000      4766 /    0
```

b) Input line

STL commands and comments can be entered in this line. The input line is divided into several editing fields.

c) Command line

The command line serves for the input of operating commands. The most important commands are also available with the function keys from [F1] to [F8].

d) Message line

Messages, that apply to the operator are displayed. A wrong command or bad operation causes an error message to be displayed. Messages that demand parameter inputs can also be displayed.

e) Softkey line

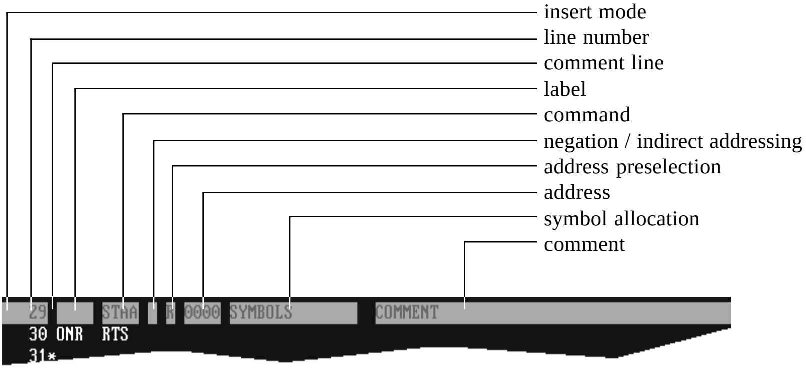
In the softkey line the STL editor functions are offered on the keys [F1] - [F8].

With the keys [F9] and [F10] the occupancy of the softkey line is changed and additional functions are offered.

4. INPUT LINE

A	B	C	PG1000	CP80	PG-PC	PC80
		●	●	●	●	●

The input line is divided into the following editing fields:



```
 29     STHA     R 0000 SYMBOLS         COMMENT
 30 ONR  RTS
 31*
```

4.1. INSERT MODE

A	B	C	PG1000	CP80	PG-PC	PC80
		●	●	●	●	●

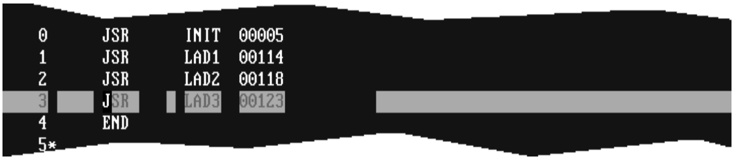
If the character “>” is in this position, the STL editor is in the insert mode. A new line is inserted with every return [↵].

If the insert mode is activated, pressing the return key [↵] will move the STL one line up and the next line can be edited.

Starting the insert mode

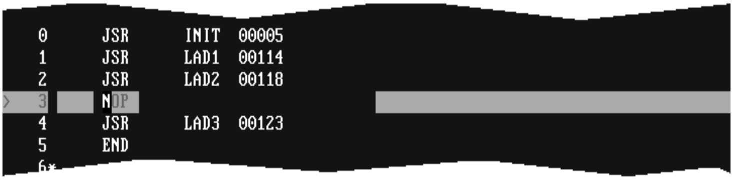
With the key combinations **[Ctrl] + [Ins]** an “NOP” line is added and **[Alt] + [Ins]** a comment line is inserted. The insert mode is started in both cases.

Example: Lines should be inserted between the line 2 and 3



```
  0      JSR      INIT  00005
  1      JSR      LAD1  00114
  2      JSR      LAD2  00118
  3      JSR      LAD3  00123
  4      END
  5*
```

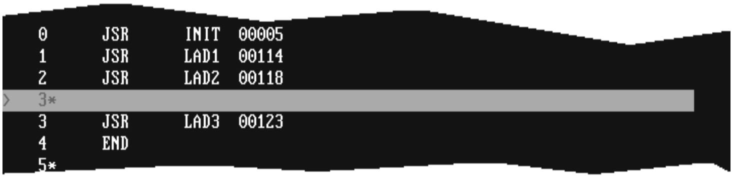
After pressing [Ctrl] + [Ins] a line is inserted:



```
  0      JSR      INIT  00005
  1      JSR      LAD1  00114
  2      JSR      LAD2  00118
> 3      NOP
  4      JSR      LAD3  00123
  5      END
  6*
```

With each pressing of [Ctrl] + [Ins], another line is inserted.

If the key combination [Alt] + [Ins] is pressed instead of [Ctrl] + [Ins] a comment line is inserted between the line 2 and 3:



```
  0      JSR      INIT  00005
  1      JSR      LAD1  00114
  2      JSR      LAD2  00118
> 3*
  3      JSR      LAD3  00123
  4      END
  5*
```

The insert mode is activated in both cases. This is identified by the character “>”.

Ending insert mode

Insert mode can be exited by pressing [↑], [↓], [PgUp], [PgDn] or [Esc].

4.2. LINE NUMBER
PROGRAM STEP NUMBER

A	B	C	PG1000	CP80	PG-PC	PC80
		●	●	●	●	●

All STL lines are provided with consecutive numbers. Comment lines are not counted. They always receive the number of the following program line.

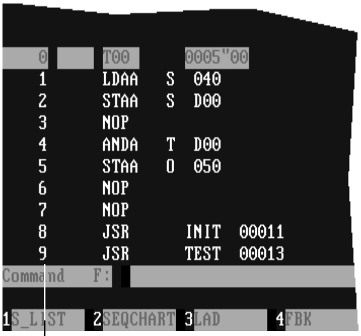
Numbering occurs automatically.

The numbering procedure depends on the programming mode:

1) Numbering in the PG1000 and PG-PC modes (CP40) succeeds after lines.

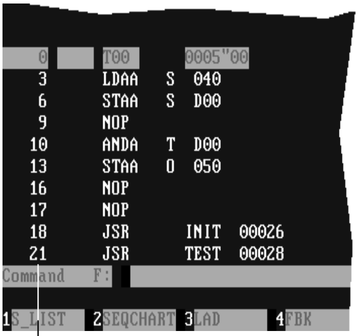
2) in the CP80 and PC80 modes (CP60, CP80) the line number is not displayed, but the program step number (number of used bytes) is displayed. The numbering proceeds in irregular intervals, since every command does not have the same opcode length.

Example: **PG1000 / PG-PC** **CP80 / PC80**



```
    0       T00      0005"00
    1       LDAA   S  040
    2       STAA   S  D00
    3       NOP
    4       ANDA   T  D00
    5       STAA   O  050
    6       NOP
    7       NOP
    8       JSR       INIT  00011
    9       JSR       TEST  00013
Command   F:
1S_LIST  2SEQCHART 3LAD      4FBK
```

Line number



```
    0       T00      0005"00
    3       LDAA   S  040
    6       STAA   S  D00
    9       NOP
   10       ANDA   T  D00
   13       STAA   O  050
   16       NOP
   17       NOP
   18       JSR       INIT  00026
   21       JSR       TEST  00028
Command   F:
1S_LIST  2SEQCHART 3LAD      4FBK
```

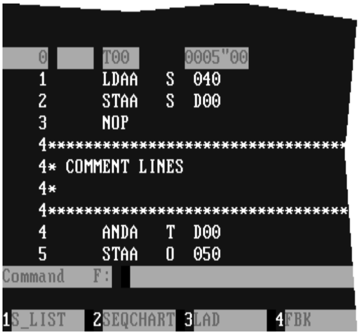
Program step number

4.3. COMMENT LINE

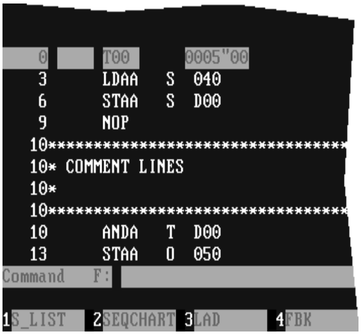
A	B	C	PG1000	CP80	PG-PC	PC80
		●	●	●	●	●

Comment lines supply the program documentation. Lines that are marked by the character “*” as comment lines, don't have an individual line number. These are assigned to the succeeding program line as a head line and, for this reason, have the same line number.

Example: **PG1000 / PG-PC** **CP80 / PC80**



```
    0        T00       0005"00
    1        LDAA   S  040
    2        STAA   S  D00
    3        NOP
    4**********************************
    4* COMMENT LINES
    4*
    4**********************************
    4        ANDA   T  D00
    5        STAA   O  050
Command   F:
1S_LIST  2SEQCHART 3LAD      4FBK
```



```
    0        T00       0005"00
    3        LDAA   S  040
    6        STAA   S  D00
    9        NOP
   10**********************************
   10* COMMENT LINES
   10*
   10**********************************
   10        ANDA   T  D00
   13        STAA   O  050
Command   F:
1S_LIST  2SEQCHART 3LAD      4FBK
```

4.4. LABEL

A	B	C	PG1000	CP80	PG-PC	PC80
		●	●	●	●	●

A label is a maximum four-digit letter-number combination that serves to distinguish certain program lines (e.g.: start of a subprogram). The first character must always be a letter. A label name can be given only once; but it can be called from the program at any time.

If a label is given, that already exists in the program, an error message is displayed:

```
E002  label already existing
```

In this case the cursor can not be moved out of the label field, until a different name is given.

4.5. COMMAND FIELD

A	B	C	PG1000	CP80	PG-PC	PC80
		●	●	●	●	●

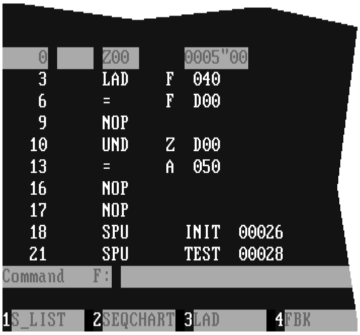
The assembler mnemonics for the processor commands are entered in this 4 character long field. Optionally B&R or MOTOROLA® mnemonics can be used.

Every input is checked for errors. If the wrong mnemonics are entered, an error message will be displayed and the command field can not be exited.

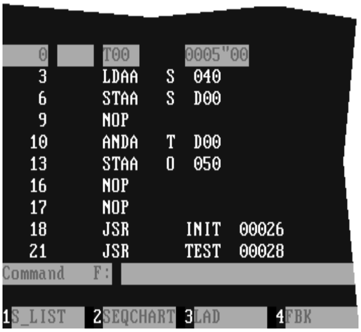
Switching the mnemonics:

instruction in the command line	function
`B&R`	input and display in B&R-mnemonics
`MOTOROLA`	input and display in MOTOROLA®-mnemonics
`MIXM`	input in B&R-mnemonics display in MOTOROLA®-mnemonics
`MIXB`	input in MOTOROLA®-mnemonics display in B&R-mnemonics

Example: **B&R-mnemonics** **MOTOROLA®-mnemonics**



```
    0          Z00       0005"00
    3          LAD    F  040
    6          =      F  D00
    9          NOP
   10          UND    Z  D00
   13          =      A  050
   16          NOP
   17          NOP
   18          SPU       INIT  00026
   21          SPU       TEST  00028
Command    F:
1S_LIST  2SEQCHART 3LAD      4FBK
```



```
    0          T00       0005"00
    3          LDAA   S  040
    6          STAA   S  D00
    9          NOP
   10          ANDA   T  D00
   13          STAA   O  050
   16          NOP
   17          NOP
   18          JSR       INIT  00026
   21          JSR       TEST  00028
Command    F:
1S_LIST  2SEQCHART 3LAD      4FBK
```

4.6. NEGATION AND EXTENDED ADDRESSING

4.6.1. Negation

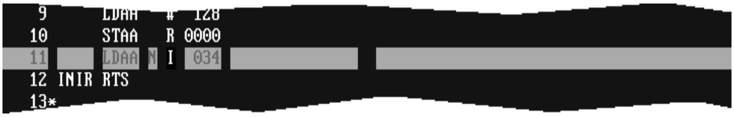
A	B	C	PG1000	CP80	PG-PC	PC80
		●	●	●	●	●

The negation can not be applied with all addresses. It is only possible for 1 bit addresses.

The following address preselections are defined:

I	digital input
O	digital output
F	flag (1-bit-memory)
S	start timer (1 bit signal to start a time)
T	timer (1 bit signal of the terminated time)

The negation occurs by pressing the key [N]; the cursor must be located in the **address preselection** field. By pressing the [N] key again, the negation is made inactive. If a certain address was negated, the respective field is marked by an **"N"** :



```
 9       LDAA   #  128
10       STAA   R 0000
11       LDAA N I  034
12 INIR RTS
13*
```

4.6.2. Extended addressing

A	B	C	PG1000	CP80	PG-PC	PC80
		●	●	●	●	●

This addressing method is only possible in connection with registers (R), index registers (X, Y) and stack pointers (U, !).

When using this addressing method, the address to which the command refers, is taken out of the memory location (2 byte) that was defined by the address preselection and the address.

Extended addressing results from pressing the key [[]; the cursor must be located in the **address preselection** field. A repeated pressing of the key [[] will end this addressing procedure. The extended addressing is displayed in the respective field by the **"["** character.

Example:	`LDAA    R 0100`	load accumulator A with the contents of R 0100.
	`LDAA [ R 0100`	load accumulator A with the contents of the memory location, as well as a 16 bit address located in R 0100 and R 0101.
	`+        X   010`	The index register X points to an address <ADDR>. The command effects that the memory location contents is added to the contents of the accumulator A<ADDR +10>.
	`+     [ X   010`	The index register X points to an address <ADDR>. The command effects, that the memory location contents along with the 16 bit address in <ADDR + 10> and <ADDR + 11> is added to the contents of the accumulator A.

4.7. ADDRESS PRESELECTION

The following address preselections are possible:

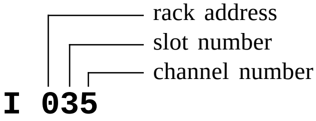
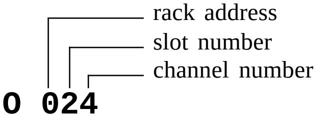
B&R	MOTOROLA®	description
E	I	digital input
A	O	digital output
C	R	register (8 bit memory location)
M	F	flag (1 bit memory location)
F	S	start timer (1 bit signal to start time)
Z	T	timer (1 bit signal of the terminated time)
P	P	peripheral address
G	G	global-RAM in the PP60 (available only in the PC80 mode) (extended dual port RAM area)
B	B	block memory in the PP60 (available only in the PC80 mode)
#	#	immediate
I	X	index register X
Y	Y	index register Y
U	U	user-stack pointer
!	!	system-stack pointer

Some statements exclude one or several address preselections. But if an inadmissible address preselection is entered, a respective error message is displayed.

```
I007   invalid address
```

4.8. ADDRESS

The address of the desired memory location is entered in the address field. This address field may be up to 17 characters long according to the entered address preselection. Advanced zeros must not be entered.

address preselection	**address range**	**notes**
I	I 000 - I 3EN I 000 - I 3FN	> with bus control > without bus control 
O	O 000 - O 3EN O 000 - O 3FN	> with bus control > without bus control 
R	R 0000 - R 7167	
	R 0800 - R 1499	This area is reserved for function blocks and system routines.
	R 0864 - R 0889	These registers are temporary storage for function blocks. The user can also apply these registers as **temporary (!)** storage.
F	F 000 - F 799	

address preselection	address range	notes
S,T		

hardware times	`S 000 - S 3E7`
	`S 000 - S 3F7`
	`T 000 - T 3E7`
	`T 000 - T 3F7`
		> with bus control
		> without bus control
		> with bus control
		> without bus control
		rack address
		slot number
		time number
		S 1E5
		T 1E5

software times	`S D00 - S D63`
	`T D00 - T D63`
		Time duration must be defined with individual commands.
		time number
		S D56
		T D56

P	`P 0xr`
	`P 000 - P 3FF`
		CPU programs: x ... slot
		r ... register
		PP-programs: the consequence of individual P-addresses is different according to the peripheral processor used.

I,Y,U,!	`I 000 - I 255`	PG1000 - and PG-PC-mode
	`I -128 - I 127`	CP80-mode

	`I -32768 - I 32767`
	`Y -32768 - Y 32767`
	`U -32768 - U 32767`
	`! -32768 - ! 32767`
		PC80-mode
		The command length in the PC80-mode depends on the size of the offset:

offset value	command length
-16 to +15	opcode + 1 byte
-128 to -17 +16 to +127	opcode + 2 byte
-32768 to -129 +128 to +32767	opcode + 3 byte

address preselection	address range	notes
B	B 0000 - B 4095	
G	G 0000 - G 4095	

address preselection	notes
#	The number format is defined by the first character of the address field.

number format	number range	
decimal “ ”		
1 byte-value	# 000	to # 255
2 byte-value	# 00000	to # 65535
hexadecimal “$”		
1 byte-value	# $00	to # $FF
2 byte-value	# $0000	to # $FFFF
binary “%”		
1 byte-value	# %00000000	to # %11111111
2 byte-value	# %0000000000000000	to # %1111111111111111
ASCII “'”		
1 byte-value	# 'x	to # 'x
2 byte-value	# 'xx	to # 'xx

4.9. SYMBOL FIELD

This field is an indicating field, which is omitted when terminating the address input; the cursor is immediately set to the comment field. To enter or change symbol assignments without selecting the SYMB menu the cursor can be set to the symbol field with [F5 SYMBINP] and a new or changed SYMB can be entered for the address.

A SYMB can have a length of up to 41 characters. According to the selected representation the entire SYMB or only the first 14 characters are displayed. With [F6 SYMBLEN] it can be switched between both possibilities.

The program print out is independent of this representation method!
The SYMB is always printed in full length!

4.10. COMMENT FIELD

Explanations for specific program lines can be entered in this field. The length of the comment is not limited by the screen width. If a text is entered that doesn't fit completely on the screen, it is moved to the left. Only a section of the text is visible. However, the comment is simultaneously displayed in it's entire length in the message line.

A comment is limited to a maximum of 64 characters.

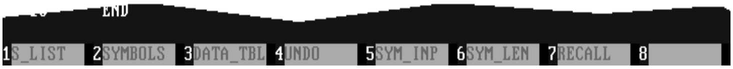
The display on the screen depends on the selected method of representing the symbols.

symbol	**comment**
max. 14 characters	A maximum of 39 characters is visible.
full length	If the SYMB is longer than 14 characters, the comment begins with a distance of two characters next to the SYMB, otherwise a maximum of 39 characters is displayed.

5. FUNCTIONS IN THE STL EDITOR

5.1. SOFTKEY-FUNCTIONS

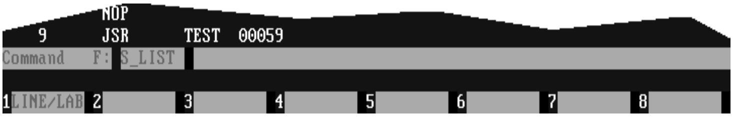
If the cursor is positioned in the input line of the STL editor, the function keys are allocated as follows:



```
        END
1S_LIST  2SYMBOLS 3DATA_TBL 4UNDO    5SYM_INP 6SYM_LEN 7RECALL  8
```

[F1 STL]

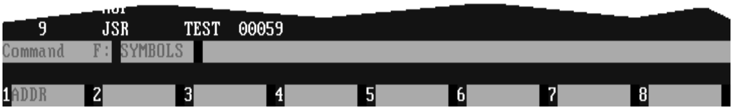
By pressing this function key the STL editor is exited and the cursor is set to the command line. The PROgramming SYStem requires the input of a line number or a label:



```
        NOP
  9     JSR      TEST  00059
Command  F: S_LIST
1LINE/LAB 2       3        4        5        6        7        8
```

[F2 SYMB]

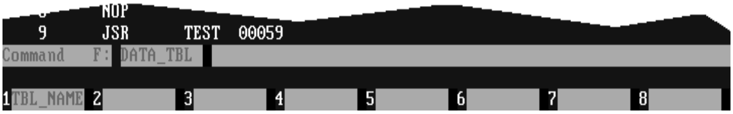
By pressing this function key the STL editor is exited and the cursor is set to the command line. The PROgramming SYStem requires the input of an address:



```
  9     JSR      TEST  00059
Command  F: SYMBOLS
1ADDR    2        3        4        5        6        7        8
```

[F3 TAB]

By pressing this function key the STL editor is exited and the cursor is set to the command line. The PROgramming SYStem requires the input of a table name:



```
        NOP
  9     JSR      TEST  00059
Command  F: DATA_TBL
1TBL_NAME 2       3        4        5        6        7        8
```

[F4 UNDO]

The STL editor is provided with a LIFO-storage (Last-In-First-Out), in which the last 10 deleted lines or characters respectively are stored. With the function [F4 UNDO] these are inserted in the text at the cursor position.

With this function e. g.: up to 10 lines can be cut out (deleted with [Ctrl] + [Del]) and inserted again at any position in the program.

[F5 SYMBINP]

In the STL editor, symbols can be assigned directly with this function (maximum length of 41 characters). The input or change is terminated with [↵].
This function is called from program lines, that are provided with the addresses I, O, R, B, G, F, P, S or T. If a wrong SYMB is entered in a line, the following error message is displayed:

```
E043  NO SYMBOLIC NAME IN THIS LINE
```

If a SYMB is changed the following question is displayed after pressing [↵]:

```
W004  CHANGE SYMBOLIC NAMES ?  ( Y / N )
```

The SYMB is changed or remains the same according to the needs of the user . The cursor is set to the next program line.

If a SYMB display is limited to 14 characters, the input of longer SYMB text is moved to the left and only the 14 section of text is displayed. The SYMB appears full length in the message line.

[F6 SYMBLEN]

According to the selected representation mode, the entire SYMB or only the first 14 characters are displayed on the screen. With [F6 SYMBLEN] it can not be switched to the active representation method.

[F7 RECALL]

If an existing program line is changed, the original line can be recalled with this function, if the changes are not terminated with [↵].

5.2. ADDITIONAL FUNCTIONS

[↑]	move STL one line down
[↓]	move STL one line up
[PgUp]	move STL one page down
[PgDn]	move STL one page up
[Ctrl] + [PgUp]	position cursor to the first program line (STL beginning)
[Ctrl] + [PgDn]	position cursor to the last program line (STL end)
[Ctrl] + [↑] or [↓]	move input line up/down
[Alt] + [↑] or [↓]	move input line to the top/bottom screen margin
[Ctrl] + [Esc]	move input line to the center of the screen
[Ctrl] + [Ins]	insert NOP-line and activate insert mode
[Alt] + [Ins]	insert comment line and activate insert mode
[Ctrl] + [Del]	delete line and disable insert mode
[Tab] or [⇑] + [Tab]	set cursor to the next right/left input field of the input line
[Ctrl] + [←] or [→]	position cursor to the comment field
[@]	

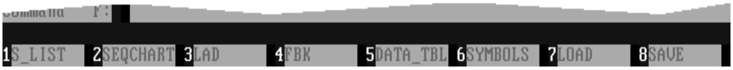
5.3. EXIT THE STL EDITOR

[Esc]	Exit the STL editor to the command line, in which different edit functions for the STL can be called.
[F2 SYMB] <ADR> [↵]	Exit the STL editor and call up the SYMB editor.
[F3 TAB] <TNAME> [↵]	Exit the STL editor and call up the TAB editor.

6. EDIT FUNCTIONS OF THE COMMAND LINE

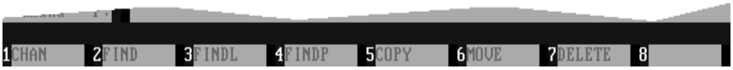
6.1. REPLACE / EXCHANGE (CHAN*)*

If the cursor is positioned in the command line, the softkey line looks like this:



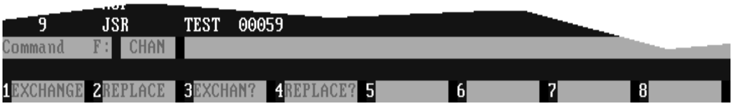
```
Command   F:
1S_LIST  2SEQCHART 3LAD      4FBK      5DATA_TBL 6SYMBOLS  7LOAD     8SAVE
```

By pressing the function key [F9] the allocation of the softkey's is changed:



```
1CHAN    2FIND     3FINDL    4FINDP    5COPY     6MOVE     7DELETE   8
```

By pressing **[F9] + [F1 CHAN]** or directly inputting ***CHAN***, different functions are offered in the softkey line:



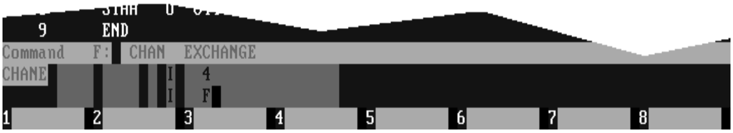
```
    9       JSR       TEST  00059
Command   F:  CHAN
1EXCHANGE 2REPLACE 3EXCHAN?  4REPLACE? 5         6         7         8
```

These functions can also be entered with the keyboard:

[F1 EXCHANGE]	or	***EXCHANGE***
[F2 REPLACE]	or	***REPLACE***
[F3 EXCHANGE?]	or	***EXCHANGE?***
[F4 REPLACE?]	or	***REPLACE?***

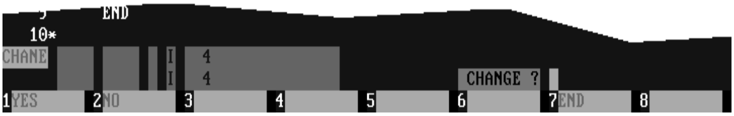
[F1 EXCHA] All program lines, within the user defined limits are searched, and compared with the two search masks, that are to be exchanged. If a search mask corresponds with the program line, it is replaced by the other mask.

This procedure is applied, for example, in the exchange of modules. The following masks must be defined for an exchange of the input card in slot 4 and the input card in slot F:



```
    9       END
Command   F:  CHAN   EXCHANGE
CHANE            I   4
                 I   F
1        2         3         4         5         6         7         8
```

[F3 EXCHA?] This function is the same as the one just described [F1 EXCHA] with a difference. With every exchange an inquiry is displayed in the message line:



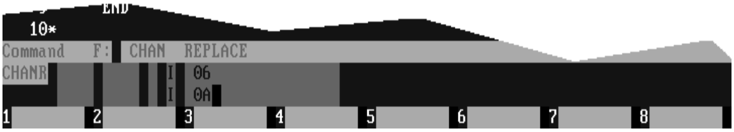
```
      END
  10*
CHANE        I   4
             I   4
                                        CHANGE ?
```

The user can now decide whether or not the exchange is to be executed:

[F1 YES], [↵] or **[Y]**	exchange
[F2 No] or **[N]**	the program line remains the same
[F7 END] or **[Esc]**	function interrupted

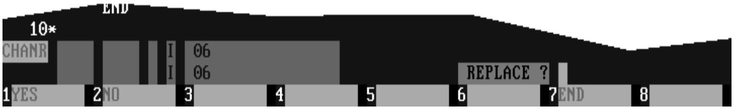
[F2 REPLA] All program lines, within the user defined limits are searched, and compared with the first search mask entered. If this corresponds with the program line, this line is then replaced by the second search mask.

This procedure is, for example, used to change the slot on modules. The following masks must be defined to change the input card in rack 0 from slot “6” to slot“ A”:



```
      END
  10*
Command   F:  CHAN  REPLACE
CHANR        I  06
             I  0A
```

[F4 REPLA?] This function is the same as the one described above [F2 REPLA] with the difference, that with every replacement an inquiry is displayed:



```
      END
  10*
CHANR        I  06
             I  06
                                        REPLACE ?
```

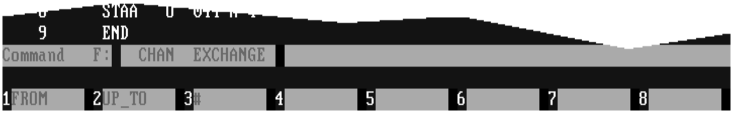
The user can now decide whether or not the replace function is executed:

[F1 YES], [↵] or **[Y]**	replace
[F2 NO] or **[N]**	the program line remains the same
[F7 END] or **[Esc]**	function interrupted

Define the search range for replace/exchange:

The softkey allocation changes after the input of one of the following:

CHAN EXCHA
CHAN REPLA
CHAN EXCHA?
CHAN REPLA?



```
          STAA    U
    9     END
Command   F:  CHAN  EXCHANGE
```

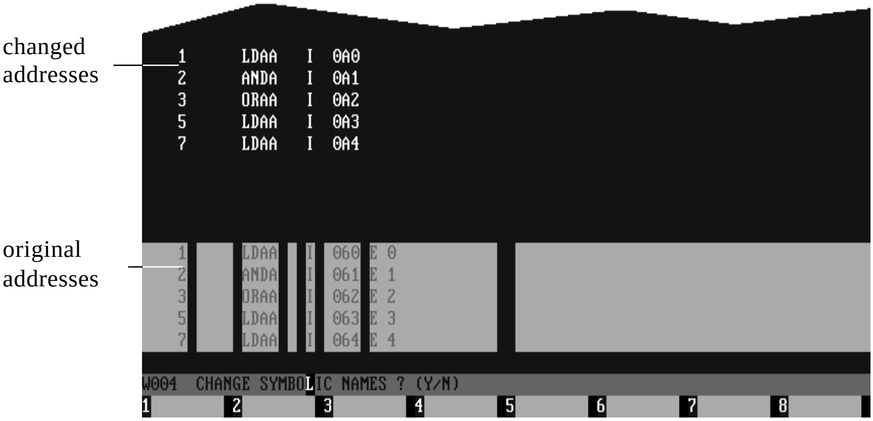
There are different ways of entering the search range:

Input	Search range
[↵]	global replacing; the entire STL is searched (except LADs).
[F1 FROM] <LN/LB> [F2 UP_TO] <LN/LB> [↵]	The STL is searched in the specified range.
[F2 UP_ TO] <LN/LB> [↵]	The STL is searched from the current input line to the specified program line <LN/LB>.
[F1 FROM] <LN/LB> [F3 #] <NR> [↵]	The STL is searched <NR> lines beginning with the specified line <LN/LB>.
[F3 #] <NR> [↵]	The STL is searched <NR> lines beginning with the current input line.
For <LN/LB> a line number or a label is to be entered.	

After the input of [↵] search masks are defined.

Change of symbols

After ending the function **CHAN**, the following question is displayed in the message line:



```
1      LDAA   I  0A0
2      ANDA   I  0A1
3      ORAA   I  0A2
5      LDAA   I  0A3
7      LDAA   I  0A4

1      LDAA   I  060 E 0
2      ANDA   I  061 E 1
3      ORAA   I  062 E 2
5      LDAA   I  063 E 3
7      LDAA   I  064 E 4

W004  CHANGE SYMBOLIC NAMES ? (Y/N)
1        2        3        4        5        6        7        8
```

The user can decide if the SYMBs are also changed.

[Y] With the function **exchange** all SYMBs, with a corresponding address to both entered masks, are exchanged. Even the SYMBs, with addresses that are not used in STL, are changed.

> With the function **replace** all SYMBs, with addresses that correspond with the search masks, are changed. If the change of a SYMB assigns an address twice, the SYMB of the original address is deleted.

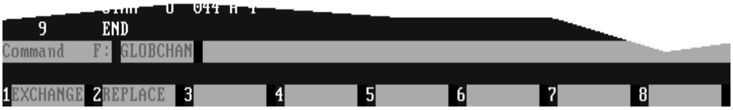
[N] The SYMBs remain the same.

> The function **CHAN** does not effect the STL of ladder diagrams.

6.2. GLOBAL EXCHANGE/REPLACEMENT (GLOBCHAN)

This function is valid for all addresses in the entire program including the LADs. But only addresses and not the address preselections can be changed (e.g.: registers can not be changed to flags).

After the input of ***GLOBCHAN*** in the command line the function key allocation is changed as follows:

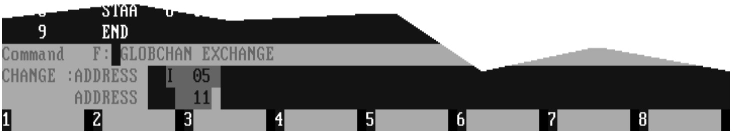


```
   9       END
Command   F: GLOBCHAN

1EXCHANGE 2REPLACE 3        4        5        6        7        8
```

[F1 EXCHA] All program lines are searched and compared with the two entered search masks to be exchanged. If the search mask corresponds with the program line, it is replaced by the other mask.

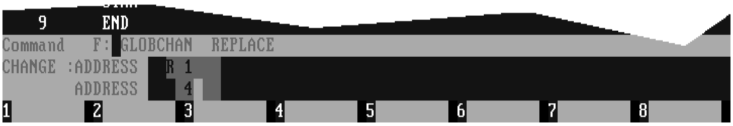
[F2 REPLA] All program lines are searched and compared with the first entered search mask. If it corresponds with the program line, the located line is replaced by the second search mask.

Example: The input card in rack 0 / slot 5 is exchanged with the input card in rack 1 / slot 1. In order to change the addresses in the program the following search masks are to be entered:



```
   9       END
Command   F: GLOBCHAN EXCHANGE
CHANGE :ADDRESS    I  05
        ADDRESS       11
1        2        3        4        5        6        7        8
```

To change all registers in the range R1000-R1999 to R4000-R4999 the following masks are to be entered:



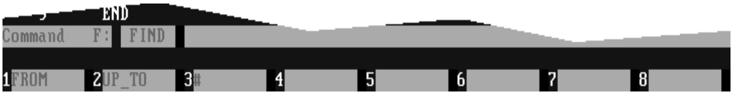
```
   9       END
Command   F: GLOBCHAN  REPLACE
CHANGE :ADDRESS    R 1
        ADDRESS      4
1        2        3        4        5        6        7        8
```

6.3. SEARCH (FIND)

With this function program lines can be searched in the entire program or in program parts just as needed (search mask).

The softkey allocation is changed after the input of:

[F 9] + [F2 FIND]

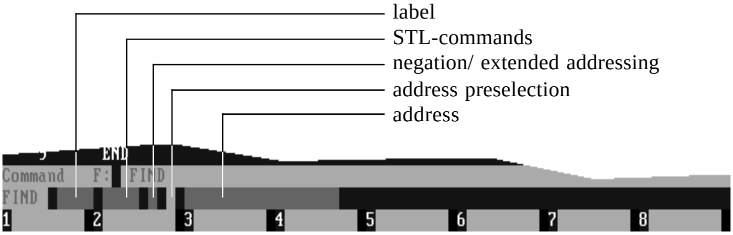


Search range

The definition of a search range occurs just as with the function **CHAN** (page 5-20).

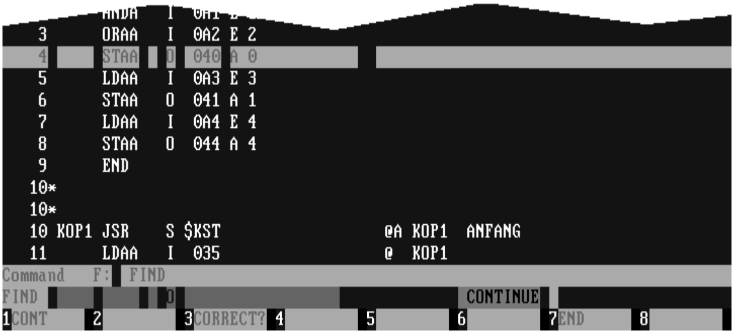
Search mask

After confirming the input with [↵] the user is asked to enter a search mask:



Spaces in the search masks are not noticed during the search process. When the input of the mask is terminated with [↵] the search is started.

The line number of the program line, that is compared with the search mask, is displayed in the message line. If a program line is located that corresponds with the search mask, the search procedure is interrupted. The softkey allocation changes and the following question appears in the message line:



```
          ANDA    I   0A1 E
   3      ORAA    I   0A2 E 2
   4      STAA    O   040 A 0
   5      LDAA    I   0A3 E 3
   6      STAA    O   041 A 1
   7      LDAA    I   0A4 E 4
   8      STAA    O   044 A 4
   9      END
  10*
  10*
  10 KOP1 JSR     S $KST                @A KOP1  ANFANG
  11      LDAA    I  035                @  KOP1
Command   F:  FIND
FIND              O                             CONTINUE
1CONT     2         3CORRECT? 4         5         6         7END      8
```

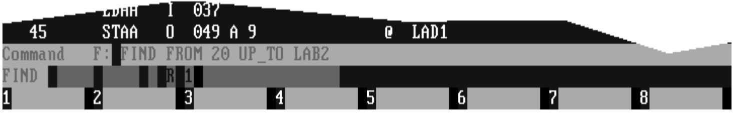
key	function
[F1 CONT] **[Y]**	The next line that is identical to the search mask is searched.
[F3 CORR]	The search procedure is interrupted and the cursor is set to the found line.
[F7 END] **[N]** **[Esc]**	The search procedure is interrupted and the cursor is set to the command line.

If no program line was found, that is identical to the search mask, the search procedure is ended and the cursor is set to the command line.

Example: In the range, line 200 to label LAB2, all registers R1000 - R1999 should be searched:

Command : **[F9] + [F2 FIND] [F1 FROM]** ***200*** **[F2 UP_ TO]** ***LAB2*** **[↵]**

Input of the search mask:



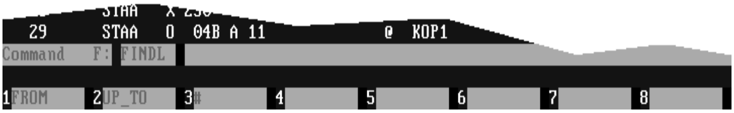
```
          LDAA    I   037
  45      STAA    O   049 A 9           @  LAD1
Command   F: FIND FROM 20 UP_TO LAB2
FIND              R 1
1         2         3         4         5         6         7         8
```

6.4. SEARCH AND LIST (FINDL) SEARCH AND PRINT (FINDP)

With these functions certain program lines, in the program or in program parts can be searched, as needed (search mask) and these can be listed/printed.

The function key allocation changes after the input of:

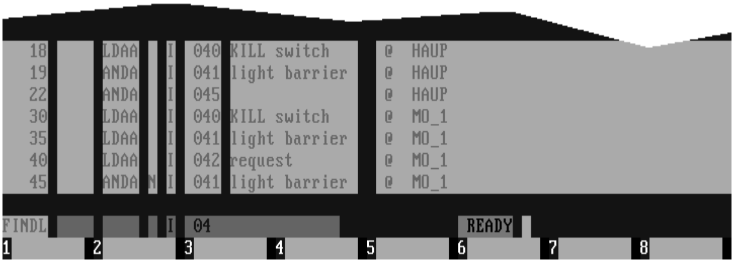
[F9] + [F3 FINDL] or **[F9] + [F4 FINDP]**



```
          STAA    X 230
  29      STAA    O  04B A 11            @  KOP1
Command    F: FINDL
1FROM     2UP_TO    3#        4         5         6         7         8
```

SEARCH MASK and **SEARCH RANGE** are defined like S**earch [F9] + [F2 FIND]** function.

Example: list all inputs applied to slot 4 (I 040 to I 04E):



```
  18      LDAA   I  040 KILL switch      @  HAUP
  19      ANDA   I  041 light barrier    @  HAUP
  22      ANDA   I  045                  @  HAUP
  30      LDAA   I  040 KILL switch      @  MO_1
  35      LDAA   I  041 light barrier    @  MO_1
  40      LDAA   I  042 request          @  MO_1
  45      ANDA N I  041 light barrier    @  MO_1

FINDL            I  04                          READY
1         2         3         4         5         6         7         8
```

Example: print out of all used input cards:

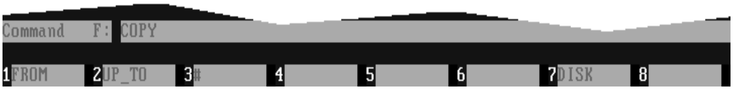
```
  237 BETS LDAA    I  0A4
  338      LDAA    I  0A3
  366 GR01 LDAA  N I  0B1
  369      ANDA  N I  0B2
  372      ANDA    I  0B8
  387      LDAA    I  0A3
  410      ANDA  N I  0A3
  472      LRK          I  050
  524      LRK     I  060
  560 HAAU LDAA    I  0A2
  569      ANDA  N I  0A1
  572      ANDA    I  0A0
  593 HAN1 LDAA    I  0A5
  599      LDAA    I  0A6
  605      LDAA    I  0B6
  611      LDAA    I  0B7
  617      LDAA    I  0B0
  620      ANDA  N I  0B2
  626      LDAA    I  0A9
  690 SCH1 LDAA  N I  0C1                                        release of
  918      LDAA    I  0B3
  924 KLEM LDAA    I  0B1
  927      ANDA    I  0C0
  951      LDAA  N I  0B9
  969      ANDA    I  0BA
 1018 TRAN LDAA    I  0C0
 1021      ANDA    I  0B1
 1036      LDAA  N I  0C0
 1039      ANDA  N I  0B1
 1057      LDAA    I  0B8
 1060      ANDA    I  0B4
 1090      ANDA    I  0B2
 1114      ANDA    I  0B8
 1117      ANDA  N I  0B2
```

6.5. COPY

A defined program range can be copied and moved to another position in the program with this function.

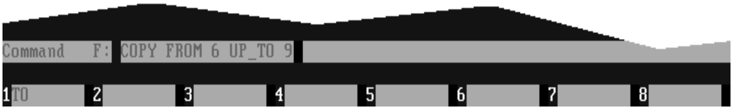
The function key allocation changes after the input of:

[F9] + [F5 COPY]



```
Command   F: COPY
```

The range that should be copied, is entered like the search range for ***CHAN*** (see page 5-20). After the definition of the desired range the function key allocations are changed:

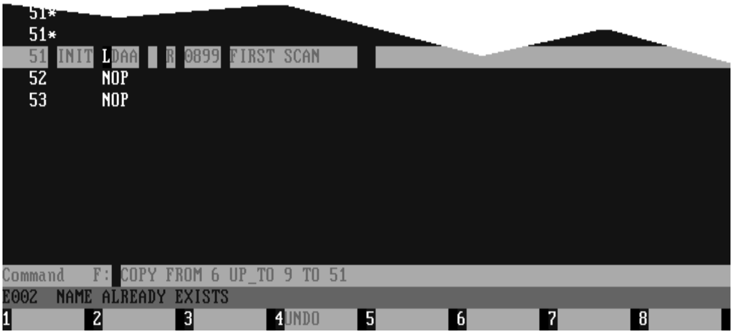


```
Command   F: COPY FROM 6 UP_TO 9

1TO   2   3   4   5   6   7   8
```

With **[F1 TO] <LN/LB>** or ***TO*** **<LN/LB>** the destination address is specified. After confirming with [↵] the entered block is copied.

Without the definition of a destination address and confirming with [↵] the specified range is copied to the cursor position.

If a program part is copied, in which labels are present, all lines are copied up to the first doubled label. An error message is then displayed and the cursor is positioned to the program line where the first double label exists:



```
   51*
   51*
   51 INIT LDAA   R 0899 FIRST SCAN
   52      NOP
   53      NOP

Command   F: COPY FROM 6 UP_TO 9 TO 51
E002  NAME ALREADY EXISTS
1     2     3     4UNDO 5     6     7     8
```

This program line can be optionally edited. If another label was distributed, the copying process can be continued with [↵].

The copying process can be interrupted with [Esc].

Program parts that include protected lines (FBK, SEQC, LAD), can not be copied. An error message is displayed:

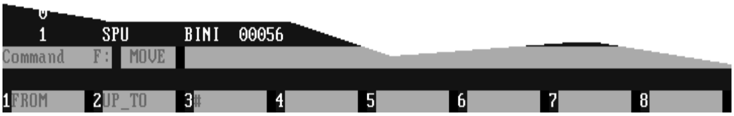
```
E127  DELETE NOT POSSIBLE
```

6.6. MOVE

A specified program range can be moved to another position in the program with this function.

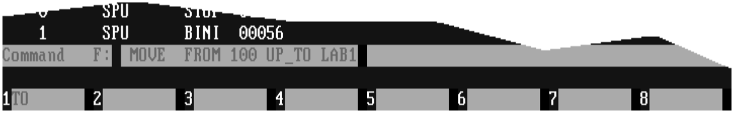
The function key allocation changes after the input of:

[F9] + [F5 MOVE]



```
  1       SPU       BINI  00056
Command   F: MOVE
1FROM  2UP_TO  3#  4  5  6  7  8
```

The range that needs to be moved, is defined like a search range with ***CHAN*** (see page 5-20). After the definition of the desired range the function key allocations are changed:



```
  1       SPU       BINI  00056
Command   F: MOVE  FROM 100 UP_TO LAB1
1TO  2  3  4  5  6  7  8
```

The destination address is entered with **[F1 TO] <LN/LB>** or ***TO*** **<LN/LB>**. The specified block is moved after pressing [↵].

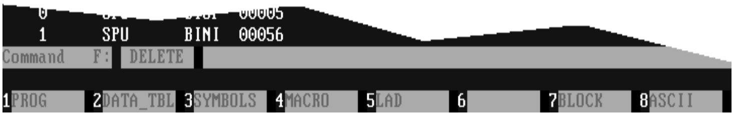
Without the definition of a destination address and confirmation with [↵] the specified range is copied to the current cursor position

6.7. DELETE BLOCK

A defined program range can be deleted from the program with this function.

The function key allocation changes after the input of:

[F9] + [F7 DELETE]



```
  1       SPU       BINI  00056
Command   F: DELETE
1PROG  2DATA_TBL  3SYMBOLS  4MACRO  5LAD  6  7BLOCK  8ASCII
```

Only [F7 BLOCK] refers to the presently active editor; the other functions files (programs, tables, SYMBs, macros, LADs, ASCII-files) can be deleted from the active disk drive. These functions are described in chapter 8 “load, save, ...”.

The function key allocations are changed after the input of [F7 BLOCK]. A program range, to be deleted, must be defined. This range is defined like the search range of the function ***CHAN***.

After confirming the input with [↵] the defined range is deleted.

7. STL STATUS TEST

7.1. GENERAL INFORMATION

The STL status test is an efficient tool for error detection and debugging before program start. STL programs can be executed in single steps, set in break points or used in trace functions.

Registers, flags, inputs etc. can be displayed in freely selectable number formats and be changed for testing purposes. The user gets an exact image of the control and software operating status.

7.2. ACCESSING THE STL STATUS TEST

7.2.1. RUN

Command: ***RUN* [↵]**

Function:

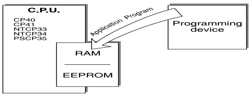
1) The program is checked for completion of the label table (compared to ***RNT***-command). If the label table is not complete or if the maximum branch width is exceeded, the program is not transferred to the PLC and an error message is displayed:

```
E051  INVALID BRANCH
```

2) The program is transferred to the RAM of the PLC program and started no matter if a program is stored in EEPROM (CP40, CP41, NTCP33, NTCP34, PSCP35) or PROM (CP60, CP80, NTCP63, NTCP64, PSCP65).

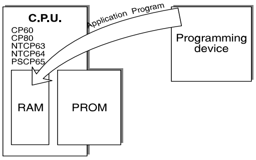
3) The STL status test is called up.

Transfer an application program to the central processing unit



CP40, CP41, NTCP33, NTCP34, PSCP35:

When transferring an application program from the programming device to the CPU, it is stored in the RAM of the EE32 and started no matter if another program is stored in the EEPROM of the EE32.



CP60, CP80, NTCP63, NTCP64, PSCP65:

When transferring an application program from the programming device to the CPU, it is stored in the internal RAM of the CPU and started no matter if an PROM memory exists.

7.2.2. STAT

Command: ***STAT* [↵]**

Function: 1) The STL status test menu is called up. The program is not transferred in the PLC-CPU.

2) If the program was changed after the last status test, this is considered by the PROgramming SYStem. The following warning is displayed in the message line:

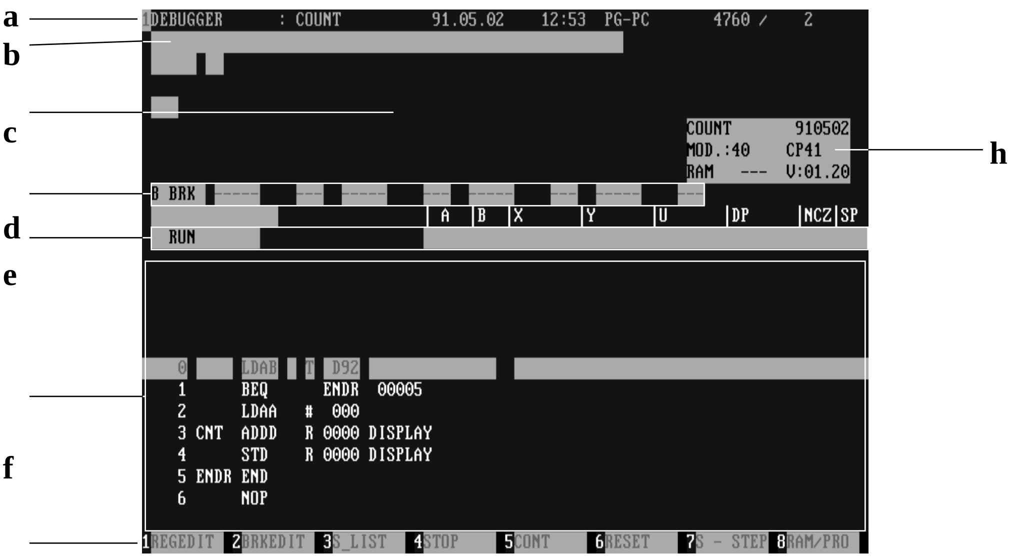
```
W233  MAKE PROGRAM INFO  ( Y / N ) ?
```

[Y] program in the PLC and PG are compared.
[N] process is interrupted.

If [N] is entered or the program in the PG is not identical with the one in the PLC , an error message is displayed:

```
E179  PROGRAM IN CPU AND PG DIFFERENT !
```

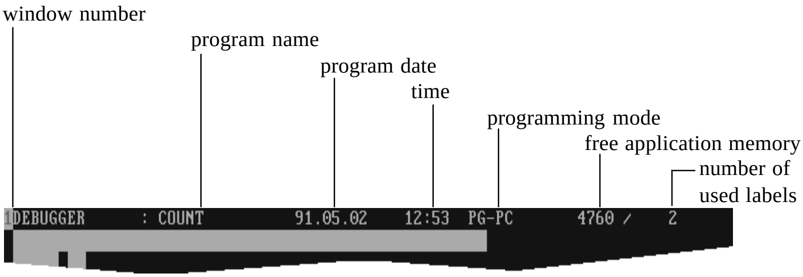
7.3. MENU STRUCTURE



```
1DEBUGGER      : COUNT          91.05.02     12:53  PG-PC       4760 /     2

                                                       COUNT       910502
                                                       MOD.:40     CP41
                                                       RAM    ---  V:01.20
B BRK
                         A    B    X        Y        U        DP       NCZ SP
 RUN

    0     LDAB   T  D92
    1     BEQ       ENDR  00005
    2     LDAA   #  000
    3 CNT ADDD   R 0000 DISPLAY
    4     STD    R 0000 DISPLAY
    5 ENDR END
    6     NOP
1REGEDIT 2BRKEDIT 3S_LIST 4STOP 5CONT 6RESET 7S - STEP 8RAM/PRO
```

a) Status line

The status line contains different information, such as:



```
1DEBUGGER      : COUNT          91.05.02     12:53  PG-PC       4760 /     2
```

b) Symbol field

In this field, symbol assignments of the address are displayed, that are defined in the status field, in which the cursor is positioned.

c) Status field

In this screen area up to 14 status fields can be applied.

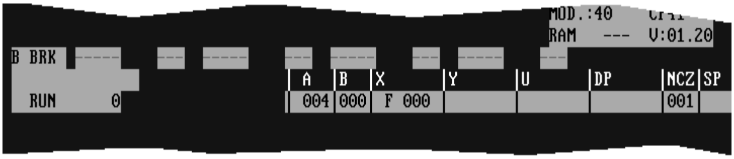
d) Break point line

In this line up to 5 breakpoints can be entered.

e) Processor status line

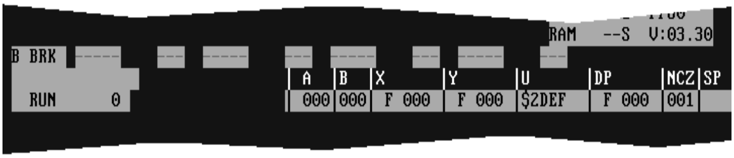
The current status of the processor and the contents of the program registers are displayed. The processor status line, to which the cursor is set, has a different appearance in the PC80-mode.

Processor status line in the PG1000-, CP80- or PG-PC-mode:



```
                                                             MOD.:40
                                                             RAM   ---  V:01.20
B BRK -----    ---  -----    ---  -----    ---  -----    ---
                             | A |B  |X      |Y      |U      |DP     |NCZ|SP
  RUN       0                | 004|000| F 000 |       |       |       |001|
```

Processor status line in the PC80-mode:



```
                                                             RAM   --S  V:03.30
B BRK -----    ---  -----    ---  -----    ---  -----    ---
                             | A |B  |X      |Y      |U      |DP     |NCZ|SP
  RUN       0                | 000|000| F 000 | F 000 |$2DEF  | F 000 |001|
```

A	accumulator A (AA)
B	accumulator B (AB)
X	index register X
Y	index register Y
U	user stack pointer
DP	direct page register
NCZ	flags: N ... negative flag C ... carry flag Z ... zero flag
!	system stack pointer

f) STL field

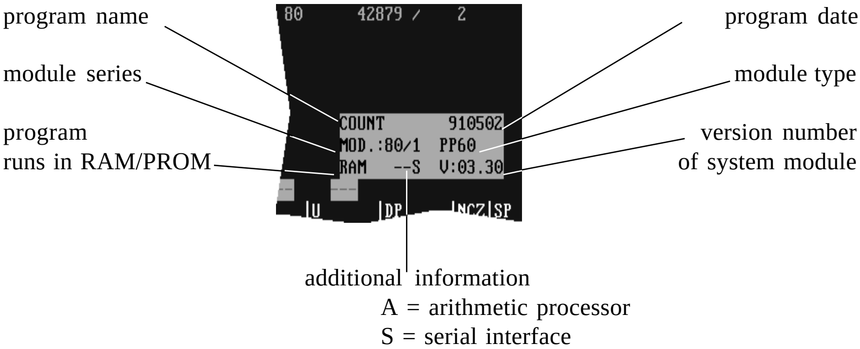
The processor status and the program register can be displayed before or after the processing of a line.

g) Softkey line

In the softkey line the most important commands of the STL status test are offered.

h) Module information

In this field the following information is displayed:



```
80      42879 /    2

COUNT        910502
MOD.:80/1  PP60
RAM   --S  V:03.30
--   ---
 |U      |DP      |NCZ|SP
```

program name

module series

program
runs in RAM/PROM

program date

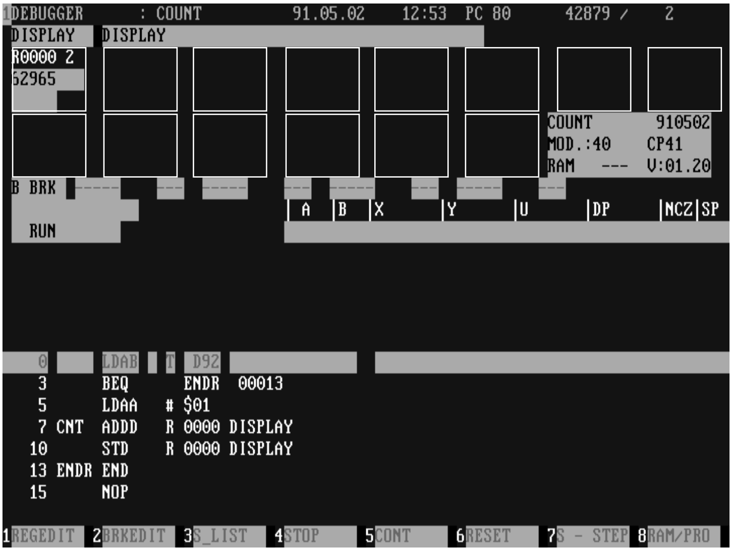
module type

version number
of system module

additional information
A = arithmetic processor
S = serial interface

7.4. STATUS FIELDS

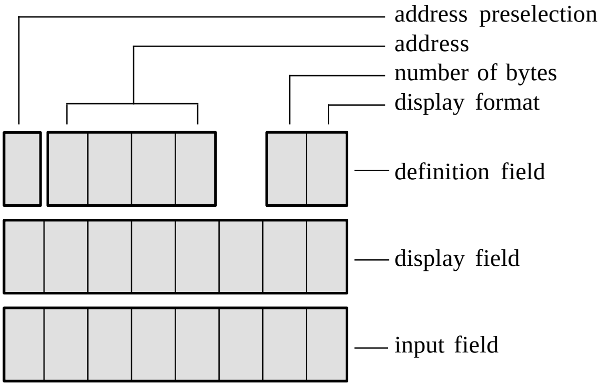
It is possible to apply up to 14 status fields, that can be contained in the following positions on the screen.



```
1DEBUGGER      : COUNT          91.05.02    12:53  PC 80      42879 /    2
DISPLAY  DISPLAY
R0000 2
62965
                                                         COUNT       910502
                                                         MOD.:40     CP41
                                                         RAM   ---   V:01.20
B BRK  -----    ---  -----    ---  -----    ---  -----    ---
                              | A |B  |X       |Y       |U       |DP      |NCZ|SP
  RUN



    0       LDAB   T  D92
    3       BEQ       ENDR  00013
    5       LDAA   # $01
    7 CNT   ADDD   R 0000 DISPLAY
   10       STD    R 0000 DISPLAY
   13 ENDR  END
   15       NOP

1REGEDIT 2BRKEDIT 3S_LIST 4STOP 5CONT 6RESET 7S - STEP 8RAM/PRO
```

If a status field requires more space because of the selected display format, the next field is used.

7.4.1. Status Field Structure



Address preselection

The possibilities for address preselection are:

I, O, F, R, P, S, T, B, G

Address range

The possible address range for a certain address preselection, is given in section “4.8. Address”.

Number of bytes

The maximum number of bytes depends on the defined address preselection and the selected display format.

address preselection	**number of bytes according to the display format**								
	$	%	'	^	I	A	F	P	no characters
for 1 bit memory location (I, O, F, S, T)	1 2-4[1]	1	x 1-8[1]	1-8	x	x	x	x	1 2-4[1]
for 8 bit memory location (R, P, B, G)	1-4	1	1-8	1-8	1-4	1-4	4[2]	2[2]	1-4

[1] These values are possible for the address preselection, but it is not very useful to apply them for 1 bit memory locations.

[2] These values are automatically set, if the respective number format is set. This display format is only relevant in connection with 8-bit memory locations.

x This number format is not efficient for the address preselection.

Display Format

In this field the number format is shown, in which the contents of the defined memory location should be displayed. 9 different formats can be selected.

If the number format in an existing status field is changed, the conversion to the new format occurs automatically. If necessary, the number of bytes is changed.

format	characters
decimal	no character
hexadecimal	$
binary	%
ASCII	'
low value bit	^
integer	I
absolute with sign	A
floating point	F
pointer	P

Decimal format

Character: **no** character
If no characters are entered in the *display format* field or the existing characters are deleted, the decimal format is selected.

Number of bytes: 1 to 4

Restrictions: 2, 3 or 4 byte values are only useful with 8 bit memory locations.

Representable number range:

number of bytes	number range	
1	0 to 255	(2^8-1)
2	0 to 65535	(2^{16}-1)
3	0 to 16777215	(2^{24}-1)
4	0 to 4294967295	(2^{32}-1)

Hexadecimal format

Character: $

Number of bytes: 1 to 4

Restrictions: 2, 3 or 4 byte numbers only useful with 8 bit memory locations.

Representable number range:

number of bytes	number range
1	0 to FF
2	0 to FFFF
3	0 to FFFFFF
4	0 to FFFFFFFF

Binary Format

Character: %

Number of bytes: 1

Representable number range: 00000000 to 11111111

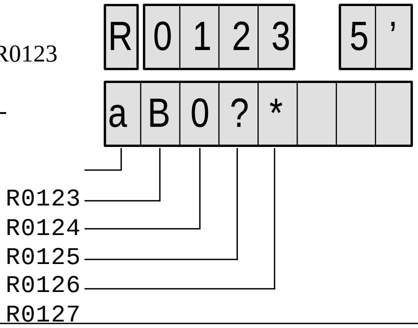
ASCII Format

Character: '

Number of bytes: 1 to 8
1-8 bytes are represented as ASCII-characters. In the display field they appear in the left, where the left character is respective of the contents of the first memory location.

Restrictions: This display format is only useful when applied with 8 bit memories.

Example: The contents of the memory locations R0123 to R0127 should be displayed in ASCII-format:



Bitwise representation

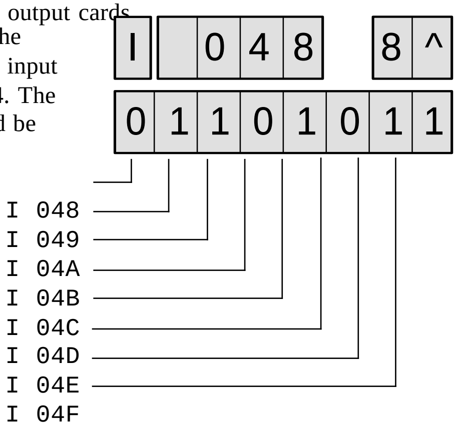
Character: **^**

Number of bytes: 1 to 8

Restrictions: Only the bit with the lowest value is taken into account in the memory location.

This representation is favourable for the control or change of the status of digital input and output cards.

Example: Representation of the states of the digital input cards rack 0 / slot 4. The inputs 8 to F should be displayed:

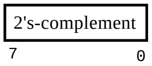
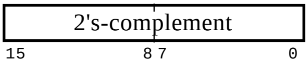
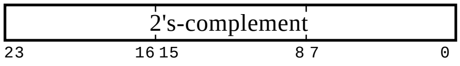
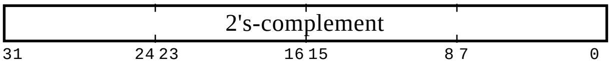


Integer Format

Character: **I**

Number of bytes: 1 to 4

Representable number range:

1 byte integer		-128 (-2^7) to 127 (2^7-1)
2 byte integer (= integer short)		-32768 (-2^{15}) to 32767 (2^{15}-1)
3 byte integer		-8388608 (-2^{23}) to 8388607 (2^{23}-1)
4 byte integer (= integer long)		-2147483648 (-2^{31}) to 2147483647 (2^{31}-1)

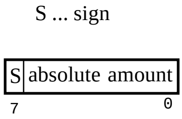
Absolute Format

Character: **A**

Number of bytes: 1 to 4

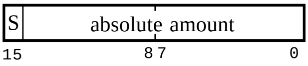
Representable number range:

1 byte absolute



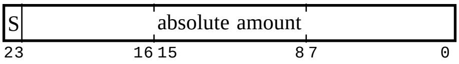
±(0 to 127)
±(2^7-1)

2 byte absolute
(= absolute short)



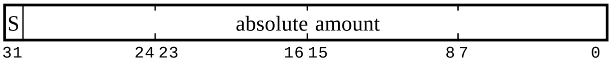
±(0 to 32767)
±(2^{15}-1)

3 byte absolute



±(0 to 8388607)
±(2^{23}-1)

4 byte absolute
(=absolute long)



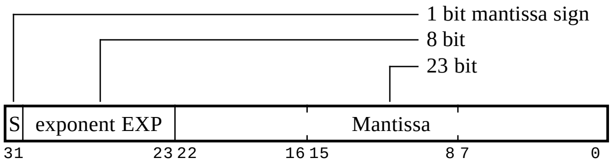
±(0 to 2147483647)
±(2^{31}-1)

Floating Point Format

Character: **F**

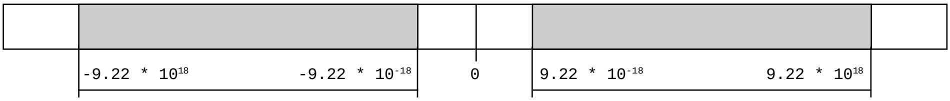
Number of bytes: 4

Format: IEEE-Standard (Single Precision Floating Point Format)



Conversion: $(-1)^S \bullet 2^{(EXP-127)} \bullet 1.$ mantissa

Representable number range:



Numbers from `-9.22 * 10`$^{-18}$ to `+9.22 * 10`$^{-18}$ except `0` can not be displayed, and are processed like `0`.

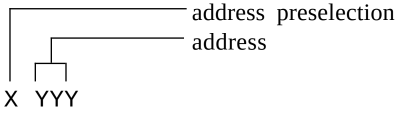
Pointer Format

Character: **P**

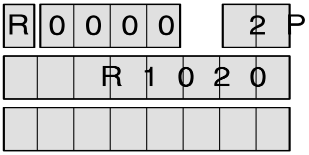
Number of bytes : 2

Restrictions: The application of this format is only useful for the contents of 8 bit memory locations.

Representation: The contents of two 8 bit memory locations is interpreted as 16 bit address with this representation method displayed in the B&R-format:
x yyy



Example: The contents of R0000 and R0001 should be displayed as a pointer:



```
R 0000   2 P
   R1020
```

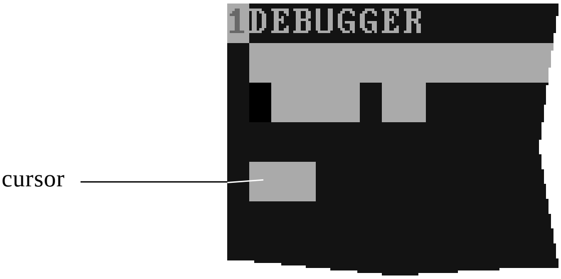
Valid for CP80 is e.g.:
```
$0000 => “F 000”
$3000 => “R0000”
```

With addresses, that don't have an B&R-equivalent, a hexadecimal value is displayed. E.g.: `$5000 => “$5000”`

A complete overview of B&R-addresses is given in the HARDWARE-MANUAL MULTI-CONTROL / MIDICONTROL / M264 in chapter 4 central processing units.

7.4.2. Edit Status Fields

After the first call of the status test, the cursor is in the field “address preselection” of the first status field. The status field, in which the input cursor is positioned, is displayed inverse:



[←] or **[→]** The cursor can be moved freely to the left and to the right.

If the cursor is moved left/right out of the status field, it is set in the next status field.

[↓] or **[↑]** The address of the memory location of the active status field is decreased or increased respectively.

[Ctrl] + [❖] The cursor is set to the field “address preselection” of the status field selected with [❖].

[Alt] + [❖] The cursor is set to the input field of the status field selected with [❖].

[Ctrl] + [Del] The active status field is deleted and the cursor is set to the left status field.

[↵] If the cursor is located in the definition field, the definition is ended and the cursor is set to the right status field.

If the cursor is in the input field, the input is terminated and the entered values are transferred to the control.
When transferring multiple byte values, without stopping the PC it asks, if the data should be transferred asynchronously:

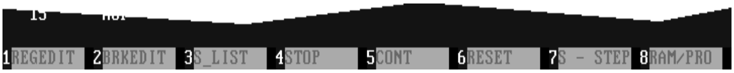
```
W235  TRANSFER ASYNCHRON !!!  CONTINUE ( Y / N ) ?
```

On newer system module versions (CP40: from V. 3.20; CP80: from V. 3.10) the data transfer is executed in the END-routine - the transfer occurs in this case without inquiry (synchronous)!

7.5. FUNCTIONS IN THE STL STATUS TEST

A	B	C	PG1000	CP80	PG-PC	PC80
		●	●	●	●	●

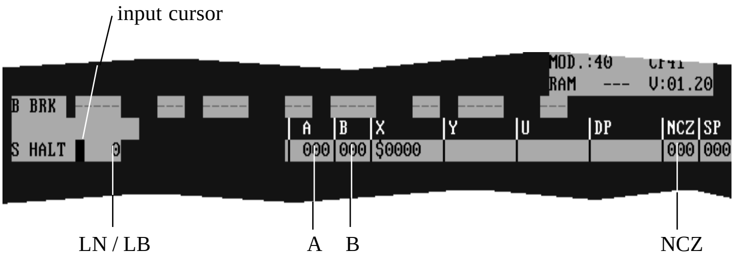
After starting the status test with ***RUN* [↵]** or ***STAT* [↵]** it is displayed on the screen of the PC. The following functions are offered in the softkey line:



7.5.1. [F1 REGEDIT] - Editing program register

This function is possible in the **HALT** state of the processor! After pressing [F1 REGEDIT] the cursor is positioned in the processor status line. The following registers and flags can be changed:

LN/LB	line number / label
A	accumulator A (AA)
B	accumulator B (AB)
NCZ	flags: N ... negative-flag
	C ... carry-flag
	Z ... zero-flag



```
                                              MOD.:40   CP41
                                              RAM   ---  V:01.20
B BRK  -----  ---  -----  ---  -----  ---  -----  ---
                       |A  |B  |X      |Y     |U     |DP    |NCZ|SP
S HALT    0            |000|000|$0000  |      |      |      |000|000
```

[→] or [←] The cursor can be moved in the fields LN/LB, A, B and NCZ.

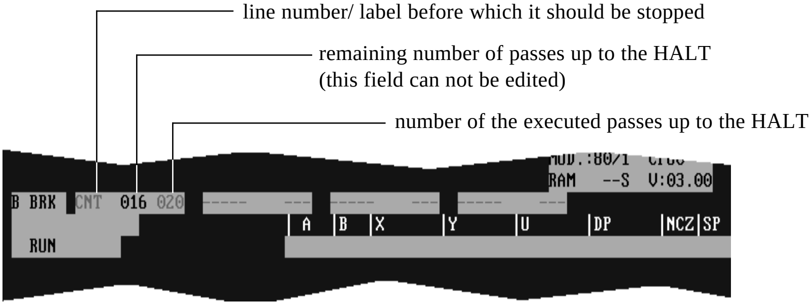
[Esc] Ending the edit procedure of the program register. The values entered by the user are saved.

7.5.2. [F2 BRKEDIT] - Breakpoint Definition

Breakpoints are used for error detection. Breakpoints are software selected program lines, at which the program is stopped after a defined number of passes. The PLC is stopped. This HALT state is displayed in the processor status line.

It is possible to define up to four breakpoints.

By pressing the function key [F2 BRKEDIT], the cursor is set to the break point line:



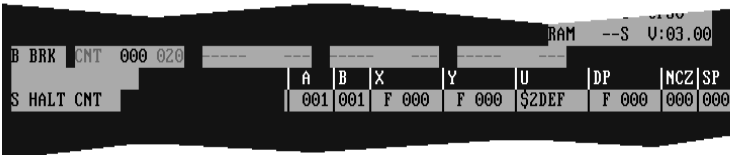
line number/ label before which it should be stopped

remaining number of passes up to the HALT
(this field can not be edited)

number of the executed passes up to the HALT

```
                                                            MOD.:80/1  CPU0
                                                            RAM   --S  V:03.00
B BRK  CNT  016 020  -----    ---  -----    ---  -----    ---
                                 | A  |B  |X     |Y     |U     |DP    |NCZ|SP
  RUN
```

If the program is stopped on a defined breakpoint, this is displayed in the processor status by the word “HALT”. In the STL field the line, in which the program was stopped, is displayed inverse:

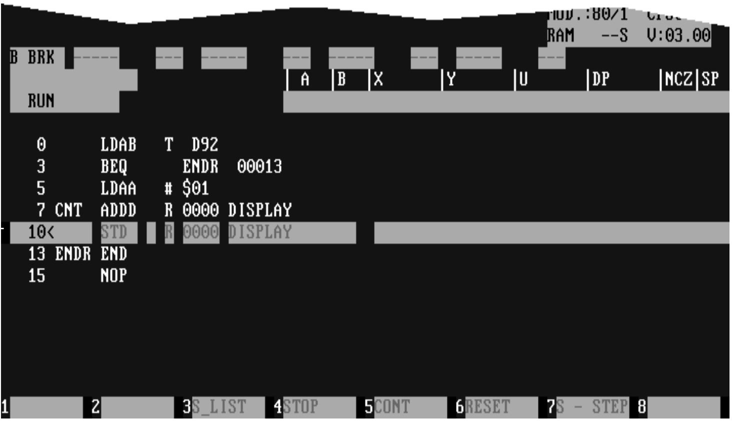


```
                                                            RAM   --S  V:03.00
B BRK  CNT  000 020  -----    ---  -----    ---  -----    ---
                                 | A  |B  |X     |Y     |U     |DP    |NCZ|SP
S HALT CNT                       | 001|001| F 000| F 000|$2DEF | F 000|000|000
```

7.5.3. [F3 STL] STL Mode

A particular program line of the STL can be watched with this function.

By pressing [F3 STL] the cursor is placed in the STL field:

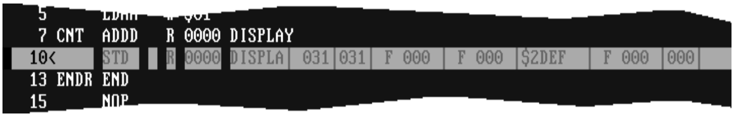


```
                                                 MOD.:80/1
                                                 RAM    --S  V:03.00
B BRK  -----   ---  -----   ---  -----   ---  -----   ---
                               | A  |B  |X      |Y      |U      |DP     |NCZ|SP
  RUN

   0      LDAB   T  D92
   3      BEQ      ENDR  00013
   5      LDAA   # $01
   7 CNT  ADDD   R 0000 DISPLAY
  10<     STD    R 0000 DISPLAY
  13 ENDR END
  15      NOP

1        2        3S_LIST  4STOP    5CONT    6RESET   7S - STEP 8
```

The STL can be moved up / down with the cursor keys [↑] / [↓].

With the input of **<LN/LB> [↵]** the cursor is positioned in the desired line.

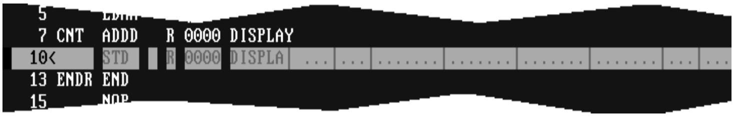
By pressing again [F3 STL] on the right side of the selected program line the program registers are displayed (AA, AB, X, Y[1], U[1], DP[1], NCZ, SZ instead of SYMB and comment). The line is marked by the character “<” on the left, i.e.: the contents of the program register is displayed **before** the execution of the selected program line:



```
   5      LDAA   # $01
   7 CNT  ADDD   R 0000 DISPLAY
  10<     STD    R 0000 DISPLA| 031|031| F 000 | F 000 |$2DEF  | F 000 |000|
  13 ENDR END
  15      NOP
```

The function [⇑] + [F3 STL] displays the states of the individual program register **after** the execution of the selected program lines. The line is marked on the left by the character “>”.

If the line, selected with the cursor, is not executed, periods are displayed instead of program register contents:



```
   5      LDAA
   7 CNT  ADDD   R 0000 DISPLAY
  10<     STD    R 0000 DISPLA| ... | ... | ....... | ....... | ....... | ....... | ... | ...
  13 ENDR END
  15      NOP
```

[1] Only in the PC80-mode

7.5.4. [F4 STOP]

The program operation in the control is interrupted with this function. This is displayed in the processor status line with “HALT”.

The status of the control remains the same, i.e. output states are not put back but remain in the current state.

ATTENTION: Control outputs for motors remain unchanged.

7.5.5. [F5 CONT]

Pressing this key continues the interrupted program. If a certain time was given by [F3 STL], the program is continued at this position.

7.5.6. [F6 RESET]

The control is stopped and set to a **defined** base status:

- all outputs are set to logic “0”
- R0000 to R0019 are deleted
- F 000 to F 499 are deleted
- all program registers are deleted

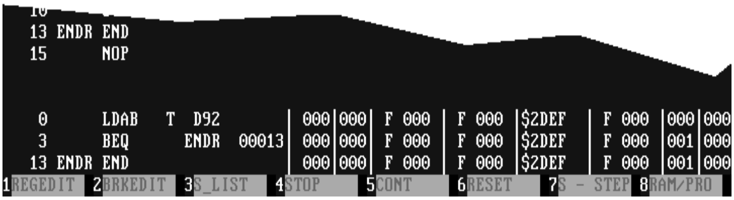
This function is also available beyond the status test. It depends on the PC, which keys are occupied with this function (see chapter 3 INTRODUCTION).

7.5.7. [F7 S-STEP] - Single Step Operation

This function is only possible in the HALT state of the processor.

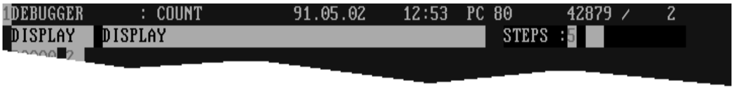
With the single step operation program assignments can be individually executed (by pressing a key). This operation is used for error detection and for error limitation.

The just processed program line is displayed with the program register:



```
 13 ENDR END
 15      NOP


  0      LDAB   T  D92         | 000|000| F 000 | F 000 |$2DEF  | F 000 |000|000
  3      BEQ       ENDR  00013| 000|000| F 000 | F 000 |$2DEF  | F 000 |001|000
 13 ENDR END                   | 000|000| F 000 | F 000 |$2DEF  | F 000 |001|000
1REGEDIT 2BRKEDIT 3S_LIST  4STOP    5CONT    6RESET   7S - STEP 8RAM/PRO
```

With the function [⇑] + [F7 S-STEP] the number of the single step operations, which need to be executed, can be entered. On screen an input request appears on the right top:



```
1DEBUGGER      : COUNT          91.05.02    12:53  PC 80      42879 /    2
 DISPLAY   DISPLAY                                   STEPS :5
```

With the key [↵] the input is terminated.

With the function [Alt] + [F7 S-STEP] the program of a subprogram jump is stopped after the jump in the subprogram.

7.5.8. [F8 RAM/PROM]

Switch from RAM to EEPROM operation and reversed. According to the choice, the program in the PLC is operated in the RAM or in the EEPROM.

8. ADDITIONAL FUNCTIONS

8.1. *RNT*

Command: ***RNT* [↵]**

Function: The program in the current window is checked for missing labels and for exceeding of the maximum branch width.

Result: If a branch instruction exists, that refers to an absent label or to an exceeding of the maximum branch width, the input cursor is set to the respective line and the following error message is displayed:

```
E051  INVALID BRANCH
```

If the label in the program was properly utilized, the test is ended with the following message:

```
W023  RNT OK
```

> The **RNT**-checking occurs automatically when transferring the program with **RUN**. Incomplete programs can not be transferred to PLC.

8.2. *RUN P*

Command: ***RUN P* [↵]**

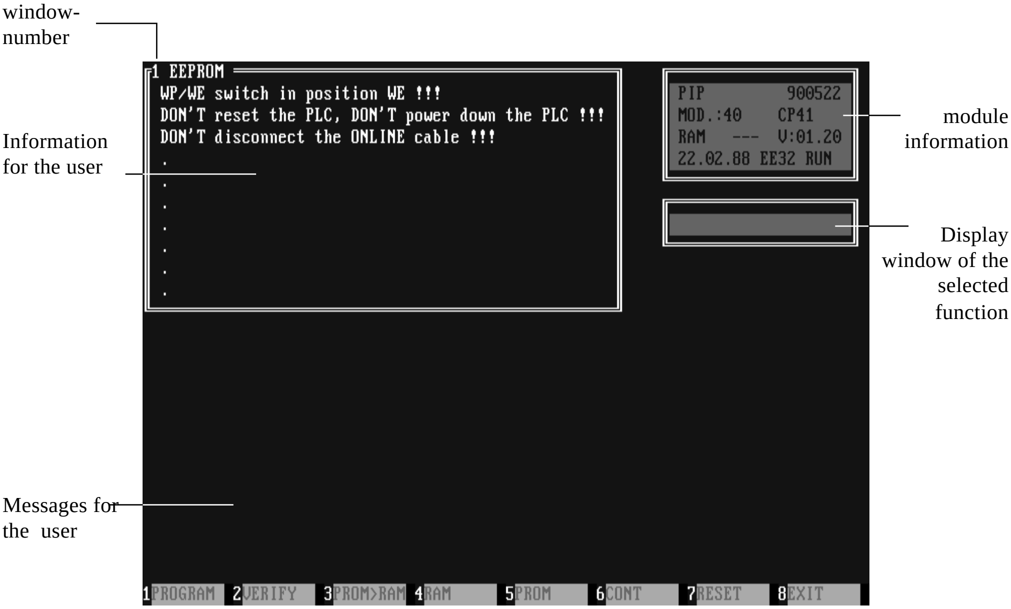
Function: This command is the same as the **RUN**-command, with the difference, that the program transfer to the PLC is ***protected.*** I.e.: The program can not be loaded to the PLC with ***LOAD PLC*** any more.

8.3. *EEPROM* - PROM PROGRAMMING

Command: ***EEPROM* [↵]**

Function:
1) The program that is located in the RAM in the PLC is programmed to the PROM-memory.
2) After accomplished programming of the PROM the program is executed from the PROM-memory.

After the function call the display on the screen is as follows:



```
1 EEPROM
 WP/WE switch in position WE !!!
 DON'T reset the PLC, DON'T power down the PLC !!!
 DON'T disconnect the ONLINE cable !!!
 .
 .
 .
 .
 .
 .
 .
```

```
PIP          900522
MOD.:40    CP41
RAM   ---  V:01.20
22.02.88 EE32 RUN
```

[F1 PROGRAM] Programming the PROM-memory

The control must be in RAM mode, otherwise an error message occurs:

E165 PROM CAN ONLY BE PROGRAMMED IF PLC IN RAM MODE

If the programming of the PROM-memory is ended, the PROgramming SYStem performs automatically a VERIFY (comparison between RAM and PROM). If the PROM was properly programmed, the following message is displayed:

W247 EEPROM VERIFY SUCCESSFUL !

[F2 VERIFY] Compare programs in PROM and RAM.

If both programs are identical, the following is displayed:

```
W247  EEPROM VERIFY SUCCESSFUL !
```

If they aren't identical, an error message is displayed:

```
E167  VERIFY NOT OK !  (WP,NOT EMPTY, DEFECT)?
```

[F3 PROM>RAM] Copy program from PROM to RAM

The control must be in PROM mode, otherwise an error message is displayed:

```
E170  COPY PROM TO RAM ONLY IF PLC IN PROM MODE
```

[F4 RAM] Switch PLC to RAM.
program is executed in the RAM

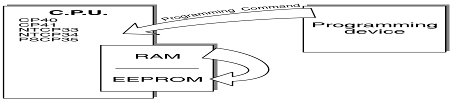
[F5 PROM] Switch PLC to PROM.
program is executed in the PROM

[F6 CONT] If the PLC-CPU is in the HALT-state, the program in the PLC is continued.

[F7 RESET] The PLC is stopped and transferred into a defined base state.

[F8 EXIT] or **[Esc]** Existing the EEPROM-menu

EEPROM Programming
(CP40, CP41, NTCP33, NTCP34, PSCP35)



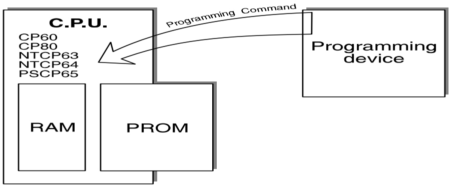
From the EEPROM-menu on your programming device the [F1 PROGRAM] command will initiate the PLC-CPU to program the EEPROM of the EE32 with the program stored in RAM. EEPROM programming can also be performed while an application program is running.

An EEPROM-program memory does not need to be deleted, it can simply be overwritten with the new program. The data in the EEPROM is non-volatile.

During EEPROM programming, the following is to be considered:

- The PLC should not be turned off.
- The WE/WP-switch of the EE32 must be set to WE (**W**rite **E**nable).

Programming the PROM-memory
(CP60, CP80, NTCP63, NTCP64, PSCP65)



From the EEPROM menu on your programming device the [F1 PROGRAM] command will initiate the PLC-CPU to program the PROM with the program stored in the RAM.

An EEPROM-program memory does not need to be deleted, it can simply be overwritten with the new program. EEPROM programming can also be performed while running application programs.

But an EPROM-memory must be deleted with UV light, before new programming is possible. EPROM programming can only occur when the PLC-CPU is in the HALT-state.

The data in the PROM is non-volatile.

During PROM programming consider the following:

- The PLC should not be turned off.
- The WE/WP-switch must be set to WE (**W**rite **E**nable).
- The PGM-LED is lit.

8.4. *LOAD PLC* - LOAD A PROGRAM OF THE PLC

Command: **[F7 LOAD] [F7 PLC] [↵]**

Function: The program that is located in the RAM, is loaded in the working memory of the PC. The program can be edited with the PROgramming SYStem.

1) After the function call it is first checked if the control is in the HALT-state:

a) **PLC in the HALT-mode:** the program is immediately loaded back into the PC.

b) **PLC in the RUN-mode:** In the message line the following enquiry is displayed:

```
STOP PLC ???  ( Y / N )
```

[Y] The execution of the program in the PLC is interrupted.

A RESET in the PLC results! Flag (F 000 to F 499), Register (R0000 to R0019) and all outputs are set back. The program is transferred from the PLC to the PC.

Continuing the program:

- In the STL status test: [F5 CONT]
- In the XFER-menu: [F6 CONT]
- transfer the new program with ***RUN* [↵]**.
- disconnect the online cable and press RESET-key of the CPU (only possible with CP40 and CP60).

[N] The program is transferred to the PC during processing. The transfer of a program takes longer than if entering [Y].

2) When loading a program the programming mode in the PC is set, in which the program was created.

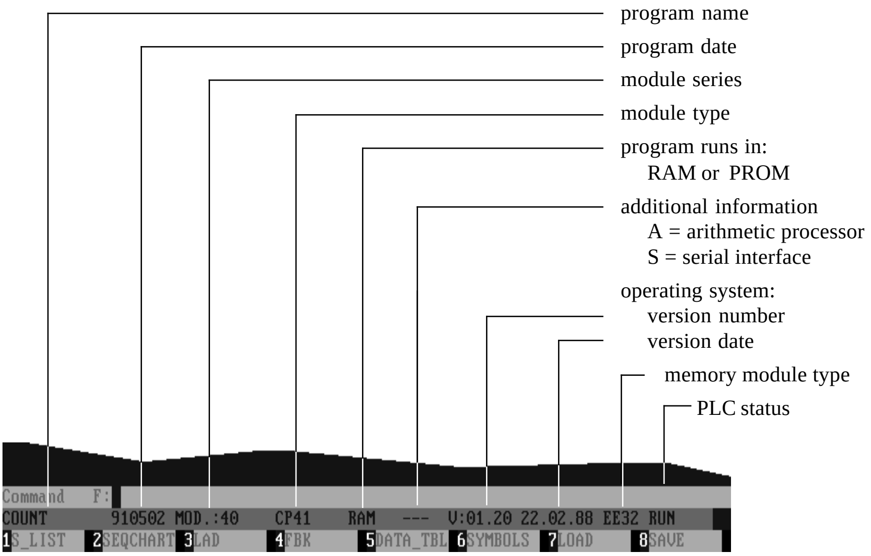
> **NOTE:**
> Comments, symbol assignments, ladder diagrams and function plans are not loaded back. They weren't transferred to the PLC with ***RUN* [↵]** during the program copy.

8.5. *P-VERS* - PLC PROGRAM INFORMATION

Command: ***P-VERS* [↵]**

Function: Information display about the program in the PLC.

After this function input program information is displayed in the 25th line on screen:



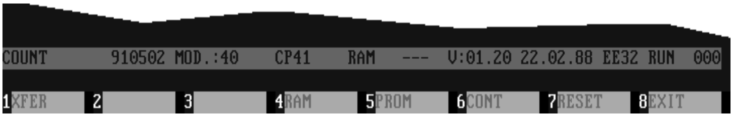
```
Command   F:
COUNT        910502 MOD.:40    CP41     RAM    ---   V:01.20 22.02.88 EE32 RUN
1S_LIST  2SEQCHART 3LAD      4FBK      5DATA_TBL 6SYMBOLS  7LOAD     8SAVE
```

8.6. *XFER*

A	B	C	PG1000	CP80	PG-PC	PC80
●	●	●	●	●	●	●

Command: **`XFER [↵]`**

Function: After entering **`XFER [↵],`** information about the module type and the running program are displayed (compare command **`P-VERS`**). The softkey allocation is changed:



```
COUNT        910502 MOD.:40    CP41    RAM   ---  V:01.20 22.02.88 EE32 RUN  000
1XFER     2         3         4RAM      5PROM     6CONT     7RESET    8EXIT
```

[F1 XFER] The application program is transferred to the RAM without interrupts of the program running in the PROM of the CPU (PP). The procedure is running completely in the background. All control functions of the processor module remain the same. For this reason the transfer can last up to a few minutes, depending on the program length.

After ending the transfer the program is still executed in the PROM, until the user switches manually to **RAM**. The command XFER allows **no RESET**, i.e. no memory locations, flags or outputs are set back.

> **NOTE:**
>
> Outputs, that were set by the old program (PROM) and that can't be processed by the new program and that are not processed any more , **remain set.**
>
> The first scan flag is not set by the first cycle of the new program.

If the processor is in the HALT state, the following error message is displayed:

`E182  XFER ONLY IF PLC IN RUN MODE`

If the program of the CPU (PP) runs in the RAM, the following error message is displayed:

`E180  XFER ONLY IF PLC IN RUN MODE`

[F4 RAM] Switching to RAM (only if PLC in RUN mode)

[F5 PROM] Switching to PROM (only if PLC in RUN mode)

[F6 CONT] If the CPU or the PP are in HALT state, the program is continued again.

[F7 RESET] The CPU or the PP are stopped and set in a defined base state.

[F8 EXIT]
[Esc] Exiting the XFER-menu.

CHAPTER 6

SYMBOL EDITOR

Contents:

1. GENERAL

A	B	C	PG1000	CP80	PG-PC	PC80
●	●	●	●	●	●	●

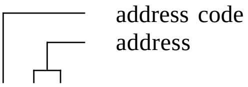
Symbols are names that are assigned to input, output, or memory location addresses . The user can use either the physical address or the symbol name. The PROgramming SYStem automatically and interactively fills in the missing information.

Symbols are an important part of documentation since the symbolic name gives information about the purpose and function of input, output or memory location.

Symbols can be given for addresses with the following address codes:

I, O, R, F, P, S, T, B, G

A symbol name may be used only once! If you try to assign the same SYMB again, an error message appears:



```
SYMB already assigned to X YYY
```

Without EMS memory, up to **2000** symbols can be used.

If there is sufficient EMS memory (at least 128 Kbytes), a total of 8100 symbols can be assigned. You must request the use of EMS memory with an environment variable (B&R$EMS=ON).

Detailed information on EMS management can be found in "Appendix C ADDITIONAL FEATURES".

2. CALL SYMB EDITOR

Command: **[F6 SYMBOLS] <ADDR> [↵]**

The SYMB editor appears after the command has been entered with [↵]. The cursor (blinking cursor) is in the command line.

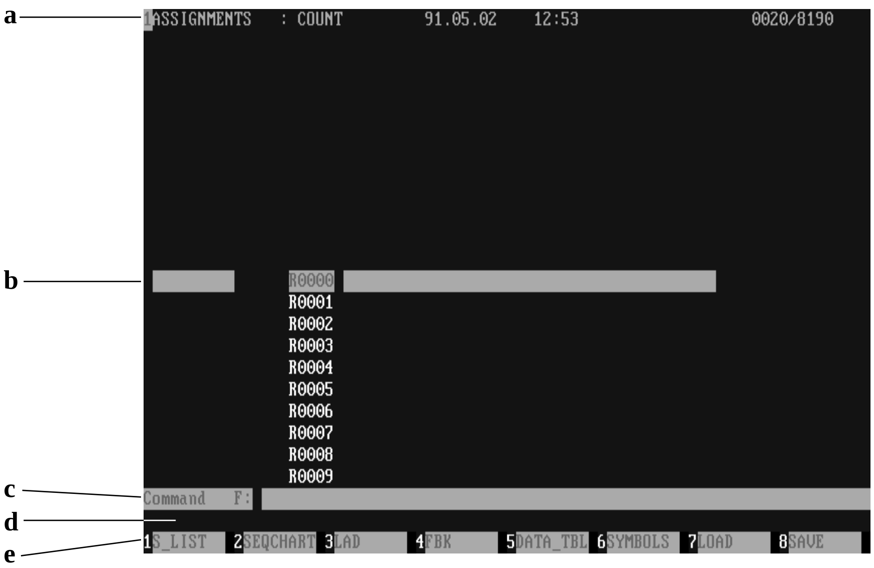
If you enter an address <ADDR>, that line in the SYMB list is displayed in inverse video (= entry line for the SYMB editor). If no address is given after the command **[F6 SYMBOLS]**, the line of the last address worked on (R0000 if SYMB editor was just called) is inverse.

If you press the [↵] key, the cursor moves to the entry line of the SYMB editor. If you would like to enter or change something in another line of the SYMB editor, you can position the cursor there by entering <**ADDR>** [↵].

If the cursor is located in the entry line, you can return to the command line with [Esc].

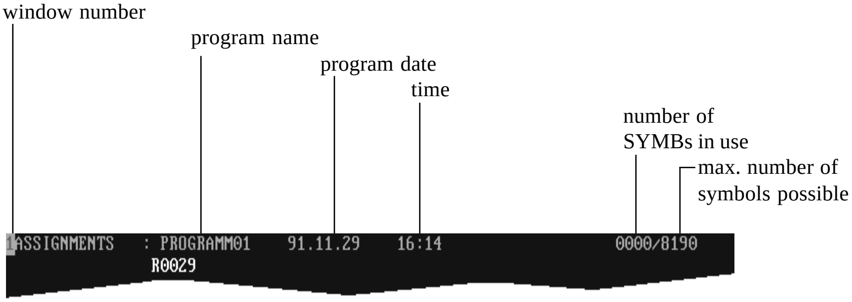
3. STRUCTURE OF THE SYMB MENU

After you have called up the SYMB editor with ***SYMBOLS*[↵]**, the following menu appears:



```
1ASSIGNMENTS   : COUNT        91.05.02    12:53              0020/8190

                R0000
                R0001
                R0002
                R0003
                R0004
                R0005
                R0006
                R0007
                R0008
                R0009
Command   F:
```

a) Status line: Various information is displayed in the status line:



```
1ASSIGNMENTS   : PROGRAMM01    91.11.29    16:14               0000/8190
                 R0029
```

b) Entry line

Symbols can be entered in this line to the left and right of the address in the same line:

- to the left of the address a symbol of up to 9 characters can be entered that will be displayed in ladder diagrams.
- to the right of the address a symbol of up to 41 characters can be entered which will be displayed to the right of the address in the STL

If the cursor is located in the command line, it is moved to the entry line by pressing [↵].

c) Command line

The command line is for entering operating commands. The most important commands are also available by function key.

d) Message line

In the message line, messages are displayed that concern the operator. These are error messages that can be the result of an operator error, but messages that prompt the user to enter parameters may also appear.

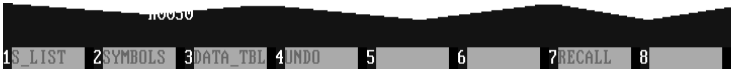
e) Softkey line

In the softkey line the functions of the SYMB editor are presented on keys [F1] to [F7].

4. FUNCTIONS IN THE SYMB EDITOR

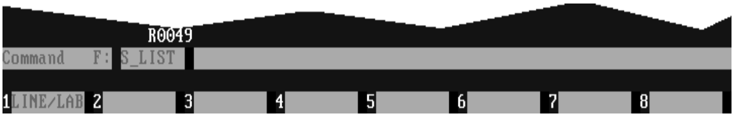
4.1. SOFTKEY FUNCTIONS

If the cursor is located in the entry line of the SYMB editor, the softkey line appears as follows:



```
R0050
1S_LIST  2SYMBOLS 3DATA_TBL 4UNDO    5        6        7RECALL  8
```

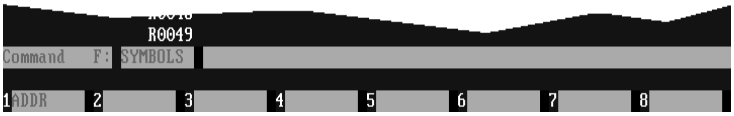
[F1 S_LIST] By pressing this function key, you exit the SYMB editor and the cursor moves to the command line. The PROgramming SYStem requests a line number or a label:



```
R0049
Command   F: S_LIST

1LINE/LAB 2        3        4        5        6        7        8
```

After you have completed the entry with [↵], the STL editor is called again.

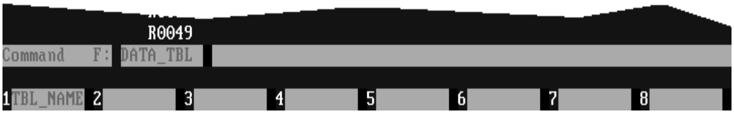
[F2 SYMBOL] If you press this function key, you exit the SYMB editor and the cursor moves to the command line. The PROgramming SYStem requests an address.



```
R0049
Command   F: SYMBOLS

1ADDR     2        3        4        5        6        7        8
```

After you have completed the entry with [↵], the SYMB editor is called again.

[F3 TAB] If you press this function key, you exit the SYMB editor and the cursor moves to the command line. The PROgramming SYStem requests the name of a data table:



```
R0049
Command   F: DATA_TBL

1TBL_NAME 2        3        4        5        6        7        8
```

After you have completed the entry with [↵], the TAB editor is called again.

[F4 UNDO] The SYMB editor has an LIFO memory (last-in-first-out) that stores the last 10 SYMBs or characters deleted. The function [F4 UNDO] calls these to the cursor position in the text.

[F7 RECALL] If you have altered a symbol and you would like to call back the original, you can use this function to do so, provided the change has not yet been completed with [↵].

4.2. ADDITIONAL FUNCTIONS

[↑]	Move SYMB list down one line.
[↓]	Move SYMB list up one line.
[PgUp]	Move SYMB list down one page.
[PgDn]	Move SYMB list up one page.
[Ctrl] + [PgUp]	Move cursor to first line (R0000) (Beginning of symbol).
[Ctrl] + [PgDn]	Move cursor to last line (Z D99) (end of SYMB).
[Ctrl] + [↑] or [↓]	Move entry line up or down.
[Alt] + [↑] or [↓]	Move entry line to upper or lower border of screen.
[Ctrl] + [Esc]	Move entry line to center of screen.
[Ctrl] + [Del]	Delete SYMB entries in the entry line.
[Tab] or [⇑] + [Tab] [Ctrl] + [←] or [→]	Move cursor to the left/right side of the address.

4.3. EXIT THE SYMB EDITOR

[Esc]	Exits the SYMB editor and enters the command line where various editing functions can be called up for the SYMB.
[F1 STL] <LINE/LAB> [↵]	Exits the SYMB editor and calls up the STL editor.
[F3 TAB] <TNAME> [↵]	Exits the STL editor and calls up the TAB editor.

5. EDITING FUNCTIONS FROM THE COMMAND LINE

5.1. MOVE

With this function the SYMBs of a specified address range can be moved to another place.

The function [**F9**] + [**F6 MOVE**] in the SYMB editor is basically the same as the one in the STL editor. When specifying the area to be moved, no line number or label is given, just start, end, and destination address.

Example: The SYMBs from address R0000 to R0099 are moved to R1300 to R1399 with the command:

```
[F9] + [F6 MOVE] [F1 FROM] R0000 [F2 UP_TO] R0099 [F3 TO] R1300
[↵]
```

If the destination addresses already contain symbols, these are overwritten.

5.2. COPY

With this function the SYMBs of a specified address range can be copied to another place.

The function **[F9] + [F5 COPY]** in the SYMB editor is basically the same as the one in the STL editor. When specifying the area to be copied, no line number or label is given, just start, end, and destination address.

Example: The SYMBs of the addresses from I 000 to I 03N are copied to I A00 to I ADN with the following command:

```
[F9] + [F5 COPY] [F1 FROM] I000 [F2 UP_TO] I03N [F3 TO] IA00 [↵]
```

Because copying would cause symbols to be used more than once, the following message appears:

```
SYMB already assigned to   X YYY
```

The cursor is simultaneously placed in the line of the first SYMB that appears twice so that it can be changed. Only after a SYMB is given that does not already exist can the next SYMB be changed.

This procedure can be interrupted at any time with the [Esc] key.

Copying is only worthwhile if the SYMBs you would like to use in another address range are nearly the same (e.g. just one character is different).

5.3. DELETE BLOCK

With this function the SYMBs of a specified address range can be deleted.

The function **[F9] + [F7 DELETE] [F7 BLOCK]** in the SYMB editor is basically the same as the one in the STL editor. When specifying the area to be deleted, no line number or label is given, just start and end address of the block to be deleted.

Example: The SYMBs of addresses F300 to F 599 are deleted with the following command:

```
[F9] + [F7 DELETE] [F7 BLOCK] [F1 FROM] F 300 [F2 UP_TO] F599 [↵]
```

5.4. FREE

Command: ***FREE*** **<ADDR/SYMB> [↵]**

Argument: For <ADDR>, either an address code, a complete address, or a symbol can be entered.

Valid address codes: I, O, R, F, S, T, P, B, G

Complete addresses:
`I 000` to `I 3FN`
`O 000` to `O 3FN`
`R0000` to `R7167`
`F 000` to `F 799`
`S 000` to `S 3F7`
`T 000` to `T 3F7`
`P 000` to `P 3FF`
`B0000` to `B4095`
`G0000` to `G4095`

Symbol: Any valid symbol is allowed.
Symbols that contain spaces must be written between quotation marks ("SYMB").

Function: Search for addresses without symbols. The search begins at the address indicated <ADDR>.

If a free address is found, the cursor is moved to the free SYMB field of that address.

Example: The search for the first digital input containing no SYMB is started with the command:

FREE I000 **[↵]** or ***FREE I*** **[↵]**

The cursor is moved to the first line of the input address that contains no SYMB.

CHAPTER 7

DATA TABLE EDITOR

Contents:

1. GENERAL INFORMATION

A	B	C	PG1000	CP80	PG-PC	PC80
	●	●	●	●	●	●

Tables are constant data blocks, which can be accessed by the application program. Entries are made by using a comfortable table editor. The number formats decimal, binary, hexadecimal and ASCII can be optionally mixed within a table or line. Tables are components of the application program and are stored in the non-volatile application program memory of the PLC.

2. CALL OF THE TAB EDITOR

Command: **[F5 TAB] <TNAME> [↵]**

Parameter: <TNAME> has a maximum length of 4 characters; the first character has to be a letter.

After quitting the command input with [↵] the TAB editor appears. The input cursor (blinking cursor) is in the command line.

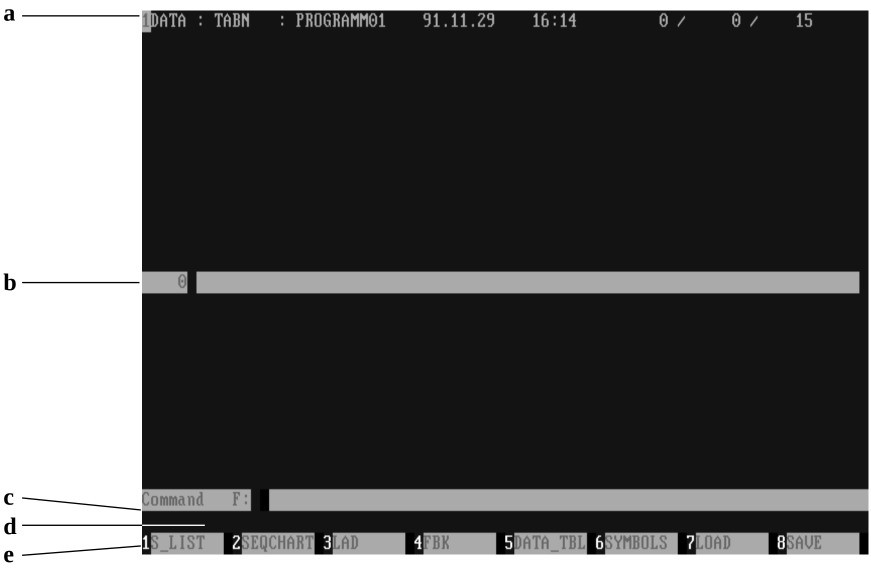
With the input of the table name <TNAME> the corresponding table is called or created anew. Without the definition of a name the last processed table is called up.

The cursor is set to the input line of the TAB editor with the input of **[↵]** in the command line. If something should be changed in a different line of the TAB editor, entering **<LINE> [↵]** will position the input cursor to the respective line.

If the cursor is in the input line, the [Esc] key will return it to the command line.

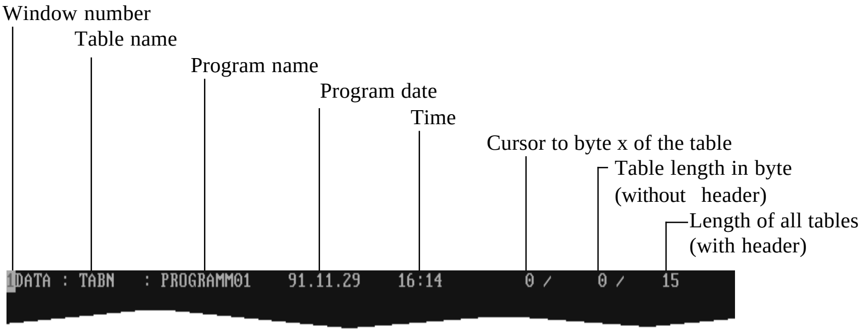
3. TAB MENU STRUCTURE

After the call of the TAB editor with ***TAB* [↵]** the following menu is displayed:



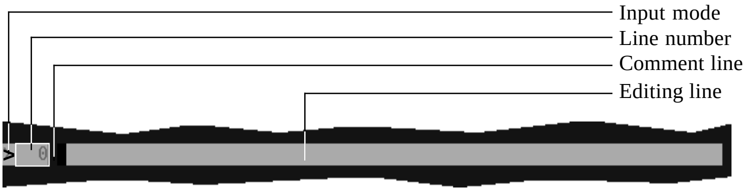
```
1DATA : TABN   : PROGRAMM01    91.11.29    16:14        0 /      0 /    15

    0

Command   F:
1S_LIST  2SEQCHART 3LAD      4FBK      5DATA_TBL 6SYMBOLS  7LOAD     8SAVE
```

a) Status line: Various information is displayed in the status line:


Window number
Table name
Program name
Program date
Time
Cursor to byte x of the table
Table length in byte (without header)
Length of all tables (with header)

```
1DATA : TABN   : PROGRAMM01    91.11.29    16:14        0 /      0 /    15
```

b) Input line:



```
>  0
```

Insert mode: If the character “>” is in this position, the TAB editor is in the insert mode. Pressing [↵] inserts a new line. If the insert mode is not turned on, the table is moved up one line, by pressing the [↵] key, and the next line can be edited.

The insert mode is **turned on**, with [Ctrl] + [Ins] or [Alt] + [Ins].
The insert mode is **turned off**, if either [↑], [↓], [PgUp] or [PgDn] is pressed.

Comment line: If the character “*” is at this position, the line is marked as comment line. Blank TAB-lines are marked with a “*” automatically.

Comments can be entered next to data (marked with “*”):

```
     :
     :
15*This is a comment line!
16 23,%01110,$1AF0,'TEXT'       *comment
     :
     :
```

c) Command line:

The command line serves for the input of operating commands. The most important commands are available with the function keys.

d) Message line

In the message line messages concerning user errors or messages requiring the input of parameters are displayed.

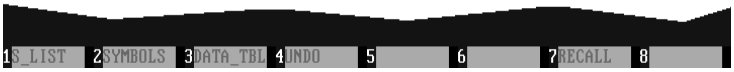
e) Softkey line

In the softkey line important functions for TAB editing are offered on keys [F1] - [F7].

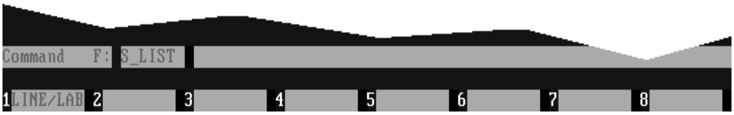
4. FUNCTIONS IN THE TAB EDITOR

4.1. SOFTKEY FUNCTIONS

If the cursor is in the input line of the TAB editor, the softkey line is allocated as follows:

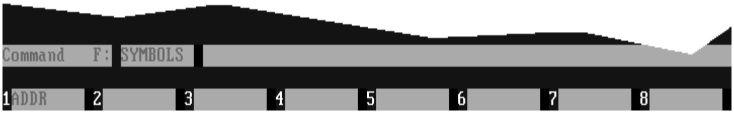


[F1 S_LIST] Pressing this function key will exit the SYMB editor and the cursor is set to the command line. The PROgramming SYStem requires the input of a line number or a label:


Command F: S_LIST

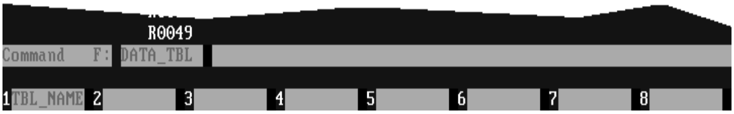
After terminating the input with [↵] the STL editor is called.

[F2 SYMBOLS] By pressing this function key the SYMB editor is exited and the cursor is set in the command line. The PROgramming SYStem demands the input of an address:


Command F: SYMBOLS

After terminating the input with [↵] the SYMB editor is called.

[F3 TAB] By pressing this function key the SYMB editor is exited and the cursor is set to the command line. The PROgramming SYStem requires the input of a line number or a table:


R0049

Command F: DATA_TBL

After terminating the input with [↵] the STL editor is called.

[F4 UNDO] The TAB editor is provided with a LIFO storage (Last-In-First-Out), where the last 10 deleted lines or characters are stored. By pressing [F4 UNDO] these are inserted in the text at the cursor position (last character deleted is first character inserted).

[F7 RECALL] If a TAB line has been changed, the original line can be recalled with this function, if the changes were not terminated with [↵].

4.2. ADDITIONAL FUNCTIONS

[↑]	Move TAB down one line.[1]
[↓]	Move TAB up one line.[1]
[PgUp]	Move TAB down one screen page.[1]
[PgDn]	Move TAB up one screen page.[1]
[Ctrl] + [PgUp]	Position cursor in the first program line (beginning of TAB).[1]
[Ctrl] + [PgDn]	Position cursor in the last program line (end of TAB).[1]
[Ctrl] + [↑] or [↓]	Move input line up/down.[1]
[Alt] + [↑] or [↓]	Position input line to top/bottom of screen.[1]
[Ctrl] + [Esc]	Position input line to the center of the screen.[1]
[Ctrl] + [Ins]	Insert line[2]
[Alt] + [Ins]	Insert a comment line.[2]
[Ctrl] + [Del]	Delete line.[1]
[Tab] or [Ctrl] + [→]	Position cursor one tabulator position right.
[⇑] + [Tab] or [Ctrl] + [←]	Position cursor one tabulator position left.

[1] The insert mode is turned off.
[2] The insert mode is turned on.

4.3. EXITING THE TAB EDITOR

[Esc] Exits the TAB editor to the command line where different editing functions for the TAB can be called.

[F1 STL] <LINE/LAB> [↵] Exits the TAB editor and calls the STL editor.

[F2 SYMB] <ADDR> [↵] Exits the TAB editor and calls the SYMB editor.

5. EDITING FUNCTIONS FROM THE COMMAND LINE

5.1. MOVE

With this function any line in a table can be moved to a different position.

The function **[F9] + [F6 MOVE]** in the TAB editor is basically identical to the same function in the STL editor. To define the range to be moved a line number is entered.

Example: The lines 10 to 15 are moved to 20 with the following:

```
[F9] + [F6 MOVE] [F1 FROM] 10 [F2 UP TO] 15 [F3 TO] 20
[↵]
```

5.2. COPY

With this function lines of a table can be copied to a different position.

The function **[F9] + [F5 COPY]** in the TAB editor is basically identical to the same function in the STL editor. To define the range to be copied line numbers are given.

Example : The lines 0 to 10 are copied with the following command and inserted at line 30:

```
[F9] + [F5 COPY] [F1 FROM] 0 [F2 UP TO] 10 [F3 TO] 30 [↵]
```

5.3. DELETE BLOCK

With this function certain lines of a table can be deleted.

The function **[F9] + [F7 DELETE] [F7 BLOCK]** in the TAB editor is basically identical to the same function in the STL editor. To define the range to be deleted a line number is entered.

Example: The lines 15 to 30 are deleted with the following command:

```
[F9] + [F7 DELETE] [F7 BLOCK] [F1 FROM] 15 [F2 TO] 30 [↵]
```

6. TABLE PROGRAMMING

6.1. DATA FORMATS

Decimal

Characteristics:	none		
Number of bytes:	1 to 3		
Number range:	0	to	16777215 (2^{24} - 1)

The PROgramming SYStem reserves the correct number of bytes for the value entered in memory:

1 byte:	0	to	255 (2^{8} - 1)
2 byte:	256	to	65535 (2^{16} - 1)
3 byte:	65536	to	16777215 (2^{24} - 1)

It is impossible to enter negative values.

Hexadecimal

Characteristic:	$		
Number of bytes:	1 to 3		
Number range:	$0	to	$FFFFFF

The PROgramming SYStem reserves the correct number of bytes for the value entered in memory:

1 byte:	$00	to	$FF
2 byte:	$0100	to	$FFFF
3 byte:	$010000	to	$FFFFFF

Binary

Characteristic: `%`

Number or bytes: 1 to 3

Number range: `%0` to `%11111111 11111111 11111111`

The PROgramming SYStem reserves the correct number of bytes for the value entered in memory:

1 byte:	`%0`	to	`%11111111`
2 byte:	`%1 00000000`	to	`%11111111 11111111`
3 byte:	`%1 00000000 00000000`	to	`%11111111 11111111 11111111`

ASCII

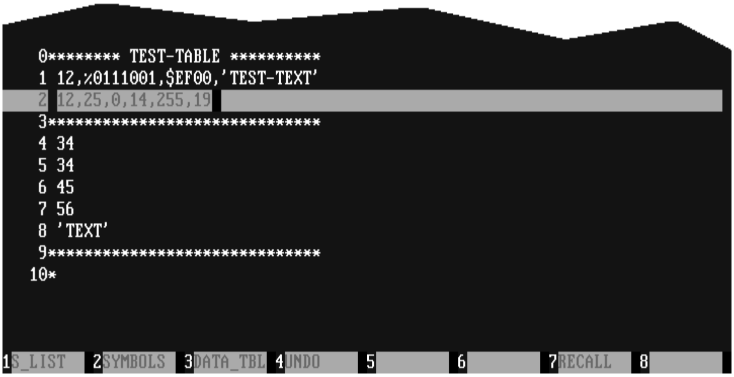
Characteristics: `'` (before and after the ASCII character)

The length of the text between the characters “`'.....'`” is optional. Every ASCII character occupies **one** byte.

Example: `'TEXTNUMBER 1'` This text requires 12 bytes.

The input of the values in different formats can be mixed! It is important, that the separation character “`,`” is inserted between individual values.

Example:



```
 0******** TEST-TABLE **********
 1 12,%0111001,$EF00,'TEST-TEXT'
 2 12,25,0,14,255,19
 3******************************
 4 34
 5 34
 6 45
 7 56
 8 'TEXT'
 9******************************
10*
```

6.2. TABLE CALL

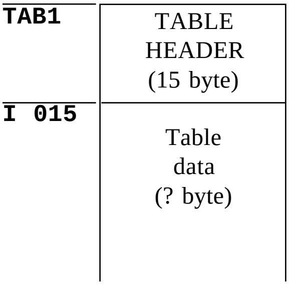
6.2.1. From an STL

A	B	C	PG1000	CP80	PG-PC	PC80
		●	●	●	●	●

Tables can be called from an STL like subprograms:

```
JSR  xxxx        xxxx .... Table name
```

When calling a table with JSR, the address of the first byte is returned to index register X and the length of the table to accumulator AB. If the table is longer than 255 bytes, the length of the table is submitted to double accumulator AD. The length, that is submitted, corresponds with the actual number of data bytes in the table. The table header of 15 bytes is not contained.



The table header is the same as a small subprogram, that stores table length and table beginning in register AB or AD and X.

The table beginning can be established in two different ways:

1) With JSR xxxx, as described above:

```
  :
JSR     TAB1
LDAA      X 000   load the first data byte
  :
```

2) Addition of an offset (=15) to the label TAB 1:

```
  :
LDXL    TAB1
LDAA      X 015   load the first data byte
  :
```

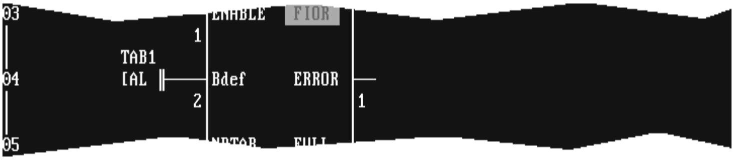
The length of a table can only be determined with choice number 1.

6.2.2. LAD CALL

A	B	C	PG1000	CP80	PG-PC	PC80
	●	●	●	●	●	●

Tables can be connected to a function block in a ladder diagram. The call of the table is made with the label contact (F6 [AL ||—).

Example:

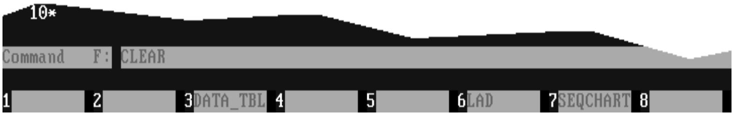


See chapter 4, section 3.6. CONTACTS AND CONNECTIONS

7. *CLEAR* - DELETING A TABLE

Command: ***CLEAR*** [F3 TAB] <TNAME> [↵]

A TAB can be deleted with the operating command ***CLEAR***. If this command is entered in the command line, the allocation of the function keys changes:



```
10*
Command   F: CLEAR
```

By selecting [F3 TAB] and entering the TAB name terminated with [↵], the respective TAB is deleted.

CHAPTER 8

LOADING, SAVING, ...

Contents:

1. SELECTING A PATH

A	B	C	PG1000	CP80	PG-PC	PC80
●	●	●	●	●	●	●

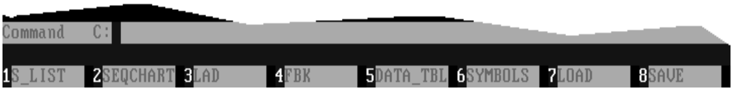
1.1. SELECTING A DRIVE

Command: ***DRIVE* <Drive Name> [↵]**

Function: With this function the drive is chosen for using commands like SAVE or LOAD where a drive name is needed in the file name.

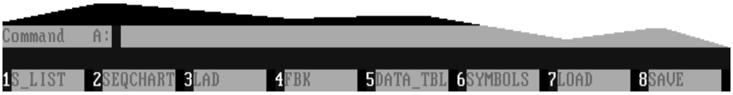
For the <Drive Name> you can give any name that the operating system of your PC knows. For example **A** for a floppy disk drive and **C** for your hard disk drive.

The name of the actual drive is displayed in the command line.



```
Command   C:
1S_LIST   2SEQCHART 3LAD      4FBK      5DATA_TBL 6SYMBOLS  7LOAD     8SAVE
```

After inputting ***DRIVE A* [↵]** in the command line the actual drive changes to drive **A**.



```
Command   A:
1S_LIST   2SEQCHART 3LAD      4FBK      5DATA_TBL 6SYMBOLS  7LOAD     8SAVE
```

1.2. CHANGING FROM DEFAULT, STANDARD AND USER DIRECTORIES

Command: ***DDIR \PATH* [↵]** Default Directories
***SDIR \PATH* [↵]** Standard Directories
***UDIR \PATH* [↵]** User Directories

A PATH is not only restricted to one directory, but can contain several subdirectories as well.
E.g.: `DDIR \PROJECT\TEST [↵]`

Function: This command determines the exact search path for the loading and saving of programs or program parts.

The commands ***DDIR***, ***SDIR*** and ***UDIR*** refer to the current designated drive. The search paths of other drives are not altered in any way. For every drive, three search paths can be defined.

If no start parameters are indicated when the PROgramming SYStem is started, the directories that were automatically set up when the software was installed are used.

`\pgdef`	Default Directory
`\pgstd`	Standard Directory
`\pguser`	User Directory

The procedure for setting other directories when starting the PROgramming SYSstem is described in Chapter 3, INTRODUCTION.

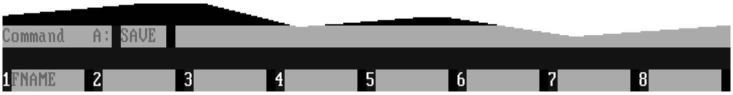
2. SAVING

A	B	C	PG1000	CP80	PG-PC	PC80
●	●	●	●	●	●	●

Command: **[F8 SAVE]**

Function: With this function, entire programs or program parts (tables, ladder diagrams, symbol lists, STL macros, ...) can be saved on diskette or hard disk.

After you have entered [**F8 SAVE**], you will be asked to indicate the file name under which you would like to save the program or program part.



Command A: SAVE

For <FNAME>, you can enter a file name either with or without the full path. If you do not indicate the entire path, the file is saved in the default directory.

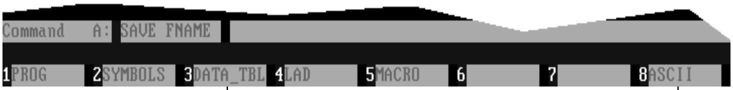
NOTE: The file name must be up to 8 characters long **without extension**. The B&R PROgramming SYStem automatically adds on the extension.

Possible file name extensions:

xxxxxxxx.PRG	Program	xxxxxxxx.MSL	Statement list macro
xxxxxxxx.SYM	Symbols	xxxxxxxx.ATA	ASCII table
xxxxxxxx.TAB	Table	xxxxxxxx.ASY	ASCII symbols
xxxxxxxx.LAD	Ladder diagram	xxxxxxxx.EMS	EMS symbols
xxxxxxxx.MSY	Symbol macro	xxxxxxxx.PRV	PROVIT files
xxxxxxxx.MTA	Table macro	xxxxxxxx.BSC	Basic files

xxxxxxxx ... max. 8-character file name

After the file name has been entered, the softkey line is altered:



Command A: SAVE FNAME

only in extension C

only in extensions B and C

SAVE [↵] without additional parameters

A	B	C	PG1000	CP80	PG-PC	PC80
●	●	●	●	●	●	●

Command: **[F8 SAVE] [↵]**

Function: The entire program is saved with the parameters that are defined by one of the following commands:

1) **[F8 SAVE] <FNAME> [F1 PROG] [↵]**
2) **[F7 LOAD] [F1 PROG] <FNAME> [↵]**

The PROgramming SYStem remembers the file name and the directory and saves the entire program according to those parameters. If you try to save the program with this command but no parameters have been saved, the following error message appears:

```
E143  MISSING FILE NAME
```

A	B	C	PG1000	CP80	PG-PC	PC80
●	●	●	●	●	●	●

[F1 PROG] - Save Program

Command: **[F8 SAVE] <FNAME> [F1 PROG] [↵]**

Function: The entire program is saved under the file name <FNAME>indicated. This includes: STL program, ladder diagrams, tables, function blocks, and symbols.

A	B	C	PG1000	CP80	PG-PC	PC80
●	●	●	●	●	●	●

[F2 SYMBOLS] - Save Symbol List

Command: **[F8 SAVE] <FNAME> [F2 SYMBOLS] [↵]**

Function: All of the symbols defined in an existing program are saved as separate program parts under the file name <FNAME> indicated. If the EMS SYMB management is activated, the file is automatically "saved with the extension ".ESM".

A	B	C	PG1000	CP80	PG-PC	PC80
	●	●	●	●	●	●

[F3 DATA_TBL] - Save Table

Command: **[F8 SAVE] <FNAME> [F3 DATA_TBL] <TBL_NAME> [↵]**

Function: The table <TBL_NAME> indicated is saved as a separate program part under the file name <FNAME>.

A	B	C	PG1000	CP80	PG-PC	PC80
●	●	●	●	●	●	●

[F4 LAD] - Save Ladder Diagram

Command: **[F8 SAVE] <FNAME> [F4 LAD] <LAD_NAME> [↵]**

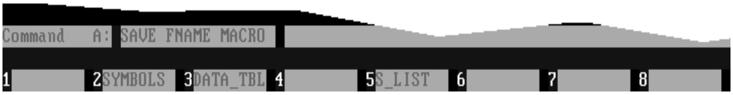
Function: The ladder diagram <LADNAME> indicated is saved as separate program part under the file name <FNAME>.

[F5 MACRO] - Save STL, TAB, or SYMB Parts as Macros

Command: **`[F8 SAVE] <FNAME> [F5 MACRO] ...`**

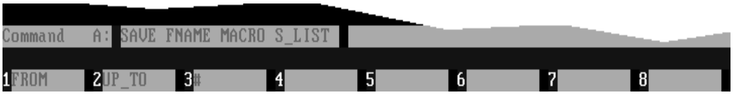
Function: With this function certain STL, TAB, or SYMB parts can be saved as macros. These macros can be used again later (e.g. in other programs).

After you have entered **[F8 SAVE] <FNAME> [F5 MACRO]** the softkey line is altered:



```
Command    A: SAVE FNAME MACRO
1        2SYMBOLS 3DATA_TBL 4        5S_LIST  6        7        8
```

You can enter, directly or by function key, whether a part of a STL, a TAB, or a SYMB is to be saved as a macro. After you have made the selection, the softkey line is altered:



```
Command    A: SAVE FNAME MACRO S_LIST
1FROM    2UP_TO   3#       4        5        6        7        8
```

Now you must define the range that is to be saved as macro. The beginning and the end of the range are defined by the following entries:

MACRO	RANGE Beginning	 End	Level A	 B	 C
SYMBOLS	**[F1 FROM] <ADDR>** from the current entry line	**[F2 UP_TO]<ADDR>** **[F3 #] <NR>**	●	●	●
DATA_TBL	**[F1 FROM] <LINE>** from the current entry line	**[F2 UP_TO]<LINE>** **[F3 #] <NR>**	●	●	
S_LIST	**[F1 FROM] <LINE/LAB>** from the current entry line	**[F2 UP_TO]<LINE/LAB>** **[F3 #] <NR>**	●		

Confirm the range definition you want to save with [↵] and it will be saved as a macro.

Example: The STL lines 100 to 300 are to be saved under the file name `TEST` in the default directory. This can be done in several different ways.

```
[F8 SAVE] TEST [F5 MACRO] [F5 S_LIST] [F1 FROM] 100 [F2 UP_TO] 300   [↵]
[F8 SAVE] TEST [F5 MACRO] [F5 S_LIST] [F1 FROM] LAB1 [F2 UP_TO] 300  [↵]
[F8 SAVE] TEST [F5 MACRO] [F5 S_LIST] [F1 FROM] 100 [F3 #] 201  [↵]
```

Set the current entry line in the STL editor to line 100:

```
[F8 SAVE] TEST [F5 MACRO] [F5 S_LIST] [F2 UP_TO] 300 [↵]
[F8 SAVE] TEST [F5 MACRO] [F5 S_LIST] [F3 #] 201 [↵]
```

Note: If TAB, LAD, or subroutine calls exist in the STL macros, the TABs, LADs and subprograms are not saved with the macro.

Symbols and comments within the macro area are saved as well.

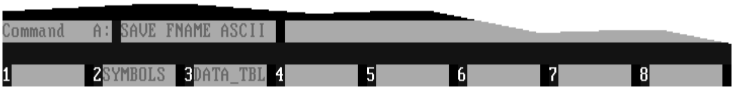
[F8 ASCII] - Save Parts of TAB or SYMB in ASCII Format

A	B	C	PG1000	CP80	PG-PC	PC80
		●	●	●	●	●

Command: **`[F8 SAVE] <FNAME> [F8 ASCII] ...`**

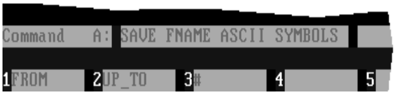
Function: With this function, certain parts from TABs (levels B and C) or SYMBs (levels A, B and C) can be saved in ASCII format so that they can also be used with other programming systems (CAD systems, editors, ...).

The softkey line is altered after this command:



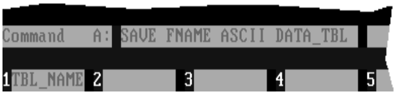
```
Command   A: SAVE FNAME ASCII
1         2SYMBOLS  3DATA_TBL 4         5         6         7         8
```

You can select whether to save an entire TAB or a part of a SYMB in ASCII format by direct entry from the keyboard or by function key. After you have made your entry, the softkey line is altered:



```
Command   A: SAVE FNAME ASCII SYMBOLS
1FROM     2UP_TO    3#        4         5
```

The PROgramming SYStem asks you to enter the STL area, which is defined in the same way as when saving macros. Confirm your entry with [↵] and the defined STL area will be saved under the file name FNAME.ASY.



```
Command   A: SAVE FNAME ASCII DATA_TBL
1TBL_NAME 2         3         4         5
```

The PROgramming SYStem asks you to enter the table name <TBL_NAME>. Confirm your entry with [↵] and the TAB <TBL_NAME> will be saved under the file name FNAME.ATA.

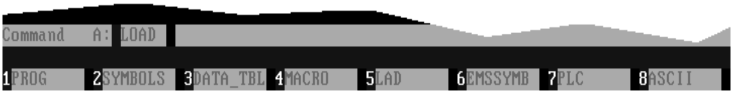
3. LOADING

A	B	C	PG1000	CP80	PG-PC	PC80
●	●	●	●	●	●	●

Command: **[F7 LOAD]**

Function: Programs or program parts (tables, ladder diagrams, etc.) can be loaded from a diskette or a hard disk with this function.

After the input of **[F7 LOAD]** the softkey line is altered:



```
Command    A: LOAD
1PROG   2SYMBOLS 3DATA_TBL 4MACRO   5LAD     6EMSSYMB 7PLC     8ASCII
```

Loading the programming device PG1000 or the PLC is described in:

[F7 PLC] chapter 5 STL Editor

[F1 PROG] - Loading a program

A	B	C	PG1000	CP80	PG-PC	PC80
●	●	●	●	●	●	●

Command: **[F7 LOAD] [F1 PROG] <FNAME> [↵]**

Function: The program FNAME.PRG is loaded from the default directory. The last edited program is deleted from memory.

A path can be entered with the file name. With the input of the following command in the command line, the program TEST.PRG is loaded from the drive A, directory \PROJECT1:

[F7 LOAD] [F1 PROG] ***A:\PROJECT1\TEST*** **[↵]**

If no filename is entered or the defined program doesn't exist, all files with the extension ".PRG" in the default directories or the defined path are displayed. The desired program can be selected and loaded from the directory with the cursor and function keys.

The following **information** is displayed by files/directories:

`Number` current number of the displayed files or directories

`File name.type` file name (with extension = mode) or directory name

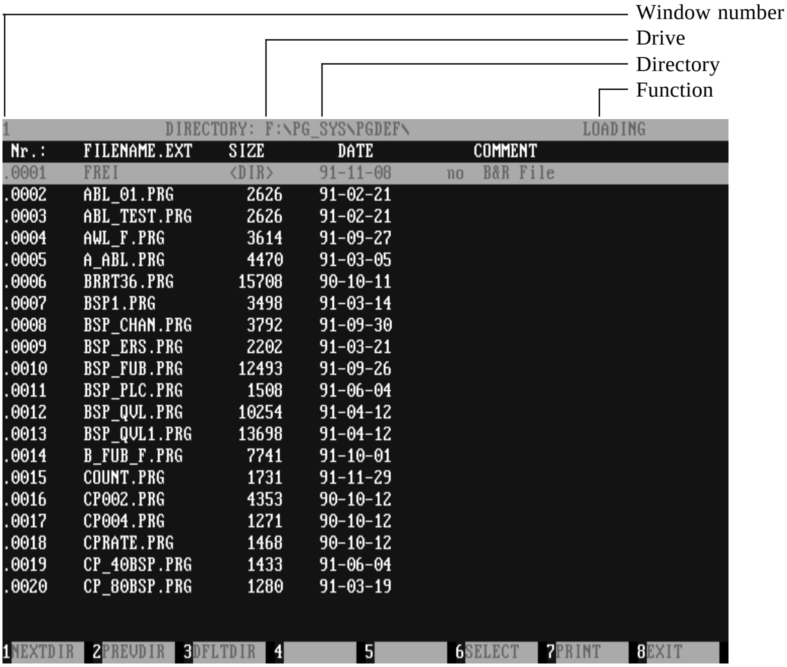
`Size` file size in byte; display, if directory: <DIR>

`Date` date of the last change

`Comment` information, if the file can be loaded by the PROgramming SYStem:

no comment - can be loaded

`no B&R-file` - can not be loaded
- also valid for directories



```
1               DIRECTORY: F:\PG_SYS\PGDEF\                    LOADING
Nr.:     FILENAME.EXT    SIZE        DATE          COMMENT
.0001    FREI            <DIR>    91-11-08      no  B&R File
.0002    ABL_01.PRG        2626   91-02-21
.0003    ABL_TEST.PRG      2626   91-02-21
.0004    AWL_F.PRG         3614   91-09-27
.0005    A_ABL.PRG         4470   91-03-05
.0006    BRRT36.PRG       15708   90-10-11
.0007    BSP1.PRG          3498   91-03-14
.0008    BSP_CHAN.PRG      3792   91-09-30
.0009    BSP_ERS.PRG       2202   91-03-21
.0010    BSP_FUB.PRG      12493   91-09-26
.0011    BSP_PLC.PRG       1508   91-06-04
.0012    BSP_QVL.PRG      10254   91-04-12
.0013    BSP_QVL1.PRG     13698   91-04-12
.0014    B_FUB_F.PRG       7741   91-10-01
.0015    COUNT.PRG         1731   91-11-29
.0016    CP002.PRG         4353   90-10-12
.0017    CP004.PRG         1271   90-10-12
.0018    CPRATE.PRG        1468   90-10-12
.0019    CP_40BSP.PRG      1433   91-06-04
.0020    CP_80BSP.PRG      1280   91-03-19

1NEXTDIR 2PREVDIR 3DFLTDIR 4        5        6SELECT  7PRINT   8EXIT
```

Key	FUNCTION
[F1 NEXTDIR]	If the cursor is in a directory (marked by <DIR>), it is opened. The contents of the directory is displayed. This function is equivalent to the DOS command “`cd <directory name>`”.
[F2 PREVDIR]	The previous directory is changed and its contents displayed. [F2 PREVDIR] is equivalent to the DOS command “`cd ..`”.
[F3 DFLTDIR]	The current directory is defined as the “default-directory”.
[F6 SELECT] **[↵]**	The program, that is selected with the [❖] keys is loaded (the respective line is displayed in inverse viedeo).
[F7 PRINT]	The displayed directory is printed.
[F8 EXIT] **[Esc]**	Exit back to the command line.

[F2 SYMB] - Load SYMBs

A	B	C	PG1000	CP80	PG-PC	PC80
●	●	●	●	●	●	●

Command: **[F7 LOAD] [F2 SYMBOLS] <FNAME> [↵]**

Function: The SYMB `FNAME.SYM` is loaded from the default directory.

A path can be entered in additon to the file name. When entering the following command in the command line the SYMB `TEST.SYM` is loaded from the directory `\PROJECT1` on the current drive:

[F7 LOAD] [F2 SYMBOLS] \*PROJECT1\\TEST* [↵]

If no file name is specified, or if the defined SYMB doesn't exist, all files with the extension "`.SYM`" in the default-directories or the defined path are displayed. The desired SYMB can be selected or loaded from this directory with the cursor or function keys (see [F1 PROG]).

All SYMBs that existed in memory before the loading procedure, are deleted from the memory.

SYMBs, that were created and stored without EMS-SYMB management, can easily be loaded, if the EMS-SYMB is turned on.

[F3 TAB] - Load Tables

A	B	C	PG1000	CP80	PG-PC	PC80
	●	●	●	●	●	●

Command: **[F7 LOAD] [F3 DATA_TBL] <TBL_NAME> <FNAME> [↵]**

Function: The SYMB `FNAME.TAB` is loaded from the default directory. The loaded table is provided with the defined name <TBL_NAME>.

A path can be entered with the file name. With the input of the following command to the command line, the TAB `TEST.TAB` is loaded from the root directory of the current drive:

```
[F7 LOAD] [F3 DATA_TBL] \TEST [↵]
```

If no file name is defined, or if the specified TAB doesn't exist, all files with the extension "`.TAB`" are displayed in the default directory or the defined path. The desired TAB can be selected and loaded of the this overview with the cursor and the function keys. (see [F1 PROG]).

If a table with this name <TBL_NAME> exists already, an error message is displayed:

```
E040  NAME USED FOR DATA TABLE
```

If a label is entered for <TBL_NAME>, that is already used in the program, the following error message is displayed:

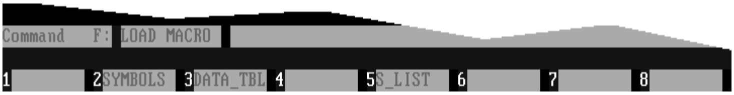
```
E022  NAME USED AS LABEL
```

[F4 MACRO] - Loading a STL, TAB or SYMB Macro

Command: **`[F7 LOAD] [F4 MACRO] [F2 SYMBOLS] <FNAME> [F1 TO] <ADDR>[↵]`**
`[F7 LOAD] [F4 MACRO] [F3 DATA_TBL] <FNAME> [F1 TO] <LINE>[↵]`
`[F7 LOAD] [F4 MACRO] [F5 S_LIST] <FNAME> [F1 TO] <LINE/LAB>[↵]`

Function: This function offers the possibility of including STL, SYMB or TAB parts, that are stored in macros, within an existing program.

After the input of **[F7 LOAD] [F4 MACRO]** the softkey line is altered:



With the function keys a STL, TAB or SYMB macro can be selected. After entering one of these three possibilities, the user is requested to enter a file name <FNAME>.

A path can be defined with the file name. With the input of the following command in the command line, the TAB macro TEST.MTA is loaded from the directory \TEST on dirve A:

`[F7 LOAD] [F4 MACRO] [F3 DATA_TBL]` ***`A:\TEST\TEST`*** **`[↵]`**

If no file name is defined, or if the specified macro doesn't exist, all files with the extension ".MTA"(TAB-Macro), ".MSY" (SYMB-Macro) or ".MSL" (STL-Macro) in the default directories or the specified paths are displayed. The desired macro can be selected and loaded from the directory with the cursor and function keys (see [F1 PROG]).

The selected macro is inserted before the TAB, SYMB or STL editor location, that is defined with **[F1 TO]**. If no parameter **[F1 TO] <ADDR/LINE/LAB>** was specified, the macro is inserted before the current input line of the respective editor.

If a macro should be loaded to a position, where this is impossible, the following error message is displayed:

```
E145  MACRO LOAD NOT ALLOWED
```

This occurs, for example, if an attempt to load a STL macro and insert it between protected lines of a LAD, is made.

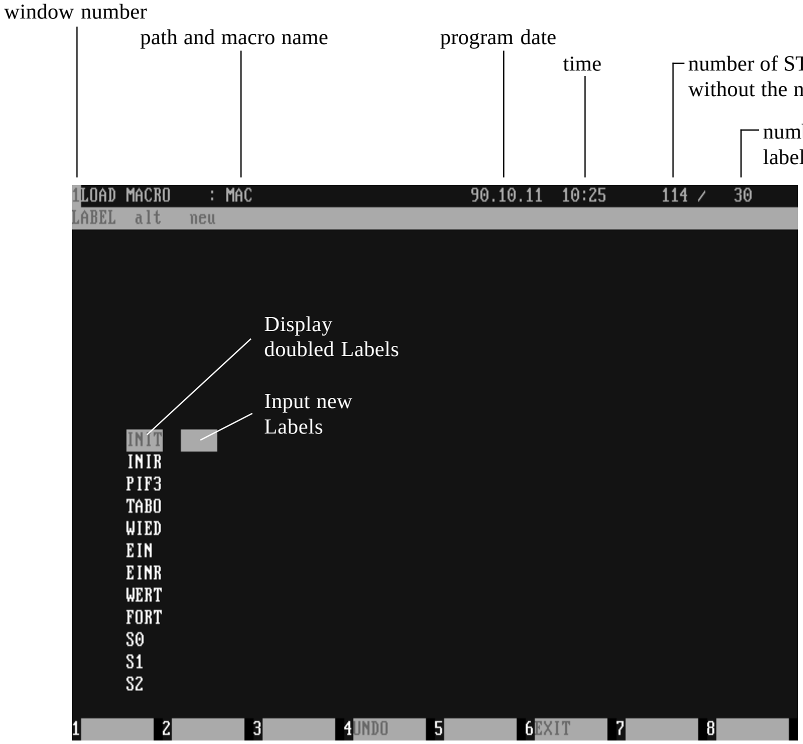
The command is available in the following expansion levels:

Macro	**Level**		
	A	**B**	**C**
SYM	●	●	●
TAB		●	●
STL			●

Correction for double allocation of labels

A	B	C	PG1000	CP80	PG-PC	PC80
		●	●	●	●	●

If the STL macro contains labels, that are already available in the program, the following picture is displayed on screen:



```
1LOAD MACRO    : MAC                    90.10.11  10:25     114 /   30
LABEL  alt   neu

      INIT
      INIR
      PIF3
      TABO
      WIED
      EIN
      EINR
      WERT
      FORT
      S0
      S1
      S2
1        2        3        4UNDO    5        6EXIT    7        8
```

The user now has to change all double labels. This menu can only be exited with **[F6 EXIT]** or **[Esc]**, if all labels were changed.

In LIFO storage (Last-In-First-Out) the 10 last deleted characters are stored, which can be inserted again in the cursor position with the function **[F4 UNDO]**.

Correction for double allocation of Symbols

A	B	C	PG1000	CP80	PG-PC	PC80
●	●	●	●	●	●	●

If the SYMB-marcro contains symbols, that were already used once, the following message is displayed:

```
W140   SYMBOL USED FOR  X YYY
```

The cursor is simultanously positioned on the first double SYMB, so it can be changed.

With the **[Esc]** key, the procedure can be interrupted any time.

[F5 LAD] - Load Ladder Diagrams

A	B	C	PG1000	CP80	PG-PC	PC80
●	●	●	●	●	●	●

Command: **[F7 LOAD] [F5 LAD] <FNAME> [F1 TO] <LAD_NAME> [↵]**

Function: The LAD FNAME.LAD is loaded from the default directory and entered in the existing program with the LAD name <LAD_NAME>.

A path can be defined with the file name. With the input of the following command in the command line, the LAD TEST.LAD is loaded from the directory \PROJECT1 on the current dirve:

```
[F7 LOAD] [F5 LAD] \PROJECT1\TEST [F1 TO] <LAD_NAME> [↵]
```

If no file name is defined or the specified LAD doesn't exist, all files with the extension ".LAD" in the default directory or the specified paths are displayed. The desired LAD can be selected and loaded of the overview with the cursor and function keys (see [F1 PROG]).

To load a LAD, a LAD name must be defined, that is not yet used as label in the program. The new LAD gets the name <LAD_NAME>. After the loading process the LAD editor is called. The loaded LAD is displayed.

In order to load a LAD, a name must be defined, that is not yet used in the program. If a LAD is already existing in the program with the defined name <LAD_NAME> or a label with the same name, an error message is displayed:

```
E022  NAME USED AS LABEL
```

If labels (conditional branches) are used in the loaded LAD, that are available elsewhere in the program and the LAD editor is exited with [Esc], the following error message is displayed:

```
E065  LABEL ALREADY USED OUTSIDE THIS LADDER
```

After this label is changed, the LAD editor can be exited.

[F6 EMSSYMB] - Loading SYMBs

A	B	C	PG1000	CP80	PG-PC	PC80
●	●	●	●	●	●	●

Command: **[F7 LOAD] [F6 EMSSYMB] <FNAME> [↵]**

Function: The SYMB `FNAME.EMS` is loaded for the default directory.

A path can be defined in with the file name. With the input of the following command in the command line, the SYMB `TEST.EMS` can be loaded from the directory `\PROJECT1` on the current drive:

[F7 LOAD] [F6 EMSSYMB] *\PROJECT1\TEST* [↵]

If no file name is defined or if the specified SYMB doesn't exist, all files with the extension "`.EMS`" in the default directories or the defined paths are displayed. The desired SYMB can be selected and loaded from the directory with the cursor and function keys (see [F1 PROG]).

All SYMBs that existed in memory before the load procedure are deleted from memory and replaced by the one selected.

A file with the extension "`*.EMS`" was saved until the SYMB editor was open. It can only be loaded in the PROgramming SYStem, if the SYMB editor is turned on, otherwise an error message is displayed:

```
E189  CANNOT LOAD SYMBOLS WITHOUT EMS!
```

[F8 ASCII] Load Files in ASCII-format

A	B	C	PG1000	CP80	PG-PC	PC80
		●	●	●	●	●

Command: **[F7 LOAD] [F8 ASCII] [F2 SYMBOLS] <FNAME> [F1 TO] <ADDR> [↵]**
[F7 LOAD] [F8 ASCII] [F3 DATA_TBL] <FNAME> [F1 TO] <LINE> [↵]

Function: This function offers the possibility of including SYMB lists and table data, that were created with external programming systems (CAD systems, editors, ...) in an existing program.

A path can be defined with the file name. With the input of the following command in the command line the SYMB TEST.ASY can be loaded from the directory \PROJECT1on the current drive and it can be inserted in the current input line of the SYMB editor:

[F7 LOAD] [F8 ASCII] [F2 SYMBOLS] *\PROJECT1\TEST* [F1 TO] <ADDR>[↵]

If no file name is defined, or if the specified LAD doesn't exist, all files with the extension “.ASY” (ASCII-SYMB) or “.ATA” (ASCII-TAB) in the default directories or in the defined path are displayed. The desired ASCII file can be selected from the directory with the cursor or function keys (see [F1 PROG]).

The load process is performed interactively, i.e.: the PROgramming SYStem checks every sentence of the SYMB or the TAB for logical errors and offers the possibility to the user, to immediately eliminate an error (within the SYMB or TAB editor) or to interrupt the load process with the [Esc] key.

Possible errors when loading an ASCII-SYMB:

- **SYMBOLIC NAME ALRADY USED FOR OTHER ADDRESS**
- **INVALID SYMBOLIC NAME**
- **INVALID ADDRESS**

Possible errors when loading an ASCII-TAB:

- **INVALID VALUE**
- **ASCII-STRING NOT MARKED WITH '**

4. DATA FORMAT OF ASCII-FILES

4.1. SYMB-DATA

File name **extension**: `*.ASY`

Data format:

The file is organized in “sentences”. Every sentence ends with control characters carriage return <CR> and line feed <LF> ($0D and $0A).

The first sentence can also be a mnemonic identification.
For every SYMB address **one** sentence is necessary.

Mnemonic identification:

Since for the address preselction in the B&R PROgramming SYStem different characters, according to the defined mnemonic, are valid, the identification must be stored in the file.

If the first sentence in the file is a “B&R<CRLF>”, the B&R address preselections are valid. If the file starts with a SYMB address, the PROgramming SYStem interprets the address preselections as MOTOROLA® mnemonics.

The PROgramming SYStem switches the mnemonic mode during the load procedure.

SYMB address sentence fomrat:

Short SYMB (LAD) is part of the Long SYMB (STL)

```
R0000 @Symbols xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx<CRLF>
```

If this SYMB is loaded, the short and long SYMB are as follows:

Short-SYMB: `Symbolx`

Long-SYMB: `Symbolxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx`

Short and long SYMB are specified separatly:

```
R0000 @ShortSYMB @Long
SYMBxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx<CRLF>
```

If this SYMB is loaded, the short and long SYMB are:

Short-SYMB: `Short SYMB`

Long-SYMB: `Long  SYMBxxxxxxxxxxxxxxxxxxxxxxxxxxxxx`

Example: The address sequence is not relevant:

```
B&R
R0000 @A        @Register 0
O 01F @S1.1     @Output x
I 000 @valve  @Pass open/closed
R0001 @         @Register 1
  :
```

4.2. TAB-DATA

File name **extension**: `*.ATA`

Data format:

The file is organized in “sentences”. Every sentence ends with control characters <CR> and <LF> ($0D and $0A).
For every TAB line **one** sentence is necessary.

Sentence format for a TAB line:

Comment line

```
*Commentline.......
```

Data line:

```
Data....  [*Comment]
```

The valid data formats are exactly listed in “chapter 7 TABLE-EDITOR”. If, for example, a table is created in the text editor, it is important to observe the data formats!

Example:

```
'ASCII-Data sentence 1','ASCII-Data sentence n'
1,2,$FF,'ASCII-Data sentence x'
*Comment line
14,23,%00110101    *Comment
15
*Comment line
  :
  :
```

5. DISPLAY DIRECTORY

A	B	C	PG1000	CP80	PG-PC	PC80
●	●	●	●	●	●	●

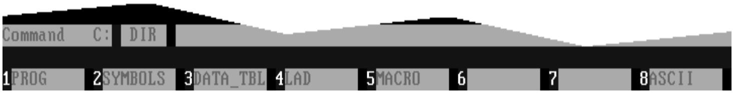
Command: **[F9] + [F6 DIR] <PATH> [↵]**
[F9] + [F6 DIR] <File type> <PATH> [↵]

Function: Display of directory conents.

Any directory path (\DIR1\DIR2\...\) including a drive name can be entered for <PATH>. Without the specification the current drive is accessed.

If no <PATH> is defined, the contents of the default directory are displayed.

Files with certain extensions can be displayed with the additional parameter <File type>. After the input of **[F9] + [F6 DIR]** the softkey line is changed:



```
Command   C: DIR
```

With these function keys one of the following extensions can be selected:

Extension	<File type>	Level		
		A	B	C
*.PRG	[F1 PROG]	●	●	●
*.SYM	[F2 SYMBOLS]	●	●	●
*.TAB	[F3 DATA_TBL]	●	●	●
*.LAD	[F4 LAD]	●	●	●
*.MSY	[F5 MACRO] [F2 SYMBOLS]	●	●	●
*.MTA	[F5 MACRO] [F3 DATA_TBL]		●	●
*.MSL	[F5 MACRO] [F5 S_LIST]			●
*.ASY	[F8 ACII] [F2 SYMBOLS]			●
*.ATA	[F8 ACII] [F3 DATA_TBL]			●

Example: Display of all programs in the directory \PROJECT1 on the disk in drive A:

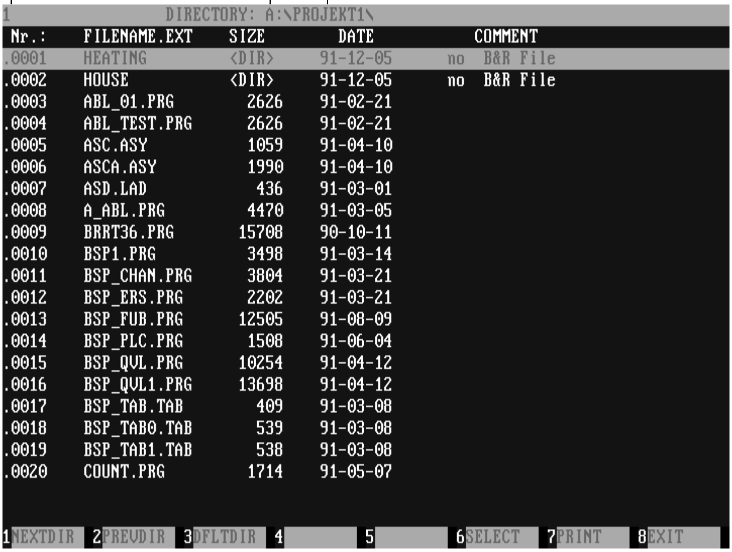
```
DIR PROG A:\PROJECT1\ [↵]
```

After terminating with the [↵] key the following picture is displayed:

Window number

Selected drive; if no drive is selected, the current dirve is accessed.

Path name



```
1                  DIRECTORY: A:\PROJEKT1\
Nr.:     FILENAME.EXT    SIZE        DATE          COMMENT
.0001    HEATING         <DIR>     91-12-05      no  B&R File
.0002    HOUSE           <DIR>     91-12-05      no  B&R File
.0003    ABL_01.PRG        2626    91-02-21
.0004    ABL_TEST.PRG      2626    91-02-21
.0005    ASC.ASY           1059    91-04-10
.0006    ASCA.ASY          1990    91-04-10
.0007    ASD.LAD            436    91-03-01
.0008    A_ABL.PRG         4470    91-03-05
.0009    BRRT36.PRG       15708    90-10-11
.0010    BSP1.PRG          3498    91-03-14
.0011    BSP_CHAN.PRG      3804    91-03-21
.0012    BSP_ERS.PRG       2202    91-03-21
.0013    BSP_FUB.PRG      12505    91-08-09
.0014    BSP_PLC.PRG       1508    91-06-04
.0015    BSP_QVL.PRG      10254    91-04-12
.0016    BSP_QVL1.PRG     13698    91-04-12
.0017    BSP_TAB.TAB        409    91-03-08
.0018    BSP_TAB0.TAB       539    91-03-08
.0019    BSP_TAB1.TAB       538    91-03-08
.0020    COUNT.PRG         1714    91-05-07

1NEXTDIR 2PREVDIR 3DFLTDIR 4         5         6SELECT  7PRINT   8EXIT
```

Key	FUNCTION
[F1 NEXTDIR]	If the cursor is located in a directory (marked <DIR>), this directory is opened. The contents of this dirctory are displayed. This function corresponds with the DOS-command "`cd <directory name>`".
[F2 PREVDIR]	The previous directory is changed and its contents are displayed. [F2 PREVDIR] corresponds with the DOS-command "`cd ..`".
[F3 DFLTDIR]	The current directory is defined as the "default-directory".
[F6 SELECT] **[↵]**	No function
[F7 PRINT]	The specified directory is printed.
[F8 EXIT] **[Esc]**	Exit to the command line.

Example: Print of a directory:

```
 1                DIRECTORY: A:\PROJECT\
  Nr.:    Filename.type        Size           Date         comment
  0001    HAUS          <DIR>     91-04-10      no B&R file
  0002    FREI            <DIR>    91-04-10      no B&R file
  0003    FLIESS          <DIR>    91-04-10      no B&R file
  0004    SOFTK.PRG        12924    90-10-11
  0005    BRRT36.PRG       15708    90-10-11
  0006    PPPROVIT.PRG     16977    90-10-11
  0007    PPPRO.PRG        16987    90-10-11
  0008    CP004.PRG         1271    90-10-12
  0009    CPRATE.PRG        1468    90-10-12
  0010    CP002.PRG         4353    90-10-12
  0011    PP002.PRG        22074    90-10-12
  0012    RATE.PRG         22423    90-10-12
  0013    PP004.PRG        24095    90-10-12
  0014    PPRATE.PRG       30164    90-10-12
  0015    MAT0.PRG          1646    90-10-22
  0016    NETW.PRG         24450    90-10-23
  0017    TI.PRG            3193    91-01-24
  0018    SEQC_01.PRG       2626    91-02-21
  0019    SEQC_TEST.PRG     2626    91-02-21
  0020    A_SEQC.PRG        4470    91-03-05
  0021    PIP.PRG          49377    91-03-05
  0022    LAD_BSP.PRG       5656    91-03-13
  0023    BSP1.PRG          3498    91-03-14
  0024    BSP_FBK.PRG       8771    91-03-15
  0025    CP_80BSP.PRG      1280    91-03-19
  0026    CP_40BSP.PRG      1280    91-03-19
  0027    BSP_ERS.PRG       2202    91-03-21
  0028    BSP_CHAN.PRG      3804    91-03-21
```

6. COPY FILES

A	B	C	PG1000	CP80	PG-PC	PC80
●	●	●	●	●	●	●

Command: **[F9] + [F5 COPY] [F7 DISK] <File type> <FNAME1> [F1 TO] <FNAME2> [↵]**

<File type> ... Description on page 8-19

Function: It is possible to copy functions with this function. The parameter <FNAME1> and <FNAME2> can also be included in a complete path name and drive name. If only a file name is specified, the defined file is searched or stored in the default directory of the current drive.

Example: The STL macro MACRO1 in the default directory on the current drive (hard disk) should be copied to drive A with the same name:

[F9] + [F5 COPY] [F7 DISK] [F5 MACRO] [F5 S_LIST] *MACRO1* [F1 TO] *A:\MACRO1* [↵]

7. RENAME FILES

A	B	C	PG1000	CP80	PG-PC	PC80
●	●	●	●	●	●	●

Command:

```
[F10] + [F5 RENAME] <File type> <FNAME1> [F1 TO] <FNAME2> [↵]
```

<File type> ... Description on page 8-19

Function: It is possible to give a file a different name with this function. The parameter <FNAME1> and <FNAME2> can also have a complete path name and a drive name. If only the file name is defined, the specified file name is searched for in the default directory of the current drive and replaced with the new name.

Attention: If the drive name and the path are completely entered, these definitions must be identical for both file names. Only the file name can be changed.

Example: To rename the STL macro MACRO1 on drive A to MACROX enter:

```
[F10] + [F5 RENAME] [F5 MACRO] [F5 S_LIST] A:\MACRO1 [F1 TO] A:\MACROX [↵]
```

8. DELETING FILES

A	B	C	PG1000	CP80	PG-PC	PC80
●	●	●	●	●	●	●

Command:

```
[F9] + [F7 DELETE] <File type> <FNAME> [↵]
```

<File type> ... Description on page 8-19

Function: It is possible to delete files with this function. The parameter <FNAME> can also have a comlete path name and a drive name. If only a file name is specified, the defined file is searched for and deleted from the default directory.

Example: To delete the STL macro MACRO1 on drive A enter:

```
[F9] + [F7 DELETE] [F5 MACRO] [F5 S_LIST] A:\MACRO1 [↵]
```

9. FORMAT DISKS

A	B	C	PG1000	CP80	PG-PC	PC80
●	●	●	●	●	●	●

Command:

```
FORMAT <drive name:> [↵]
```

Function: It is possible to format disks with this function.

Example: To format the disk in drive A enter:

```
FORMAT A: [↵]
```

CHAPTER 9
PRINTING

Contents:

1. CALL PRINTER MENU

The printer menu can be called up with various commands:

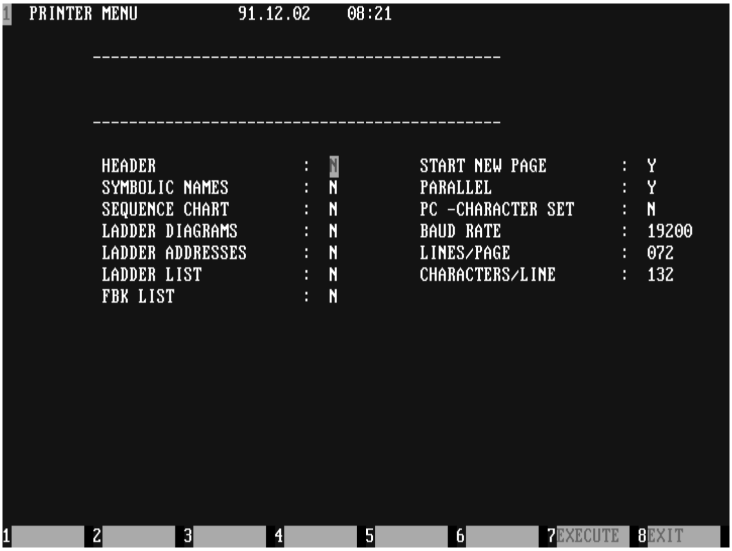
Command: ***PMEN*** **[↵]**
[F10] + [F1 PRINT] [↵]
[F10] + [F2 LIST] [↵]

After one of the commands listed above has been given, the printer menu appears.

2. STRUCTURE OF THE PRINTER MENU

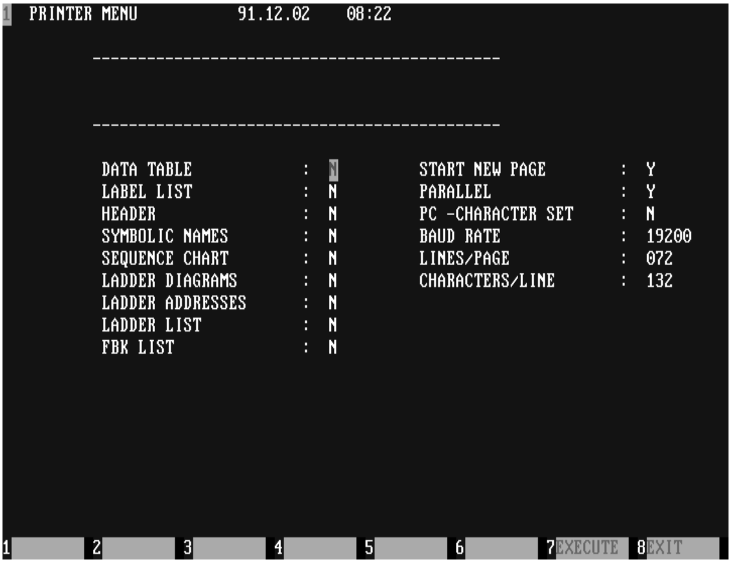
The structure of the printer menu depends on the level of the PROgramming SYStem. Levels A and B do not offer all options.

2.1. LEVEL A



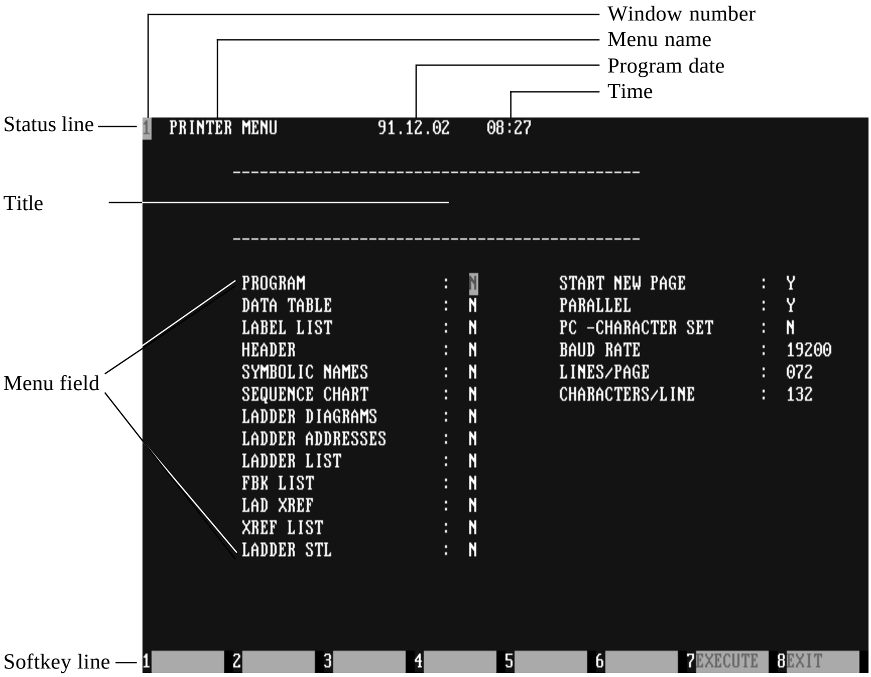
```
1  PRINTER MENU          91.12.02    08:21

          -------------------------------------------


          -------------------------------------------

           HEADER                : N         START NEW PAGE         :  Y
           SYMBOLIC NAMES        : N         PARALLEL               :  Y
           SEQUENCE CHART        : N         PC -CHARACTER SET      :  N
           LADDER DIAGRAMS       : N         BAUD RATE              :  19200
           LADDER ADDRESSES      : N         LINES/PAGE             :  072
           LADDER LIST           : N         CHARACTERS/LINE        :  132
           FBK LIST              : N
```

2.2. LEVEL B



```
1 PRINTER MENU          91.12.02    08:22

        --------------------------------------------

        --------------------------------------------

         DATA TABLE            :  N        START NEW PAGE        :  Y
         LABEL LIST            :  N        PARALLEL              :  Y
         HEADER                :  N        PC -CHARACTER SET     :  N
         SYMBOLIC NAMES        :  N        BAUD RATE             :  19200
         SEQUENCE CHART        :  N        LINES/PAGE            :  072
         LADDER DIAGRAMS       :  N        CHARACTERS/LINE       :  132
         LADDER ADDRESSES      :  N
         LADDER LIST           :  N
         FBK LIST              :  N

1        2        3        4        5        6        7EXECUTE 8EXIT
```

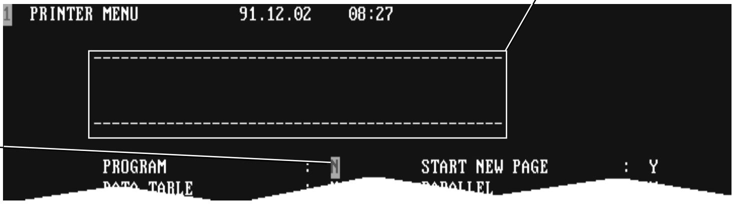
2.3. LEVEL C



```
1 PRINTER MENU          91.12.02    08:27

        --------------------------------------------

        --------------------------------------------

         PROGRAM               :  N        START NEW PAGE        :  Y
         DATA TABLE            :  N        PARALLEL              :  Y
         LABEL LIST            :  N        PC -CHARACTER SET     :  N
         HEADER                :  N        BAUD RATE             :  19200
         SYMBOLIC NAMES        :  N        LINES/PAGE            :  072
         SEQUENCE CHART        :  N        CHARACTERS/LINE       :  132
         LADDER DIAGRAMS       :  N
         LADDER ADDRESSES      :  N
         LADDER LIST           :  N
         FBK LIST              :  N
         LAD XREF              :  N
         XREF LIST             :  N
         LADDER STL            :  N

1        2        3        4        5        6        7EXECUTE 8EXIT
```

Title: A text can be entered in this field (e.g. project name, date, etc.). This text can be printed on every page as a heading.

After the PROgramming SYStem has been started this Text field is defined as follows by default:



```
1  PRINTER MENU          91.12.02    08:27

        ---------------------------------------------


        ---------------------------------------------

          PROGRAM              : N        START NEW PAGE        :  Y
          DATA TABLE           :          PARALLEL
```

blinking cursor

The cursor is moved to the desired position for text entry with the [❖] keys. The "-
"
characters can be replaced by any ASCII characters.

Up to 4 lines of 45 characters each can be entered.

Menu field: Various printing options can be set in this menu field:

- What is to be printed?
- Interface selection
- Selection of character set
- Layout

Softkey line: Functions available in the print menu:

[F7 EXECUTE]	Print/list according to parameters set.
[F8 EXIT] or **[Esc]**	The print menu is exited. The paramters and values set by the user are retained.

3. THE MENU FIELD

3.1. EDIT MENU FIELD

The cursor is moved to the desired position with the [❖] keys.

If the cursor is located in the text field for the title, all ASCII characters that can be represented can be entered. All of the functions for processing the text, like [BkSpc], [Ins], [Del] etc., are available.

If the cursor is located in a selection field, only certain keys are accepted:

[Y] The menu point is activated and the cursor is moved to the next selection field.

[N] The menu point is deactivated and the cursor is moved to the next selection field.

[↵] The menu point is not altered. The cursor is moved to the next selection field

[0] ... [9] Entry of decimal values where allowed (lines/page, characters/ lines).

[→] or **[↓]** The cursor is moved to the next selection field.

[←] or **[↑]** The cursor is moved to the previous selection field.

[Ctrl] + [↑] In the baud rate selection field the next higher value is set.

[Ctrl] + [↓] In the baud rate selection field the next lower value is set.

3.2. PRINT FROM ...

... Program

A	B	C	PG1000	CP80	PG-PC	PC80
		●	●	●	●	●

If this menu point is activated with [Y], the entire STL program is printed. The LAD translations are not printed.

STL lines containing NOP commands are not shown on the printout. The following line is inserted for every block of NOP lines:

“---” (see example)

Example:

```
 *** PROGRAM ***

00000       JSR      TEST  00130
00001       END
00002 *
00002 *
00002 -----------------------------------------
00011       BNE          DEC1  00008
00012       CLR                O 005
00013       LDD      #    00000
00014       STD           R 1000
```

(each of the lines from 2 to 9 contains the STL command NOP)

... Tables

A	B	C	PG1000	CP80	PG-PC	PC80
	●	●	●	●	●	●

If this menu point is activated with [Y], all of the tables in the program are printed out.

Example:

```
 *** DATA TABLE***

TABLE:TIN2

00000  2,                 * Node number 2
00001  3,                 * 19,200 baud
00002  1,0,0,0,0,0,0,0,    * Group 1
00003  0,                 * reception of node 0
```

... Symbol Tables

A	B	C	PG1000	CP80	PG-PC	PC80
	●	●	●	●	●	●

If this menu point is activated with [Y], all of the labels in the program are printed out.

- STL labels
- LAD names
- TAB names (marked with T)

Example:

```
 *** LABEL ***

$KST    S       $US2    S       R016  0062      R017  0086      R018  0098
R019  0109      R020  0093      R021  0125      R022  0114      R023  0111
CM0   0027      CM1   0033      CM2   0037      CMPA2  0022     CMPAH  0054
CMPAR  0039     DEC1  0008      DEC2  0012      DEC3  0016      DECD  0003
DECR  0021      FEK0  0043      FEK1  0049      FEKO  0040      FEKR  0053
TEST  0130      TIN2    T
```

The following can appear next to the label:

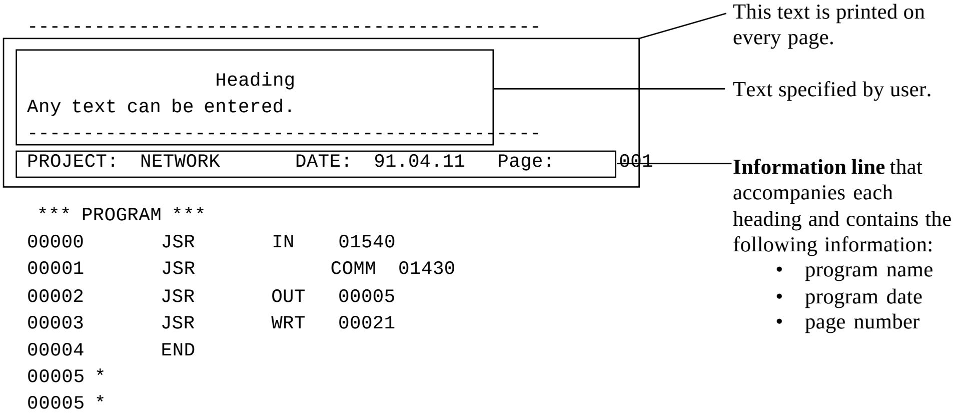
- xxxx Lines or number of program steps
- S System call
- T Table

... Title

A	B	C	PG1000	CP80	PG-PC	PC80
●	●	●	●	●	●	●

If this menu point is activated with [Y], the text defined in the text field of the print menu is printed as the heading on every page. In addition to the specified text, an information line is printed on every page and a table of contents is printed on the last page.

Example:



```
---------------------------------------------
                 Heading
Any text can be entered.
---------------------------------------------
PROJECT:  NETWORK       DATE:  91.04.11   Page:       001

 *** PROGRAM ***
00000       JSR       IN    01540
00001       JSR            COMM  01430
00002       JSR       OUT   00005
00003       JSR       WRT   00021
00004       END
00005 *
00005 *
```

This text is printed on every page.

Text specified by user.

Information line that accompanies each heading and contains the following information:

- program name
- program date
- page number

... Symbols

A	B	C	PG1000	CP80	PG-PC	PC80
●	●	●	●	●	●	●

If this menu point is activated with [Y], all addresses with SYMBs assigned to them are printed out with the symbols.

Example:

```
*** ASSIGNMENTS ***

      DISPL. 0       R0000 DISPLAY 0
      FBK ADDR.      R0864 & FBK DATA ADDR
      FBK DEST0      R0868 & FBK DEST 0
                        :
```

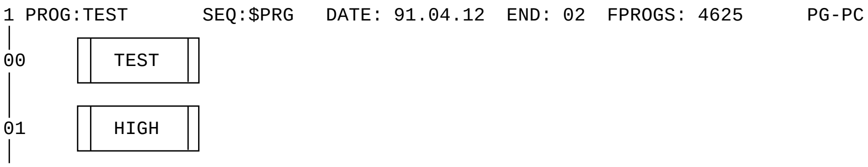
The SYMB for the LAD is to the left of the address, and the SYMB for the STL to the right.

... Sequence Charts

A	B	C	PG1000	CP80	PG-PC	PC80
●	●	●	●	●	●	●

If this menu point is activated with [Y], all of the sequence charts in the program are printed out.

Example:



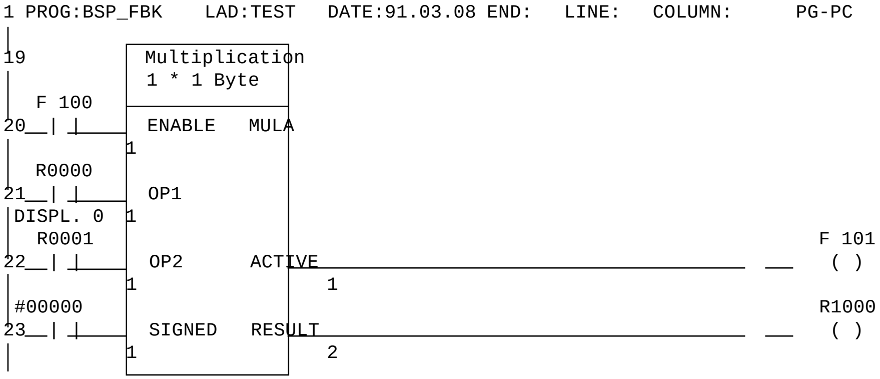
```
1 PROG:TEST      SEQ:$PRG   DATE: 91.04.12  END: 02  FPROGS: 4625     PG-PC
|
00      TEST
|
01      HIGH
|
```

... Ladder Diagrams

A	B	C	PG1000	CP80	PG-PC	PC80
●	●	●	●	●	●	●

If this menu point is activated with [Y], all of the ladder diagrams in the program are printed out.

Example:



```
1 PROG:BSP_FBK    LAD:TEST   DATE:91.03.08 END:   LINE:   COLUMN:      PG-PC
|
19          Multiplication
|           1 * 1 Byte
  F 100
20__| |___  ENABLE   MULA
|          1
  R0000
21__| |___  OP1
|DISPL. 0  1
  R0001                                                              F 101
22__| |___  OP2      ACTIVE___________________________________  ___   ( )
|          1              1
 #00000                                                              R1000
23__| |___  SIGNED   RESULT___________________________________  ___   ( )
|          1              2
```

... LAD Addresses

A	B	C	PG1000	CP80	PG-PC	PC80
●	●	●	●	●	●	●

[Y] In LADs, addresses are printed above the ladder and SYMBs below it.

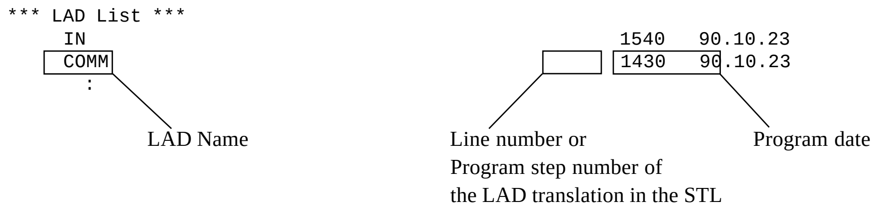
[N] In LADs only the SYMBs above the ladder are printed out. If there is no SYMB defined, the address is printed (see SYMB mode in "Symbols Editor - Chapter 4).

... LAD List

A	B	C	PG1000	CP80	PG-PC	PC80
●	●	●	●	●	●	●

If this menu point is activated with [Y], a list of all of the LADs in the program is printed out.

Example:



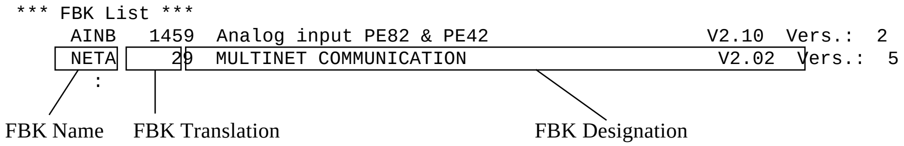
```
*** LAD List ***
     IN                                                1540   90.10.23
     COMM                                              1430   90.10.23
       :
```

... FBK List

A	B	C	PG1000	CP80	PG-PC	PC80
●	●	●	●	●	●	●

If this menu point is activated with [Y], a list of all of the FBKs in the program is printed out.

Example:



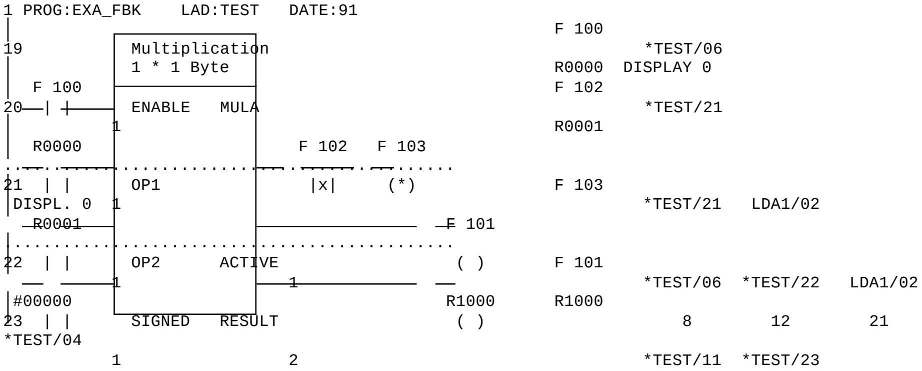
```
*** FBK List ***
     AINB   1459  Analog input PE82 & PE42                V2.10  Vers.:  2
     NETA     29  MULTINET COMMUNICATION                   V2.02  Vers.:  5
       :
```

... LAD - XREF

A	B	C	PG1000	CP80	PG-PC	PC80
		●	●	●	●	●

When activating this menu with [Y] all existing address cross references in this line are listed to the next LAD line.

Example:



```
1 PROG:EXA_FBK    LAD:TEST   DATE:91
|                                                       F 100
19         | Multiplication                                     *TEST/06
|          | 1 * 1 Byte  |                              R0000  DISPLAY 0
|  F 100   +-------------+                              F 102
20--| |----| ENABLE   MULA                                      *TEST/21
|          1             |                              R0001
|  R0000   |             |    F 102   F 103
..--.------..............---.------.--......
21  | |    | OP1         |     |x|     (*)              F 103
|DISPL. 0  1             |                                      *TEST/21   LDA1/02
 -R0001----|             |----------------  -F 101
|..........................................
22  | |    | OP2      ACTIVE                ( )         F 101
  --  -----1             1----------------  --                  *TEST/06  *TEST/22   LDA1/02
|#00000    |             |                   R1000      R1000
23  | |    +-SIGNED---RESULT                ( )                     8        12         21
*TEST/04
           1                 2                                  *TEST/11  *TEST/23
```

The LAD-XREF is only printed if the menu item “ladder diagrams” is activated.

Since the LAD-XREF is printed next, on the right to the LAD, a printer that can print at least 132 characters / line is required. The corresponding font (e.g.: condensed font) must be set directly on the printer.

To print a LAD-XREF a LAD is divided into blocks. A block consists of all components, that are directly connected with each other by a connecting line (vertical and horizontal). Comments are assigned to the subsequent block, except a comment still starts within the lower block limit. A space remains between the blocks.

The LAD-XREF contains all addresses with their long SYMBs, that are used in a block. All address locations are given in the entire program, on which the memory location is changes. The following references are possible:

- STL line number
- LAD name / line number

The LAD-XREF next to a block is divided in maximal 3 parts, that are separated from each other by a dotted line:

```
                                      P A R T   1
                                      ...............................................
      LAD block                       P A R T   2
                                      ...............................................
                                      P A R T   3
```

Example explanation:

Part 1: In this section all memory location addresses of which the contents is loaded, are listed, (load command: all contacts with the angular contact ~~sym~~bol:——
| |).

The contents of the following memory location is loaded:
F 100, R0000, F 102, R0001

Part 2: In this section all memory location addresses are listed, that are present in a FBK (internal memory locationds) and of which the contents within a block is changed (interm~~edia~~te~~ sto~~rage with address specification: contacts with the following symbol: (*)).

F 103 in changed in the following ladder diagrams:
*TEST / line 21
LAD1 / line 2

Part 3: In this section all output addresses (contacts in the last column) and locations, where they were changed, are listed.

F 101 is changed in the following ladder diagrams:
*TEST / line 6 and 22
LAD1/ line 2

R1000 is changed in the following STL lines: 8, 12, 21,
are changed in the following ladder diagrams:
*TEST / line 4, 11 and 23

All LADs that are marked with a “*”, are references to the corresponding LAD.

... Cross Reference List

A	B	C	PG1000	CP80	PG-PC	PC80
		●	●	●	●	●

When activating the menu item with [Y] the cross reference list is printed. This list consists of two parts:

1) Address list with specification of the following data:
2) Label list with specification of the following data:

1) Cross reference list:

```
 *** CROSS REFERENCE LIST***

YXXX  SYMB-SHORT SYMB-LONG....
         Reference 1     Reference 2     Reference 3     ...
  :
YXXX
  :
```

YXXX Address with address preselection and address part
SYMB-SHORT short symbols for LADs
SYMB-LONG long symbols for STL
Reference At this location it is listed, at which position a program address exists and how it is accessed (Load or Storage instuction, Contact type, ...).

- LN STL instruction
  The address is used together with the specified STL instruction in the STL line LN .

- e.g.: NAME/LAD-LN Contact type
  The address is used in the ladder diagram LNAME in the line LN. The address is used with the specified contact type.

- e.g.: NAME/LAD-LN >>>>
  The address is used in the ladder diagram NAME in the line LN within a FBK (internal memory).

Example:

```
*** CROSS REFERENCE LIST ***

R0000  DISPL. 0   DISPLAY 0
         LAD1/04  ¦─¦    LAD2/04 ─¦/¦    LAD3/04  ¦ ¦
R0100
         LAD1/02  ¦─¦    LAD2/02 [ADR
R0300
          LAD4/03 >>>>
R0864  FBK ADDR.  & FBK DATA ADDR
            38 =R           93 LR
F 100  FLANK+    POSITIVE FLANK
         LDA1/02  ¦+¦
```

2) Label Cross Reference List:

```
 *** LABEL CROSS REFERENCE LIST ***

LABL  LN/*TAB*
         Reference 1     Reference 2     Reference 3     ...
LABL
  :
```

`LABL` Label used in the program

`LN` STL line, that is assigned to the label

`*TAB*` Label is a table name

`Reference` The program location is specified where a label is used and how it is accessed.

- e.g.: `STL LN  STL instruction`
  The label is called with the specified STL instruction in the STL line.
- e.g.: `NAME/LAD-LN FBK`
  A FBK that corresponds with the label name, is called up in the ladder diagram NAME, line LN.
- e.g.: `SEQC/LN  LAD`
  The LAD, that corresponds with the label name, is called in the Sequence chart SEQC, line LN.
- e.g.: `NAME/LN  >LTN`
  The label exists within a FBK - in the LAD LNAME , line LN (**L**abel-, **T**able- or **N**ame input in the FBK).

Example:

```
 *** LABEL CROSS REFERENCE LIST ***

ADR1 *TAB*

CMPH    38
          LAD1/01 FBK     LAD2/01 FBK     LAD3/01 FBK
CMPR    23
             10 BEQ          16 JMP          20 JMP
LDA1   117
          $PRG/00 LAD
MIN    212
          LAD1/25 >LTN
NEW    158
          LAD2/00 >>      LAD2/35 >>
```

... Print LAD Code

A	B	C	PG1000	CP80	PG-PC	PC80
		●	●	●	●	●

While activating the menu item with [Y] all STL transfers of all LADs are printed. While printing the LAD Code, the function “program” must also be selected with [Y].

3.3. ADDITIONAL MENU OPTIONS

A	B	C	PG1000	CP80	PG-PC	PC80
●	●	●	●	●	●	●

Page beginning

The option activation with [Y] effects a page feed after the print of the selected menu item.

Line / Page

At this location the maximum number of lines per page is entered. According to the setting of this value, the PROgramming SYStem formats the printer output.

Default setting: 72 lines/ page

Characters / Line

At this location the maximum number of characters per line is entered. According to the setting of this value, the PROgramming SYStem formats the printer output.

Defualt setting: 132 characters / line

Parallel

[Y] The print comes from the paralle interface LPT1.
[N] The print comes frome the serial interface COM1 corresponding with the set baud rate.

Baud Rate

If the cursor is in the input field for the baud rate, the desired baud rate for the serial interface can be set with the key combination [Ctrl] + [↑] or [↓]. The following baud rates are possible:

300 baud
600 baud
1200 baud
2400 baud
4800 baud
9600 baud
19200 baud

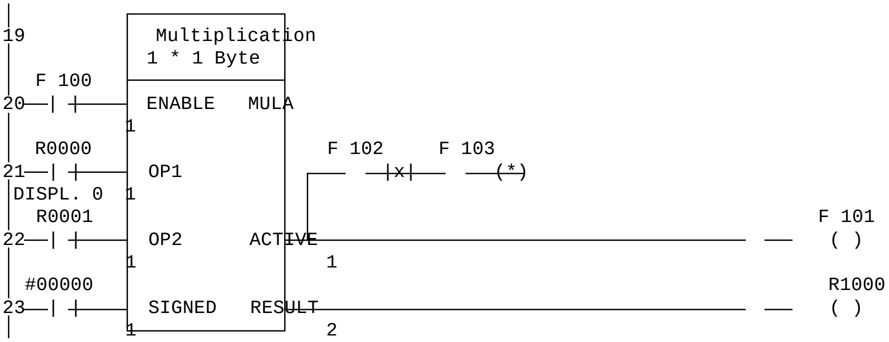
PC Output Record

[Y] The PC output record is applied.
[N] The ASCII output record is used.

Example: ASCII output record

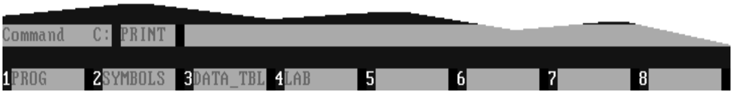
```
!           +---------------+
19          !Multiplication !
!           !1 * 1 Byte     !
!  F 100    I---------------I
20--I I-----IENABLE   MULA  !
!          1!               !
!  R0000    !               !    F 102     F 103
21--I I-----IOP1            ! +---IxI-------(*)----
!DISPL. 0  1!               ! !
!  R0001    !               ! !                                          F 101
22--I I-----IOP2      ACTIVEI-+------------------------------------------( )--
!          1!               !1
!#00000     !               !                                            R1000
23--I I-----ISIGNED   RESULTI--------------------------------------------( )--
!          1+---------------+2
```

Example: PC output record



4. PRINTING PROGRAMS OR PROGRAM PARTS

For the printing process the printer menu doesn't necessarily need to be called up. After the input of [F10] + [F1 PRINT] the sofkeyline is changed:



```
Command   C: PRINT
```

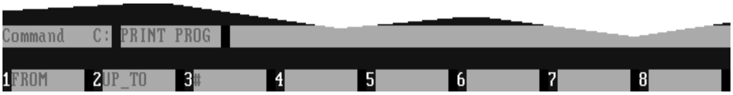
If [↵] is pressed immediately after the input of [F10] + [F1 PRINT], the printer menu is called up.

[F1 PROG]

A	B	C	PG1000	CP80	PG-PC	PC80
●	●	●	●	●	●	●

Printing a STL program or STL program parts. The STL transfer from LADs is not printed out.

After the input of [F1 PROG] the user is asked to set the program range, that should be printed. The softkey line is changed:



```
Command   C: PRINT PROG
```

Input	Print range
[↵]	The entire STL program is printed.
***FROM* <LN/LB> *TO* <LN/LB> [↵]** **[F1 FROM] <LN/LB> [F2 To] <LN/LB> [↵]**	The specified range is printed.
***TO* <LN/LB> [↵]** **[F2 TO] <LN/LB> [↵]**	The STL range beginning with the current input line up to the specified program line <LN/LB> is printed.
***FROM* <LN/LB> *NUM* <NR> [↵]** **[F1 FROM] <LN/LB> [F3 NUM] <NR> [↵]**	Beginning with the specified line <LN/LB> <NR> lines are printed.
***NUM* <NR> [↵]** **[F3 NUM] <NR> [↵]**	Beginning with the current input line <NR> lines are printed.
A line number or a label is to be entered for <LN/LB>.	

After pressing [↵] the specified range is printed out. When the print procedure is finished, the message "`List / Print ended`" is displayed on screen. The cursor is positioned in the command line again after pressing any key.

[F2 SYMB]

A	B	C	PG1000	CP80	PG-PC	PC80
●	●	●	●	●	●	●

Printing the SYMBs or a SYMB range.

After the input of [F2 SYMB] the user is asked to set a SYMB range, that should be printed. The softkeyline is changed as with [F1 PROG].

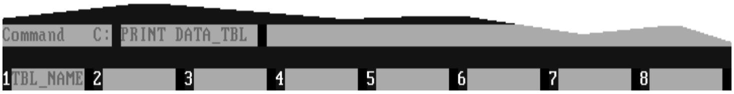
The print range definition is the same as with [F1 PROG]. Instead of <LN/LB> an address <ADR> is entered.

[F3 TAB]

A	B	C	PG1000	CP80	PG-PC	PC80
	●	●	●	●	●	●

Table print-out.

After the input of [F3 TAB] the user has to enter the TAB, that should be printed. The softkeyline is changed:



```
Command   C: PRINT DATA_TBL
```

If no TAB name is entered, all existing tables are printed.

[F4 LAB]

A	B	C	PG1000	CP80	PG-PC	PC80
●	●	●	●	●	●	●

Print-out of a list of all labels used in the program.

This list corresponds with the symbol table in the printer menu.

5. DISPLAY ON SCREEN

A	B	C	PG1000	CP80	PG-PC	PC80
●	●	●	●	●	●	●

5.1. GENERAL INFORMATION

The output speed on screen can be changed with the keyboard during the output:

[L]	Slow output speed
[Any key]	normal output speed
[W]	Stopping the output on screen
[Any key]	Continuing the output on screen

5.2. OUTPUT OF PROGRAMS AND PROGRAM PARTS

All outputs, that are possible on a printer, can also be displayed on screen. The instruction **[F10] + [F2 LIST]** is equivalent to the already explained ***PRINT*** command.

After the input of **[F10] + [F2 LIST] [↵]** the printer menu is displayed. With the selection of the desired items in this menu the program or program parts are printed.

The following menu item settings do not influence the display on screen:

- page start
- parallel (interface)
- PC output record (only the PC output record is used, not the ASCII output record)
- baud rate
- character / page

Also the screen display of program parts and the definition of the print range with the commands ***LIST PROG***, ***LIST SYMB***, ***LIST TAB*** and ***LIST LAB*** are handled the same as the ***PRINT*** command.

5.3. CROSS REFERENCE LIST

Command: **XREF [↵]**

Function: Cross refernce output on screen.

This function corresponds to the menu item cross reference in the printer menu. The output of cross reference list on screen with the command `LIST [↵]` is identical to the command `XREF [↵]`.

6. “PRINTING” IN A FILE

A	B	C	PG1000	CP80	PG-PC	PC80
●	●	●	●	●	●	●

It is possible to transfer all print commands, that refer to a printer in a file. This can not be done directly from the PROgramming SYStem. It must be defined before the program start by a MS-DOS environment vaiable.

Before the start of a B&R-PROgramming SYStem the following must be entered in the MS-DOS input line:

```
C:\>SET B&R$PRINTFILE=<file name> [↵]
```

`<file name>` is the name of the file, in which all print outputs occur. An entire path, including the disk drive specification, can be entered.

Example:

```
C:\> SET B&R$PRINTFILE=C:\OUTPUT\PRINTER.PRN [↵]
C:\> PG [↵]
```

All printer outputs go in the file `PRINTER.PRN` on drive `C` in the directory `\OUTPUT`.

To delete this environment variable, the following command is entered to the MS-DOS input line:

```
C:\>SET B&R$PRINTFILE= [↵]
```

The environment variable `B&R$PRINTFILE` is deleted.

7. DEFINE INTERFACE

A	B	C	PG1000	CP80	PG-PC	PC80
●	●	●	●	●	●	●

The selection of the interface and baud rate for a serial interface is described in 3.3. ADDITIONAL MENU OPTIONS. The setting can also be performed with the commands in in the command line:

- ***PAR* [↵]**
  Selection of a parallel interface LPT1 as printer interface.
- ***SER* [↵]**
  Selection of a serial interface COM1 as printer interface.
- ***BAUD* <Baud-Rate> [↵]**
  Creating the baud-rate for the serial interface.

  The following values can be entered for <baudrate> :

  300, 600, 1200, 2400, 4800, 9600, 19200

CHAPTER 10

ESTABLISHMENT OF FUNCTION BLOCKS

Contents:

1. GENERAL INFORMATION

A	B	C	PG1000	CP80	PG-PC	PC80
		●	●	●	●	●

An extensive collection of standard functions blocks is available at B&R. These standard FBKs support the user in program creation and shorten software development time.

In expansion level C there is additionally the possibility of defining and programming new function blocks with the FBK editor. The user can create individual, problem specific function blocks.

In the first step the block form is specified - the number of inputs and outputs. In the second step the FBK function is defined with a STL program.

The B&R standard software collection includes efficient function blocks, that are divided in clearly arranged groups according to their function.

Group	**Contents**
1. Hardware	FBKs for a software operation of modules.
2. Utilities	Useful programming devices (shift register, drum sequencer ...)
3. System	FBKs for the application of operating system mathematic.
4. Operating panels	FBKs for Input/Output programs for operator panels as well as the operation of the graphic panels.
5. Printer	Printer activation FBKs.
6. MININET	FBKs to operate a Master-Slave Network.
7. MULTINET	FBKs to operate a Token Ring Network.
8. PID-controller	PID-controller FBKs
9. Positioning	Positioning FBKs
10. Communication	FBKs for the coupling with other systems (Siemens, Modbus, A-B, ...)
11. Device driver	Support of peripheral devices (BRMEC).
12. PROVIT	FBKs for visualization programs with PROVIT terminals

There are several software packages available:

Package	**Group**														
	1	2	3	4	5	6	7	8	9	10-1	10-2	10-3	10-4	11	12
1	●	●	●												
2	●	●	●	●	●									●	●
3	●	●	●					●							
4	●	●	●						●						
5	●	●	●			●	●								
6	●	●	●							●					
7	●	●	●								●				
8	●	●	●									●			
9	●	●	●										●		

Model numbers:

Package	**Model number**	
	German	English
1	SWSPSSTD01-x	SWPLCSTD01-x
2	SWSPSSTD02-x	SWPLCSTD02-x
3	SWSPSPID01-x	SWPLCPID01-x
4	SWSPSPOS01-x	SWPLCPOS01-x
5	SWSPSCOM01-x	SWPLCCOM01-x
6	SWSPSDRV01-x	SWPLCDRV01-x
7	SWSPSDRV02-x	SWPLCDRV02-x
8	SWSPSDRV03-x	SWPLCDRV03-x
9	SWSPSDRV04-x	SWPLCDRV04-x

x = 0 Software on 3.5" Diskettes
x = 1 Software on 5.25" Diskettes

The SOFTWARE MANUAL (2 volumes) is delivered with every package.

2. FBK PROGRAMMING BASICS

2.1. PREPARATIONS

Before a new FBK is defined and programmed, it is necessary to design it first on paper with all inputs and outputs **exactly**, to avoid subsequent changes and redefinition of a created FBK.

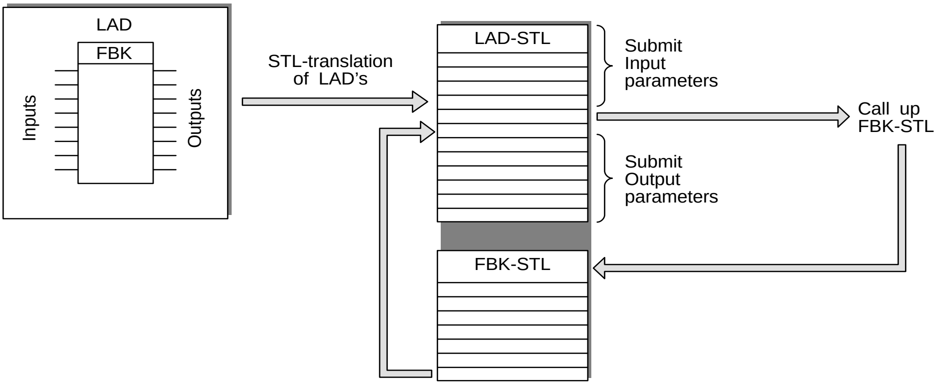
2.2. FBK FUNCTION

A FBK has been referred to as a so called “Black-Box”. The user knows the tasks of a FBK (Software-description) and which functions its inputs and outputs have and which data needs to be connected. The function of this “Black-Box” is defined in the STL subprogram. This STL program processes the inputs and changes the outputs according to the input data.

A FBK is included in LAD [1], in which the inputs and outputs are connected. This part of the ladder diagram is translated to the STL:

- Input submittal
- FBK program call
- Output submittal

Schematic structure of the STL translation:



[1] Chapter 4, section 6. Function plan programming

2.3. INTERNAL MEMORY LOCATIONS

A FBK uses, according to the input and output definition in the FBK editor, internal and/or private memory locations. In this memory location the input parameters are submitted to the FBK program which also returns the output parameters to the same memory location.

The internal memory location is in the system memory of the PLC, that can not be accessed by the B&R address preselections.

Input parameter submittal: Before the call of a FBK- STL, ...

... the address of the internal memory range is determined.
... the input parameters are stored in this memory range (①).
... the start address is put onto the stack.
... the start address of this memory range is filed in index register X.

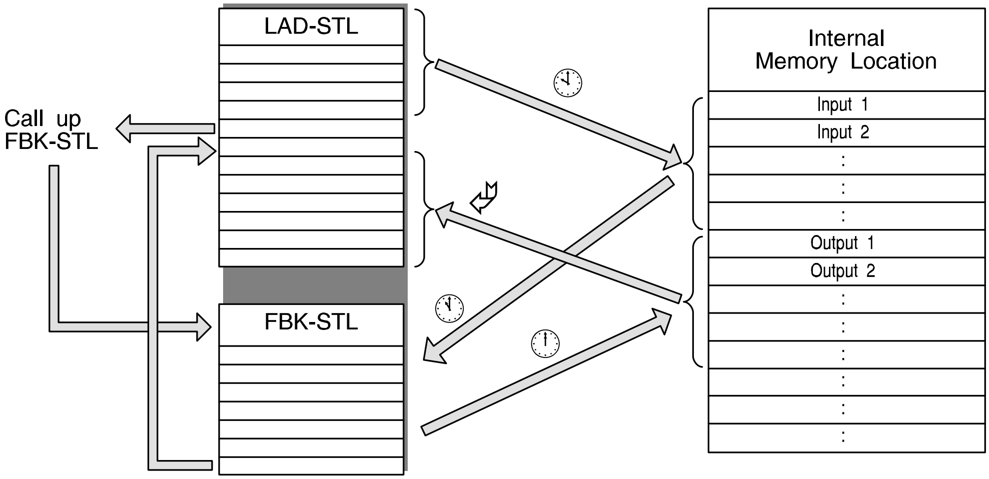
FBK-STL call:

The FBK program can now read and work using the indicated addressing of the input parameter (②) and store with the indicated addressing of the output parameter (③).

Output parameter addressing: After the call of a FBK-STL, ...

... the start address is taken from the stack.
... the output parameters are loaded from the internal memory range (④) and assigned to the outputs.

Schematic structure:



2.4. PRIVATE MEMORY LOCATIONS

The private memory locations are in the PLC user memory range. These are memory locations, that the user can access with the B&R address preselections (e.g.: R, F, S, T, ...). These **private** memory locations should not be overwritten by other program parts.

The parameters that is defined by the user is submitted to this memory location, (FBK-definition editor). No internal memory locations are required.

The submittal of input/output parameters is the same as the submittal of internal memory locations.

Note: **Private** and **internal** memory locations can be mixed with the definition of the inputs and outputs. I.e. the parameter submittal ensues according to the FBK definition either in private or internal memory locations.

3. FBK EDITOR CALL

Command: **`[F4 FBK] <NAME> [↵]`**

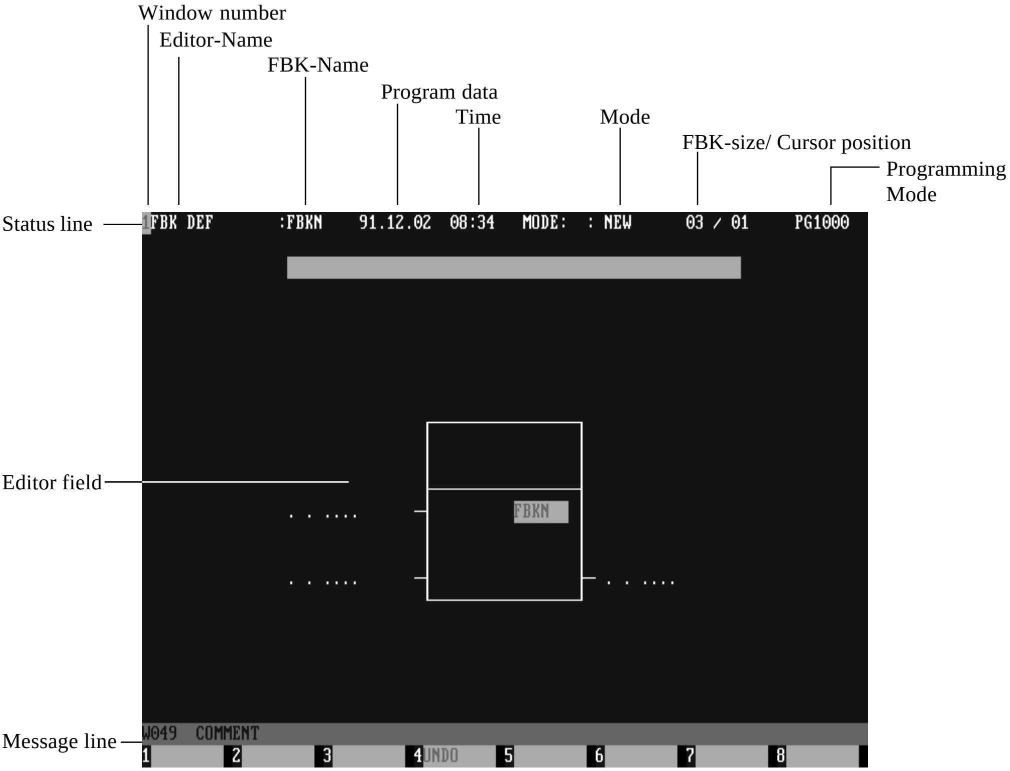
<NAME> may have a maximum length of 4 characters and has to start with a letter.

If no name is entered, the FBK definition menu of the last processed FBK is displayed. If no FBK is defined, the following message is displayed:

```
E174  DEFAULT NAME NOT DEFINED OR WRONG
```

After terminating with [↵] the FBK definition menu is displayed.

4. MENU STRUCTURE



```
1FBK DEF       :FBKN    91.12.02  08:34   MODE:  : NEW     03 / 01     PG1000

                              FBKN
   . . ....                                         
   . . ....                                      . . ....

W049  COMMENT
1         2         3         4UNDO     5         6         7         8
```

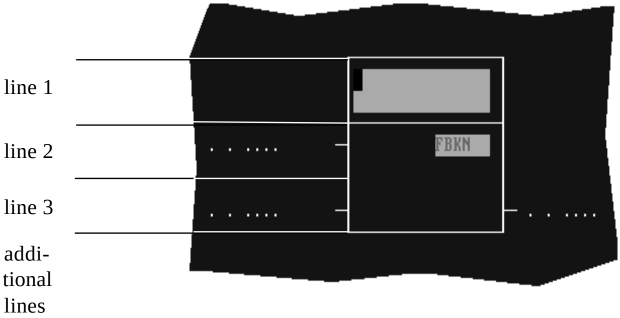
Status line: In this line different information is displayed.

Mode: NEW — This information is displayed if a FBK is newly defined.

CHANGE — This information is displayed if an existing FBK is changed.

The FBK consists of 3 lines.
The cursor is located in line 1.

FBK size / cursor position: e.g.: **03/01**



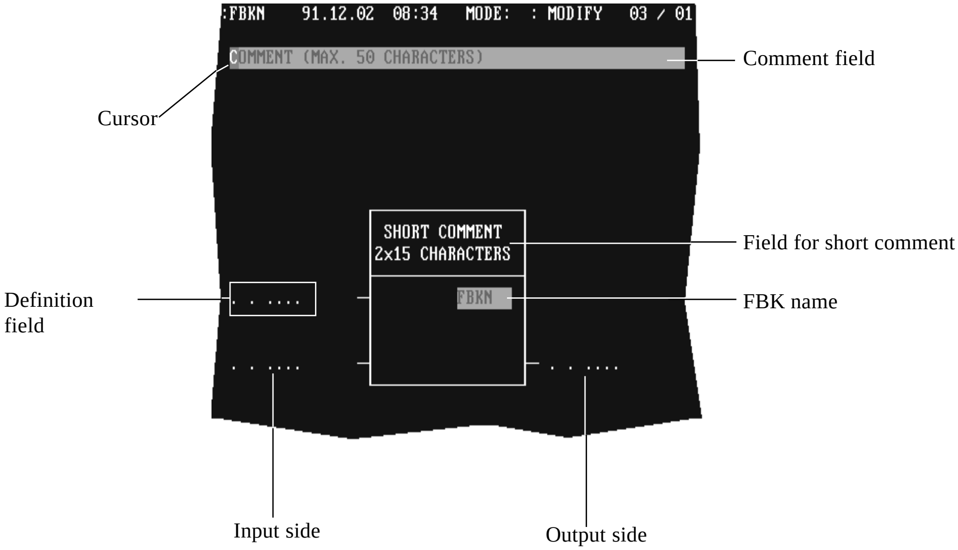
Editor field: In this field the FBK, to be established, is defined:

- comment and short comment (short statement about the FBK function)
- input page (number and size of the inputs)
- output page (number and size of the outputs)

Message line In the message line messages concerning user errors or messages requiring the input of parameters are displayed.

5. FBK DEFINITION

If a new FBK is defined, a standard FBK with two inputs and one output is displayed after calling it up with ***FBK* <NAME> [↵]**.



```
:FBKN    91.12.02  08:34   MODE:  : MODIFY   03 / 01

COMMENT (MAX. 50 CHARACTERS)

                    SHORT COMMENT
                    2x15 CHARACTERS
. . ....                      FBKN

. . ....                                 . . ....
```

5.1. COMMENT FIELD

The cursor is in the comment field after the call of the FBK editor. Any text can be entered in this comment field (a maximum of 50 characters), that briefly describes the FBK functions.

This comment line is displayed in the FBK overview (when loading FBKs) next to the FBK names and is printed out with the FBK list.

After pressing the [↵] or [↓] key the cursor is positioned in the field for the short comment.

5.2. SHORT COMMENT

Any short piece of text (2 lines with 15 characters each) can be entered in this field, similar to the comment field, which briefly describes the FBK functions.

This text is displayed in the LAD together with the FBK (**only in PC-mode!**).

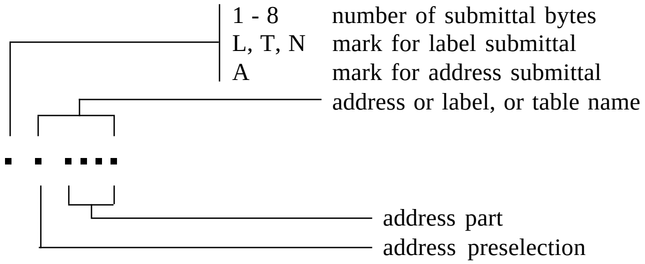
After pressing the [↵] or [↓] key the cursor is positioned at the first input.

5.3. FBK NAME

The FBK name is displayed in inverse video in this field. This name corresponds to the label of the subprogram pertinent to the FBK.

5.4. DEFINITION FIELD

Every Input/Output is assigned to a definition field, that contains information on the values, that are to be committed:


1 - 8   number of submittal bytes
L, T, N   mark for label submittal
A   mark for address submittal
address or label, or table name
address part
address preselection

5.4.1. Submittal of a certain number of bytes

A	B	C	PG1000	CP80	PG-PC	PC80
		●	●	●	●	●

At the **first** position of the definition field is a value **b**, that indicates, how many bytes are submitted in a input/output.

Address preselection or address don't need to necessarily be specified. Possible definitions for inputs/outputs are:

`b . ....` A **b**-byte value is submitted to the internal memory range.

`b v ....` A **b**-byte value is submitted to the memory range. This value is stored in a memory location with the address preselection **v** and its complete address is entered in the LAD during the FBK call.

`b v aaaa` A **b**-byte value is submitted to memory location **v aaaa**. No internal memory is used.

address selection **v**	address **aaaa**	PG1000	CP80	PG-PC	PC80
.	....	I/O	I/O	I/O	I/O
I					
O					
R	0000 - 7167	I/O	I/O	I/O	I/O
F	000 - 799	I/O	I/O	I/O	I/O
S	000 - 3E7	I/O	I/O	I/O	I/O
T	000 - 3E7	I/O	I/O	I/O	I/O
P					
G	0000 - 4095				I/O
B	0000 - 4095				I
#	00000 - 65535 %0 - %111 1111 $0000 - $FFFF			I	I

I Address preselection in only possible for the input page
O Address preselection is only possible for the output page

If an immediate-value (#) is distributed in the FBK definition, nothing can be connected at this input. The defined value is exclusively submitted to the FBK program.

5.4.2. Characteristic for the address submittal

A	B	C	PG1000	CP80	PG-PC	PC80
		●			●	●

If the letter “A” is in the **first** position of the definition field, an address can be submitted for the FBK program with this input. For this address submittal 2 bytes of the internal memory range are always used.

The address preselection must be defined, to exit the definition field. Possible input definitions are:

`A v ....` The memory location address with the address preselection **v** is submitted to the FBK program. The complete address of this memory location must be entered in the LAD during the FBK call.

`A v aaaa` The memory location address **v aaaa** is submitted to the FBK program.

address selection v	address **aaaa**	PG1000	CP80	PG-PC	PC80
.					
I	000 - 3FF			I	I
O	000 - 3FF			I	I
R	0000 - 7167	I	I	I	I
F	000 - 799	I	I	I	I
S	000 - 3FF	I	I	I	I
T	000 - 3FF	I	I	I	I
P	000 - 3FF		I	I	I
B					I
G					I
#					

I Address preselection is only possible for the input page

> No contact or path can be connected with this input. The predefined address **v aaaa** is submitted to the FBK program. The addressing ensues either entirely in the FBK definition or partially during the call of a FBK in a LAD.

5.4.3. Characters used for submitting Labels

A	B	C	PG1000	CP80	PG-PC	PC80
		●			●	●

If the letter “L”, “N” or “T” is in the **first** position of the definition field, the label address can be submitted to the FBK program with this input. 2 bytes are always used for the submittal.

The following possibilities are available, to submit an input to the label. Either “L”, “N” or “T” can be entered instead of “x”:

x The label address is submitted to the FBK program in the internal memory range. The label name must be entered in the LAD when calling the FBK.

x > The label address is submitted to the FBK program in the internal memory range. A label contact must be connected to this input in the LAD. Other contacts are not permitted by the PROgramming SYStem.

x NAME The label address NAME is submitted to the FBK program in the internal memory range. NAME is specified by the FBK definition and can not be changed any more, when calling the FBK in the LAD. NAME may have a maximum length of 4 characters. The first character must be a letter.

“L”, “T” and “N” have the same function. They can be applied by the user for the structuring of the program. E.g.:

L	Labels of subprograms, that are called by the FBK program.
T	Labels of tables, that are used by the FBK program.
N	Other labels.

> At inputs “**x**” and “**x NAME**” no label contact can be connected. The predefined label **NAME** is submitted to the FBK program. .The label definition ensues either completely in the FBK definition or when calling a FBK in a LAD.

5.5. INPUT/OUTPUT NAMES

Every input is provided with a name field, where any text can be entered. If the cursor is moved in the FBK with the [❖]-keys, one of the name fields can be entered. The name can be up to 4 or 6 characters long, depending on the programming mode:

PG1000	CP80	PG-PC	PC80	Name
●	●			4 characters
		●	●	6 characters

5.6. NUMBER OF INPUTS/OUTPUTS

If a new FBK is defined, a standard FBK with two inputs and one output is displayed after the call of the FBK editor. The number of inputs /outputs can be changed with the following commands:

[Ctrl] + [Ins] Inserts a line (1 input and 1 output)
A FBK can have a maximum of 48 inputs and 47 outputs.

[Ctrl] + [Del] Deletes a line (1 input and 1 output).

The cursor must be at the position, where a line should be inserted or deleted.

6. EXITING THE FBK EDITOR

[Esc] Exits the editor in the command line

[Ctrl] + [Esc] Exits the editor in the command line without consideration of any changes made since the beginning.

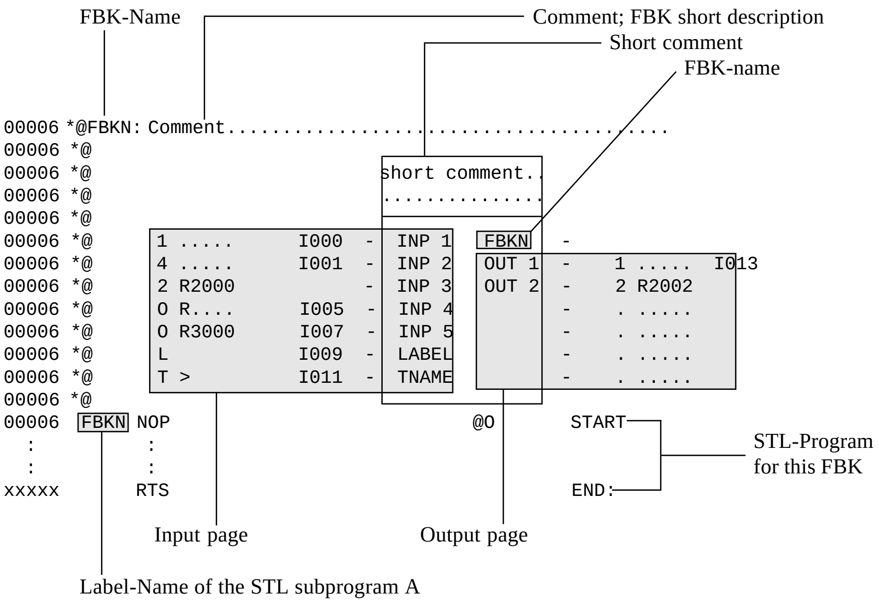
After exiting with [Esc] the PROgramming SYStem creates or changes the FBK header in the STL. This FBK header contains all information, that is required by the user for the creation of a FBK program.

7. THE STL PROGRAM OF A FBK

The STL program of a FBK includes a header and the pertinent STL program. All information about the inputs and outputs, as defined in the FBK editor, is located in the header.

The STL subprogram of the FBK is to be inserted with the label **FBKN**. The input/output parameter must be submitted according to the information given in the FBK header. Index register X contains the start address of the range in system memory, to which the parameters are submitted.

Example:



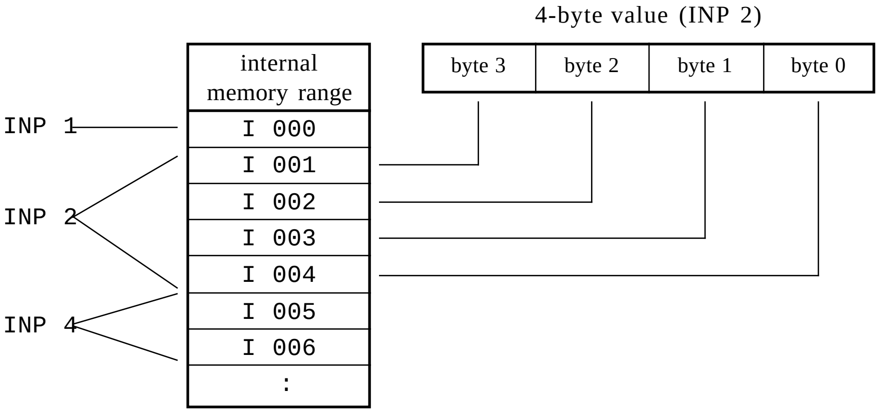
```
00006 *@FBKN: Comment.........................................
00006 *@
00006 *@                                   short comment..
00006 *@                                   ...............
00006 *@
00006 *@       1 .....      I000  -  INP 1   FBKN   -
00006 *@       4 .....      I001  -  INP 2   OUT 1  -    1 .....  I013
00006 *@       2 R2000            -  INP 3   OUT 2  -    2 R2002
00006 *@       O R....      I005  -  INP 4          -    . .....
00006 *@       O R3000      I007  -  INP 5          -    . .....
00006 *@       L            I009  -  LABEL          -    . .....
00006 *@       T >          I011  -  TNAME          -    . .....
00006 *@
00006  FBKN  NOP                          @O       START
  :           :
  :           :
xxxxx        RTS                                END:
```

ADDRESSING for the input/output parameters in the FBK program:

Input/output	Addressing	Number of bytes
INP 1	I 000	1
INP 2	I 001	4
INP 3	R2000	2
INP 4	I 005	2
INP 5	I 007	2
LABEL	I 009	2
TNAME	I 011	2
OUT 1	I 013	1
OUT 2	R2002	2

The most significant byte is located on a lower address in multi byte values.

Example: Input INP2: A 4-byte value is submitted to the FBK program.
This value includes the following:



Important

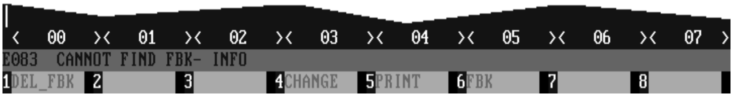
The internal memory range may not be used for data storage, since this range is used by the system. So it is not guaranteed, that the memory contents won't change during the next program execution.

This is to be considered especially concerning outputs. All output states, submitted to the internal memory range by the FBK program, absolutely must be processed. The contacts, connected to FBK outputs, could otherwise accept undefined states.

8. CHANGE AN EXISTING FBK

The defined inputs/outputs of existing FBKs can be changed. With the command ***FBK*** **<NAME> [↵]** the FBK editor is called, where the inputs and outputs can be redefined.

After terminating all changes and exiting the FBK editor the first LAD is automatically called, in which this FBK is installed. The respective FBK is displayed in inverse video (on a colored screen the FBK is displayed with red background) and the softkey line allocation is altered.



```
<  00  ><  01  ><  02  ><  03  ><  04  ><  05  ><  06  ><  07  >
E083  CANNOT FIND FBK- INFO
1DEL_FBK 2        3        4CHANGE  5PRINT   6FBK     7        8
```

[F1 DEL_FBK] The unchanged FBK is deleted from the LAD.

[F4 CHANGE] The original FBK is replaced by the altered FBK.

[F5 PRINT] The current LAD is printed.

[F6 FBK] The FBK overview is displayed.

The FBK displayed in inverse video has to either be deleted from the LAD or be replaced by the newly defined FBK. This procedure is repeated by all “old” existing FBKs in the LAD. Only if no old FBK is available in a LAD, can the lad be edited and changed.

By exiting the first LAD with [Esc], the next LAD is called, where an old FBK still exists.

This is repeated until all original FBKs, that are available in the program, are deleted or replaced by a new FBK.

If all inputs/outputs of a FBK are redefined, in most cases the pertinent STL program must be changed as well.

CHAPTER 11

ONLINE PROGRAMMING VIA MODEM

Contents:

1. INTRODUCTION

The application program memory module with networking capabilities and built-in modem interface (briefly referred to as "the modem interface" in this manual) allows you to make a remote diagnosis of PLC processors with the B&R PROgramming SYStem. Up to 16 PLC processors (CPUs and parallel processors) can be monitored or programmed on-line with the programming device. However, you must distinguish between two operating modes[1]:

1) Operation with a modem
2) Operation without a modem

You require a B&R PROgramming SYStem version higher than 5.4 for on-line programming via the modem interface. This version contains some new instructions for the on-line programming via modem:

-	**MODEM**	Preselection of modem-specific parameters (configuration menu)
-	**DIAL**	Manages up to 200 names and telephone numbers. Establishes a telephone link.
-	**ONL COM1**	Activates the PC's serial COM1 port and displays details about the dialed station.
-	**STANDBY**	The B&R PROgramming SYStem waits for alarms from the dialed station and saves these into a file.

The B&R PROgramming SYStem from version 5.4 upwards presently only supports the RS232 links to the busmaster (modem interface with node number 0). An on-line link via a modem interface with the node numbers 1 to 15 is only supported through the busmaster's RS485 interface.

FEATURES

The modem on-line link provides you with the same features to intervene in the PLC as those in a direct link between PC and PLC. All operational commands that require an existing on-line link can be used:

```
RUN, RUN P, DEBUG, LDMO, XFER, P-VERS, LOAD PLC, EEPROM
```

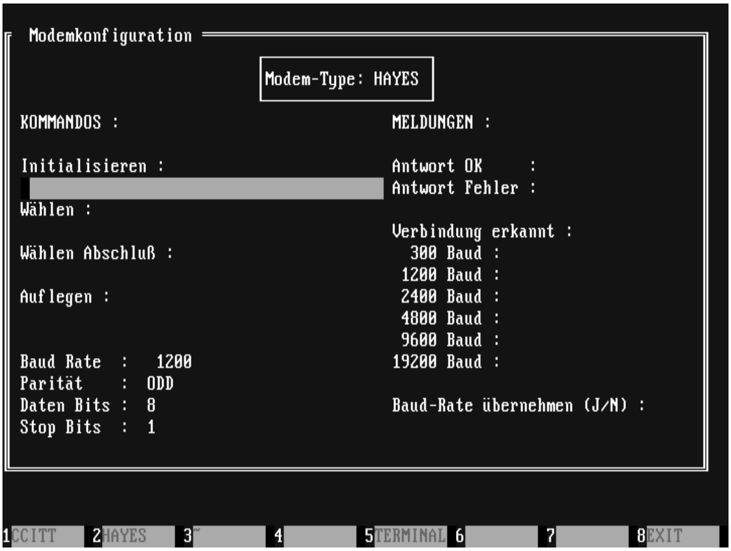
In addition, there is also the possibility that the modem interface monitors the PLC and forwards alarm messages to a superior station, which will then protocol these messages and, if necessary, take remedial action in the PLC.

[1] See section 12 "Online Interfaces"

2. MODEM SETTINGS

Command: *MODEM* **[↲]**

Function: This command calls up the "modem settings" menu. In this menu you enter the specific parameters of the modem that is connected to the PC.



```
Modemkonfiguration

                              Modem-Type: HAYES

KOMMANDOS :                                MELDUNGEN :

Initialisieren :                           Antwort OK     :
                                           Antwort Fehler :
Wählen :
                                           Verbindung erkannt :
Wählen Abschluß :                            300 Baud :
                                            1200 Baud :
Auflegen :                                  2400 Baud :
                                            4800 Baud :
                                            9600 Baud :
Baud Rate  :   1200                        19200 Baud :
Parität    :  ODD
Daten Bits :  8                            Baud-Rate übernehmen (J/N) :
Stop Bits  :  1
```

2.1. ASSIGNMENT OF FUNCTION KEYS

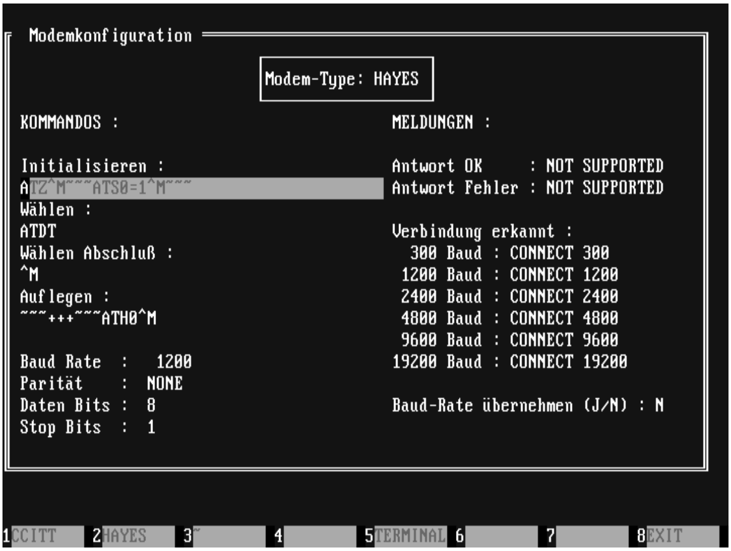
[F1 CCITT]

Preliminary setting of the modem configuration with standard parameters for CCITT-compatible modems.

This function is not supported by the B&R PROgramming SYStem version 5.4.

[F2 HAYES]

Preliminary setting of the modem configuration with the standard parameters for HAYES-compatible modems (AT instruction set).



```
Modemkonfiguration

                         Modem-Type: HAYES

KOMMANDOS :                              MELDUNGEN :

Initialisieren :                         Antwort OK     : NOT SUPPORTED
ATZ^M~~~ATS0=1^M~~~                      Antwort Fehler : NOT SUPPORTED
Wählen :
ATDT                                     Verbindung erkannt :
Wählen Abschluß :                          300 Baud : CONNECT 300
^M                                        1200 Baud : CONNECT 1200
Auflegen :                                2400 Baud : CONNECT 2400
~~~+++~~~ATH0^M                           4800 Baud : CONNECT 4800
                                          9600 Baud : CONNECT 9600
Baud Rate  :   1200                      19200 Baud : CONNECT 19200
Parität    :  NONE
Daten Bits :  8                          Baud-Rate übernehmen (J/N) : N
Stop Bits  :  1
```

[F3 ~]

This function key allows you to place the "~" character (wait) at your current cursor position. This character makes the modem wait 0.5 seconds before executing the next command.

[F5 TERMINAL]

Enter the terminal emulation for operating the modem directly through the keyboard. After calling up this function, the screen is cleared and the cursor is placed on the left hand-side at the top of the screen.

[F8 EXIT] or [ESC]

Exit from the "modem settings". After activating this function, the following request is made in the menu line:

W300 write data ? (Y/N)

If you press... [Y] the entered data are accepted and saved to the user configuration file;

If you press... [N], [F8 EXIT] or [ESC] the menu "modem settings" is exited without writing the data into the user configuration file.

2.2 MODEM SETTING PARAMETERS

Select the field to be altered with the cursor keys [up], [down], [PgUp] or [PgDn]. The field selected is displayed in inverse video.

2.2.1COMMANDS

Initialization Command: String of max. 40 characters with which the modem is initialized in each dialing process or when the standby menu is entered.

Dialing: Command: Dialing command consisting of max. 30 characters for initiating the dialing process.

Dialing end: Command: String of max. 30 characters that is sent to the modem at the end of each dialing process.

Hang-up: Command: String of max. 30 characters that makes the modem disconnect from an existing link.

Entry of special characters:

"~" Wait for 0.5 secs.

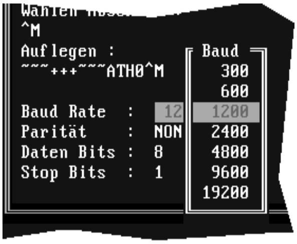
"^" Characterizes a control character
Example: ^M is interpreted hexadecimally as $0D (=Carriage Return)
and sent to the modem[1]

"^^" The character "^" is sent to the modem.

[1] The function of each control character is outlined in the modem´s operating manual.

2.2.2. PARAMETERS FOR COM1-INTERFACE

Baud rate: If [↵] is pressed while the cursor is placed on this field, a window is opened in which the following baud rates can be selected by means of the cursor keys [up], [down].



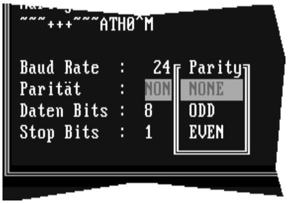
```
Wahlen
^M
Auflegen :
~~~+++~~~ATH0^M

Baud Rate   :  12  Baud
Parität     :  NON   300
Daten Bits  :  8     600
Stop Bits   :  1    1200
                    2400
                    4800
                    9600
                   19200
```

If you press....

... [↵] the respective baud rate is selected and confirmed.

... [F8 EXIT] or [ESC] the window for selecting the baud rate is exited without accepting the selected value.

Parity: If the [↵] key is pressed while the cursor is placed on this field, a window is opened in which the following features can be selected with the cursor keys [up], [down]:



```
~~~+++~~~ATH0^M

Baud Rate   :  24  Parity
Parität     :  NON  NONE
Daten Bits  :  8    ODD
Stop Bits   :  1    EVEN
```

If you press....

... [↵] the parity is selected and confirmed.

... [F8 EXIT] or [ESC] the window for selecting the parity is exited without accepting the selected value.

Data bits: The number of data bits is selected with the following keys:

[5]	5 Data bits
[6]	6 Data bits
[7]	7 Data bits
[8]	8 Data bits

Stop bits: The number of stop bits is selected with the following keys:

[1]	1 Stop bit
[2]	2 Stop bits

2.2.3. MESSAGES

Response OK: This response is not used by the B&R PROgramming SYStem.

Response Error: This response is not used by the B&R PROgramming SYStem.

Connect Message: If a link is established with another modem, the modem connected to the PC sends a "connect message". This message contains the baud rate with which the link is established with the remote modem.

Any message to be displayed by the modem in the event of an established link should be entered here. For further information please refer to your modem's operating manual.

CAUTION: If incorrect messages are entered in the modem configuration (connect message), B&R PROgramming SYStem is not able to recognize an established link.

Auto Baud Rate Detect:

[Y] The PC's COM1 interface is parameterized with the baud rate contained in the "connect message".

[N] The baud rate of the COM1 interface is not changed. This is only possible in modems that are able to communicate with the remote modem in another baud rate than with the PC.

2.3 PRINCIPAL SETTINGS IN THE MODEM

Some settings have to be made in the modem connected to the PC, so that the B&R PROgramming SYStem is able to communicate with it.

1) Set the AT instruction set, because it is presently the only one supported by the B&R PROgramming SYStem. Usually you can select between the AT and the CCITT instruction set in a modem.

2) Determine how the modem is connected in your telephone system:

   a) Main line
   b) Extension

3) Set the data format:

   a) 8 Data bits
   b) No parity
   c) 1 Stop bit

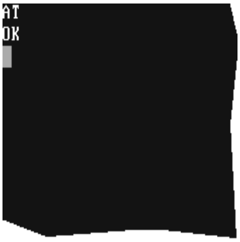
4) Select the type of dialing:

   a) Pulse dialing
   b) Tone dialing

5) Hardware handshake is active.

6) Check the connection PC - Modem by means of the terminal emulation that is built into the B&R PROgramming SYStem ([F5 TERMINAL] in the menu "modem configuration"). Enter the modem commands directly (e.g. dialing commands, hang-up, ...) in order to check the link. If necessary, make alterations to the modem parameters.

Example: After entering AT (the AT command for attention) in the terminal emulation, the modem should respond with OK:



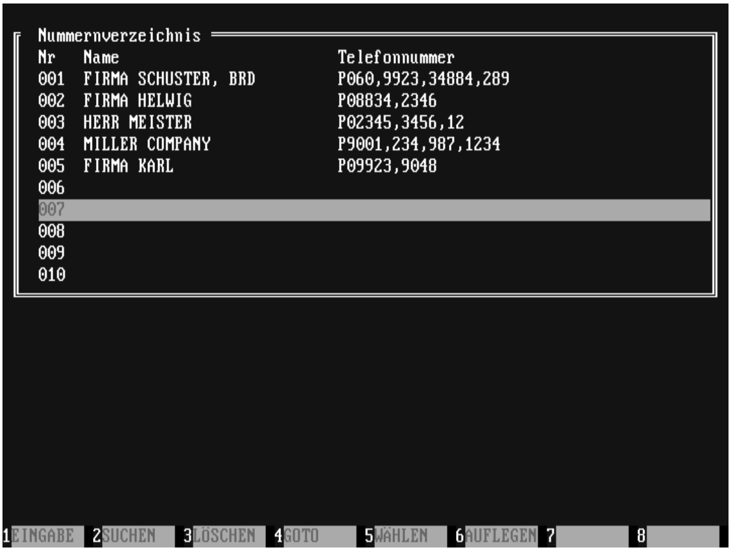
```
AT
OK
```

3. DIALING DIRECTORY

Command: ***DIAL [↵]***

Function: This command calls up the "Dialing directory" menu. This menu allows you to enter up to 200 station names with the respective telephone numbers, establish (DIAL) and terminate (HANG-UP) links.

After entering DIAL[↵] in the command line, the following menu appears:

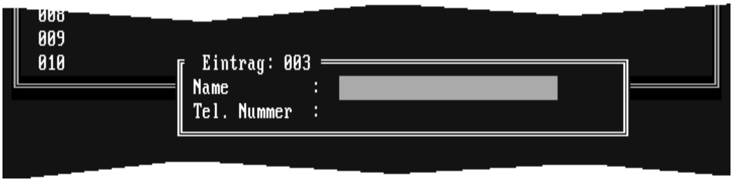


```
Nummernverzeichnis
Nr   Name                          Telefonnummer
001  FIRMA SCHUSTER, BRD           P060,9923,34884,289
002  FIRMA HELWIG                  P08834,2346
003  HERR MEISTER                  P02345,3456,12
004  MILLER COMPANY                P9001,234,987,1234
005  FIRMA KARL                    P09923,9048
006
007
008
009
010
```

Select the desired number with the cursor keys [up], [down] or [PgUp], [PgDn]. The inverse entry can be edited, altered, newly entered or "invoked" through the modem connected to the PC.

[F1 REVISE]

Input or change of an entry. If this function is selected, a window is opened in which the "name" and "telephone number" fields can be altered:



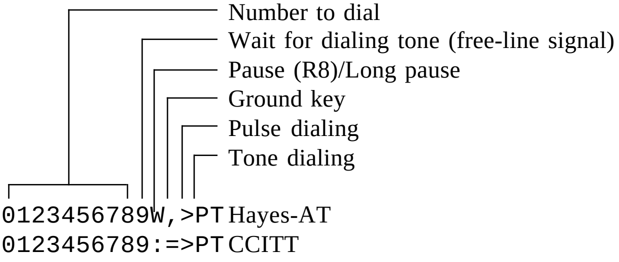
```
008
009
010          Eintrag: 003
           Name        :
           Tel. Nummer :
```

[ESC] Exit from the window and accept entered data.

Entering special characters in the "telephone number" field:

"~" Wait for 0.5 secs.
"^" Characterizes a control character
Example: ^M is interpreted hexadecimally as $0D (=Carriage Return)
and sent to the modem[1]
"^^" The character "^" is sent to the modem.

Recognized control characters within the telephone number:



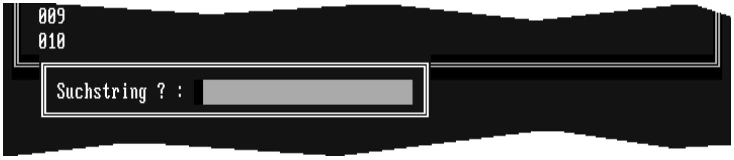
Example: Tone dial the number 09989 7602 521 through an AT-modem from an extension (get main line first with "0" and wait for free-line signal dialing tone) and pause
between area code, number and extension:

T0W09989,7602,521

[1] The function of each control character is outlined in the modem´s operating manual.

[F2 SEARCH]

Search for an entry. After pressing [F2 SEARCH], a window is opened:



```
009
010
Suchstring ? :
```

The user can now enter a search mask according to which a search is made. The mask is compared with the name and telephone number. The first entry found is shown in inverse video. The dialing directory is searched from NO. 001 to 200. If necessary, the search begins again at the first entry. No difference is made between upper case and lower case characters.

If no entry was found with the mask, the following error message is displayed:

```
E212 FAILED SEARCH !
```

Example: Search mask: ... STAT
The following entries are found

Name	**Telephone number**
STATION 1	092388836
8. **Stat**ion	P2
Co. Salmond	**STAT**0911

[F3 ERASE]

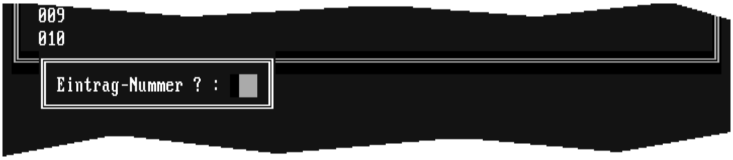
The selected entry is deleted from the directory. If this function was selected, the user is asked for a confirmation:

```
W299 DELETE ENTRY ? (Y / N)
```

[Y] The entry is deleted
[N] The delete function is terminated

[F4 GOTO]

Select the entry with the desired number. When calling up this function, a dialogue window appears in which the desired number can be entered (1 to 200):



```
009
010
Eintrag-Nummer ? :
```

[F5 DIAL]

The B&R-PROgrammingSYStem tries to establish a link with the selected number (telephone number). A window is opened in which the modem's last message is displayed. The dial process can be interrupted at any time by pressing [F8 EXIT] or [ESC]. After the link has been established successfully, the program automatically switches to the menu "node list".

[F6 HANG UP]

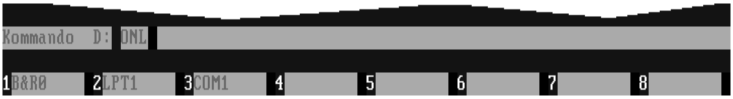
The existing link is disconnected. A window is opened which shows the modem's last message to the PC. Close the window again by pressing [F8 EXIT] or [ESC]. The following message appears:

```
W283  PLEASE WAIT 10 SECONDS !
```

4. NODE LIST

Command: ***ONL COM1* [↵]**

Function: After entering ONL in the command line, the assignment of the softkey line is changed:

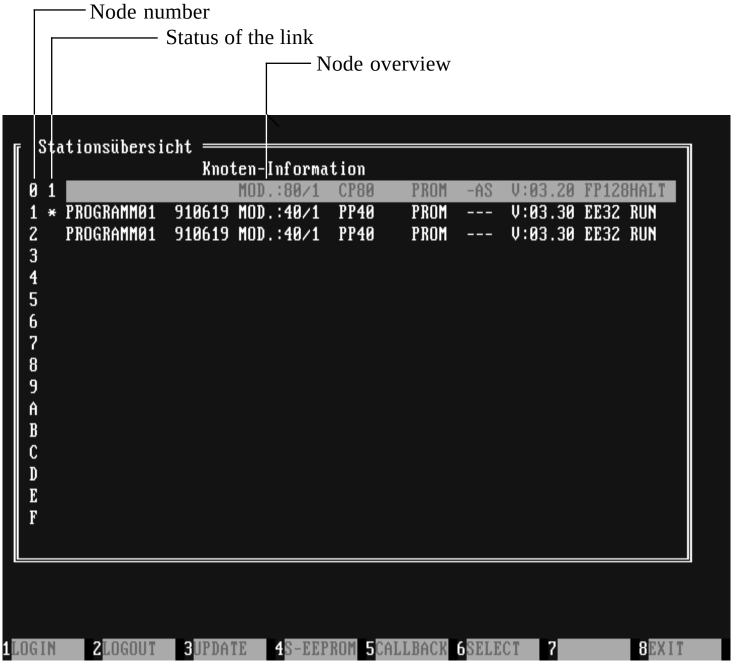


```
Kommando  D: ONL
1B&R0    2LPT1    3COM1    4        5        6        7        8
```

After entering [F3 COM1] and [↵] the program tries to establish a link to the PLC (modem interface) through the PC's COM1 interface. In modem operation it is necessary to establish the respective telephone link (command: DIAL) first before calling up the node list, as otherwise the following error message is displayed:

```
E052 MISSING ONLINE INTERFACE CARD
```

After the link is established successfully, the station overview is displayed:



```
 Stationsübersicht
                  Knoten-Information
0 1                    MOD.:80/1  CP80    PROM  -AS  V:03.20 FP128HALT
1 * PROGRAMM01  910619 MOD.:40/1  PP40    PROM  ---  V:03.30 EE32 RUN
2   PROGRAMM01  910619 MOD.:40/1  PP40    PROM  ---  V:03.30 EE32 RUN
3
4
5
6
7
8
9
A
B
C
D
E
F

1LOGIN   2LOGOUT  3UPDATE  4S-EEPROM 5CALLBACK 6SELECT  7        8EXIT
```

Node number:

Up to 16 CPUs or parallel processors can be linked within a network by means of the modem interface. The node number is set on the modem interface through a turn switch (see section 12 Online Interfaces).

Status of the link:

""	Not logged in
"*"	Logged in (no on-line link between modem interface and PLC!)
"!"	On-line link established, but not active (For example: after switching to a local interface with ONL B&R0, without previous logout)
"1"	Station is activated for window 1 (On-line link exists between modem interface and PLC)
"2"	Station is activated for window 2 (On-line link exists between modem interface and PLC)

Node information:

Five different messages are possible:

1) `!!!! SAFETY CALLBACK ACTIVE !!!!`
After logging in and selecting one of the two callback numbers, the modem interface interrupts the link and tries to establish the link with the selected callback number.

2) `!!!! PASSWORD ACTIVE !!!!`
The station is protected by a password.

3) `PLC ERROR`
PLC is out of order.

4) `PLC DISCONNECTED`
The on-line connection between the modem interface and the PLC is either not inserted properly or out of order.

5) Information on the program that is currently being executed in the PLC is equivalent to the information line that is sent in the command P-VERS [1)].

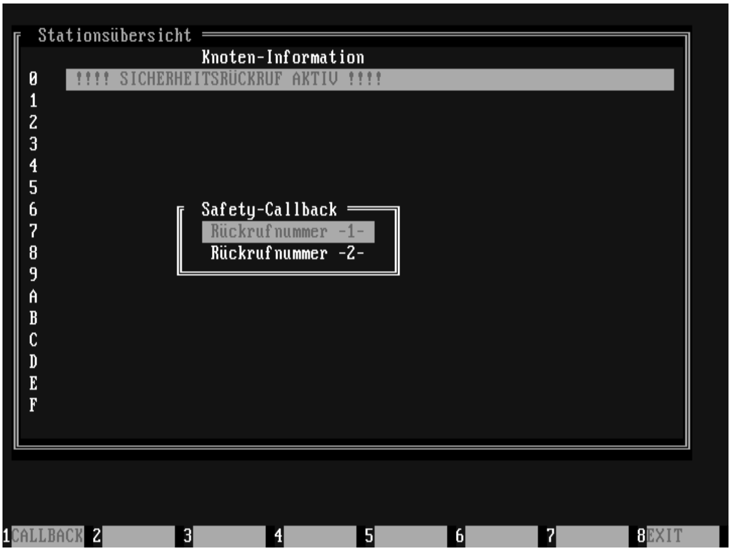
Select the displayed stations with the cursor keys [up] or [down]. The station is displayed inversely on the screen.

[1)] See section 5 "STATEMENT LIST PROGRAMMING", 8.5 P-VERS - Information on the PLC program.

[F1 LOGIN]

The B&R-PROgrammingSYStem logs into the selected station. It should be proceeded as follows:

1) If the safety callback is active, the user is asked in a dialogue window to select one of the two callback numbers when a link is established via the modem.



The assignment of the softkey line was also altered:

[F1 CALLBACK]

Call back with the selected number (see also [F5 CALLBACK]).

The B&R-PROgrammingSYStem switches into the STANDBY-mode and waits for the callback from the modem interface. The last message from the modem is shown in a window on the screen. You can leave the STANDBY-mode at any time by pressing [ESC] or [F8 EXIT].

After a successful callback and the establishment of the link, the program switches back to the station overview mask and displays all stations.

[F8 EXIT] or [ESC]

Interrupt the function and return to the node list.

2) If the password check system is active, the user is asked in a window to enter his password. If a wrong password was entered, the following message is displayed:

```
E204  INVALID PASSWORD !
```

If the password was correct, the character "*" for "logged in" is displayed in the station overview.

3) If none of the safety systems is active, the node is marked by the "*" character as "logged in".

[F2 LOGOUT]

Log out from the selected station.

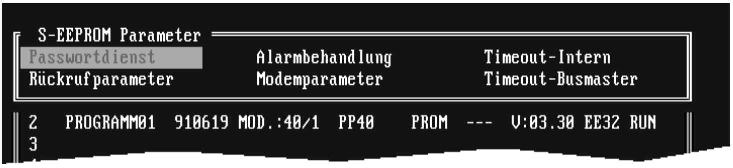
After logging out from the master (node number 0), all slaves are automatically logged off (node numbers 1 to F). Then the link is interrupted.

[F3 UPDATE]

All the information on the stations (status of links, node information) is updated. This function is used, for example, when the station overview is called up with ONL COM1 [↵]. For runtime reasons, the node information can only be updated if the PLC status has changed.

[F4 S-EEPROM]

Call up the menu for entering the parameters in the S-EEPROM (serial EEPROM in the modem interface). The following menu appears.

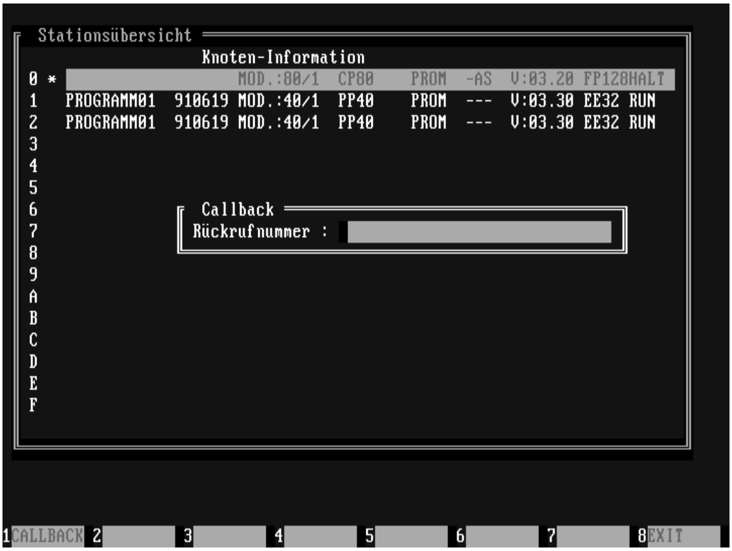


```
S-EEPROM Parameter
Passwortdienst       Alarmbehandlung      Timeout-Intern
Rückrufparameter     Modemparameter       Timeout-Busmaster

2   PROGRAMM01  910619 MOD.:40/1  PP40    PROM  ---  V:03.30 EE32 RUN
3
```

For further details on this menu, please see "5. Programming the S-EEPROM".

[F5 CALLBACK]

Initiate a callback from the modem interface. After activating this function, the user is requested in a window to enter the callback string:



```
 Stationsübersicht
                    Knoten-Information
0 *                     MOD.:80/1  CP80    PROM  -AS  V:03.20 FP128HALT
1   PROGRAMM01   910619 MOD.:40/1  PP40    PROM  ---  V:03.30 EE32 RUN
2   PROGRAMM01   910619 MOD.:40/1  PP40    PROM  ---  V:03.30 EE32 RUN
3
4
5
6                   Callback
7                  Rückrufnummer :
8
9
A
B
C
D
E
F

1CALLBACK 2        3        4        5        6        7        8EXIT
```

If the function key [F5 CALLBACK] is pressed, the link is interrupted, the B&R-PROgramming-SYStem switches over to standby and waits for the callback. The last message from the modem is displayed in a window on the screen. The standby mode can be left at any time by pressing [F8 EXIT] or [ESC]. After the callback and the successful establishment of a link, the program switches back to the node list, which displays all stations. A CALLBACK is only possible if...

...the user is only logged into the master.
...the link was established through a modem.

[F6 SELECT]

The marked node is selected for the current window. The window number is displayed on the right hand-side next to the node number. A node that has already been selected for the same window is marked by "*"; the on-line link of this node is terminated.

All functions of the B&R-PROgrammingSYStem in the respective window now refer to this node (e.g. RUN, DEBUG, hardware reset,....).

[F8 EXIT] or [ESC]

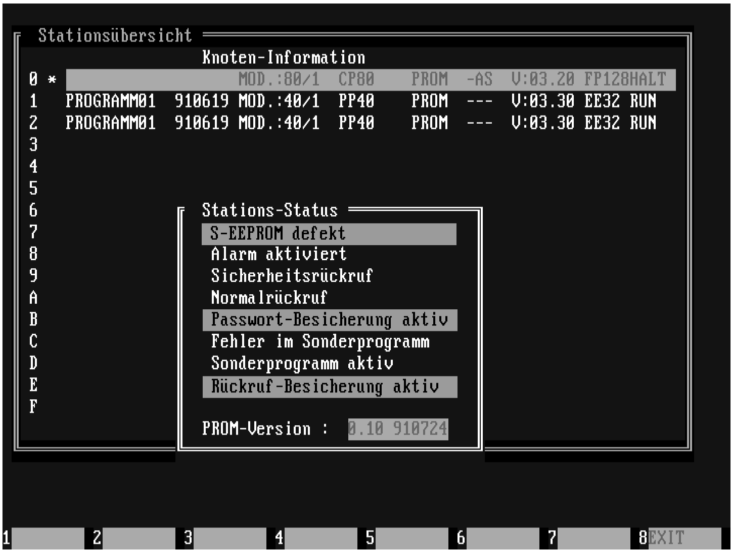
Exit from the node list and return to the command line in B&R-PROgrammingSYStem.

[F9] or [F10]

The assignment of the softkey lines can be changed with these two function keys.

[F9] + [F1 STATUS]

Display station status. Various conditions of the marked node as well as the PROM version are displayed in the window. The parameters marked by the bar are active:



```
 Stationsübersicht
                   Knoten-Information
0 *                     MOD.:80/1  CP80    PROM  -AS  V:03.20 FP128HALT
1   PROGRAMM01  910619 MOD.:40/1  PP40    PROM  ---  V:03.30 EE32 RUN
2   PROGRAMM01  910619 MOD.:40/1  PP40    PROM  ---  V:03.30 EE32 RUN
3
4
5
6                  Stations-Status
7                   S-EEPROM defekt
8                   Alarm aktiviert
9                   Sicherheitsrückruf
A                   Normalrückruf
B                   Passwort-Besicherung aktiv
C                   Fehler im Sonderprogramm
D                   Sonderprogramm aktiv
E                   Rückruf-Besicherung aktiv
F
                   PROM-Version :  0.10 910724
```

S-EEPROM Defective
This message will be displayed as long as the S-EEPROM is not programmed by the user or if the S-EEPROM is defective.

Alarm Active
An alarm is imminent.

Safety Callback
Link was established by safety callback with one of the two callback numbers stored in the modem interface.

Callback
Link was established through a normal callback with the number entered in the B&R-PROgrammingSYStem.

Password Protection Active
Station is protected by a password.

S-EEPROM Programming Error
There was an error in programming the S-EEPROM.

Programming S-EEPROM
S-EEPROM is programmed.

Callback Protection Active
A safety callback will be made when a login is made at this station.

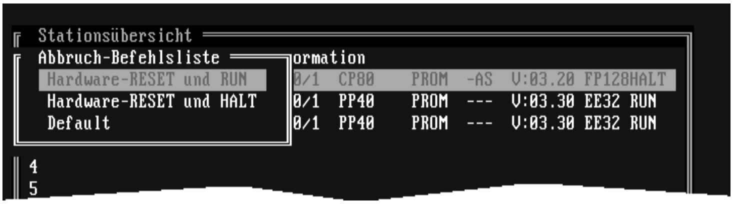
PROM Version
The modem's operating system version

[F8 EXIT] or [ESC]

Close the window and return to the menu "Node List".

[F9] + [F2 ONL-ABRL]

The "Disconnect Command List" determines the PLC's behaviour after a link was interrupted. After calling this function, the user has three options:



```
Stationsübersicht
Abbruch-Befehlsliste        ormation
  Hardware-RESET und RUN    0/1  CP80   PROM  -AS  V:03.20 FP128HALT
  Hardware-RESET und HALT   0/1  PP40   PROM  ---  V:03.30 EE32 RUN
  Default                   0/1  PP40   PROM  ---  V:03.30 EE32 RUN

4
5
```

1) Hardware RESET and STOP

After the link is disconnected, a hardware reset is carried out in the PLC. The PLC remains in a halted condition (it behaves as if an on-line cable were attached).

2) Hardware RESET and RUN

After the link is terminated, a hardware reset is carried out in the PLC. However, the PLC is started again (it behaves as if no on-line cable were attached).

3) Default

The PLC remains in the condition in which it was at the time when the link was disconnected.

[F1 PROGRAM]

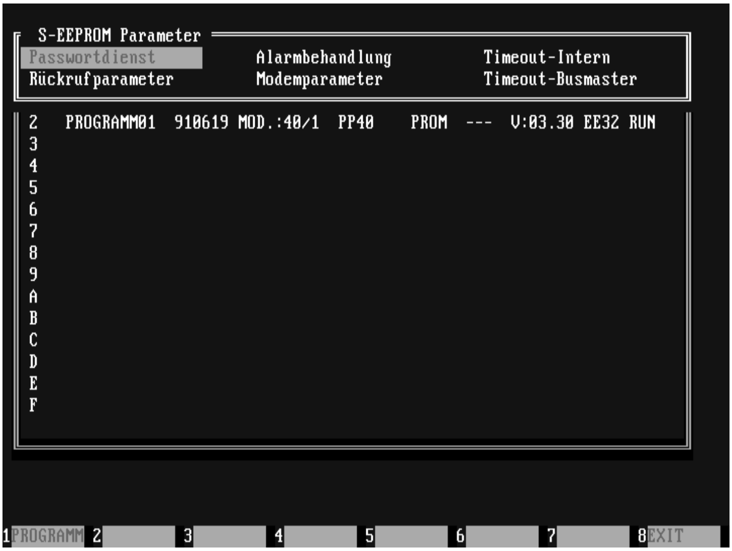
Transfer the settings to the modem interface and store them in the S-EEPROM.

[F8 EXIT] or [ESC]

Exit from the menu without transmitting the command list.

5. PROGRAMMING THE S-EEPROM

After pressing [F4 S-EEPROM] in the menu "node list", the menu for the S-EEPROM parameter is displayed.



```
2   PROGRAMM01  910619 MOD.:40/1  PP40    PROM  ---  V:03.30 EE32 RUN
3
4
5
6
7
8
9
A
B
C
D
E
F
```

Select one of the menu items with the cursor keys [up], [down], [left] and [right]. The function is called up with the [↵] key.

[F1 PROGRAM]

The S-EEPROM is programmed with the new or altered parameters. Programming the S-EEPROM takes approx. 10 secs. After pressing [F1 PROGRAM] the following message appears:

```
W277 S-EEPROM PROGRAMMING ACTIVE (10 SEC) !
```

After the programming is completed, the following message is displayed:

```
W278 S-EEPROM PROGRAMMING OK !
```

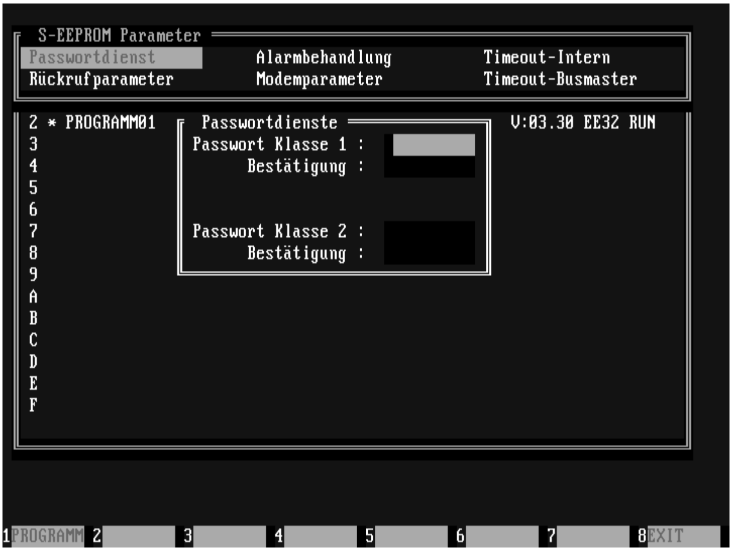
[F8 EXIT] or [Esc]

The menu for the S-EEPROM parameters is exited. The B&R-PROgrammingSYStem returns to the node list.

5.1 PASSWORD OPTION

Passwords protect the PLC against unauthorized use. One password can be defined for each of the user levels in the modem interface's S-EEPROM. Before a user can log into the system and thus can manipulate the PLC, he has to enter one of the two passwords.

This menu item allows you to alter an existing password or to add a new one. After selecting the menu item "password option" with the cursor keys [up], [down], [left] and [right] and confirming it with [↵], a window for entering and altering the passwords is opened:



For each of the two user levels one password with a maximum length of ten characters can be entered.

Enter/alter password:

The entry is made "blind", i.e. the screen does not show the password entered by the user.

1) Enter new/altered password
2) Confirm entered password (re-enter password)

[F1 PROGRAM]

The S-EEPROM is programmed with the new or altered passwords. The cursor has to be located on one of the confirmation fields when this function key is activated. If this is not the case, the following error message is displayed:

```
E205 PROGRAM PASSWORD ONLY FROM ACKNOWLEDGEMENT !
```

If another password was entered than in the field "password class x", the B&R-PROgrammingSYStem issues the following error message:

```
E203 PASSWORD / ACKNOWLEDGEMENT MISMATCH !
```

The programming of the S-EEPROM takes approx. 10 secs. Depending on which password was changed, one of the following messages is issued:

```
W274 PASSWORD CHANGE FOR LEVEL 1 ACTIVE (10 SEC)!
W275 PASSWORD CHANGE FOR LEVEL 2 ACTIVE (10 SEC)!
```

After the programming is completed, the following confirming message is displayed:

```
W276 PASSWORD CHANGED !
```

[F8 EXIT] or [ESC]

The window for changing/entering passwords is closed. The B&R-PROgrammingSYStem returns to the S-EEPROM menu.

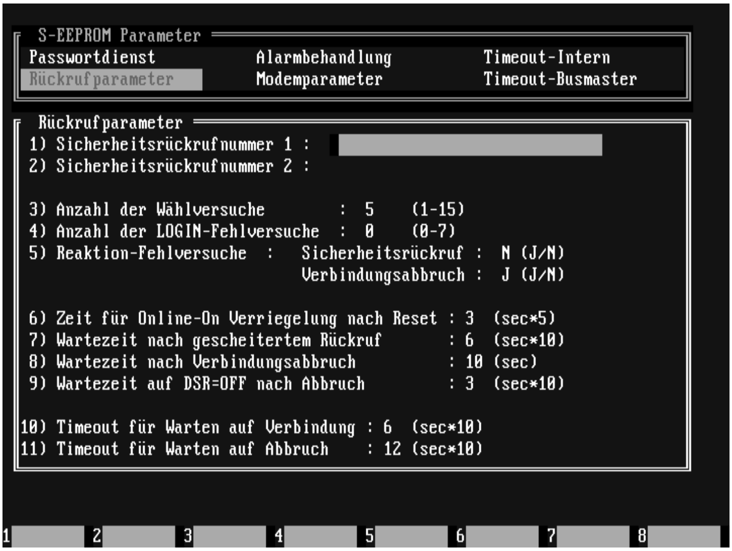
User Levels and their Rights:

Function	Level	Remarks
RUN	2	-
Hardware Reset	2	-
Disconnect Command List	2	-
Program EEPROM	2	First time switch from PROM to RAM only allowed after... ... verify Y/N = Y ... or after previous switch from RAM to PROM ... RUN with EEPROM programming
XFER	2	-
LOAD PLC	2	-
Program S-EEPROM	2 1	- Call only possible for information purposes
Change Passwords	2 1	Passwords of both user levels can be changed Only passwords of user level 1 can be changed
Functions in the Debug Mode		
Display memory locations	1/2	-
Change memory locations	2	-
Edit register [F1 REGEDIT]	2	-
Stop program [F4 STOP]	2	-
Continue program [F5 CONT]	1/2	-
Edit breaks [F2 BRKEDIT]	2	-
STL debug mode [F3 STL]	1/2	-
Single steps [F7 S-STEP]	2	-
Trace [SHIFT] + [F7 S-STEP]	2	-
Reset [F6 RESET]	2	-

5.2 CALLBACK PARAMETERS

The PLC can be protected against unauthorized users by means of the safety callback. Two telephone numbers can be stored in the modem interface's S-EEPROM, which are called back in the event that a login is attempted. Only users who know one of the two telephone numbers are able to log into the system.

The parameter for the safety callback can be entered or altered in this menu item. After selecting the menu item "callback parameter" with the cursor keys [up], [down], [left] and [right] and the respective confirmation with [↵], a window is opened for entering or altering the parameters.



```
Rückrufparameter
 1) Sicherheitsrückrufnummer 1 :
 2) Sicherheitsrückrufnummer 2 :

 3) Anzahl der Wählversuche       :  5    (1-15)
 4) Anzahl der LOGIN-Fehlversuche  :  0    (0-7)
 5) Reaktion-Fehlversuche  :   Sicherheitsrückruf :  N (J/N)
                               Verbindungsabbruch :  J (J/N)

 6) Zeit für Online-On Verriegelung nach Reset : 3  (sec*5)
 7) Wartezeit nach gescheitertem Rückruf       : 6  (sec*10)
 8) Wartezeit nach Verbindungsabbruch          : 10 (sec)
 9) Wartezeit auf DSR=OFF nach Abbruch         : 3  (sec*10)

10) Timeout für Warten auf Verbindung : 6  (sec*10)
11) Timeout für Warten auf Abbruch    : 12 (sec*10)
```

[ESC]

Closes the window and returns to the menu "S-EEPROM Parameters".

Select and alter any one of the parameters with the cursor keys [up] or [down].

1/2) Safety callback number 1/2
Enter any international telephone number (max. 30 characters) (control characters are allowed; see section 3 "Dialing directory"). The modem interface tries to establish a link with one of the two stored numbers in the event of a safety callback.

3) Number of redial attempts:
Enter the number of attempts the modem will make for calling one of the two numbers. If after the preselected number of attempts no link is established, the modem returns to its standy mode.

4) Number of LOGIN attempts:
After the set number of login attempts the response entered under item 5) is initiated.

5) Action LOGIN failure:
Two possible responses can be set:

- Safety callback
- Termination of the link

6) Time for online-on after reset:
The on-line link to the PLC is maintained for the set period after a reset, so that the PLC is allowed to start up properly. During this period it is not possible to ...
... scan alarms
... establish an on-line link.

7) Pause after failed callback:
If a callback fails, the set time has to pass before a new call back is made.

8) Pause after disconnect:
After a link is terminated, the set time has to pass before a callback is carried out ("settling period" for the respective device).

9) Pause for DSR drop after disconnect:
Within the set period the modem interface waits for the interruption of the link to become effective (acknowledgement from the modem: link is terminated).

10) Wait for connect:
The modem interface waits for the establishment of a link after a dial request. If within the set period a link is not established, the callback is deemed as having failed.

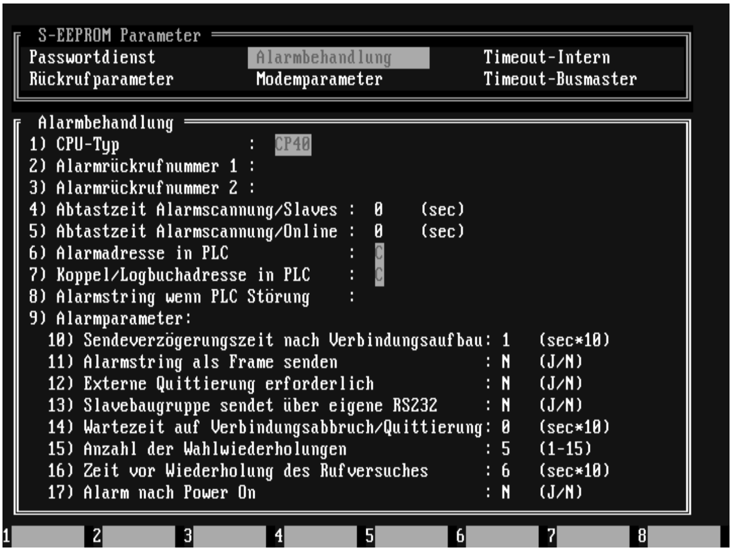
11) Wait for disconnect:
This waiting time is superior to the one defined under item 8). Before a callback can take place, the current link must be interrupted. If this does not take place within the set period, the callback is not carried out.

5.3 ALARM OPTION

The modem interface is able to scan a dual-port RAM area with various alarm parameters via the on-line link to the attached PLC. The PLC can deposit alarm messages in this area, which are then sent by the modem interface via modem to the remote station that is called through one of the two alarm callback numbers.

Practical example: A system with several PLC stations sends messages and alarms via the modem interface to a master station, which logs these messages and, if necessary, initiates certain reactions to them.

This menu item allows you to enter and alter the parameters for handling an alarm . After selecting the menu item "alarm handling" with the cursor keys [up], [down], [left] and [right] keys and confirming it with [↵], a window is opened for entering and altering the parameters:



```
Alarmbehandlung
1) CPU-Typ                  :  CP40
2) Alarmrückrufnummer 1 :
3) Alarmrückrufnummer 2 :
4) Abtastzeit Alarmscannung/Slaves :  0    (sec)
5) Abtastzeit Alarmscannung/Online :  0    (sec)
6) Alarmadresse in PLC             :  C
7) Koppel/Logbuchadresse in PLC    :  C
8) Alarmstring wenn PLC Störung    :
9) Alarmparameter:
  10) Sendeverzögerungszeit nach Verbindungsaufbau: 1   (sec*10)
  11) Alarmstring als Frame senden                : N   (J/N)
  12) Externe Quittierung erforderlich            : N   (J/N)
  13) Slavebaugruppe sendet über eigene RS232     : N   (J/N)
  14) Wartezeit auf Verbindungsabbruch/Quittierung: 0   (sec*10)
  15) Anzahl der Wahlwiederholungen               : 5   (1-15)
  16) Zeit vor Wiederholung des Rufversuches      : 6   (sec*10)
  17) Alarm nach Power On                         : N   (J/N)
```

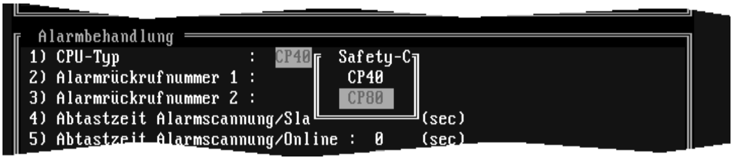
[ESC]

Closes the window and returns to the menu "S-EEPROM parameters".

Select and alter any one of the parameters with the cursor keys [up] or [down].

1) Type of CPU:

Enter the type of CPU (CP40/CP80) to which the modem interface is connected.



```
Alarmbehandlung
1) CPU-Typ              : CP40  Safety-C
2) Alarmrückrufnummer 1 :        CP40
3) Alarmrückrufnummer 2 :        CP80
4) Abtastzeit Alarmscannung/Sla         (sec)
5) Abtastzeit Alarmscannung/Online :  0    (sec)
```

2/3) Alarm callback number 1/2:

Enter any international telephone number (with a maximum of 30 characters) (control characters are allowed; see chapter "3. Dialing directory").

4) Time base alarm scanning / slaves:

Scanning interval within which the bus master (node number 0) scans the slaves (node numbers 1 to F) for imminent alarms.

Caution: If this time is changed, it also has to be changed in all slaves with alarm scanning.

5) Time base alarm scanning / online:

Scanning interval within which the PLC, which is connected via an on-line link with the modem interface, is scanned. If time "0" is entered, the alarm scan is deactivated.

6) Alarm address in PLC:

This alarm address is scanned in defined intervals. This address is the base address of the alarm parameter block in the PLC. If no alarm address is entered, the alarm scan is deactivated.

7) Log book address in PLC:

Coupling address: The set memory area of the PLC contains the coupling parameters for the data transfer between the two PLCs (transmission takes place via the modem interface's RS485 or RS232).

Log-book address: This address determines the beginning of the memory area in the PLC that contains a brief protocol (log-book) on the modem interface's last action. For example, the following results are entered into the log-book:

```
Establishment of link recognized.       Login attempt invalid.
Rejection of unauthorized command.            Login has occurred.
Safety callback was activated.                Alarm callback was initiated.
```

If no coupling/log-book address is entered, the log-book function is deactivated.

Caution: The address areas of 6) and 7) must not overlap.

8) Alarm text for PLC error:
Message with max. 10 characters that is sent in the event of an on-line PLC error.

9) Alarm parameters:
Send parameters for alarm string of item 8). The parameters 10) to 16) are only used if there is a malfunction in the PLC or a defective on-line link to the PLC. Otherwise, the alarm parameters defined in the PLC are used (see 6.1 Alarm parameters).

10) Wait for transmission after CONNECT:
Delay period before sending the alarm signal after establishing the link. If during this period valid data are received, the send process is blocked. If the link is terminated within this time, a renewed dial attempt is made. If a link already exists, the alarm string is sent within at least one second without a previous dial attempt.

11) Transmit alarm text as frame:
The alarm string is sent "protected" (MININET protocol).

12) External acknowledgement required:
The modem interface expects the remote station's acknowledgement of the alarm string. This item can only be selected with [Y] if item 11 is activated.

13) Slave node transmits via own RS232:
In the event of an alarm, the slave is able to transmit independently through its own RS232 interface:

14) Time for disconnect / acknowledge:
Depending on the selection made in item 11), this time has the following meanings:

* Item 11 not active:	Waiting time after sending the alarm string before the link is terminated.
* Item 11 active:	Waiting time until alarm string was acknowledged. If the alarm string is not acknowledged within the set period, an error message is sent to the CPU.

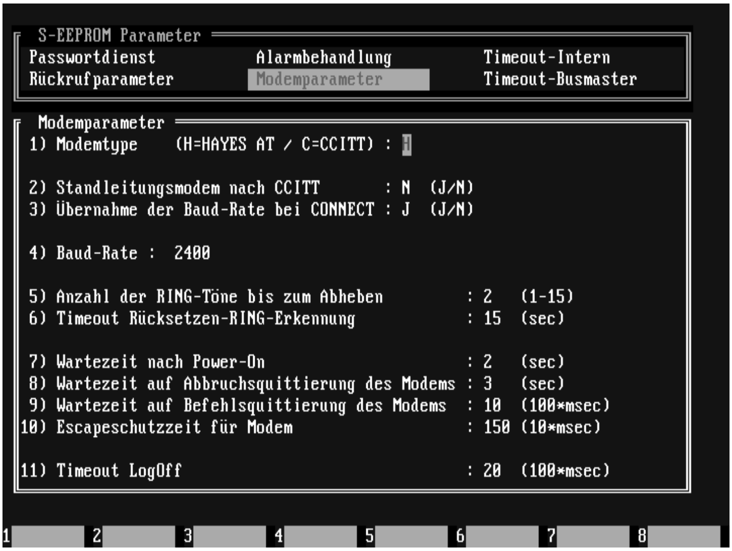
15) Number of redial attempts during establishment of the link.

16) Pause between calls:
Waiting time before a renewed dial attempt is made after the failure to establish a link.

17) Alarm after Power On:
An alarm is issued after a power on in the PLC.

5.4 MODEM PARAMETER

In this menu you can enter or alter the parameters for the modem that is connected to the modem interface. After selecting the menu item "modem parameter" with the cursor keys [up], [down], [left] and [right] and confirming it with [↵], a window is opened for entering and altering the parameters:



```
 Modemparameter
 1) Modemtype    (H=HAYES AT / C=CCITT) : H

 2) Standleitungsmodem nach CCITT       : N  (J/N)
 3) Übernahme der Baud-Rate bei CONNECT : J  (J/N)

 4) Baud-Rate :  2400

 5) Anzahl der RING-Töne bis zum Abheben         : 2   (1-15)
 6) Timeout Rücksetzen-RING-Erkennung            : 15  (sec)

 7) Wartezeit nach Power-On                      : 2   (sec)
 8) Wartezeit auf Abbruchsquittierung des Modems : 3   (sec)
 9) Wartezeit auf Befehlsquittierung des Modems  : 10  (100*msec)
10) Escapeschutzzeit für Modem                   : 150 (10*msec)

11) Timeout LogOff                               : 20  (100*msec)
```

[ESC]

Closes the window and returns to the menu "S-EEPROM parameters".

Select and alter one of the 11 parameters with [up] or [down]:

1) Type of modem:

Set the type of modem: C CCITT
H Hayes-AT

2) Connect if DTR=ON / disconnect if DTR=OFF:

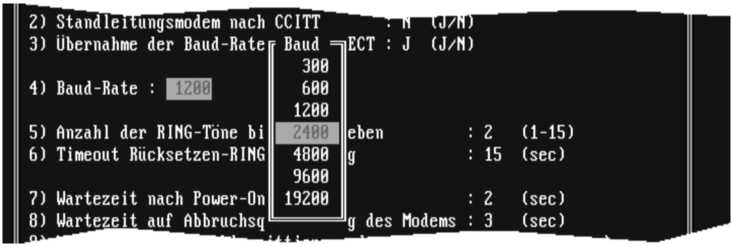
Link is established by DTR=ON and terminated by DTR=OFF.

3) Auto baud rate detect:

The baud rate is gained from the CONNECT message (see item 2.2.3 - Messages).

4) Baud rate:

If the [↵] key is pressed when the cursor is placed on this field, the following baud rates can be selected with the cursors keys [up] or [down]:



```
2) Standleitungsmodem nach CCITT        : N  (J/N)
3) Übernahme der Baud-Rate╒ Baud ╕ECT : J  (J/N)
                          │  300 │
4) Baud-Rate :  1200      │  600 │
                          │ 1200 │
5) Anzahl der RING-Töne bi│ 2400 │eben          : 2   (1-15)
6) Timeout Rücksetzen-RING│ 4800 │g             : 15  (sec)
                          │ 9600 │
7) Wartezeit nach Power-On│19200 │              : 2   (sec)
8) Wartezeit auf Abbruchsq└──────┘g des Modems : 3   (sec)
```

5) Number of rings:

If the modem is called, the <number> of rings is waited until the modem goes off hook to establish the link.

6) Timeout reset - RING indicator:

If within the stated period the desired rings are not recognized, the ring counter is reset to zero and the modem returns to the standby mode.

7) Wait after power on:

Waiting time before the modem is initialized after the PLC is switched on.

8) Wait after disconnect ACK from modem:

After a request for a disconnect, the modem interface waits the period set until the link is terminated (message from the modem: disconnect).

9) Wait for command ACK from modem:

After a general command to the modem, the modem interface waits for the acknowledgement of the command from the modem within the period set.

10) Escape timeout for modem:

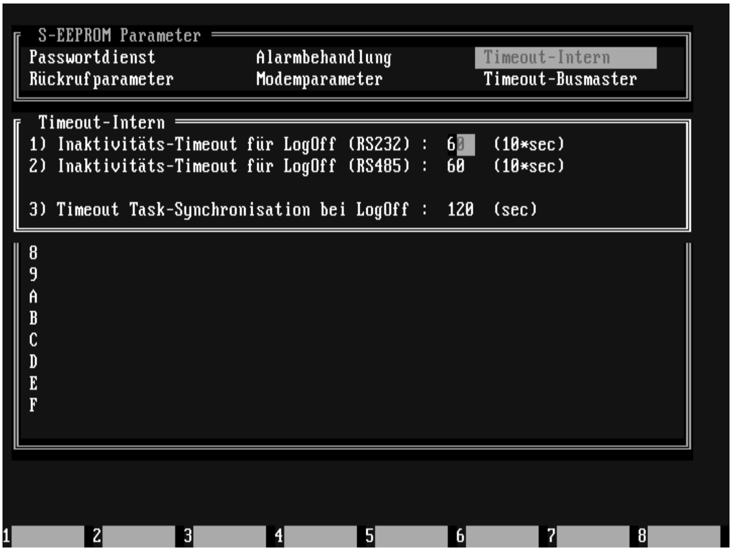
This concerns a time within which nothing must be sent to the modem before or after sending the escape sequence (+++).

11) LOFOUT timeout:

Carrier loss time without causing a logoff. If the data carrier between the two modem exceeds the set time, the link is disconnected. It is assumed that the link was terminated externally (i.e. from the remote station).

5.5 INTERNAL TIMEOUT

Select and alter the internal timeout parameters for the modem interface. After selecting the menu item "internal timeout" with the cursor keys [up], [down], [left] and [right] and confirming it with [↵], a window is opened for entering and altering the parameters:



```
1) Inaktivitäts-Timeout für LogOff (RS232) :  60   (10*sec)
2) Inaktivitäts-Timeout für LogOff (RS485) :  60   (10*sec)

3) Timeout Task-Synchronisation bei LogOff :  120  (sec)
```

[ESC] Closes the window and returns to the menu "S-EEPROM parameters".

Select and alter one of the 3 parameters with the cursor keys [up] or [down]:

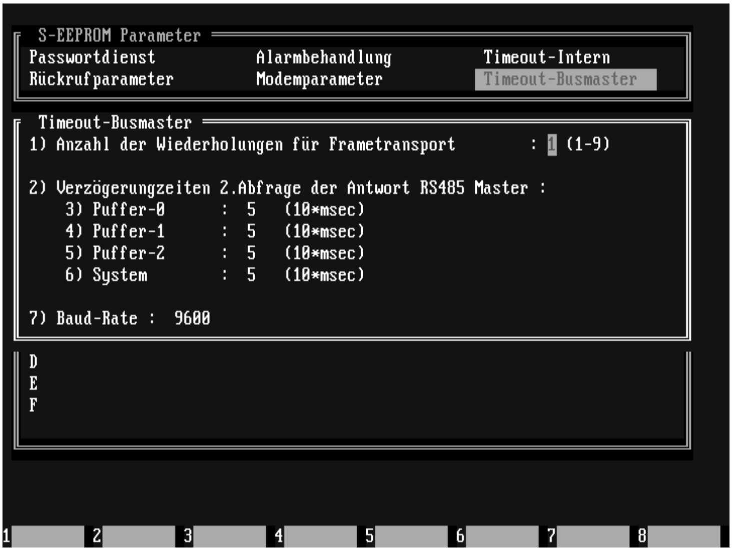
1) Inactivity timeout for LOGOUT (RS232):
If no valid data are received during the set period, the link is disconnected. This time is to be adjusted to the modem's inactivity timeout, if activated. If time "0" is entered, the timeout is deactivated.

2) Inactivity timeout for LOGOUT (RS485):
If no valid data are received during the set period, the link to the master is disconnected. If time "0" is entered, the timeout is deactivated.

3) Timeout task synchronization at LOGOUT :
After the time has passed, a limited logoff is carried out:
- Termination of the on-line and modem link
- Software reset of the modem interface

5.6 BUSMASTER TIMEOUT

This item allows you to enter or alter the timeout parameters of the busmaster (node number 0). These parameters influence the data transmission through the RS485 (MININET). After selecting the menu item "busmaster timeout" with the cursor keys [up], [down], [left] and [right] and confirming it with [↵], a window is opened for entering and altering the parameters:



```
S-EEPROM Parameter
Passwortdienst          Alarmbehandlung         Timeout-Intern
Rückrufparameter        Modemparameter          Timeout-Busmaster

Timeout-Busmaster
1) Anzahl der Wiederholungen für Frametransport       : 1 (1-9)

2) Verzögerungzeiten 2.Abfrage der Antwort RS485 Master :
    3) Puffer-0      :  5   (10*msec)
    4) Puffer-1      :  5   (10*msec)
    5) Puffer-2      :  5   (10*msec)
    6) System        :  5   (10*msec)

7) Baud-Rate :  9600

D
E
F

1         2         3         4         5         6         7         8
```

[ESC] Closes the windows and returns to the menu "S-EEPROM parameters".

Select and alter one of the 7 parameters with the cursor keys [up] or [down]:

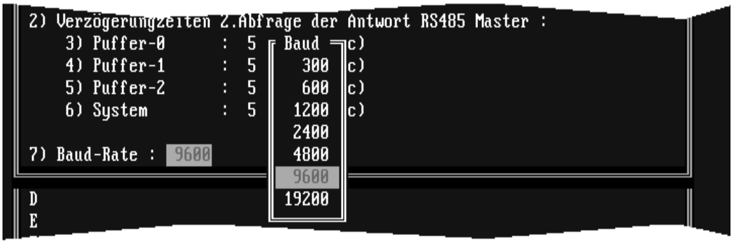
1) Number of retries for frame transmit:
Number of attempts by the busmaster for sending a frame to a slave. If the transmission is not successful within the set number, the busmaster invokes a bus error.

2) Delay time for second request of RS485 master response:
The master's scan period for the query whether the slave has processed (completed) the current command frame. This time can be defined for each of the four logical input/ output channels.

3) Buffer-0:
4) Buffer-1:
5) Buffer-2:
6) System:

7) Baud rate:

If the [↵] key is pressed when the cursor is placed on this field, the following baud rates can be selected with the cursors keys [up] or [down]:



```
2) Verzögerungzeiten 2.Abfrage der Antwort RS485 Master :
    3) Puffer-0      :  5  Baud   c)
    4) Puffer-1      :  5    300  c)
    5) Puffer-2      :  5    600  c)
    6) System        :  5   1200  c)
                            2400
7) Baud-Rate : 9600         4800
                            9600
D                          19200
E
```

When pressing...

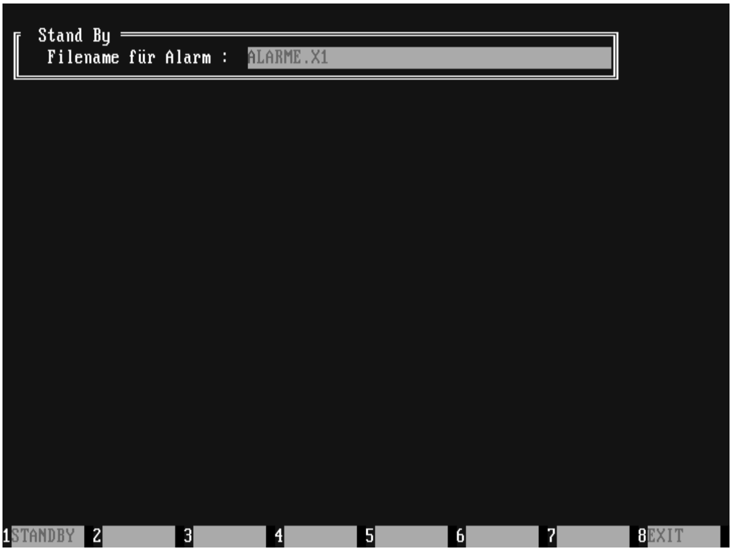
... [↵] the respective baud rate is selected and confirmed.

... [F8 EXIT] or [ESC] the window for selecting the baud rate is exited without accepting the selected value.

6. STANDBY OPERATION

Command: ***STANDBY [↵]***

Function: Call up the standby menu:



Stand By
Filename für Alarm : ALARME.X1

The B&R-PROgrammingSYStem opens a window in which the current file name is shown. Any incoming alarms are stored in this file. The file name can be changed. If no file name is entered (or the current file name is deleted), the incoming alarms are not stored in a file.

Default file name: ALARME.X1

The file name can be entered with path name and drive. If the path name is not entered, the file is automatically stored in the directory in which the B&R-PROgrammingSYStem was installed (usually: C:\PG).

The B&R-PROgrammingSYStem automatically initializes the modem in accordance with the settings made in the menu "modem settings".

The following functions are offered in the softkey line:

[F1 STANDBY]

Activate the actual standby mode. The window for entering a file name is closed and the window "last message" is opened. This window shows the last message from the modem to the B&R-PROgrammingSYStem. The B&R-PROgrammingSYStem then waits for the following events:

- Message from the modem: connect message, i.e. the link was established successfully.
- Termination of the standby mode by pressing [F8 EXIT] or [ESC].

If a link was established (B&R-PROgrammingSYStem receives a connect message from the modem), the type of call is determined. The following options are possible:

- Alarm callback
- Safety callback
- Normal callback
- Call cannot be identified

[F8 EXIT] or [ESC]

The standby mode is terminated. The first standby menu appears, in which the file name can be altered.

6.1 RESPONSE TO ESTABLISHMENT OF LINK

The B&R-PROgrammingSYStem determines the type of the call and responds pertinently:

Alarm callback

1) The alarm string is displayed on the screen and, if desired, written to an alarm file.

2) If the acknowledgement of the alarm is required, it is carried out. The necessity of an acknowledgement is defined by the user in the alarm parameters (see "5.3 Alarm handling" and "6.1 Alarm parameters).

Safety callback

1) If at the same time an alarm is imminent, it is treated like an alarm callback.

2) B&R-PROgrammingSYStem switches to the node list.

Normal callback

1) If at the same time an alarm is imminent, it is treated like an alarm callback.

2) B&R-PROgrammingSYStem switches to the node list.

Call cannot be identified

1) The link is disconnected.

2) An error message is issued.

3) The B&R-PROgrammingSYStem remains in the standby mode.

6.2 STANDBY AFTER INITIATING A CALLBACK

After initiating a normal or safety callback from the node list, the B&R-PROgrammingSYStem automatically switches to the standby mode. However, note has to be taken of the following:

1) The alarm file's name cannot be changed.

2) The modem is not initialized, as this was already done in the previous dialing process.

After the callback was received, the standby mode is exited and the program switches to the menu station overview.

Remark:

For the operation in the standby mode the modem connected to the PC must be set to the **auto answering mode**, i.e. the modem must go off-hook automatically. For example, this can be deterined within the modem initialization settings (modem configuration). The command for setting this mode is described in your modem's manual.

7. EXCHANGING DATA WITH THE PLC

Two addresses of registers can be defined in the S-EEPROM, which constitute the base address of a memory area reserved by the PLC:

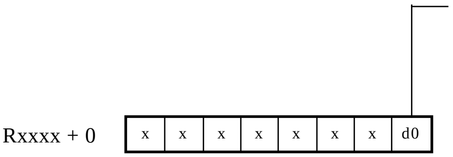
1) Alarm parameter block
2) Dual-port RAM

Remark: The memory areas for the dual-port RAM and the alarm parameter must not overlap!

7.1 ALARM PARAMETERS

The base address Rxxxx for the memory area in which the alarm parameters are stored is defined in the menu alarm handling (6th alarm address in PLC) and saved to the modem interface's S-EEPROM. The modem interface scans this address within certain intervals and invokes an alarm in the event of an alarm callback.

The PLC sends alarm to the modem interface

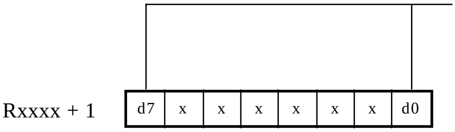


The PLC informs the modem interface of an alarm message through bit 0:

```
xxxxxxx0  no alarm
xxxxxxx1  alarm
```

If the PLC intends to send an alarm to the modem interface, it first has to set bit 0 from Rxxxx + 1 to logical 0, as the modem interface will otherwise not recognize the alarm message.

The modem interface acknowledges the alarm message

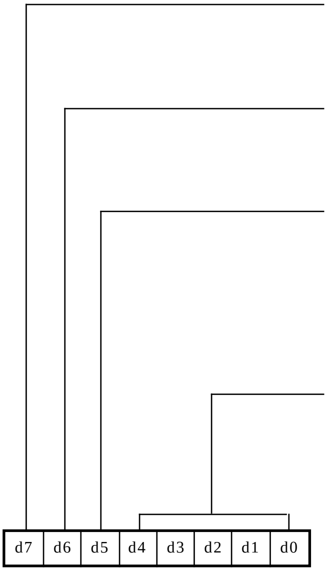


The modem interface returns an acknowledgement to the PLC:

```
0xxxxxx1  Acknowledgement without error message
1xxxxxx1  Acknowledgement with error message
```

Possible errors:
- Alarm call was unsuccessfully terminated / Establishment of link not possible
- Alarm was not confirmed by remote station
- Link was terminated during the transmission of the alarm string

Alarm parameters



Slave node transmits via own RS232:

0xxxxxxx	Yes
1xxxxxxx	No (sends via RS485 to busmaster)

Transmit alarm text as frame:

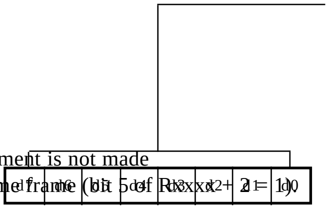
x0xxxxxx	No
x1xxxxxx	Yes (MININET protocol)

External acknowledge required (only possible if the alarm string was sent as frame):

xx0xxxxx	No
xx1xxxxx	Yes (the modem interface expects the acknowledgement from the remote station and issues an error to the PLC if it is not made)

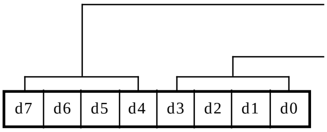
Wait for transmission after CONNECT (x x 10 secs.): Delay before sending the alarm string after establishing the link. If during this period valid data are received, the sending process is blocked. If the link is terminated within this time, a renewed dialing attempt is made.

Rxxxx + 2



Time for disconnect / acknowledge:
(x x 1 sec.): Waiting time after sending the alarm string before the link is disconnected (bit 5 of Rxxxx + 2 = 0). Waiting time for acknowledgement of the alarm string. The modem interface reports errors to the PLC if the acknowledgement is not made within the predefined time frame (bit 5 of Rxxxx + 2 = 1)

Rxxxx + 3



Number of redials when establishing the link.

Time before redial after a failed alarm callback (x x 10 secs)

Rxxxx + 4

These three registers contain the alarm parameters. These parameters must be defined by the user within the PLC program (see "5.3 Alarm handling").

Alarm message string

Rxxxx + 5	d7	d6	d5	d4	d3	d2	d1	d0	Address of the alarm message string MSB
Rxxxx + 6	d7	d6	d5	d4	d3	d2	d1	d0	Address of the alarm message string LSB
Rxxxx + 7	d7	d6	d5	d4	d3	d2	d1	d0	Length of the alarm message string (max. 80 characters)

In these three registers the PLC hands over the address and the length of the alarm message string. Hence, the PLC is able to supply a certain text to the modem interface according to the respective alarm. The modem interface, on the other hand, can send this message to a remote station via an established link (alarm callback).

7.2 LOG BOOK / DUAL-PORT RAM

The base address Rxxxx for the memory area of the dual-port RAM is defined in the menu alarm handling (7th log book address in PLC) and saved to the S-EEPROM of the modem interface.

The first register contains the command:

Rxxxx + 0	x	x	x	x	x	x	d1	d0

Command:
xxxxx00 No command/Acknowledgement from PLC
xxxxx01 Load through dual-port RAM
xxxxx10 Send through dual-port RAM
xxxxx11 Log book function

7.2.1 Log book

The modem interface makes a log book entry into a ring buffer with 16 entries when special events occur. This ring buffer is transmitted to the PLC's coupling area (coupling/log book address) depending on on-line traffic (link between modem interface and the PLC). As the log book is realized as a ring buffer, the oldest entries are lost in the event of an overflow. Hence, only the 16 latest entries are retained.

Log book command

xxxxxx11	The modem interface writes this log book command into the PLC as soon as it has saved a complete entry to the dual-port RAM area, i.e. the entries from memory locations Rxxxx + 3 are now valid.
xxxxxx00	The PLC overwrites the log book command with this value after the log book entry was evaluated. The acknowledgement informs the modem interface that the dual-port RAM area can be written into again. The PLC should react to this log book entry within 500 ms, otherwise an error is generated by the modem interface.

Rxxxx + 0	x	x	x	x	x	x	1	1

Source-/Destination address

Rxxxx+1	d7	d6	d5	d4	d3	d2	d1	d0	not used
Rxxxx+2	d7	d6	d5	d4	d3	d2	d1	d0	not used

Logbook entry

Rxxxx + 3	d7	d6	d5	d4	d3	d2	d1	d0	length of log book entry incl. type
Rxxxx + 4	d7	d6	d5	d4	d3	d2	d1	d0	type of entry
Rxxxx + 5	d7	d6	d5	d4	d3	d2	d1	d0	logbook entry
	d7	d6	d5	d4	d3	d2	d1	d0	
	d7	d6	d5	d4	d3	d2	d1	d0	

The log book entry consists of the three parts length, type and entry.

Type	Description	Interface	Data	Length
1	Establishment of link recognized	-	-	1
2	Login attempt invalid Login attempt invalid	RS232 RS485	RS232 RS485	6 6
3	Command rejected as unauthorized Command rejected as unauthorized	RS232 RS485	RS232 RS485	6 6
4	LOGIN has occurred, user level 1 LOGIN has occurred, user level 2 LOGIN has occurred, user level 1 LOGIN has occurred, user level 2	RS232 RS232 RS485 RS485	RS232 1 RS232 2 RS485 1 RS485 2	8 8 8 8
5	Safety callback invoked with <tel. no.>	-	<tel. no.>	16
6	Normal callback invoked with <tel. no.>	-	<tel. no.>	16
7	Alarm callback invoked with <tel. no.>	-	<tel. no.>	16
8	Disconnect through internal event	-	-	1
9	Disconnect through missing data carrier	-	-	1
10	Disconnect because of inactivity timeout	-	-	1

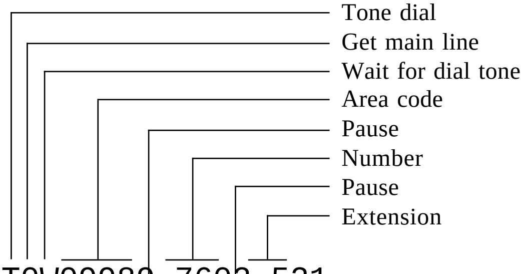
Data format of the telephone numbers <tel. no.>

The telephone numbers and the control characters contained in the dial string are compressed. The length of the dial string is limited to max. 15 characters (30 nibbels).

Character	Coding	Number of Nibbels
Characters from "0" to "9"	$0 to $9	1
ASCII characters from $20 to $7F	$80 + $xx	2
Control characters from $00 to $1F	$B00 + $Bxx	3
String end character as $FF	$FF	2

Example: Normal callback from an extension (get main line with 0 and wait for dial tone); tone dial the number 09989 7602 521 with an AT modem and pause between area code, number and extension.

Control character: (Hayes, AT instr. set)	**Function:**
T	Control character for tone dial
W	Control character for waiting for the dial tone
'	Control character for pause



Dial string: T0W09989,7602,521

The dial string is coded as follows:

Nibbel	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Contents	D	4	0	D	7	0	9	9	8	9	A	C	7	6	0	2	A	C	5	2	1	F	F	F	F	F	F	F	F	F

T	⇨	ASCII-value $54 + $80 = $D4	⇨	D4	⇨
0	⇨	number 0 = $0	⇨	0	⇨
W	⇨	ASCII-value $57 + $80 = $D7	⇨	D7	⇨
0	⇨	number 0 = $0	⇨	0	⇨
9	⇨	number 9 = $9	⇨	9	⇨
9	⇨	number 9 = $9	⇨	9	⇨
8	⇨	number 8 = $8	⇨	8	⇨
9	⇨	number 9 = $9	⇨	9	⇨
,	⇨	ASCII-value $2C + $80 = $AC	⇨	AC	⇨
7	⇨	number 7 = $7	⇨	7	⇨
6	⇨	number 6 = $6	⇨	6	⇨
0	⇨	number 0 = $0	⇨	0	⇨
2	⇨	number 2 = $2	⇨	2	⇨
,	⇨	ASCII-value $2C + $80 = $AC	⇨	AC	⇨
5	⇨	number 5 = $5	⇨	5	⇨
2	⇨	number 2 = $2	⇨	2	⇨
1	⇨	number 1 = $1	⇨	1	⇨
		end of string = $FF [1]	⇨	FF	⇨
The remainder of the 30 nibbels is filled with $Fs.					⇨

[1] If the length of the coded telephone number is precisely 15 bytes (= 30 nibbles) no end-of-string character ($FF) is added.

The log book entry in the PLC would look like this:

Rxxxx + 0	$03	Log book command
Rxxxx + 1	$xx	Not used
Rxxxx + 2	$xx	Not used
Rxxxx + 3	$10	Length of the entry incl. type = 16
Rxxxx + 4	$06	Type: Normal callback invoked with the following number.
Rxxxx + 5	$D4	Coded telephone number
Rxxxx + 6	$0D	
Rxxxx + 7	$70	
Rxxxx + 8	$99	
Rxxxx + 9	$89	
Rxxxx + 10	$AC	
Rxxxx + 11	$76	
Rxxxx + 12	$02	
Rxxxx + 13	$AC	
Rxxxx + 14	$52	
Rxxxx + 15	$1F	
Rxxxx + 16	$FF	
Rxxxx + 17	$FF	
Rxxxx + 18	$FF	
Rxxxx + 19	$FF	

7.2.2 DUAL-PORT RAM

It is possible to exchange blocks of data of up to 248 bytes between the PLC and the modem interface in the same memory area in which the log book entries are stored. In this case the modem interface is superior to the PLC. The modem interface can carry out the following two functions:

Load via dual-port RAM: The modem interface can require the PLC to save a certain memory area (source address) of the PLC to the dual-port RAM. The modem interface independently collects these data after the acknowledgement by the PLC and can ....

.... send these via the RS232 (via modem or directly) to a PC.
.... send these via the RS485 to the busmaster.

Send via dual-port RAM: The modem interface can save data received via the RS485 or RS232 to the PLC's dual-port RAM and inform the PLC about the destination address of these data.

These two functions allow an "error-protected" exchange of data with the PLC. A superior system (PC, MAESTRO,...) can request data from an PLC and send these to an other one. This exchange of data can be protected by safety callback.

Remark:

As the modem accesses the PLC's dual-port RAM through the normal debug mode functions (this takes place every 10 msecs in one of the PLC's timer interrupt routines), there is an effective baud rate of 200 baud.

Standard software for controlling this function is currently in preparation.

CHAPTER 12

ONLINE INTERFACES - HARDWARE

Contents:

1. GENERAL INFORMATION

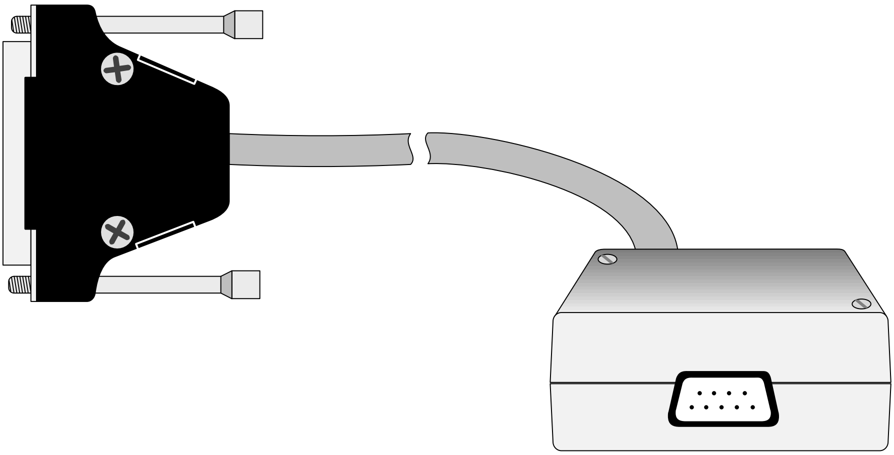
In order to be able to communicate with the programming device, all CPUs are equipped with an ONLINE interface. There are various possibilities for the establishment of an ONLINE connection between the programming device and the PLC:

- Via a **CENTRONICS ONLINE interface** to the parallel interface of the PC.
- Via a **B&R ONLINE interface** built into the PC.
- Via a **Modem** connected to the serial interface COM1 on the PC,
  or directly via the COM1 to a Modem Online Interface.

Note:
In this chapter the abbreviation "PD" will be used in place of the term PROGRAMMING DEVICE.

2. CENTRONICS ONLINE INTERFACE

2.1. MODEL NUMBERS - DESCRIPTIONS



Centronics ONLINE Interface for ONLINE operation with a PLC via the parallel interface of the PC (incl. cable to the PLC BRKAOL-0)	**BRKAOL5-1**

2.2. TECHNICAL DATA

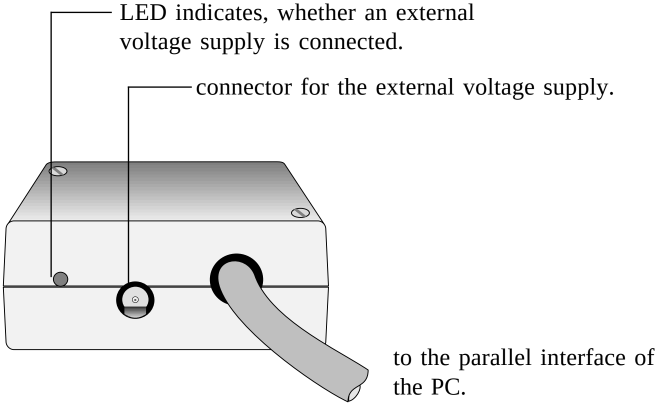
Description	CENTRONICS ONLINE interface
Model number	BRKAOL5-1
Voltage supply	via the parallel interface of the PC or external supply with 8 - 30 VDC / max. 250 mA
Interfaces	CENTRONICS parallel interface to the PC PD interface to the PLC
LED	indication of external supply

2.3. CONNECTION TO PC AND PLC

The CENTRONICS ONLINE interface is connected to the parallel interface of the PC.

> **Note:** The CENTRONICS ONLINE Interface may only be connected/disconnected when the PC is off.

If the CENTRONICS ONLINE interface is used together with battery supplied Laptops (e.g.: Toshiba T1200, T1600, T3100SX), the interface has to be supplied externally (e.g.: B&R power supply BRPS220904-0).

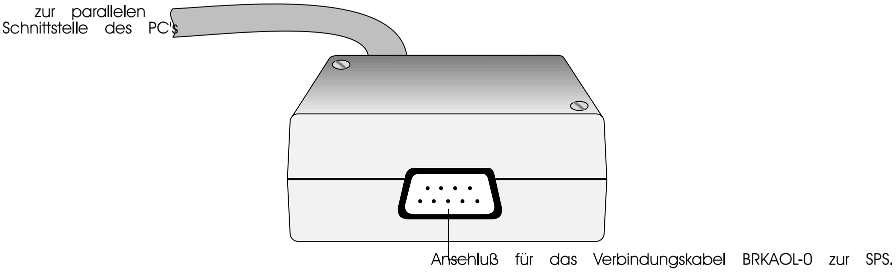


LED indicates, whether an external voltage supply is connected.

connector for the external voltage supply.

to the parallel interface of the PC.

The CENTRONICS ONLINE interface is connected with the PD interface of the PLC (labelled PG) by means of the ONLINE cable BRKAOL5-1:


zur parallelen Schnittstelle des PC's
Anschluß für das Verbindungskabel BRKAOL-0 zur SPS.

> **Note:** When the PC is switched on/off, the CENTRONICS ONLINE interface must be disconnected from the PLC.

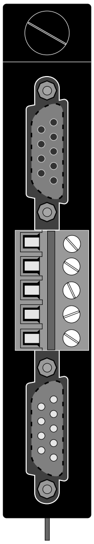
3. B&R ONLINE INTERFACE

3.1. MODEL NUMBERS - DESCRIPTIONS

B&R ONLINE Interface with two ONLINE interfaces...	
...for PCs with AT-compatible slots (e.g.:Toshiba T3200)	**BRIFPC-0**
...for Toshiba T 3100e, T 5100	**BRIFTO-0**
...for battery supplied Laptop Computers	
Toshiba T 1200	**BRIFTO-1**
Toshiba T 1600, T 3100SX	**BRIFTO-2**
...for COMPAQ Portable III	**BRIFCO-0**
ONLINE cable for BRIFPC-0 and BRIFTO-x, PD - PLC, 2,5 m	**BRKAOL-0**
ONLINE cable for BRIFCO-0, PD - PLC, 2,5 m	**BRKAOL-2**

4. MODEM ONLINE INTERFACE

4.1. MODEL NUMBERS - DESCRIPTIONS



User program memory module with Modem interface, for the connection of CCITT- or HAYES-Modems serial RS232 interface, serial RS485 interface, online interface, node-number switch and status-LED	
RAM/EEPROM module for CPU/PP modules equipped with a 6303-Processor (CP30, CP31, CP40, CP41, NTCP3# and PP40), 16 KB RAM, 16 KB EEPROM	**ECEE32MP-0**
Flash PROM module for CPU/PP modules equipped with a 6809-Processor (CP60, CP80, NTCP6#, PP60), Flash PROM memory for 42 KB of application program and 52 KB of application data memory	**ECFP128MP-0**

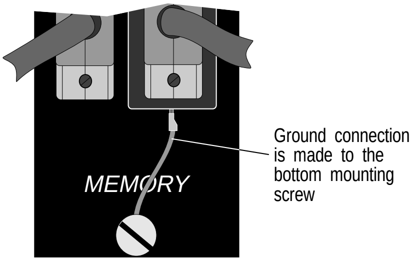
4.2. GENERAL INFORMATION

The network-suited user program memory module equipped with a Modem interface (hereinafter referred to as "Modem interface") is used for the remote diagnostic analysis of PLC. processors (CPUs or parallel processors) by means of the B&R PROgramming SYStem. It permits the online programming of up to 16 PLC. processors (CPUs or parallel processors), which are connected by means of a RS485 two-wire network.

Instead of an user program memory module (EE32, EP128, FP128, ...), the module is plugged into the processor module. The processor also supplies the modem interface with the required voltage.

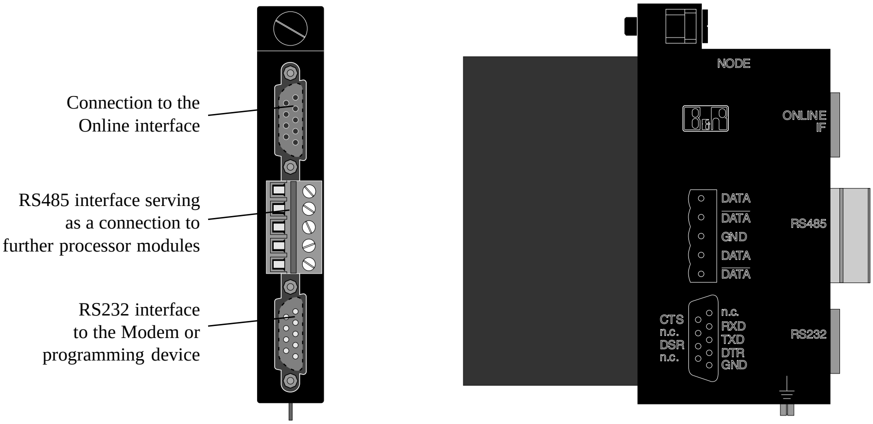
4.3. GROUNDING

On the bottom of the module the Modem interface is provided with a flat mounting clip. During operation, this mounting clip has to be connected with the grounding terminal strip by using a very short wire (at least 0.75 mm^2 cross section). In the case of MULTICONTROL, MIDICONTROL and M264 systems the grounding connection is attached to the bottom mounting screw of the processor module. In the case of MINICONTROL systems the grounding connection is attached to the ground terminal of the power supply module.

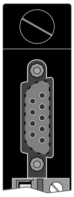


4.4. INTERFACES

The Modem interface is equipped with three serial interfaces. The pin configuration of the RS485 and RS232 interfaces is imprinted on the left side of the housing of the Modem interface:


Connection to the Online interface
RS485 interface serving as a connection to further processor modules
RS232 interface to the Modem or programming device
NODE
ONLINE IF
DATA
$\overline{\text{DATA}}$
GND
DATA
$\overline{\text{DATA}}$
RS485
CTS
n.c.
DSR
n.c.
n.c.
RXD
TXD
DTR
GND
RS232

CONNECTION TO THE ONLINE INTERFACE

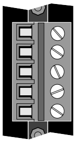


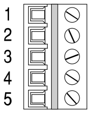
The 9-pin DSHELL connector is connected with the Online interface of the processor module (CPU or PP). For this purpose, one of the following Online cables can be used:

BRKAOL-0	3 m length
BRKAOL-3	0.3 m length

RS485 INTERFACE AS A CONNECTION TO FURTHER PROCESSOR MODULES

By means of the electrically isolated RS485 interface, up to 16 processor modules (CPUs or PPs) can be connected in a two-wire network. The desired processor module can be selected via a selection function in the B&R PROgramming SYStem.



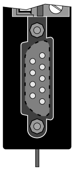


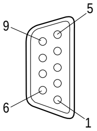
RS485	
Connection	**Function**
1	DATA
2	$\overline{\text{DATA}}$
3	GND
4	DATA
5	$\overline{\text{DATA}}$

The upper or lower DATA/$\overline{\text{DATA}}$ connections can be used alternatively. The ground (GND) need not be connected. For the network, a two-wire line may be sufficient. If a shielded cable is used, the cable shield is to be connected to the ground (GND) at both ends.

RS232 INTERFACE TO THE MODEM OR TO THE PROGRAMMING DEVICE

The programming device (PC, COM1) or a Modem can alternatively be connected to the RS232 interface.





Pin	Function
1	GND
2	DTR
3	TXD
4	RXD
5	
6	
7	DSR
8	
9	CTS

4.5. MODES OF OPERATION

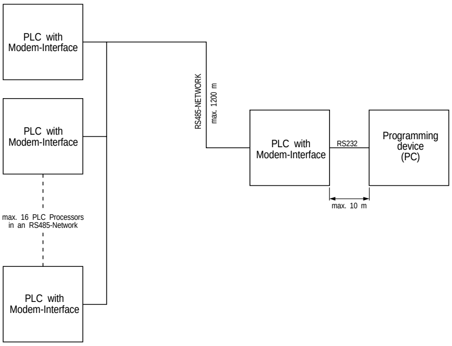
Two modes of operation need to be distinguished:

a) Operation without Modem
b) Operation with Modem

a) OPERATION WITHOUT MODEM

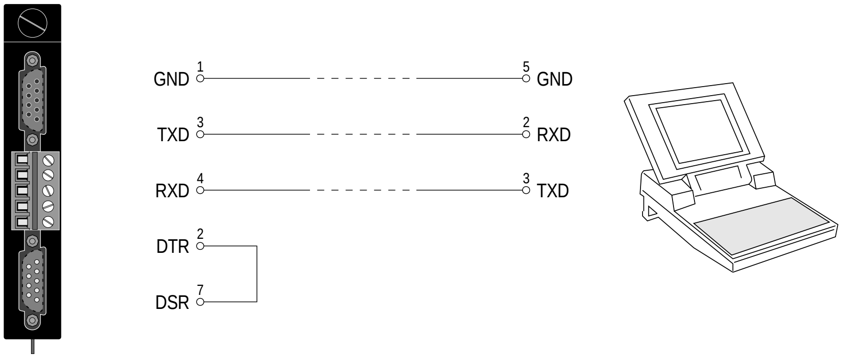
In the case of an operation without Modem, the Modem interface of a PLC. Processor is connected to the serial interface (COM1) of the programming device by means of a RS232 connection. Via a RS485 network, up to 15 other PLC. processors can be connected to the first Modem interface.

Diagram:



The operation without Modem may be advantageous, if several PLCs, which are locally separated way and/or difficult to access, should be programmed from a central programming device.

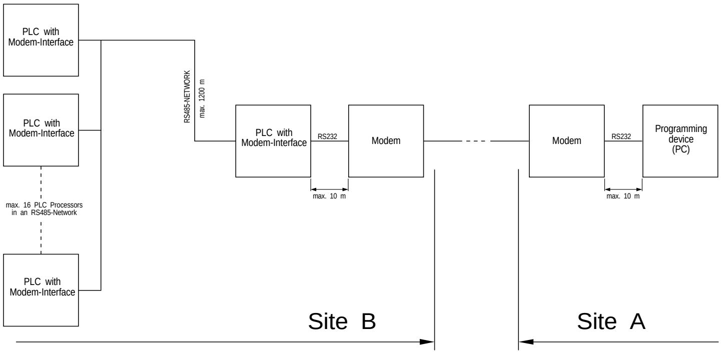
CONNECTING CABLE MODEM INTERFACE ↔ PROGRAMMING DEVICE (PC, COM1)



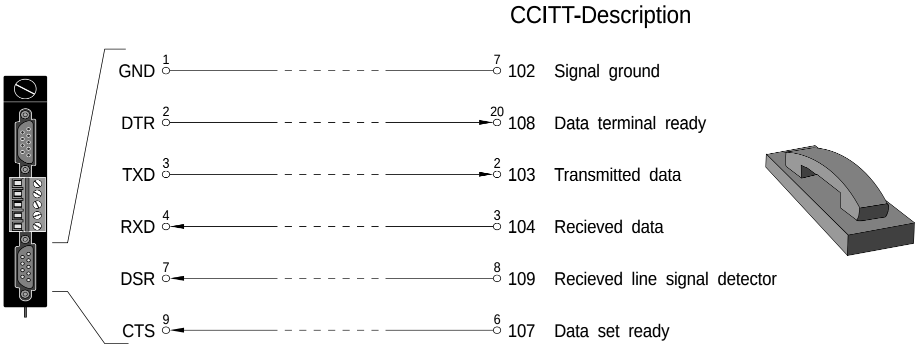
b) OPERATION WITH MODEM

This mode of operation is used for the remote diagnostic analysis via telephone and Modem. The programming device at place A is connected to the first Modem, and the second Modem at place B to the Modem interface in the processor module.

Diagram:

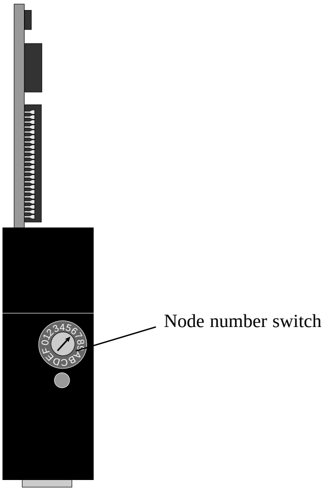


CONNECTING CABLE MODEM INTERFACE ↔ MODEM



4.6. NODE NUMBER SWITCH

On the top of the Modem interface there is a 16 position switch. The node number of the PLC. processor can be selected by means of this switch. All PLC. processors must have different node numbers. Node number 0 must be assigned to the Modem interface, which is connected to the PC or to the Modem.

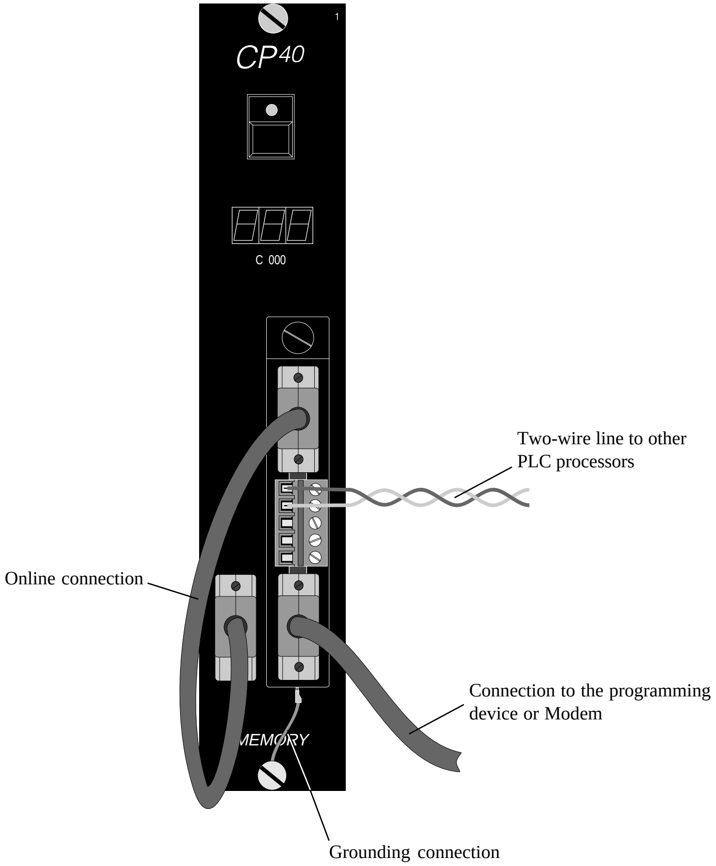


4.7. STATUS LED

On the top of the Modem interface there is a status LED: Various blinking sequences indicate certain operating states of the PLC. processor (further details can be found in: Hardware manual, section 4, subsection "Status-LED").

In the case of CPUs or PPs equipped with a 6303 processor (CP30, CP31, CP40, CP41, NTCP3#, PP40), the status LED is green. In the case of CPUs or PPs with a 6809 processor (CP60, CP80, NTCP6#, PP60), the status LED is red.

4.8. WIRING EXAMPLE


1
CP40
C 000
MEMORY
Two-wire line to other PLC processors
Online connection
Connection to the programming device or Modem
Grounding connection

5. ACCESSORIES

5.1. CABLE

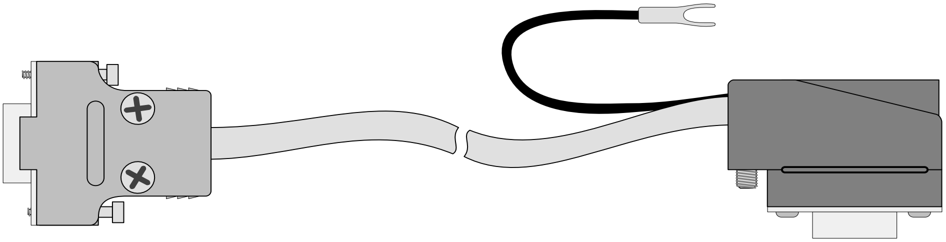
ONLINE cable for BRIFPC-0 and BRIFTO-x, PD - PLC, 2.5 m	**BRKAOL-0**
ONLINE cable for Modem interface - PLC, 0.3 m	**BRKAOL-3**
Dual Port ONLINE cable for BRIFCO-0, PD - PLC, 2.5 m	**BRKAOL-2**
FOL ONLINE cable, 4 m	**BRKAOL3-0**
FOL ONLINE cable, 50 m	**BRKAOL3-1**

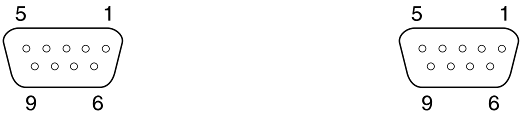
5.1.1. ONLINE cable Programming device - control

Model number: BRKAOL-0 or **BRKAOL-3**

Programming device:
- **B&R ONLINE interface** (BRIFPC-0, BRIFTO-x)
- **CENTRONICS ONLINE interface**
- **Modem ONLINE interface**

Control:
- **PD interface**





RXD	1	←	1	TXD
TXD RET	3	←	3	RXD RET
Reset RET (GND)	4	—	4	Reset RET (GND)
RXD RET	6	→	6	TXD RET
TXD	7	→	7	RXD
Reset	8	—	8	Reset

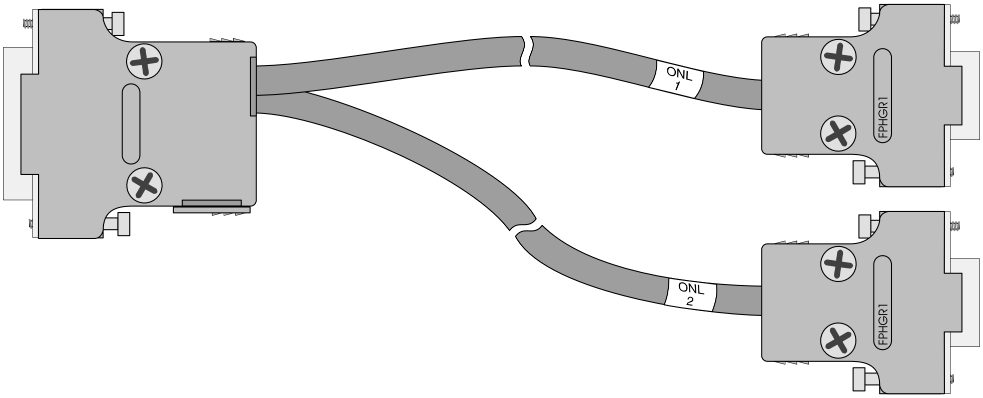
ONLINE interface   PD interface of the PLC

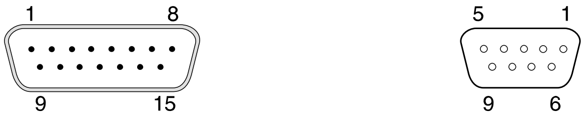
5.1.2. ONLINE cable COMPAQ Portable III - control

Model number: BRKAOL2-0

Programming device:
- **B&R ONLINE interface** (BRIFCO-0)

Control:
- **PD interface**





ONLINE 1	RXD	14	←	1	TXD	PD interface of the PLC
	TXD RET	15	←	3	RXD RET	
	Reset RET (GND)	16	—	4	Reset RET (GND)	
	RXD RET	17	→	6	TXD RET	
	TXD	18	→	7	RXD	
	Reset	19	—	8	Reset	
ONLINE 2	RXD	14	←	1	TXD	PD interface of the PLC.
	TXD RET	15	←	3	RXD RET	
	Reset RET (GND)	16	—	4	Reset RET (GND)	
	RXD RET	17	→	6	TXD RET	
	TXD	18	→	7	RXD	
	Reset	19	—	8	Reset	

COMPAQ interface

5.1.3. FOL Cable

FOL Adapter - FOL Adapter
CP80 - FOL Adapter

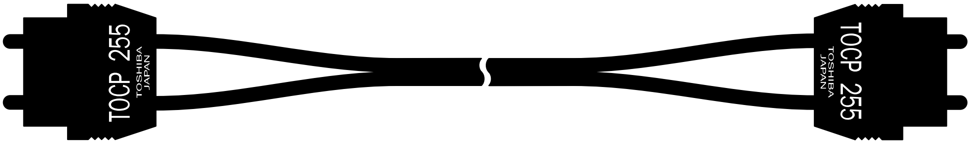
Model number: BRKAOL3-0, BRKAOL3-1

Programming device:

- **FOL Adapter**

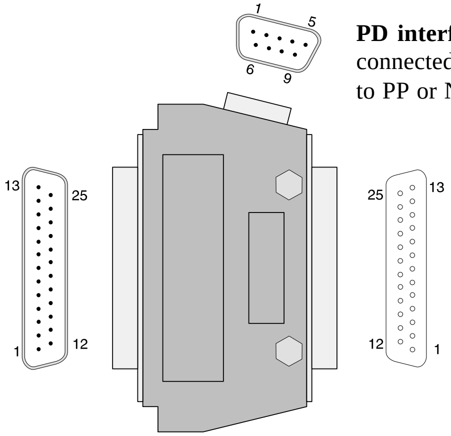
Control:

- **FOL Adapter**
- **FOL interface of theCP80**



5.2. ONLINE ADAPTER

ONLINE Adapter for PP## or NTCP6# **ECPAD1-0**



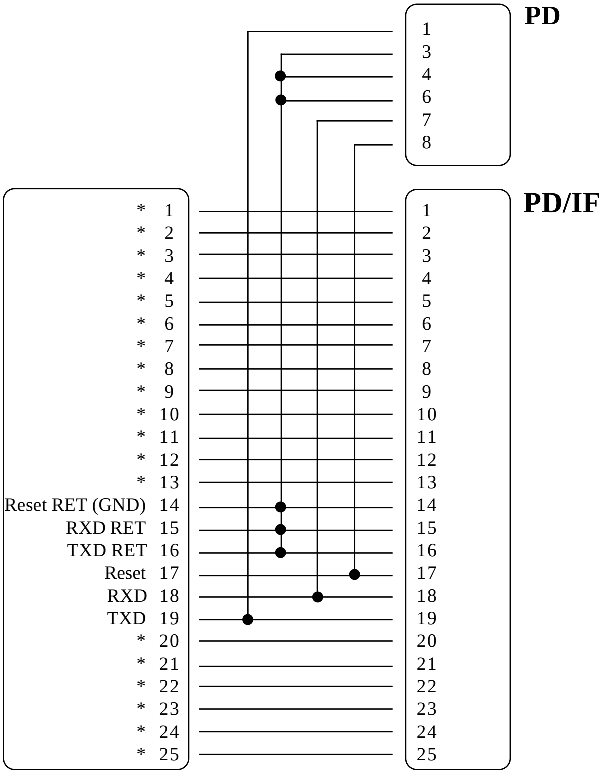
PD interface
connected directly
to PP or NTCP6#

PD/IF interface
of a parallel processor
or a NTCP6#

PD/IF interface
connected directly
to PP or NTCP6#

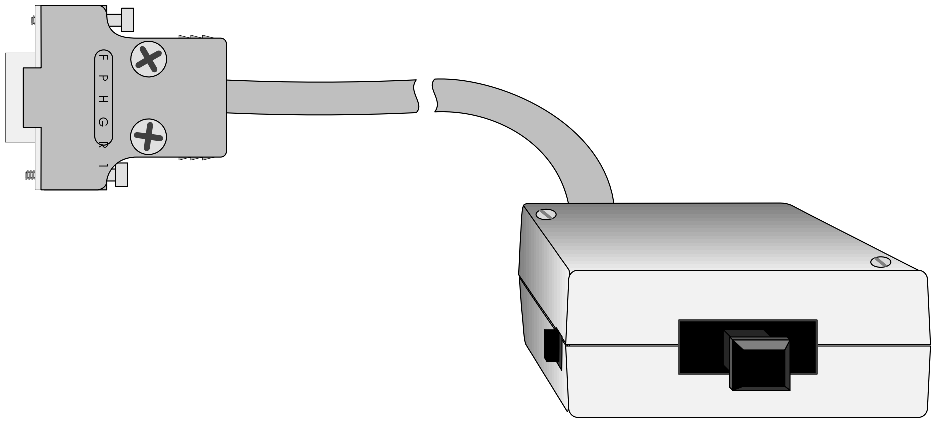
PD/IF

The configuration (*) of the user interface IF can be taken from the description of the PP## and the NTCP6#.



5.3. FOL ADAPTER

5.3.1. Model number - Description



PD ONLINE FOL Adapter, **BRADFOL-0**
with connection cable for external power supply

5.3.2. Technical data

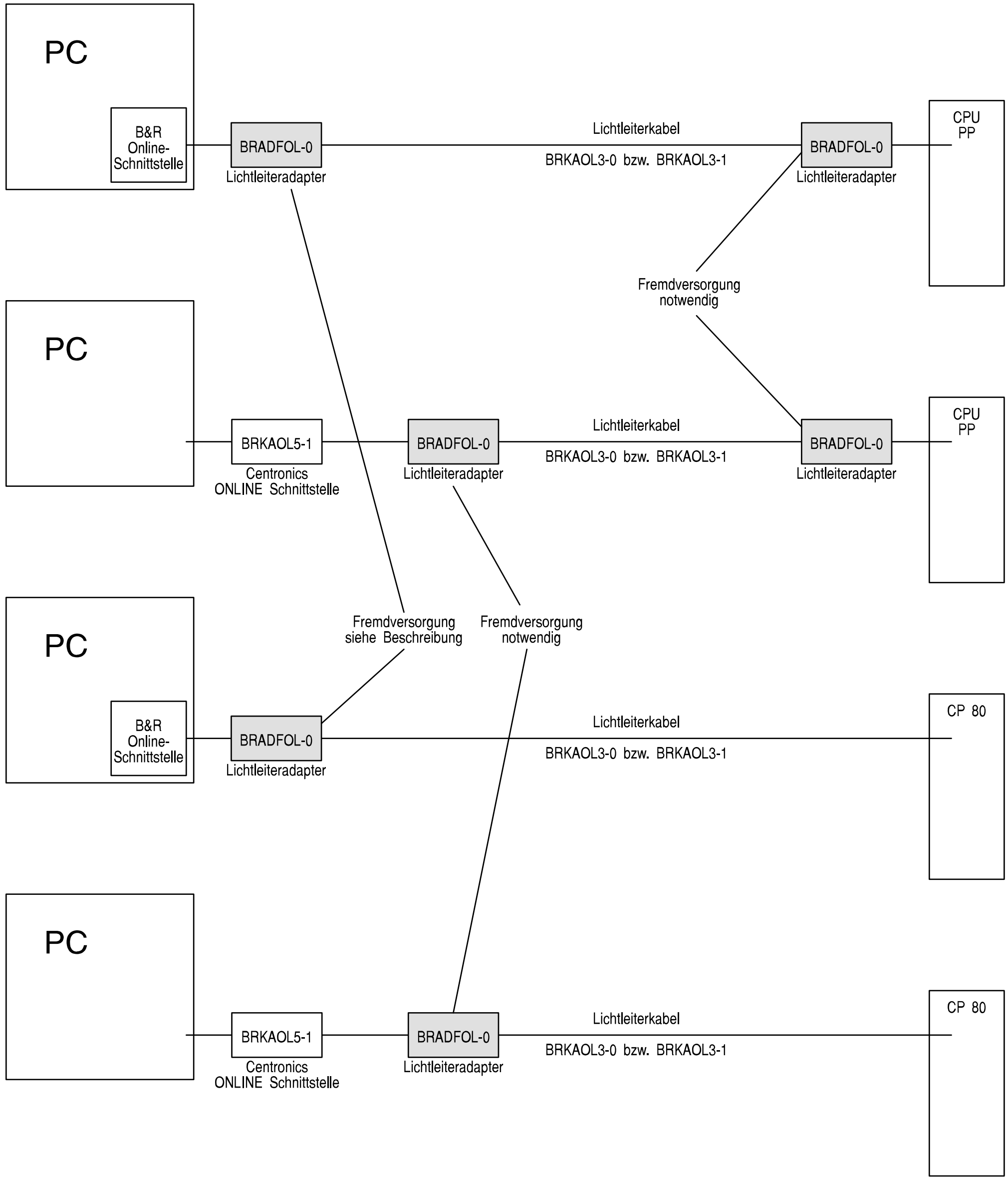
Description	PLC. Online FOL Adapter
Model number	BRADFOL-0
Voltage supply	via the PD interface of the PLC or PC, or external supply with 8 - 30 VDC / max. 250 mA
Interfaces	PD interface to the PLC or to the PC FOL connection

5.3.3. General information

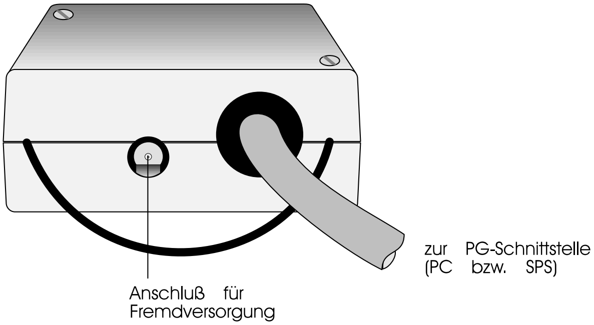
The adapter can be used for the ONLINE connection via FOL. Due to the electrical isolation, electrical or magnetic interferences are eliminated

5.3.4. Connection

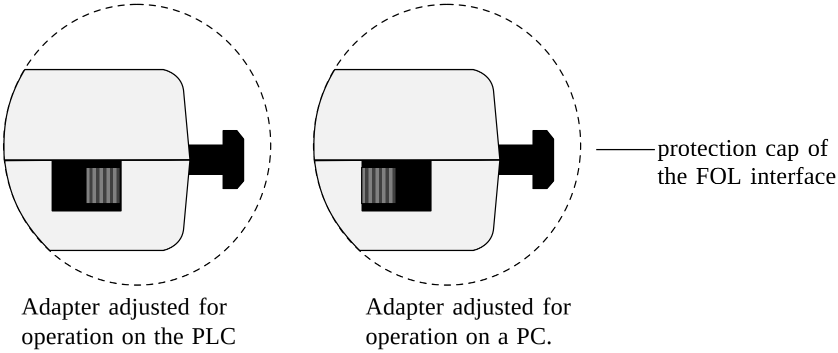
Schematic structure:



The 9-pin DSHELL connector is connected to the PD interface of the PC or the PLC:



Depending on the fact whether the adapter is connected to the ONLINE interface of a PC or to the PLC, the switch located on the adapter must be in the right position:



In order to be able to connect the FOL cable to the FOL interface (FOL = **F**ibre **O**ptic **L**ink), the protection cap has to be removed first.

External supply

By using the connection cable, which is included in the scope of delivery, the FOL adapter can be supplied with an external voltage (in most cases an industrial voltage of 24 VDC). In addition, B&R offers the power supply BRPS220904-0.

Online Interface	**external supply necessary**
BRIFPC-0	NO
BRIFTO-0	NO
BRIFTO-1	YES
BRIFTO-2	YES
BRIFCO-0	YES
BRKAOL5-1	YES

5.4. POWER SUPPLY

Description	9V Power supply
Model number	BRPS220904-0
Primary	220VAC / 50Hz
Secondary	9VDC / 400mA
Wattage	3,6W

APPENDIXA

SUMMARY - OPERATING COMMANDS

Contents:

GENERAL INFORMATION

The following explanations give an overview of the operating commands of the PROgramming SYStem:

<NAME> **<TBL_NAME>** **<LAD_NAME>**	4 character long name, the first character must be a letter.
<FNAME>	file name with a maximum length of 8 characters without extension
<FNAME.EXT>	file name with a maximum length of 8 characters with extension
<range>	a range must be defined for block operations (e.g.: COPY, MOVE, ...): **FROM<LINE/LAB/ADDR> TO<LINE/LAB/ADDR>** The range starts at FROM <...> and ends at TO<...>. **FROM<LINE/LAB/ADDR> DISP<NR>** The range starts at FROM <...> and includes DISP <...> lines. **TO<LINE/LAB/ADDR>** The range starts at the current cursor position and ends at TO<...>. **DISP<NR>** The range starts at the current cursor position and includes DISP<...> lines.
<destination>	defines range where to copy or move to. **[↵]** the destination is indicated by the cursor. **TO <LINE/LAB/ADDR> [↵]** The destination is indicated by the defined files (TO<...>).
<File type>	Defines the extension of the file name (<FNAME>)

Extension	<File type>		Level	
		A	B	C
*.PRG	PROG	●	●	●
*.SYM	SYM	●	●	●
*.TAB	TAB	●	●	●
*.LAD	LAD	●	●	●
*.MSY	MACRO SYM	●	●	●
*.MTA	MACRO TAB		●	●
*.MSL	MACRO STL			●
*.ASY	ASCII KTZ			●
*.ATA	ASCII TAB			●

PROGRAMMING SYSTEM OPERATING COMMANDS

Command	additional parameter	Function	Page
#LEARN#	**<FNAME.EXT> [↵]**	The LEARN mode is turned on. All subsequent key strokes are registered and written to <FNAME.EXT> .	C-3
#LOFF#	**[↵]**	The LEARN mode is switched off.	C-3
B&R	**[↵]**	Switch to B&R mnemonics	3-11
BAUD	**<Baud rate> [↵]**	Sets the baud rate for the serial interface. Valid baud rates are: 300, 600, 1200, 2400, 4800, 9600, 19200	3-11
CHAN	**...**	Replace / change	5-19
	EXCHANGE <range> [↵]	All program lines within the range are searched and compared with two defined search masks. If one of the two search masks corresponds with the program line, it is replaced by the other mask.	5-19
	EXCHAN? <range> [↵]	This function is the same as the CHAN described above, with the difference that with every change procedure, an inquiry is displayed in the message line.	5-19
	REPLACE <range> [↵]	All program lines within the defined range are searched and compared with the first specified search mask. If it corresponds with a program line, it is replaced by the second mask.	5-20
	REPLACE? <range> [↵]	This function is the same as REPLACE described above, with the difference that with every replace procedure, an inquiry is displayed in the message line.	5-20
	Without range definition	If no <range> is entered, all program lines are searched, except the LAD translation.	5-20
CLEAR	**...**	Deletes tables, ladder diagrams and sequence charts from the PC program memory.	
	DATA_TBL <TBL_NAME> [↵]	The table with the name <TBL_NAME> is deleted.	7-12
	LAD<LAD_NAME> [↵]	The ladder diagram with the name <LAD_NAME> is deleted.	4-42
	SEQCHART <NAME> [↵]	The sequence chart with the name <NAME> is deleted.	4-12
COPY	**<range> <destination>**	Copy STL, SYMB or TAB ranges and insert them in a certain position (destination).	5-27, 6-8, 7-8
	DISK <File type> <FNAME1> TO <FNAME2> [↵]	The file <FNAME1> is copied and provided with the name <FNAME2>.	8-20
CP40	**[↵]**	Switch from CP80 to PG1000 mode or from PC80 to PG-PC mode.	3-11
CP80	**[↵]**	Switch from PG1000 to CP80 mode or from PG-PC to PC80 mode.	3-11
DATA_TBL	**[↵]**	Call of the DATA_TBL editor and the last processed table.	7-3
	<TBL_NAME> [↵]	Call of an already existing table or a new defined DATA_TBL with the specified name <TBL_NAME>.	

DATE	**<program date> [↵]**	The current program date is replaced by the <program date> .	3-25
DDIR	**<PATH> [↵]**	The default directory of the current drive is set to <PATH> .	8-4
DEBUG	**[↵]**	S_LIST status test call.	5-31
DELETE	**<File type> <FNAME> [↵]**	Delete the file <FNAME>.	8-21
	BLOCK <range> [↵]	Delete STL, SYM or TAB ranges.	4-60,5-29, 6-9,7-8
DIR	**<File type> <PATH> [↵]**	Display of the directory contents <PATH>. Files with definite extensions can be displayed with the parameter <file type>. If this parameter is not specified, all existing files are displayed. Without entering <PATH> the contents of the default directory are displayed.	8-18
DOS	**[↵]**	Call of the MS-DOS command interpreter. With the command EXIT it can be returned to the PROgramming SYStem.	3-27
	"Command" [↵]	Executes a MS-DOS command "command" and automatically returns to the PROgramming SYStem.	
DRIVE	**<Drive name> [↵]**	The current drive is changed to <drive name>.	8-3
DUMPS	**<FNAME> [↵]**	A PLC program is stored as MOTOROLA S-record file under the name FNAME.S .	C-10
EEPROM	**[↵]**	EEPROM menu call.	5-49
END	**[↵]**	quits the PROgramming SYStem.	3-20
EXE	**<FNAME.EXT> [↵]**	The LEARN-File <FNAME.EXT> is executed.	C-4
FBK	**<NAME> [↵]**	Accesses the FBK definition editor for the definition of the FBK with the name <NAME>.	10-8
FIND	**<range> [↵]**	Searches STL program lines, that correspond with the specified search mask. If no <range> is specified, the entire STL program is searched.	5-24
FINDL	**<range> [↵]**	All program lines that correspond with the specified search mask, are listed on screen.	5-26
FINDP	**<range> [↵]**	All program lines that correspond with the specified search mask, are printed.	5-26
FORMAT	**<Drive name:> [↵]**	Formats the disk in the specified drive.	8-21
FREE	**<ADDR/SYMBOLS> [↵]**	Searches for addresses that are not allocated with symbols, beginning with address <ADD/SYM>. If no start address is entered, the search starts at R 000.	6-9
GLOBCHAN	**...**	Global exchange/replace with all addresses in the entire S_LIST program including the LAD translation.	5-23
	EXCHANGE[↵]	All program lines are changed and compared with two defined search masks. If a search mask corresponds with a program line, it is replaced by the other mask.	
	REPLACE[↵]	All program lines are searched and compared with the first defined search mask. If it corresponds with the program line, it is replaced by the second mask.	

LAD	**[↵]**	Call of the last processed LAD.	4-13
	<LAD_NAME> [↵]	Call of the LAD with the name <LAD_NAME>.	
LANG	**<Language> [↵]**	selection of the dialogue language:	3-11
	GR	German	
	USA	English	
	F	French	
	ITA	Italian	
	SP	Spanish	
LARGE	**[↵]**	Program memory definition on 4766 program lines.	3-26
LDMO	**[↵]**	Call of the last LDMO.	4-44
	<LAD_NAME> [↵]	LDMO call (LAD with the name <LAD_NAME>).	
LIST	**[↵]**	Call of the printer menu, where the screen display is defined.	9-3, 9-19
	SYM <range> [↵]	Output of the SYMBOL range. If no <range> is define, all SYMBOLs are output.	9-19
	LAB [↵]	Output of a list of all labels that are used in the program .	9-19
	PROG <range> [↵]	Output of the specified S_LIST range. If no <range> is entered, the entire S_LIST is distributed.	9-19
	DATA_TBL <TBL_NAME> [↵]	Output of the table <TBL_NAME>. If no DATA_TBL name is specified, all tables are distributed.	9-19
LOAD	**...**	Load programs or program parts	8-9
	ASCII SYMBOL <FNAME> [↵]	Load a SYMBOL, that is saved in ASCII format.	8-15
	ASCII DATA_TBL <FNAME> [↵]	Load a DATA_TBL that is stored in ASCII format .	8-15
	LAD <FNAME> TO <LAD_NAME> [↵]	The ladder diagram <FNAME> is loaded and included in the existing program with the LAD name <LAD_NAME> .	8-14
	SYMBOL <FNAME> [↵]	The SYMBOL file <FNAME> is loaded. All SYMBOLs, that existed before the loading, are deleted.	8-12
	MACRO S_LIST <FNAME> <destination> [↵]	The S_LIST macro <FNAME> is loaded and included in the existing S_LIST of the specified location <LINE/LAB> .	8-12
	MACRO SYMBOL <FNAME> <destination> [↵]	The SYMBOL macro <FNAME> is loaded and included in the existing SYMB at the specified position <ADDR> .	8-12
	MACRO DATA_TBL <FNAME> <destination> [↵]	The DATA_TBL macro <FNAME> is loaded and in the existing DATA_TBL at the specified position <LINE> .	8-12

	PG1000 [↵]	The PROgramming SYStem switches to the PG1000 mode and waits until the programming device PG1000 submits a program through the serial interface to the PC.	3-22
	PLC [↵]	The program that is in the PLC RAM, is loaded in the work memory of the PC.	5-52
	PROG <FNAME> [↵]	The program <FNAME> is loaded. The program in the memory is deleted.	8-9
	TAB <TBL_NAME> <FNAME> [↵]	The table <FNAME> is loaded and the loaded table receives the name <TBL_NAME>.	8-11
MAESTRO	**[↵]**	Emulation of a PROVIT 500/600 with ext. ASCII-keyboard.	C-9
MOVE	**<range> <destination> [↵]**	Move a S_LIST, SYMBOL or DATA_TBL range.	5-30, 6-8, 7-8
NAME	**<Program name> [↵]**	The current program name is replaced by <program name> .	3-24
NEW	**[↵]**	Deleting the program in the current window, that is in the PC working memory.	3-25
ONL	**<interface> [↵]**	Selection of the online interface that should be used, to program a PLC:	3-26
	B&R0 ...	Selection of the installed B&R online interface.	
	LPT1 ...	Selection of the Centronics online-interface.	
	COM1 ...	Modem selection for the serial interface COM1.	
P-VERS	**[↵]**	Information display about the program in the PLC.	5-23
PAR	**[↵]**	Selection of the parallel interface as printer interface.	9-20
PMEN	**[↵]**	Call of the printer menu, where it is defined, what should be printed.	9-3
PRINT	**[↵]**	Call of the printer menu, where it should be defined, what needs to be printed.	9-3,9-17
	SYMBOL <range> [↵]	Print of the defined SYM range. If no <range> is entered, all SYMBs are printed.	9-18
	LAB [↵]	Print of a list of all labels used in a program.	9-18
	PROG <range> [↵]	Print of the specified S_LIST range. If no <range> is entered, the entire S_LIST is printed.	9-17
	DATA_TBL <TBL_NAME> [↵]	Table print out <TBL_NAME>. If no TAB name is entered, all tables are printed.	9-18
RENAME	**<File type> <FNAME1> TO <FNAME2> [↵]**	Rename the file FNAME1 to FNAME2.	8-21
RNT	**[↵]**	The program in the current window is checked for missing labels and exceeding of the maximum branch width.	5-48
RUN	**[↵]**	The program is transmitted to PLC RAM and started. The S_LIST status test is called.	4-11,5-30

RUN P	**[↵]**	This command is the same as the RUN command, with the difference, that the program is protected when submitted to the PLC. I.e.: The program can not be loaded to the PLC with LOAD PLC.	5-48
S_LIST	**[↵]**	S_LIST editor call.	C-3
	<LINE/LAB> [↵]	The input line is set to the specified S_LIST line <LINE/LAB>.	4-5
SAVE	**[↵]**	The entire program is saved with the name, that was previously defined with a SAVE or LOAD command.	8-5
	<FNAME> ASCII SYMBOL <range> [↵]	Store a SYMBOL range in ASCII format.	8-8
	<FNAME> ASCII DATA_TBL <TBL_NAME> [↵]	Store a DATA_TBL range in ASCII format.	8-8
	<FNAME> LAD <LAD_NAME> [↵]	The ladder diagram <LAD_NAME> is saved.	8-6
	<FNAME> SYMBOL [↵]	All defined symbols are saved.	8-6
	<FNAME> MACRO S_LIST <range> [↵]	The defined S_LIST range is saved as a macro.	8-7
	<FNAME> MACRO SYMBOL <range> [↵]	The defined SYMBOL range is saved as a macro.	8-7
	<FNAME> MACRO DATA_TBL <range> [↵]	The defined SYMBOL range is saved as a macro.	8-7
	PG1000 <Mnemonics> [↵]	Submits a program to the PG1000. There are two possibilities for <Mnemonics> : • INTER (MOTOROLA-Mnemonics) • GERMAN (B&R Mnemonics)	3-23
	<FNAME> PROG [↵]	The entire program is saved.	8-6
	<FNAME> DATA_TBL <TBL_NAME> [↵]	The table <TBL_NAME> is stored.	8-6
SDIR	**<PATH> [↵]**	The standard directory of the current drive is set to <PATH>.	8-4
SER	**[↵]**	The serial interface COM1 is defined as printer interface.	9-20
SEQCHART	**[↵]**	SEQCHART editor call and call of the last edited SEQCHARTs.	4-5
	<NAME> [↵]	Call of an existing or new SEQCHART with the specified name <NAME>.	4-5
SMALL	**[↵]**	Program memory definition on 2719 program lines.	3-26
SYMBOL	**[↵]**	SYMBOLS editor call.	6-4
	<ADDR> [↵]	The input line is set to a particular address.	
TERM	**[↵]**	PC can be used as input terminal for the operation of a PROVIT 900/1000.	C-6
UDIR	**<PATH> [↵]**	The user directory of the current drive is set to <PATH>.	8-4
WINDOW	**[↵]**	Window definition menu call.	3-18
XFER	**[↵]**	Program submittal to the PLC, without interruption of the program running in the PLC.	5-55
XREF	**[↵]**	Output of a list of all cross references on screen.	9-19

APPENDIX B

STL-INSTRUCTIONS

Contents:

SYNTAX EXPLANATIONS

The STL instruction overview includes various information:

- Mnemonic (Motorola, B&R, B&R short version)
- short function description
- operating mode, in which commands are available
- addresssing modes with execution times and opcode length
- address preselections (Motorola and B&R)
- status register (status after command execution)

The table is divided into two pages, i. e. the left and the right page belong together. To increase the readability of the STL overview, the left and the right page were provided with “consecutive numbers”.

Left Page

Cont. Nr.	Mnemonics			Operation	Mode			
	Motorola	B&R	short		PG1000	PG-PC	CP80	PC80
1	ABA	A+B		A + B ⇨ A	❍	❍	❍	❍
2	ABX	B+R		B + X ⇨ X	❍	❍	❍	❍
3	ADCA	ADD	A, ++	A + M + C ⇨ A	❍	❍	❍	❍
4	ADCB	++B		B + M + C ⇨ B	❍	❍	❍	❍

A	Accumulator A (8 bit)	M	Memory, 8 bit
B	Accumulator B (8 bit)	r_8	8-bit Register (A, B, CCR, DP)
D	Accumulator D (16 Bit; A = MSB, B = LSB)	r_{16}	16-bit Register (D, R, Y, SP!, SPU)
		IIB	Interrupt Inhibit Bit
SR	Condition Code Register (8 bit)	∧	Log. AND
	N Negative Flag (data bit 3)	∨	Log. OR
	C Carry Flag (data bit 2)	⊕	Log. EXOR
	Z Zero Flag (data bit 0)	EA	Effective Address
X	Index register X (16 bit)	IMM	Test Mask
Y	Index register Y (16 bit)	d0	Data bit 0 of Accumulator A
SP!	System Stack pointer (16 bit)		
SPU	User Stack pointer (16 bit)		

❍ STL instruction is available in this expansion level

Right page

Cont. Nr.	Addressing Modes 6303/6809												Address Preselections															Condition Code					
	IMPL.		DIR.		EXT.		IMMED.		IND.		REL.		I	O	F	S	T	#	P	R	X	Y	D	U	!	B	G	6303			6809		
	~	#	~	#	~	#	~	#	~	#	~	#	E	A	M	F	Z	#	P	C	I	Y	D	U	!	B	G	N	Z	C	N	Z	C
1	1/25	1/3																										●	●	●	●	●	●
2	1/3	1/1																										❍	❍	❍	❍	❍	❍
3			-/4	-/2	4/5	3/3	2/2	2/2	$4/4^+$	$2/2^+$			❍	❍	❍	❍	❍	❍	❍	❍	❍	●	●	●	●	●	●	●	●	●	●	●	●
4			-/4	-/2	4/5	3/3	2/2	2/2	$4/4^+$	$2/2^+$			❍	❍	❍	❍	❍	❍	❍	❍	❍	●	●	●	●	●	●	●	●	●	●	●	●

IMPL.	Implied
DIR.	Direct
EXT.	Extended
IMMED.	Immediate
IND.	Indexed (Ind. indexed)
REL.	Relative
~	Processor Cycles
#	Bytes

-/4	4/5		$2/2^+$
not available at 6303	6303	6809	2 or more cycles/bytes

I	E	1 Bit Input
O	A	1 Bit Output
F	M	1 Bit Flag
S	F	Start Timer
T	Z	1 Bit Timer
#	#	Immediate
P	P	Peripheral address
R	C	8 Bit Register
X	I	Index register X
Y	Y	Index register Y
D	D	Direct page addressing
U	U	User Stack Pointer
!	!	System Stack Pointer
B	B	Block Memory in PP60
G	G	Global RAM in PP60
	❍	available
	●	available at 6809 with PROSYS V 5.00 or later
	■	available at 6303

N	Negative Flag
Z	Zero Flag
C	Carry Flag
❍	not changed
●	changes according to operation
▲	set
▼	reset
✕	not valid

Cont. Nr.	Mnemonics			Operation	Mode			
	Motorola	B&R	short		PG1000	PG-PC	CP80	PC80
1	ABA	A+B		A + B ⇨ A	❍	❍	❍	❍
2	ABX	B+R		B + X ⇨ X	❍	❍	❍	❍
3	ADCA	ADD	A, ++	A + M + C ⇨ A	❍	❍	❍	❍
4	ADCB	++B		B + M + C ⇨ B	❍	❍	❍	❍
5	ADDA	+		A + M ⇨ A	❍	❍	❍	❍
6	ADDB	+B		B + M ⇨ B	❍	❍	❍	❍
7	ADDD	+D		A:B + M:M+1 ⇨ A:B	❍	❍	❍	❍
8	AIM	AIM		M ∧ IMM ⇨ M		❍		
9	ANDA	UND	U	A ∧ M ⇨ A	❍	❍	❍	❍
10	ANDB	UB		B ∧ M ⇨ B	❍	❍	❍	❍
11	ASL	SL		Shift M left	❍	❍	❍	❍
12	ASLA	SLA		Shift A left	❍	❍	❍	❍
13	ASLB	SLB		Shift B left	❍	❍	❍	❍
14	ASLD	SLD		Shift D left	❍	❍	❍	❍
15	BCC	JC0		Branch, if C = 0	❍	❍	❍	❍
16	BCCL	JC0L		Branch, if C = 0				❍
17	BEQ	SP0	J0	Branch, if Z = 1	❍	❍	❍	❍
18	BEQL	SP0L		Branch, if Z = 1				❍
19	BHI	SP>	J>	Branch, if C + Z = 0	❍	❍	❍	❍
20	BHIL	SP>L		Branch, if C + Z = 0				❍
21	BITA	B		A ∧ M	❍	❍	❍	❍
22	BITB	BB		B ∧ M	❍	❍	❍	❍
23	BCS	SP<	J<	Branch, if C = 1	❍	❍	❍	❍
24	BCSL	SP<L		Branch, if C = 1				❍
25	BLS	J<=		Branch, if C + Z = 1	❍	❍	❍	❍
26	BLSL	J<=L		Branch, if C + Z = 1				❍
27	BMI	J-		Branch, if N = 1	❍	❍	❍	❍
28	BMIL	J-L		Branch, if N = 1				❍
29	BNE	SN0	J1	Branch, if Z = 0	❍	❍	❍	❍
30	BNEL	SN0L		Branch, if Z = 0				❍
31	BPL	J+		Branch, if N = 0	❍	❍	❍	❍
32	BPLL	J+L		Branch, if N = 0				❍
33	CBA	AVB		A - B	❍	❍	❍	❍
34	CLC	CLC		0 ⇨ C	❍	❍	❍	❍
35	CLI	CLI		0 ⇨ IIB	❍	❍	❍	❍

Cont. Nr.	Addressing Modes 6303/6809												Address Preselections															Condition Code					
	IMPL.		DIR.*		EXT.		IMMED.		IND.		REL.		I	O	F	S	T	#	P	R	X	Y	D	U	!	B	G	6303			6809		
	~	#	~	#	~	#	~	#	~	#	~	#	E	A	M	F	Z	#	P	C	I	Y	D	U	!	B	G	N	Z	C	N	Z	C
1	1/25	1/3																										●	●	●	●	●	●
2	1/3	1/1																										❍	❍	❍	❍	❍	❍
3			-/4	-/2	4/5	3/3	2/2	2/2	$4/4^+$	$2/2^+$			❍	❍	❍	❍	❍	❍	❍	❍	❍	●	●	●	●	●	●	●	●	●	●	●	●
4			-/4	-/2	4/5	3/3	2/2	2/2	$4/4^+$	$2/2^+$			❍	❍	❍	❍	❍	❍	❍	❍	❍	●	●	●	●	●	●	●	●	●	●	●	●
5			-/4	-/2	4/5	3/3	2/2	2/2	$4/4^+$	$2/2^+$			❍	❍	❍	❍	❍	❍	❍	❍	❍	●	●	●	●	●	●	●	●	●	●	●	●
6			-/4	-/2	4/5	3/3	2/2	2/2	$4/4^+$	$2/2^+$			❍	❍	❍	❍	❍	❍	❍	❍	❍	●	●	●	●	●	●	●	●	●	●	●	●
7			-/6	-/2	5/7	3/3	3/4	3/3	$5/6^+$	$2/2^+$								❍	❍	❍	❍	●	●	●	●	●	●	●	●	●	●	●	●
8									7/-	3-								■										●	●	❍			
9			-/4	-/2	4/5	3/3	2/2	2/2	$4/4^+$	$2/2^+$			❍	❍	❍	❍	❍	❍	❍	❍	❍	●	●	●	●	●	●	●	●	❍	●	●	❍
10			-/4	-/2	4/5	3/3	2/2	2/2	$4/4^+$	$2/2^+$			❍	❍	❍	❍	❍	❍	❍	❍	❍	●	●	●	●	●	●	●	●	❍	●	●	❍
11			-/6	-/2	6/7	3/3			$6/6^+$	$2/2^+$				❍	❍	❍			❍	❍	❍	●	●	●	●		●	●	●	●	●	●	●
12	1/2	1/1																										●	●	●	●	●	●
13	1/2	1/1																										●	●	●	●	●	●
14	1/4	1/2																										●	●	●	●	×	●
15											3/3	2/2																❍	❍	❍	❍	❍	❍
16											$-/5^+$	-/4																			❍	❍	❍
17											3/3	2/2																❍	❍	❍	❍	❍	❍
18											$-/5^+$	-/4																			❍	❍	❍
19											3/3	2/2																❍	❍	❍	❍	❍	❍
20											$-/5^+$	-/4																			❍	❍	❍
21			-/4	-/2	4/5	3/3	2/2	2/2	$4/4^+$	$2/2^+$			❍	❍	❍	❍	❍	❍	❍	❍	❍	●	●	●	●	●	●	●	●	❍	●	●	❍
22			-/4	-/2	4/5	3/3	2/2	2/2	$4/4^+$	$2/2^+$			❍	❍	❍	❍	❍	❍	❍	❍	❍	●	●	●	●	●	●	●	●	❍	●	●	❍
23											3/3	2/2																❍	❍	❍	❍	❍	❍
24											$-/5^+$	-/4																			❍	❍	❍
25											3/3	2/2																❍	❍	❍	❍	❍	❍
26											$-/5^+$	-/4																			❍	❍	❍
27											3/3	2/2																❍	❍	❍	❍	❍	❍
28											$-/5^+$	-/4																			❍	❍	❍
29											3/3	2/2																❍	❍	❍	❍	❍	❍
30											$-/5^+$	-/4																			❍	❍	❍
31											3/3	2/2																❍	❍	❍	❍	❍	❍
32											$-/5^+$	-/4																			❍	❍	❍
33	1/25	1/3																										●	●	●	●	●	●
34	1/3	1/2																										❍	❍	▼	❍	❍	▼
35	1/3	1/2																										❍	❍	❍	❍	❍	❍

Cont. Nr.	Mnemonics			Operation	Mode			
	Motorola	B&R	Short		PG1000	PG-PC	CP80	PC80
36	CLR	CLR	C	00 ⇨ M	❍	❍	❍	❍
37	CLRA	CLA		00 ⇨ A		❍		❍
38	CLRB	CLB		00 ⇨ B		❍		❍
39	CMPA	CMP	V	A - M	❍	❍	❍	❍
40	CMPB	VB		B - M	❍	❍	❍	❍
41	COM	K		M ⊕ #$FF ⇨ M (Negation)	❍	❍	❍	❍
42	COMA	COA	KA	A ⊕ #$FF ⇨ A (Negation)		❍		❍
43	COMB	COB	KB	B ⊕ #$FF ⇨ B (Negation)		❍		❍
44	CPX	VR		X - M:M+1	❍	❍	❍	❍
45	CPX#	VRK		X - M:M+1	❍	❍	❍	❍
46	CPY	VY		Y - M:M+1				❍
47	CPY#	VYK		Y - M:M+1				❍
48	DAA	DK		Decimal adjust of A	❍	❍	❍	❍
49	DEC	DEC	D	M - 1 ⇨ M	❍	❍	❍	❍
50	DECA	DA		A - 1 ⇨ A	❍	❍	❍	❍
51	DECB	DB		B - 1 ⇨ B	❍	❍	❍	❍
52	DES	DS		SP! - 1 ⇨ SP!	❍	❍	❍	❍
53	DEX	DR		X - 1 ⇨ X	❍	❍	❍	❍
54	EIM	EIM		M ⊕ IMM ⇨ M		❍		
55	END	END		End of Program, start program line 0000	❍	❍	❍	❍
56	EORA	EXO	E	A ⊕ M ⇨ A	❍	❍	❍	❍
57	EORB	EB		B ⊕ M ⇨ B	❍	❍	❍	❍
58	EXG	EXG		r_8 ⟺ r_8 or r_{16} ⟺ r_{16}				❍
59	INC	INC		M + 1 ⇨ M	❍	❍	❍	❍
60	INCA	IA		A + 1 ⇨ A	❍	❍	❍	❍
61	INCB	IB		B + 1 ⇨ B	❍	❍	❍	❍
62	INS	IS		SP! + 1 ⇨ SP!	❍	❍	❍	❍
63	INX	IR		X + 1 ⇨ X	❍	❍	❍	❍
64	JMP	SPI	J	Unconditional jump	❍	❍	❍	❍
65	JSR	SPU	JU, JS	Jump to subroutine	❍	❍	❍	❍
66	LDAA	LAD	L	M ⇨ A	❍	❍	❍	❍
67	LDAB	LB		M ⇨ B	❍	❍	❍	❍
68	LDD	LD		M:M+1 ⇨ D	❍	❍	❍	❍
69	LDK	LDK		M:M+1 ⇨ D		❍		❍

Cont. Nr.	Addressing Modes 6303/6809												Address Preselections															Condition Code					
	IMPL.		DIR.		EXT.		IMMED.		IND.		REL.		I	O	F	S	T	#	P	R	X	Y	D	U	!	B	G	6303			6809		
	~	#	~	#	~	#	~	#	~	#	~	#	E	A	M	F	Z	#	P	C	I	Y	D	U	!	B	G	N	Z	C	N	Z	C
36			-/6	-/2	5/7	3/3			5/6$^+$	2/2$^+$				❍	❍	❍			❍	❍	❍	●	●	●	●		●	▼	▲	▼	▼	▲	▼
37	1/2	1/1																										▼	▲	▼	▼	▲	▼
38	1/2	1/1																										▼	▲	▼	▼	▲	▼
39			-/4	-/2	4/5	3/3	2/2	2/2	4/4$^+$	2/2$^+$			❍	❍	❍	❍	❍	❍	❍	❍	❍	●	●	●	●	●	●	●	●	●	●	●	●
40			-/4	-/2	4/5	3/3	2/2	2/2	4/4$^+$	2/2$^+$			❍	❍	❍	❍	❍	❍	❍	❍	❍	●	●	●	●	●	●	●	●	●	●	●	●
41			-/6	-/2	6/7	3/3			6/6$^+$	2/2$^+$				❍	❍	❍			❍	❍	❍	●	●	●	●		●	●	●	▲	●	●	▲
42	1/2	1/1																										●	●	▲	●	●	▲
43	1/2	1/1																										●	●	▲	●	●	▲
44			-/6	-/2	5/7	3/3			5/6$^+$	2/2$^+$									❍	❍	❍	●		●	●	●	●	●	●	●	●	●	●
45							3/4	3/3					❍	❍	❍	❍	❍		❍	❍						●	●	●	●	●	●	●	●
46			-/6	-/2	-/7	-/3			-/6$^+$	-/2$^+$									●	●	●	●		●	●	●	●				●	●	●
47							-/4	-/3					●	●	●	●	●		●	●						●	●				●	●	●
48	2/2	1/1																										●	●	●	●	●	●
49			-/6	-/2	6/7	3/3			6/6$^+$	2/2$^+$				❍	❍	❍			❍	❍	❍	●	●	●	●		●	●	●	❍	●	●	❍
50	1/2	1/1																										●	●	❍	●	●	❍
51	1/2	1/1																										●	●	❍	●	●	❍
52	1/6	1/2																										❍	❍	❍	❍	❍	❍
53	1/6	1/2																										❍	●	❍	❍	●	❍
54									7/-	3/-								■										●	●	❍			
55																												●	●	●	●	●	●
56			-/4	-/2	4/5	3/3	2/2	2/2	4/4$^+$	2/2$^+$			❍	❍	❍	❍	❍	❍	❍	❍	❍	●	●	●	●	●	●	●	●	❍	●	●	❍
57			-/4	-/2	4/5	3/3	2/2	2/2	4/4$^+$	2/2$^+$			❍	❍	❍	❍	❍	❍	❍	❍	❍	●	●	●	●	●	●	●	●	❍	●	●	❍
58	-/7	-/2																													❍	❍	❍
59			-/6	-/2	6/7	3/3			6/6$^+$	2/2$^+$				❍	❍	❍			❍	❍	❍	●	●	●	●		●	●	●	❍	●	●	❍
60	1/2	1/1																										●	●	❍	●	●	❍
61	1/2	1/1																										●	●	❍	●	●	❍
62	1/6	1/2																										❍	❍	❍	❍	❍	❍
63	1/6	1/2																										❍	●	❍	❍	●	❍
64					3/4	3/3			3/6$^+$	2/2$^+$											❍	●		●	●			❍	❍	❍	❍	❍	❍
65					6/8	3/3			5/7$^+$	2/2$^+$											❍	●		●	●			❍	❍	❍	❍	❍	❍
66			-/4	-/2	4/5	3/3	2/2	2/2	4/4$^+$	2/2$^+$			❍	❍	❍	❍	❍	❍	❍	❍	❍	●	●	●	●	●	●	●	●	❍	●	●	❍
67			-/4	-/2	4/5	3/3	2/2	2/2	4/4$^+$	2/2$^+$			❍	❍	❍	❍	❍	❍	❍	❍	❍	●	●	●	●	●	●	●	●	❍	●	●	❍
68			-/5	-/2	5/6	3/3	3/3	3/3	5/5$^+$	2/2$^+$			❍	❍	❍	❍	❍	❍	❍	❍	❍	●	●	●	●	●	●	●	●	❍	●	●	❍
69							3/3	3/3					❍	❍	❍	❍	❍		❍	❍						●	●	●	●	❍	●	●	❍

Cont. Nr.	Mnemonics			Operation	Mode			
	Motorola	B&R	Short		PG1000	PG-PC	CP80	PC80
70	LDL	LDL		M:M+1 ⇨ D		❍		❍
71	LDS	LS		M:M+1 ⇨ SP!	❍	❍	❍	❍
72	LDX	LR		M:M+1 ⇨ X	❍	❍	❍	❍
73	LDX#	LRK		M:M+1 ⇨ X	❍	❍	❍	❍
74	LDXL	LRL		M:M+1 ⇨ X		❍		❍
75	LDY	LY		M:M+1 ⇨ Y				❍
76	LDY#	LYK		M:M+1 ⇨ Y				❍
77	LDYL	LYL		M:M+1 ⇨ Y				❍
78	LEA!	LE!		EA ⇨ SP!				❍
79	LEAU	LEU		EA ⇨ SPU				❍
80	LEAX	LER		EA ⇨ X			❍	❍
81	LEAY	LEY		EA ⇨ Y				❍
82	LSR	SR		Shift M right	❍	❍	❍	❍
83	LSRA	SRA		Shift A right	❍	❍	❍	❍
84	LSRB	SRB		Shift B right	❍	❍	❍	❍
85	LSRD	SRD		Shift D right	❍	❍	❍	❍
86	MUL	A*B		A ✕ B ⇨ D	❍	❍	❍	❍
87	NOP	NOP	N	No Operation	❍	❍	❍	❍
88	OIM	OIM		M ∨ IMM ⇨ M		❍		
89	ORAA	OD	O	A ∨ M ⇨ A	❍	❍	❍	❍
90	ORAB	OB		B ∨ M ⇨ B	❍	❍	❍	❍
91	PRS	PRS	P	If d0 = 1, 1 ⇨ M	❍	❍	❍	❍
92	PSH	PSH		Put register (r_8 or r_{16}) on the stack (SP! or SPU) r_8 ⇨ (SP!); SP! - 1 ⇨ SP! ... r_{16} ⇨ (SP!:SP!+1); SP! - 2 ⇨ SP! ...				❍
93	PSHA	ANS		A ⇨ (SP!); SP! - 1 ⇨ SP!	❍	❍	❍	❍
94	PSHB	BNS		B ⇨ (SP!); SP! - 1 ⇨ SP!	❍	❍	❍	❍
95	PSHX	RNS		X ⇨ (SP!:SP!+1); SP! - 2 ⇨ SP!	❍	❍	❍	❍
96	PUL	PUL		Take register (r_8 or r_{16}) from stack (SP! oder SPU) SP! + 1 ⇨ SP!; (SP!) ⇨ r_8 ... SP! + 2 ⇨ SP!; (SP!:SP!+1) ⇨ r_{16} ...				❍
97	PULA	AVS		SP! + 1 ⇨ SP!; (SP!) ⇨ A	❍	❍	❍	❍
98	PULB	BVS		SP! + 1 ⇨ SP!; (SP!) ⇨ B	❍	❍	❍	❍
99	PULX	RVS		SP! + 2 ⇨ SP!; (SP!:SP!+1) ⇨ X	❍	❍	❍	❍

Cont. Nr.	Addressing Modes 6303/6809												Address Preselections															Condition Code					
	IMPL.		DIR.		EXT.		IMMED.		IND.		REL.		I	O	F	S	T	#	P	R	X	Y	D	U	!	B	G	6303			6809		
	~	#	~	#	~	#	~	#	~	#	~	#	E	A	M	F	Z	#	P	C	I	Y	D	U	!	B	G	N	Z	C	N	Z	C
70							3/3	3/3																				●	●	❍	●	●	❍
71			-/6	-/3	5/7	3/4			5/6$^+$	2/3$^+$									❍	❍	❍	●		●	●		●	●	●	❍	●	●	❍
72			-/5	-/2	5/6	3/3			5/5$^+$	2/2$^+$									❍	❍	❍	●		●	●	●	●	●	●	❍	●	●	❍
73							3/3	3/3					❍	❍	❍	❍	❍		❍	❍						●	●	●	●	❍	●	●	❍
74							3/3	3/3																				●	●	❍	●	●	❍
75			-/5	-/2	-/6	-/3			-/5$^+$	-/2$^+$									●	●	●	●		●	●	●	●				●	●	❍
76							3/3	3/3					●	●	●	●	●		●	●						●	●				●	●	❍
77							-/3	-/3																							●	●	❍
78									-/4$^+$	-/2$^+$											●	●		●	●						❍	❍	❍
79									-/4$^+$	-/2$^+$											●	●		●	●						❍	❍	❍
80									-/4$^+$	-/2$^+$											❍	●		●	●						❍	●	❍
81									-/4$^+$	-/2$^+$											●	●		●	●						❍	●	❍
82			-/6	-/2	6/7	3/3			6/6$^+$	2/2$^+$			❍	❍	❍	❍	❍		❍	❍	❍	●	●	●	●		●	▼	●	●	▼	●	●
83	1/2	1/1																										▼	●	●	▼	●	●
84	1/2	1/1																										▼	●	●	▼	●	●
85	1/4	1/2																										▼	●	●	▼	×	●
86	7/11	1/1																										❍	❍	●	❍	●	●
87	1/2	1/1																										❍	❍	❍	❍	❍	❍
88									7/-	3/-								■										●	●	❍			
89			-/4	-/2	4/5	3/3	2/2	2/2	4/4$^+$	2/2$^+$			❍	❍	❍	❍	❍	❍	❍	❍	❍	●	●	●	●	●	●	●	●	❍	●	●	❍
90			-/4	-/2	4/5	3/3	2/2	2/2	4/4$^+$	2/2$^+$			❍	❍	❍	❍	❍	❍	❍	❍	❍	●	●	●	●	●	●	●	●	❍	●	●	❍
91					4/5	3/3								❍	❍	❍												●	●	❍	●	●	❍
92	-/5$^+$	-/2																										❍	❍	❍	❍	❍	❍
93	4/6	1/2																										❍	❍	❍	❍	❍	❍
94	4/6	1/2																										❍	❍	❍	❍	❍	❍
95	5/7	1/2																										❍	❍	❍	❍	❍	❍
96	-/5$^+$	-/2																										❍	❍	❍	❍	❍	❍
97	3/6	1/2																										❍	❍	❍	❍	❍	❍
98	3/6	1/2																										❍	❍	❍	❍	❍	❍
99	4/7	1/2																										❍	❍	❍	❍	❍	❍

Cont. Nr.	Mnemonics			Operation	Mode			
	Motorola	B&R	Short		PG1000	PG-PC	CP80	PC80
100	ROL	SLI	RL	Rotate M left	❍	❍	❍	❍
101	ROLA	RLA		Rotate A left	❍	❍	❍	❍
102	ROLB	RLB		Rotate B left	❍	❍	❍	❍
103	ROR	SRE	RR	Rotate M right	❍	❍	❍	❍
104	RORA	RRA		Rotate A right	❍	❍	❍	❍
105	RORB	RRB		Rotate B right	❍	❍	❍	❍
106	RST	RST	R	If d0 = 1, 0 ⇨ M	❍	❍	❍	❍
107	RTS	RET		Return from subroutine	❍	❍	❍	❍
108	SBA	A-B		A - B ⇨ A	❍	❍	❍	❍
109	SBCA	SUB	--	A - M - C ⇨ A	❍	❍	❍	❍
110	SBCB	--B		B - M - C ⇨ B	❍	❍	❍	❍
111	SEC	SEC		1 ⇨ C	❍	❍	❍	❍
112	SEI	SEI		1 ⇨ IIB	❍	❍	❍	❍
113	SET	SET	S	1 ⇨ M	❍	❍	❍	❍
114	SK0	SK0		Skip next instruction, if Z = 1		❍		❍
115	SK1	SK1		Skip next instruction, if Z = 0		❍		❍
116	STAA	=	I	A ⇨ M	❍	❍	❍	❍
117	STAB	=B		B ⇨ M	❍	❍	❍	❍
118	STD	=D		D ⇨ M:M+1	❍	❍	❍	❍
119	STS	=S		SP! ⇨ M:M+1	❍	❍	❍	❍
120	STX	=R		X ⇨ M:M+1	❍	❍	❍	❍
121	STY	=Y		Y ⇨ M:M+1				❍
122	SUBA	-		A - M ⇨ A	❍	❍	❍	❍
123	SUBB	-B		B - M ⇨ B	❍	❍	❍	❍
124	SUBD	-D		D - M:M+1 ⇨ D	❍	❍	❍	❍
125	TAB	MAB		A ⇨ B	❍	❍	❍	❍
126	TAP	MAC		A ⇨ CCR	❍	❍	❍	❍
127	TBA	MBA		B ⇨ A	❍	❍	❍	❍
128	TFR	TFR		r_8 ⇨ r_8 or r_{16} ⇨ r_{16}				❍
129	TIM	TIM		M $\wedge$ IMM		❍		
130	TPA	MCA		CCR ⇨ A	❍	❍	❍	❍
131	TSX	MSR		SP! + 1 ⇨ X	❍	❍	❍	❍
132	TXS	MRS		X - 1 ⇨ SP!	❍	❍	❍	❍
133	XGDX	DXR		D ⇔ X	❍	❍	❍	❍
134								

Cont. Nr.	Addressing Modes 6303/6809												Address Preselections															Condition Code					
	IMPL.		DIR.		EXT.		IMMED.		IND.		REL.		I	O	F	S	T	#	P	R	X	Y	D	U	!	B	G	6303			6809		
	~	#	~	#	~	#	~	#	~	#	~	#	E	A	M	F	Z	#	P	C	I	Y	D	U	!	B	G	N	Z	C	N	Z	C
100			-/6	-/2	6/7	3/3			6/6+	2/2+				❍	❍	❍	❍		❍	❍	❍	●	●	●	●		●	●	●	●	●	●	●
101	1/2	1/1																										●	●	●	●	●	●
102	1/2	1/1																										●	●	●	●	●	●
103			-/6	-/2	6/7	3/3			6/6+	2/2+				❍	❍	❍	❍		❍	❍	❍	●	●	●	●		●	●	●	●	●	●	●
104	1/2	1/1																										●	●	●	●	●	●
105	1/2	1/1																										●	●	●	●	●	●
106					4/5	3/3								❍	❍	❍												●	●	❍	●	●	❍
107	5/5	1/1																										❍	❍	❍	❍	❍	❍
108	1/25	1/3																										●	●	●	●	●	●
109			-/4	-/2	4/5	3/3	2/2	2/2	4/4+	2/2+			❍	❍	❍	❍	❍	❍	❍	❍	❍	●	●	●	●	●	●	●	●	●	●	●	●
110			-/4	-/2	4/5	3/3	2/2	2/2	4/4+	2/2+			❍	❍	❍	❍	❍	❍	❍	❍	❍	●	●	●	●	●	●	●	●	●	●	●	●
111	1/3	1/2																										❍	❍	▲	❍	❍	▲
112	1/3	1/2																										❍	❍	❍	❍	❍	❍
113					4/5	3/3								❍	❍	❍												●	●	❍	●	●	❍
114											3/3	2/2																❍	❍	❍	❍	❍	❍
115											3/3	2/2																❍	❍	❍	❍	❍	❍
116			-/4	-/2	4/5	3/3			4/4+	2/2+				❍	❍	❍			❍	❍	❍	●	●	●	●		●	●	●	❍	●	●	❍
117			-/4	-/2	4/5	3/3			4/4+	2/2+				❍	❍	❍			❍	❍	❍	●	●	●	●		●	●	●	❍	●	●	❍
118			-/5	-/2	5/6	3/3			5/5+	2/2+				❍	❍	❍			❍	❍	❍	●	●	●	●		●	●	●	❍	●	●	❍
119			-/6	-/3	5/7	3/4			5/6+	2/3+									❍	❍	❍	●		●	●		●	●	●	❍	●	●	❍
120			-/5	-/2	5/6	3/3			5/5+	2/2+									❍	❍	❍	●		●	●		●	●	●	❍	●	●	❍
121			-/6	-/3	-/7	-/4			-/6+	-/3+									●	●	●	●		●	●		●				●	●	❍
122			-/4	-/2	4/5	3/3	2/2	2/2	4/4+	2/2+			❍	❍	❍	❍	❍	❍	❍	❍	❍	●	●	●	●	●	●	●	●	●	●	●	●
123			-/4	-/2	4/5	3/3	2/2	2/2	4/4+	2/2+			❍	❍	❍	❍	❍	❍	❍	❍	❍	●	●	●	●	●	●	●	●	●	●	●	●
124			-/6	-/2	5/7	3/3	3/4	3/3	5/6+	2/2+								❍	❍	❍	❍	●	●	●	●	●	●	●	●	●	●	●	●
125	1/6	1/2																										●	●	❍	❍	❍	❍
126	1/6	1/2																										●	●	●	●	●	●
127	1/6	1/2																										●	●	❍	❍	❍	❍
128	-/6	-/2																													❍	❍	❍
129									5/-	3/-								■										●	●	❍			
130	1/6	1/2																										❍	❍	❍	❍	❍	❍
131	1/6	1/2																										❍	❍	❍	❍	❍	❍
132	1/6	1/2																										❍	❍	❍	❍	❍	❍
133	2/7	1/2																										❍	❍	❍	❍	❍	❍
134																																	

B&R Mnemonics

B&R	Motorola	Short	B&R	Motorola	Short	B&R	Motorola	Short
+	ADDA		EXG	EXG		PUL	PUL	
++B	ADCB		EXO	EORA	E	RET	RTS	
+B	ADDB		IA	INCA		RLA	ROLA	
+D	ADDD		IB	INCB		RLB	ROLB	
-	SUBA		INC	INC		RNS	PSHX	
—B	SBCB		IR	INX		RRA	RORA	
-B	SUBB		IS	INS		RRB	RORB	
-D	SUBD		J+	BPL		RST	RST	R
=	STAA	I	J+L	BPLL		RVS	PULX	
=B	STAB		J-	BMI		SEC	SEC	
=D	STD		J-L	BMIL		SEI	SEI	
=R	STX		J<=	BLS		SET	SET	S
=S	STS		J<=L	BLSL		SK0	SK0	
=Y	STY		JC0	BCC		SK1	SK1	
A*B	MUL		JC0L	BCCL		SL	ASL	
A+B	ABA		K	COM		SLA	ASLA	
A-B	SBA		LAD	LDAA	L	SLB	ASLB	
ADD	ADCA	A, ++	LB	LDAB		SLD	ASLD	
AIM	AIM		LD	LDD		SLI	ROL	RL
ANS	PSHA		LDK	LDK		SN0	BNE	J1
AVB	CBA		LDL	LDL		SN0L	BNEL	
AVS	PULA		LE!	LEA!		SP0	BEQ	J0
B	BITA		LER	LEAX		SP0L	BEQL	
B+R	ABX		LEU	LEAU		SP<	BCS	J<
BB	BITB		LEY	LEAY		SP<L	BCSL	
BNS	PSHB		LR	LDX		SP>	BHI	J>
BVS	PULB		LRK	LDX#		SP>L	BHIL	
CLA	CLRA		LRL	LDXL		SPI	JMP	J
CLB	CLRB		LS	LDS		SPU	JSR	JU, JS
CLC	CLC		LY	LDY		SR	LSR	
CLI	CLI		LYK	LDY#		SRA	LSRA	
CLR	CLR	C	LYL	LDYL		SRB	LSRB	
CMP	CMPA	V	MAB	TAB		SRD	LSRD	
COA	COMA	KA	MAC	TAP		SRE	ROR	RR
COB	COMB	KB	MBA	TBA		SUB	SBCA	—
DA	DECA		MCA	TPA		TFR	TFR	
DB	DECB		MRS	TXS		TIM	TIM	
DEC	DEC	D	MSR	TSX		UB	ANDB	
DK	DAA		NOP	NOP	N	UND	ANDA	U
DR	DEX		OB	ORAB		VB	CMPB	
DS	DES		OD	ORAA	O	VR	CPX	
DXR	XGDX		OIM	OIM		VRK	CPX#	
EB	EORB		PRS	PRS	P	VY	CPY	
EIM	EIM		PSH	PSH		VYK	CPY#	
END	END							

Motorola Mnemonics

Motorola	B&R	Short
ABA	A+B	
ABX	B+R	
ADCA	ADD	A, ++
ADCB	++B	
ADDA	+	
ADDB	+B	
ADDD	+D	
AIM	AIM	
ANDA	UND	U
ANDB	UB	
ASL	SL	
ASLA	SLA	
ASLB	SLB	
ASLD	SLD	
BCC	JC0	
BCCL	JC0L	
BEQ	SP0	J0
BEQL	SP0L	
BHI	SP>	J>
BHIL	SP>L	
BITA	B	
BITB	BB	
BCS	SP<	J<
BCSL	SP<L	
BLS	J<=	
BLSL	J<=L	
BMI	J-	
BMIL	J-L	
BNE	SN0	J1
BNEL	SN0L	
BPL	J+	
BPLL	J+L	
CBA	AVB	
CLC	CLC	
CLI	CLI	
CLR	CLR	C
CLRA	CLA	
CLRB	CLB	
CMPA	CMP	V
CMPB	VB	
COM	K	
COMA	COA	KA
COMB	COB	KB
CPX	VR	
CPX#	VRK	
CPY	VY	
CPY#	VYK	
DAA	DK	
DEC	DEC	D
DECA	DA	
DECB	DB	
DES	DS	
DEX	DR	
EIM	EIM	
END	END	
EORA	EXO	E
EORB	EB	
EXG	EXG	
INC	INC	
INCA	IA	
INCB	IB	
INS	IS	
INX	IR	
JMP	SPI	J
JSR	SPU	JU, JS
LDAA	LAD	L
LDAB	LB	
LDD	LD	
LDK	LDK	
LDL	LDL	
LDS	LS	
LDX	LR	
LDX#	LRK	
LDXL	LRL	
LDY	LY	
LDY#	LYK	
LDYL	LYL	
LEA!	LE!	
LEAU	LEU	
LEAX	LER	
LEAY	LEY	
LSR	SR	
LSRA	SRA	
LSRB	SRB	
LSRD	SRD	
MUL	A*B	
NOP	NOP	N
OIM	OIM	
ORAA	OD	O
ORAB	OB	
PRS	PRS	P
PSH	PSH	
PSHA	ANS	
PSHB	BNS	
PSHX	RNS	
PUL	PUL	
PULA	AVS	
PULB	BVS	
PULX	RVS	
ROL	SLI	RL
ROLA	RLA	
ROLB	RLB	
ROR	SRE	RR
RORA	RRA	
RORB	RRB	
RST	RST	R
RTS	RET	
SBA	A-B	
SBCA	SUB	—
SBCB	—B	
SEC	SEC	
SEI	SEI	
SET	SET	S
SK0	SK0	
SK1	SK1	
STAA	=	I
STAB	=B	
STD	=D	
STS	=S	
STX	=R	
STY	=Y	
SUBA	-	
SUBB	-B	
SUBD	-D	
TAB	MAB	
TAP	MAC	
TBA	MBA	
TFR	TFR	
TIM	TIM	
TPA	MCA	
TSX	MSR	
TXS	MRS	
XGDX	DXR	

STL instruction Groups

Load and Storage instructions

B&R	Motorola	Short
LAD	LDAA	L
LB	LDAB	
LD	LDD	
LDK	LDK	
LDL	LDL	
LR	LDX	
LRK	LDX#	
LRL	LDXL	
LS	LDS	
LY	LDY	
LYK	LDY#	
LYL	LDYL	
LE!	LEA!	
LEU	LEAU	
LER	LEAX	
LEY	LEAY	
=	STAA	I
=B	STAB	
=D	STD	
=R	STX	
=S	STS	
=Y	STY	
MAB	TAB	
MBA	TBA	
MAC	TAP	
MCA	TPA	
MSR	TSX	
MRS	TXS	
TFR	TFR	
DXR	XGDX	
EXG	EXG	
PSH	PSH	
ANS	PSHA	
BNS	PSHB	
RNS	PSHX	
PUL	PUL	
AVS	PULA	
BVS	PULB	
RVS	PULX	

Logic Connections

B&R	Motorola	Short
UND	ANDA	U
UB	ANDB	
AIM	AIM	
OD	ORAA	O
OB	ORAB	
OIM	OIM	
EXO	EORA	E
EB	EORB	
EIM	EIM	

Comparison instructions

B&R	Motorola	Short
AVB	CBA	
CMP	CMPA	V
VB	CMPB	
VR	CPX	
VRK	CPX#	
VY	CPY	
VYK	CPY#	
B	BITA	
BB	BITB	
TIM	TIM	

Comparison instructions

B&R	Motorola	Short
AVB	CBA	
CMP	CMPA	V
VB	CMPB	
VR	CPX	
VRK	CPX#	
VY	CPY	
VYK	CPY#	
B	BITA	
BB	BITB	
TIM	TIM	

Arithmetic Operations

B&R	Motorola	Short
+	ADDA	
ADD	ADCA	A, ++
+B	ADDB	
++B	ADCB	
+D	ADDD	
A+B	ABA	
B+R	ABX	
-	SUBA	
SUB	SBCA	--
-B	SUBB	
--B	SBCB	
-D	SUBD	
A-B	SBA	
A*B	MUL	

Shift and Rotation instruction

B&R	Motorola	Short
SL	ASL	
SLA	ASLA	
SLB	ASLB	
SLD	ASLD	
SR	LSR	
SRA	LSRA	
SRB	LSRB	
SRD	LSRD	
SLI	ROL	RL
RLA	ROLA	
RLB	ROLB	
SRE	ROR	RR
RRA	RORA	
RRB	RORB	

Decrement- and Increment instructions

B&R	Motorola	Short
INC	INC	
IA	INCA	
IB	INCB	
IR	INX	
IS	INS	
DEC	DEC	D
DA	DECA	
DB	DECB	
DR	DEX	
DS	DES	

Branch instructions

B&R	Motorola	Short
JC0	BCC	
JC0L	BCCL	
SP<	BCS	J<
SP<L	BCSL	
J+	BPL	
J+L	BPLL	
J-	BMI	
J-L	BMIL	
SN0	BNE	J1
SN0L	BNEL	
SP0	BEQ	J0
SP0L	BEQL	
SP>	BHI	J>
SP>L	BHIL	
J<=	BLS	
J<=L	BLSL	
SK0	SK0	
SK1	SK1	
SPI	JMP	J
SPU	JSR	JU, JS
RET	RTS	
NOP	NOP	N
END	END	

Other

B&R	Motorola	Short
PRS	PRS	P
RST	RST	R
CLA	CLRA	
CLB	CLRB	
CLR	CLR	C
SET	SET	S
CLC	CLC	
SEC	SEC	
CLI	CLI	
SEI	SEI	
K	COM	
COA	COMA	KA
COB	COMB	KB
DK	DAA	

APPENDIX C
ADDITIONAL INSTALLATIONS

Contents:

1. LEARN - FUNCTION

With the LEARN-function any sequence of key combinations can be recorded and repeated later. There is the possibility to record a sequence of commands that are needed more often, and to return those with the input of a command.

1.1. START RECORDING A LEARN FILE

Command: **`#LEARN# <FNAME> [↵]`**

Function: The LEARN mode is turned on. All subsequent keys operations are written in the defined FNAME file.

<FNAME> is a file name with a maximum length of 8 characters with an extension, that can be selected optionally. The extensions that are used by PROgramming SYStem shouldn't be applied:

```
*.PRG     *.SYMBOLS *.DATA_TBL*.LAD
*.MSY     *.MTA     *.MSL     *.BSC
*.PRV     *.ATA     *.ASY     *.EMS
```

A path can be entered with the file name (drive and directory). If no path is defined, the LEARN file is stored in the directory configured when the PROgramming SYStem was installed. (default setting for the software installation: **`C:\PG`**)

1.2. END THE RECORDING

Command: **`#LOFF# [↵]`**

Function: The LEARN mode is turned off.

1.3. RETURN OF A LEARN FILE

Command: **EXE <FNAME> [↵]**

Function: All key operations, that are stored in the specified file FNAME are returned according to the record.

A LEARN file can also be defined as start parameter for the PROgramming SYStem (chapter 3 INTRODUCTION).

The subdivision of LEARN files is impossible, i.e.: If a LEARN-file is called by another one, a route is formed, but there is no jump back to the first one.

> The processing of a LEARN file can be interrupted with the [Esc]-key.

1.4. FORMAT OF A LEARN FILE

A LEARN file contains only ASCII characters. ASCII characters, that are not representable on screen (e.g.: cursor-movements, CR, LF etc.), are entered with a 3 digit decimal number and a preceding “\”. The format “\xxx” must be adhered to exactly, to allow the learn file to be created and edited with any processing program.

If the character “*” is in the LEARN file, all subsequent characters in the same line are interpreted as comment.

If both characters “\“ and “*” are directly submitted to the PROgramming SYStem, these have to be entered in decimal format:

```
\  ...  \092
*  ...  \042
```

All remaining control characters CR, LF, ESC etc. are to be entered in decimal form.

On the following page all keys with the corresponding decimal values are listed in a table.

Key	Character sequence	Key	Character sequence	Key	Character sequence
EXIT (ESC)	\027	F1	\000\059	REL_F1	\000\200
TAB	\009	F2	\000\060	REL_F2	\000\201
SH_TAB	\015	F3	\000\061	REL_F3	\000\202
ALT_TAB	\000\133	F4	\000\062	REL_F4	\000\203
CTRL_TAB	\000\134	F5	\000\063	REL_F5	\000\204
		F6	\000\064	REL_F6	\000\205
		F7	\000\065	REL_F7	\000\206
		F8	\000\066	REL_F8	\000\207
		F9	\000\067	REL_F9	\000\208
		F10	\000\068	REL_F10	\000\209
SH_F1	\000\084	CTRL_F1	\000\094	ALT_F1	\000\104
SH_F2	\000\085	CTRL_F2	\000\095	ALT_F2	\000\105
SH_F3	\000\086	CTRL_F3	\000\096	ALT_F3	\000\106
SH_F4	\000\087	CTRL_F4	\000\097	ALT_F4	\000\107
SH_F5	\000\088	CTRL_F5	\000\098	ALT_F5	\000\108
SH_F6	\000\089	CTRL_F6	\000\099	ALT_F6	\000\109
SH_F7	\000\090	CTRL_F7	\000\100	ALT_F7	\000\110
SH_F8	\000\091	CTRL_F8	\000\101	ALT_F8	\000\111
SH_F9	\000\092	CTRL_F9	\000\102	ALT_F9	\000\112
SH_F10	\000\093	CTRL_F10	\000\103	ALT_F10	\000\113
HOME	\000\071	CTRL_CUL	\000\115	ALT_INS	\000\140
CUP	\000\072	CTRL_CUR	\000\116	ALT_DEL	\000\141
PGUP	\000\073	CTRL_END	\000\117	ALT_END	\000\142
CUL	\000\075	CTRL_PGDN	\000\118	ALT_CUD	\000\143
CUR	\000\077	CTRL_HOME	\000\119	ALT_PGDN	\000\144
END	\000\079	CTRL_PGUP	\000\132	ALT_CUL	\000\145
CUD	\000\080	CTRL_CUP	\000\135	ALT_EXIT	\000\146
PGDN	\000\081	CTRL_CUD	\000\136	ALT_CUR	\000\147
INS	\000\082	CTRL_INS	\000\137	ALT_HOME	\000\148
DEL	\000\083	CTRL_DEL	\000\138	ALT_CUP	\000\149
SPACE	\032	CTRL_EXIT	\000\139	ALT_PGUP	\000\150
CTRLALT_INS	\000\151	CTRL_A	\001	CTRL_N	\014
CTRLALT_DEL	\000\152	CTRL_B	\002	CTRL_O	\015
CTRLALT_END	\000\153	CTRL_C	\003	CTRL_P	\016
CTRLALT_CUD	\000\154	CTRL_D	\004	CTRL_Q	\017
CTRLALT_PGDN	\000\155	CTRL_E	\005	CTRL_R	\018
CTRLALT_CUL	\000\156	CTRL_F	\006	CTRL_S	\019
CTRLALT_EXIT	\000\157	CTRL_G	\007	CTRL_T	\020
CTRLALT_CUR	\000\158	CTRL_H	\008	CTRL_U	\021
CTRLALT_HOME	\000\159	CTRL_I	\009	CTRL_V	\022
CTRLALT_CUP	\000\160	CTRL_J	\010	CTRL_W	\023
CTRLALT_PGUP	\000\161	CTRL_K	\011	CTRL_X	\024
		CTRL_L	\012	CTRL_Y	\025
		CTRL_M	\013	CTRL_Z	\026

SH_ SHIFT-key is pressed
ALT_ ALT-key is pressed
CTRL_ CTRL-key is pressed
CTRLALT_ CTRL- and ALT-key are pressed together
REL_ Key is released (only function keys)

2. PROVIT EMULATIONS

2.1. PROVIT 1000 - KEYBOARD EMULATION

The B&R PROgramming SYStem software package makes use of the PC as an input terminal for operating the PROVIT 900/1000. The PC keyboard is allocated as an external ASCII-keyboard. Pictures can only be created, edited and loaded/stored from/to diskette or hard disk with the keyboard on the installed PROVIT terminal.

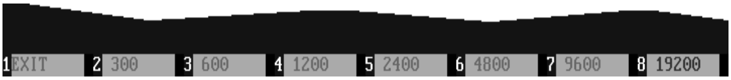
The terminal is connected to the PC COM1 interface with one of the following cables:

- BRKAPC-1
- BRKAPC-3

Command: **TERM [↵]**

Function: After the input of this command in the command line the screen is cleared and the cursor is set to the top left corner.

The function key allocation alters as follows:



The displayed functions can only be called in connection with the [⇑] key:

[⇑] + [F1 EXIT] Exit the terminal mode back to the command line of the PROgramming SYStem.

[⇑] + [F2] to **[F8]** Adjusting the baud rate of the interface COM1.

> **Note:**
> PC and PROVIT 900/1000 baud rates must correspond.

The function key allocation without simultaneous operation with [⇑] is the same as the softkey allocation of the PROVIT 900/1000 (displayed on the terminal).

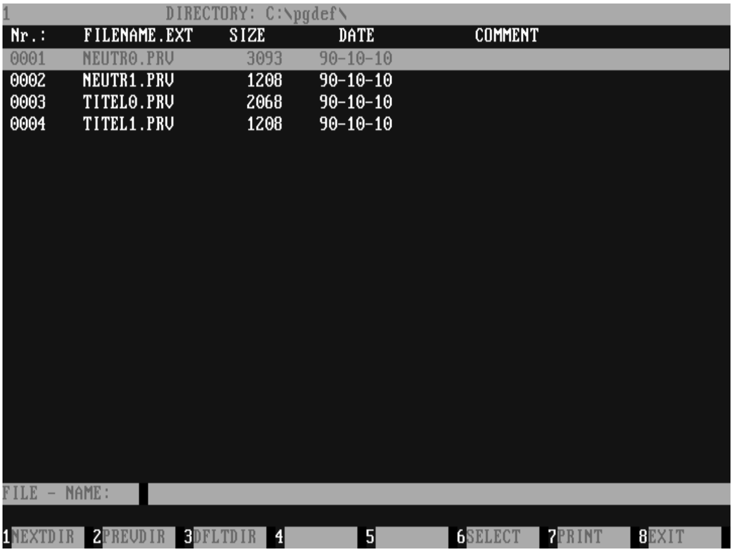
The sofkey functions and the operations of the PROVIT 900/1000 are to be found in:

- Operating Terminals - USER'S MANUAL (MATERMINAL-0)
- PROVIT 1000 Short Description (MAP1000KB-0)

If the PROVIT function [F5 SAVE] or [F4 LOAD] is called, three alternatives are offered in the softkey line of the PROVIT terminal:

- **`[F1 MASK]`** store/load individual PROVIT masks
- **`[F2 TEXT]`** store/load the PROVIT global texts
- **`[F3 B-UP]`** store/load all 28 PROVIT masks
  (Backup/Restore; from PROVIT V. 2.50)

All files of the default-directories with the extension `*.PRV` (for selecting [F1 MASK] or [F2 TEXT]) or `*.PBK` (for selecting [F3 B-UP]) are displayed on screen:



```
1                  DIRECTORY: C:\pgdef\
Nr.:     FILENAME.EXT     SIZE        DATE           COMMENT
0001     NEUTR0.PRV        3093     90-10-10
0002     NEUTR1.PRV        1208     90-10-10
0003     TITEL0.PRV        2068     90-10-10
0004     TITEL1.PRV        1208     90-10-10

FILE - NAME:

1NEXTDIR 2PREVDIR 3DFLTDIR 4        5        6SELECT  7PRINT   8EXIT
```

Load

The desired file can be selected with the [❖] keys. After pressing [↵] the selected file is loaded from diskette or hard disk into PROVIT. During the load procedure the following message is displayed:

`--- LOAD active ---`   **`Length: xxxx Bytes`**   xxxx ... file length

Store

A file name with a length of up to 8 characters may be defined. After pressing the [↵] key, the screen, the global texts and the 28 pictures are stored with the specified file name on disk or hard disk. During the storage procedure the following message is displayed:

```
--- SAVE active ---          Length: xxxx Bytes
```

xxxx ... File length

Function key allocation during the store/load procedure:

Key	FUNCTION
[F1 NEXTDIR]	If the cursor is set to a directory (marked by <DIR>), this directory is opened. The contents of this directory are displayed. This function is the same as the DOS command "`cd <directory name>`".
[F2 PREVDIR]	The previous directory is changed and its contents are displayed. [F2 PREVDIR] is the same as the DOS command "`cd ..`".
[F3 DEFDIR]	The current directory is defined as the "default directory".
[↵]	The file selected with the [❖] keys (the corresponding line is displayed in inverse video in the overview) or the files with the defined file name (without extension) is loaded/stored.
[F7 PRINT]	The displayed directory is printed.
[F8 EXIT] **[Esc]**	Quits the load/store menu.

If the load or storage function should apply to a different drive, the PROVIT emulation must be exited. A different drive can be set in the command line of the PROgramming SYStem with the command **`DRIVE <drive name> [↵]`**.

If an existing file name is selected for storage, the following message is displayed on screen:

```
W008  FILE NAME EXISTS - CONTINUE ?  ( Y / N )
```

[N] The storage procedure is interrupted.
[Y] The existing file is overwritten.

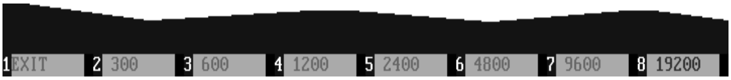
2.2. PROVIT 500/600 (MAESTRO) EMULATION

Command: **`MAESTRO [↵]`**

Function: Emulates a PROVIT 500/600 with external ASCII keyboard. The PC keyboard layout is the same as the ASCII keyboard.

After inputting this command in the command line the screen is deleted and the cursor is set in the top left corner.

The softkey line allocation changes as follows:



```
1EXIT  2 300  3 600  4 1200  5 2400  6 4800  7 9600  8 19200
```

The displayed functions can only be called by simultaneously pressing the [⇑] key:

`[⇑] + [F1 EXIT]` Exits the terminal mode into the command line of the PROgramming SYStem.

`[⇑] + [F2]` to **`[F8]`** Adjusts the COM1 interface baud rate.

> **Note:**
> The baud rates of PC and PROVIT must correspond.

The following PROVIT control characters are supported during terminal emulation (beginning from PROgramming SYStem Version 5.00):

Ctrl A Ctrl B Ctrl C Ctrl D Ctrl E Ctrl F Ctrl H Ctrl I
Ctrl J Ctrl K Ctrl L Ctrl M Ctrl N Ctrl O Ctrl P Ctrl Q
Ctrl R Ctrl S Ctrl U Ctrl V Ctrl W Ctrl X Ctrl Y Ctrl Z

Esc A Esc D Esc F Esc H Esc I Esc L Esc M Esc N
Esc P Esc R Esc U Esc W Esc X

Esc 0 Esc 1 Esc 2 Esc 3 Esc 4 Esc 5 Esc 6 Esc 7

$ 1E $ 1D $ 7F

The functions for these control sequences are described in :

- OPERATOR INTERFACE TERMINALS USER'S MANUAL (MATERMINAL-0)

  Section 6 PROVIT control characters and escape sequences

2.2.1. Restrictions

Display resolution: The 24 x 40 or 12 x 40 display resolution characters are simulated by character repetition and additional blanks.

Screen background: Only **one** display background color can be defined.
8 colors are given for selecting the background.
When using a color screen the following environment variable must be set:

```
set B&R$SCREEN=COL [↵]
```

If the variable is not set, the emulation defaults to monochrome mode - colors are ignored.

Bars, Rectangles: The creation of bars, rectangles and special chatacters is not directly possible, but can be done by redefining, positioning and outputting certain characters. .

2.2.2. software change of the output record - Esc P c x_0 ... x_n

For the redefinition of characters certain preconditions must be met:

- An EGA or VGA graphic card must be available in the PC.
- The following environment variable must be set before the start of the PROgramming SYStem:

```
set B&R$CDEF=ON [↵]
```

If one of these two conditions is not correct, no characters can be redefined!

The values x_0 to x_n contain the bit pattern of the new character. The terminal emulation expects, according to which graphic card is used, either 14 bytes (EGA) or 16 bytes (VGA).

Loading back the original output record - ESC P $20

The original PC output record is loaded back with this escape sequence. All character redefinitions are canceled.

If the environment variable **B&R$FDEF** is set in the file defined, the output record is loaded back.

2.2.3. Read output records of a file

In order to have several output records available for emualtion, they must be defined in a file. When using the MAESTRO terminal emulation, this file is read. To turn on this function the following environment variable must be set before the start of the PROgramming SYStem:

```
set B&R$FDEF=ON [↵]
```

If this function is activated, the emulation (when calling with the operating command **`MAESTRO`**) expects the file with the output record to be in the directory, where the PROgramming SYStem was installed (default setting for the software installation: `C:\PG\`). This file has a different name depending on which graphic card is installed:

Graphic card	File name
EGA	`PRV_AEGA.DEF`
VGA	`PRV_AVGA.DEF`

Formatting the definition file

The definition file of the output record is an ASCII file, that has to correspond with a certain format. The base format is:

```
<n>
<cc>
<x_0> <x_1> ... <x_n>
<cc>
<x_0> <x_1> ... <x_n>
       :
       :
       :
```

<n> The EGA card requires a resolution of 14x8 pixels for one character and the VGA card needs 16x8 pixels. Corresponding to the installed card, the **decimal** value 14 or 16 must be enterd.

<cc> The output code, that needs to be redefined, in decimal.

<x$_0$> to <x$_n$> Character definition bytes. 14 or 16 hexadecimal values, that are divided by the so called “white spaces” (blanks, tabs or line feed) depending on which card is installed.

* Comments are marked with this chatacter. All subsequent chatacters to the line end are comments. This character must not be at the beginning of a line (see example).

Blank lines Blank lines are definitely ignored.

In a character definition file, a maximum of 256 (0 to 255) characters can be defined.

Example:

```
******************************************************************************
*      PROSYS PROVIT 600 EMULATION - OUTPUT RECORD A
******************************************************************************
14                                * number of character definition bytes
***  Character definition  **************************************************
148                                     * corner top left
00 00 00 00 00 00 1F 1F 18 18 18 18 18 18
*            12345678
*      01:    ........ = 00
*      02:    ........ = 00
*      03:    ........ = 00
*      04:    ........ = 00
*      05:    ........ = 00
*      06:    ........ = 00
*      07:    ...xxxxx = 1F
*      08:    ...xxxxx = 1F
*      09:    ...xx... = 18
*      10:    ...xx... = 18
*      11:    ...xx... = 18
*      12:    ...xx... = 18
*      13:    ...xx... = 18
*      14:    ...xx... = 18
******************************************************************************
149                                     * corner top right
00 00 00 00 00 00 F8 F8 18 18 18 18 18 18
*            12345678
*      01:    ........ = 00
*      02:    ........ = 00
*      03:    ........ = 00
*      04:    ........ = 00
*      05:    ........ = 00
*      06:    ........ = 00
*      07:    xxxxx... = F8
*      08:    xxxxx... = F8
*      09:    ...xx... = 18
*      10:    ...xx... = 18
*      11:    ...xx... = 18
*      12:    ...xx... = 18
*      13:    ...xx... = 18
*      14:    ...xx... = 18
******************************************************************************
150                                     * corner bottom left
18 18 18 18 18 18 1F 1F 00 00 00 00 00 00
*            12345678
*      01:    ...xx... = 18
:                   :
:                   :
```

Note:

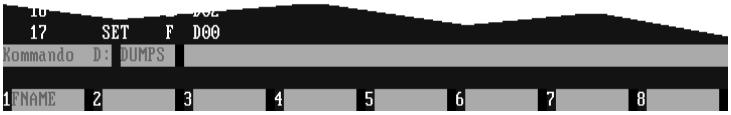
The file is easier to survey and read if comments are added.

3. EDITING THE PROGRAM IMAGE FOR EPROM-PROGRAMMER

Command: **DUMPS <FNAME> [↵]**

Function: A PLC program can be stored in the PROgramming SYStem with the command **DUMPS** as MOTOROLA S-Record file. This record file can now be copied with an EPROM programmer, that has knowledge of MOTOROLA S-Records.

After entering DUMPS the PROgramming SYStem demands the definition of a file name:



```
 17        SET     F  D00
Kommando  D: DUMPS
1FNAME    2         3         4         5         6         7         8
```

A 8 character long file name <FNAME> with drive name and path can be entered, which should be stored in the record file. If no drive name and path is defined, the file is stored in the default directory of the current drive.

MOTOROLA S - RECORD FILE FORMAT

Three different record types S0, S1 and S9 are used in a MOTOROLA S-Record file.

Record type S0

S0 is the Start record, and has the following format:

```
S0 bb yyyyyyyy xx
```

`bb`	...	byte counter
`yy..yy`	...	random hex. data
`xx`	...	check sum

Record type S1

S1 is a Data record, that starts at a 16-bit load address with the following format:

```
S1 bb aaaa dddddddddddddddddddddddddd xx
```

bb	...	data bytes+address bytes+1 check sum
aaaa	...	16-bit load address
dd..dd	...	data bytes (or words) up to 24 bytes per record
xx	...	check sum

Record type S9

S9 is the End record and has the following format:

```
S9 bb yyyyyyyy xx
```

bb	...	byte counter
yy..yy	...	random hex. data
xx	...	check sum

B&R offers a small, compact EPROM Programmer, that consists of three components: programming device, adapter for memory modules and software. These are available with the following model numbers:

Programming device:	PGEP01-0	(including software SWPEP01-0)
Software:	SWPEP01-0	
Adapter for memory modules:	PGEPAD01-0	

4. MS-DOS ENVIRONMENT VARIABLES

Before starting PROgramming SYStem, parameters can be submitted with environment variables at the MS-DOS level .

The environment variables are set or deleted respectively with the DOS command **`set`**:

`set ENVIRONMENT=<Parameter>`	set environment variable
`set ENVIRONMENT=`	delete environment variable

4.1. TURN OFF TEST OF PARALLEL ONLINE INTERFACE

Environment variable: **`B&R$LPT1=NO`**

Function: The PROgramming SYStem quits testing the parallel online interface.

This variable is used in Novell Networks with parallel Ethernet adapters.

4.2. USING A DIFFERENT PARALLEL PORT

Environment variable: **`B&R$LPT_OVER=<Base address>`**
`B&R$LPT_IRQ=<IRQ>`

Parameter: <Base address> is the 4 digit hexadecimal base address of the parallel port.[1]

<IRQ> IRQ can have one of the following values: 3,4,5,7[1]

Function: To operate the parallel interface BRKAOL5-1 from a different parallel port, the base address of the interface and the interrupt used must be submitted to the PROgramming SYStem with the variables that are listed above.

Example: Parallel interface operation to LPT2:

```
set  B&R$LPT_OVER=0378
set  B&R$LPT_IRQ=5
```

[1] More information about the base address and IRQ of the parallel port is given in the User' Manual of the PC or is available from the dealer.

4.3. LOADING ANOTHER KEYBOARD DRIVER

Environment variable: **`B&R$KBD=<File name>`**

Parameter: <File name> ... File name of the Keyboard Driver.

The following keyboard drivers are delivered with the PROgramming SYStem version 5.0:

`KEYB.101`	German (MF 101)
`KEYB.CH`	Swiss
`KEYB.F`	French
`KEYB.GR`	German
`KEYB.ITA`	Italian
`KEYB.SP`	Spanish
`KEYB.SU`	Swedish
`KEYB.UK`	English
`KEYB.US`	U.S.A.
`KEYB.DEF`	installed keyboard driver

Function: The PROgramming SYStem loads the defined keyboard driver <file name>.

This variable is used in networks, which connect with different types of computers.

Example: The American keyboard driver is loaded by the PROgramming SYStem.

```
set B&R$KBD=KEYB.US [↵]
```

4.4. PRINTING A FILE

Environment variable: **`B&R$PRINTFILE=<File name>`**

Parameter: <File name> ... max. 8 character file name with path.

Function: All prints of the PROgramming SYStem are by-passed to the file <file name>
(Chapter 9 PRINT, 6. "printing" in a file).

4.5. MAESTRO TERMINAL EMULATION VARIABLES

Environment variable: **`B&R$SCREEN=COL`**

Function: If this environment variable si set, the representation of the terminal emulation is in color, if a color screen is connected.
If the variable has not been set, or has been deleted, the display defaults to monochrome.

Environment variable: **`B&R$CDEF=ON`**

Function: If this environment variable is set, a software redefinition of the chatacters in the terminal emulation is possible.

Environment variable: **`B&R$FDEF=ON`**

Function: If this environment variable is set, the output record is loaded from two definition files.

4.6. USER SPECIFIC FILES

Environment variable: **`B&R$NET=<Directory>`**

Function: All user specific files are searched or installed in the defined directory. This environment variable is mainly used, if the PROgramming SYStem in a network is accessable for several users.

Example:

```
set B&R$NET=c:\user1\private\ [↵]
```

4.7. TEMPORARY FILES

Environment variable: **`B&R$VIRT=<Directory>`**

Function: The temporary files of the PROgramming SYStem are installed in the defined directory.

Example: The temporary files will be installed on the virtual disk e:\ (RAM disk):

```
set B&R$NET=e:\ [↵]
```

4.8. EMS-SYMBOL MANAGEMENT

Environment variable:**B&R$EMS=ON**

Function: The SYMBOLs of the PROgramming SYStem are installed and controlled in the EMS memory. At least 128 KBytes must be free in the EMS memory and a respective driver must be installed.

Requirements

- EMS memory expansion must be available in the PC. At least 128 KBytes must be available for the PROgramming SYStem to be able to control 8100 SYMBOLs. The following configurations are basically necessary, to use EMS:

    - 8088/8086 CPU with Expanded Memory Card and corresponding driver. (e.g.: EMM.SYS; this driver is usually delivered with standard DOS).

    - 80286 CPU with Expanded Memory Card and corresponding driver.
      80286 CPU with Extended Memory, which can also be used as Expanded Memory. The following is possible...
      ...hardwarewise using NEAT chips
      ...softwarewise with an EMS driver

    - 80386/80486 CPU with Expanded Memory CARD and respective driver.
      80386/80486 CPU with Extended Memory and respective driver, which supports the paging possibilities of the CPU, so simulating EMS.
      (e.g.: 386MAX or QEMM386).

  The EMS which can be used in your computer is explained in the PC description.

- Before starting PROgramming SYStem the environment variable B&R$EMS=ON must be set.

APPENDIX D
ERROR MESSAGES

Contents:

FATAL ERROR MESSAGES

When working with PROgramming SYStem the following FATAL ERROR messages can occur. If a FATAL error does occur, the PROgramming SYStem ceases the program procedure and returns to the DOS input line.

Error number	Error description Error elimination
1	V_MEM.TMx can't be opened. If necessary increase `files` and `buffers` in the `config.sys` .
2	Memory access error.
3	LAD_CALL no stack.
4	SYMBOL overflow V_MEM.

5	FKEY.BEL missed (function key allocation).
	Copying the file FKEY.BEL from the installation disk (insert in drive A) to \pg directory:
	`C:\> cd \pg [↵]`
	`C:\PG\> copy a:\fkey.bel [↵]`

6	EDITOR.PAR missed (error description, color adjustment). The file EDITOR.PAR can be newly created with the INSTALL program. After the selection of the menu item “screen reconfiguration” and the selection of the adequate screen the file EDITOR.PAR is created (Chapter 2 INSTALLATION, 6.1. Changing the configuration).

7	WARN.PG or ERROR.PG is missing.
	Copying the file WARN.PG or ERROR.PG from the installation diskette (insert in drive A) to the \pg directory:
	`C:\> cd \pg [↵]`
	`C:\PG\> copy a:\warn.pg [↵]`
	`C:\PG\> copy a:\error.pg [↵]`

8	Program name is missing.

9	not enough RAM free Memory resident programs like FDO, CHAD, Device Driver (e.g.: mouse driver) and loaded output records (DOS 3.X) are, if necessary, to be deleted.
10	KEYB.DEF is missing (keyboard allocation). The file KEYB.DEF can be newly created with the INSTALL-program. After the selection of the menu item “reconfigure keyboard” and selection of the corresponding keyboard driver, the file KEYB.DEF is created (chapter 2 INSTALLATION, 6.1. Changing the configuration).

11	EDITOR.CTL is missing (control strings).
	Copying the file EDITOR.CTL from the installation diskette (insert in drive A) to the \pg directory:
	`C:\> cd \pg [↵]`
	`C:\PG\> copy a:\editor.ctl [↵]`

12	LAD_COM no stack
13	free
14	V_MEM.TMN can't be opened (undo-buffer) If necessary increase `files` and `buffers` in the `config.sys`.
15	free

16	CMDS.PG can't be loaded (PROgramming SYStem KEY-WORDS)
	Copying the file CMDS.PG from the installation diskette (insert in drive A) to the \pg directory:
	`C:\> cd \pg [↵]`
	`C:\PG\> copy a:\cmds.pg [↵]`

17	Not sufficient space for virtual files (at least 300.000 bytes) Delete files not required from the hard disk.

APPENDIX F

PROGRAMMING SYSTEM CHANGES VERSION 5.50

Contents:

CORRECTIONS AND ADDITIONS for the User's Manual B&R PROgramming SYStem (5th Edition)

CHAPTER 2 - INSTALLATION

4. PORTABLE PC CONFIGURATIONS (Page2-5)

One very important step in having a problem free B&R PROgramming SYStem is the proper configuration in the PC's setup menu. With the wrong configuration, the B&R PROgramming SYStem will only run partially or not at all.

The next few pages show configuration menus for programming devices which are offered by B&R.

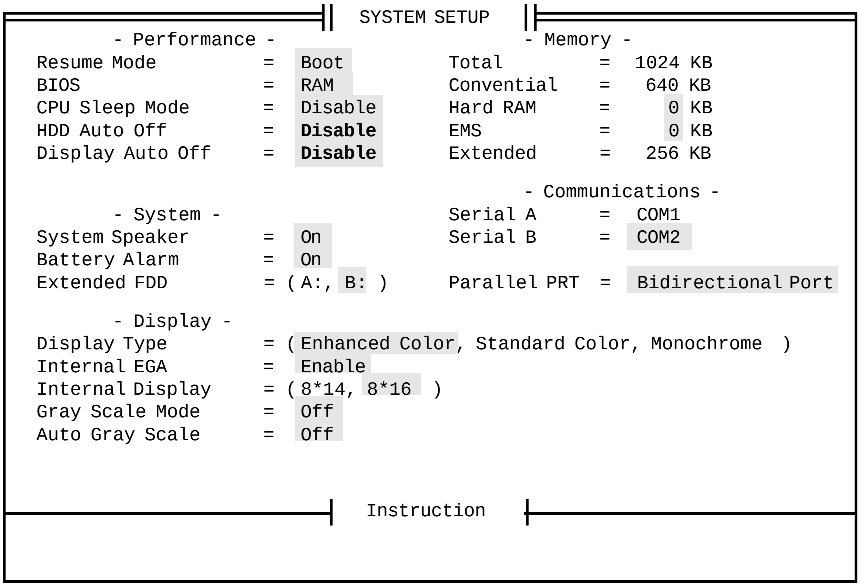
- Toshiba T1200
- Toshiba T1600
- Toshiba T3100e
- Toshiba T5200 with PC-Interface
- Toshiba T5200 with Toshiba-Interface

Parameters to be used for trouble free operation of the B&R PROgramming SYStem are printed within the configuration in **bold** !

4.1. SETUP MENU FOR THE TOSHIBA T1200

```
Internal Floppy Disk Drives:  1 Drive      2 Drives
Built-in Modem:               Disable      Enable
Power On Resume:              Disable      Enable
Primary Display Type:
           Other Type    Color[40*25]  Monochrome    Color[80*25]
Built-in RS-232C Port Assign.: COM1    COM2
Printer Port Bi-directional:  Disable      Enable
Lower Battery Speaker Alarm:  Disable      Enable
System Speaker:               Disable      Enable
Built-in Expanded Memory Port Address:
           208H   218H   258H   268H   2B8H   2E8H   Disable
Hard Ram Size:
           0KB    128KB  192KB  256KB  320KB  384KB
Keyboard Type:                84-keys      101-keys
```

4.2. SETUP MENU FOR THE TOSHIBA T1600



```
                               SYSTEM SETUP
       - Performance -                         - Memory -
Resume Mode          =  Boot         Total          =  1024 KB
BIOS                 =  RAM          Convential     =   640 KB
CPU Sleep Mode       =  Disable      Hard RAM       =     0 KB
HDD Auto Off         =  Disable      EMS            =     0 KB
Display Auto Off     =  Disable      Extended       =   256 KB

                                           - Communications -
       - System -                    Serial A       =  COM1
System Speaker       =  On           Serial B       =  COM2
Battery Alarm        =  On
Extended FDD         = ( A:, B: )    Parallel PRT   =  Bidirectional Port

       - Display -
Display Type         = ( Enhanced Color, Standard Color, Monochrome  )
Internal EGA         =  Enable
Internal Display     = ( 8*14, 8*16  )
Gray Scale Mode      =  Off
Auto Gray Scale      =  Off


                                Instruction
```

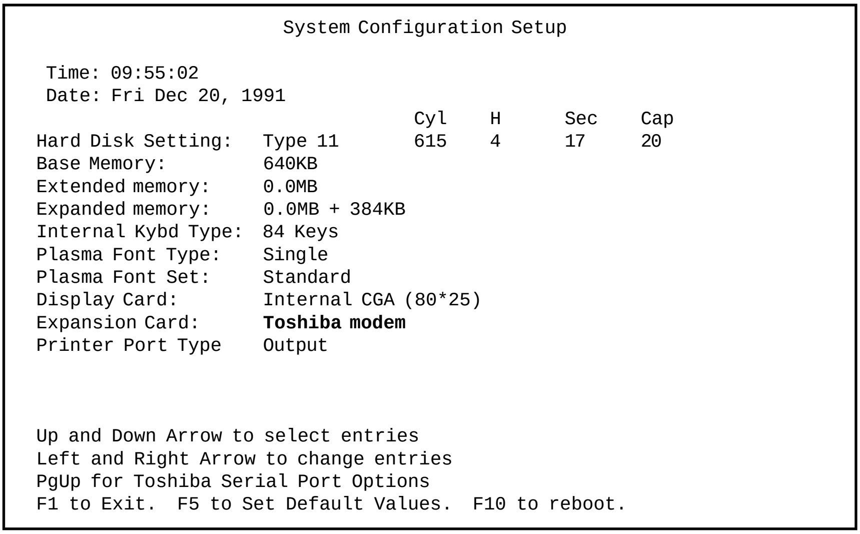
The selections "HDD Auto Off" and "Display Auto Off" must stay set to **"Disable"**.

4.3. SETUPMENU FOR THE TOSHIBA T3100e

4.3.1 T3100e with DOS 3.30

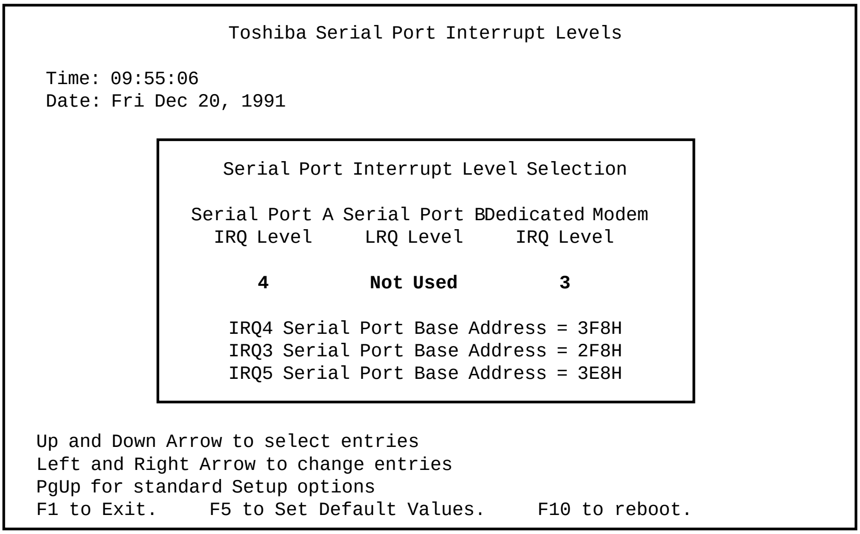
```
1. Hard disk type    = 11 - Cyl= 615, H= 4, S/T= 17, Cap= 20MB
2. Memory size
       System memory                = 640KB
       Extended memory              = 0MB
       Expanded memory              = 0MB + 384KB
3. Internal KB type                 = 84 Keys
4. Plasma font type                 = Single
5. Plasma font set                  = Standard
6. Display card                     = Internal CGA (80*25)
7. Card type in Expansion slot      = Toshiba modem
8. Printer port type                = Output port
9. Serial port
       IRQ5 SIO base address        = 3E8h
       Serial A IRQ level           = 4 (I/O base address = 3F8h)
       Serial B IRQ level           = Not used
       Optional port IRQ level      = 3 (I/O base address = 2F8h)
```

4.3.2. T3100e with DOS 4.00



```
                          System Configuration Setup

 Time: 09:55:02
 Date: Fri Dec 20, 1991
                                        Cyl    H      Sec    Cap
Hard Disk Setting:   Type 11       615    4      17     20
Base Memory:         640KB
Extended memory:     0.0MB
Expanded memory:     0.0MB + 384KB
Internal Kybd Type:  84 Keys
Plasma Font Type:    Single
Plasma Font Set:     Standard
Display Card:        Internal CGA (80*25)
Expansion Card:      Toshiba modem
Printer Port Type    Output



Up and Down Arrow to select entries
Left and Right Arrow to change entries
PgUp for Toshiba Serial Port Options
F1 to Exit.  F5 to Set Default Values.  F10 to reboot.
```

By pressing the [PgUp] key the following menu appears:



```
                     Toshiba Serial Port Interrupt Levels

 Time: 09:55:06
 Date: Fri Dec 20, 1991

              Serial Port Interrupt Level Selection

          Serial Port A Serial Port BDedicated Modem
            IRQ Level     LRQ Level     IRQ Level

                4         Not Used         3

              IRQ4 Serial Port Base Address = 3F8H
              IRQ3 Serial Port Base Address = 2F8H
              IRQ5 Serial Port Base Address = 3E8H

Up and Down Arrow to select entries
Left and Right Arrow to change entries
PgUp for standard Setup options
F1 to Exit.     F5 to Set Default Values.     F10 to reboot.
```

4.4. SETUP MENU FOR THE TOSHIBA T5200 WITH PC-INTERFACE BRIFPC-0

```
1. Hard disk type    = 11 - Cyl= 805, H= 4, S/T= 26, Cap= 40MB
2. Memory size
       System memory              = 640KB
       Extended memory            = 1MB
       Expanded memory            = 0MB + 288KB (95KB used as a fast ROM)
3. Display
       Plasma display adapter     = VGA compatible
       Display device             = Plasma
       Plasma display mode        = Color
       Plasma gray scale          = Normal: Semi-bright     Intense: Bright
4. Printer port type              = Output Port
5. Serial port
       IRQ5 SIO base address      = 3E8h
       Serial A IRQ level         = 4 (I/O base address = 3F8h)
       Serial B IRQ level         = Not used
       Toshiba modem IRQ level    = Not used
6. Bus speed                      = Normal
```

4.5. SETUP MENU FOR THE TOSHIBA T5200 WITH TOSHIBA-INTERFACE BRIFTO-0

```
1. Hard disk type    = 11 - Cyl= 805, H= 4, S/T= 26, Cap= 40MB
2. Memory size
       System memory              = 640KB
       Extended memory            = 1MB
       Expanded memory            = 0MB + 288KB (95KB used as a fast ROM)
3. Display
       Plasma display adapter     = VGA compatible
       Display device             = Plasma
       Plasma display mode        = Color
       Plasma gray scale          = Normal: Semi-bright     Intense: Bright
4. Printer port type              = Output Port
5. Serial port
       IRQ5 SIO base address      = 3E8h
       Serial A IRQ level         = 4 (I/O base address = 3F8h)
       Serial B IRQ level         = Not used
       Toshiba modem IRQ level    = 3 (I/O base address = 2F8h)
6. Bus speed                      = Normal
```

CHAPTER 3 - INTRODUCTION

10.8 UTILIZING LABELS (Page 3-28)

Labels are symbolic addresses which are necessary for structured programming. A PROgramming SYStem user utilizes labels in many different places:

- STL-labels, can be assigned to every program line.
- LAD-names are labels, which show up in an STL translation.
- Labels can be used in LADs => Label Contact (see Chapter 4 “3.6. Contacts and Connections”)
- TAB-names are labels, which do not show up in an STL.
- SEQC-names are labels, which show up in an STL translation.
- FBK-names are labels, which sow up in an STL translation.

> **NOTE:**
>
> A label must begin with a letter and is up to four characters long.
>
> The number of labels is restricted to max. 1499. When labels are deleted (e.g. by editing a program line and deleting a label field), they remain in the PROgramming SYStem memory. This memory area only becomes free after a Run-Test (RNT) or a RUN is executed.

Operation commands such as LIST, RUN, COPY, MOVE etc. can also be used as label names. These names can be used freely in the STL editor. To enable these label entries in the command line, you must insert the name between quotation marks .

Example: To save a program part from the label LIST until line 1000 as a macro the following line must be entered:

```
[F8 SAVE] <FNAME> [F5 MACRO] [F1 FROM] "LIST" [F2 UP_TO] 1000 [↵]
```

To go directly to the STL program line with the label LIST for STL editing, you must enter *"LIST"* in the command line.

12. INCOMPATIBILITY OF PROGRAM TRANSLATION TO CP80 MODE

Address range with indexed addressing:

X -128 to X +127

Make sure with STL commands that:

MUL	Zero Flag is influenced
TAB	Status register is not changed
TBA	Status register is not changed
LSRD	Zero Flag is not defined
ASLD	Zero Flag is not defined

Step sequencer:

If a CP40 program uses a step sequencer and is translated to CP60/CP80 mode, care should be taken that steps which are not used in the step sequence are not executed with RTS but with JMP instead. In using the 1 byte command RTS, step sequencing is no longer guaranteed. With the 3 byte command JMP <LABEL> and an RTS with the <LABEL>, the step sequence functions correctly again. .

Program in CP40:

```
 0       LDAA      R0000
 1       JSR       SSW
 2       END
 3       JMP       S1
 4       NOP
 5       RTS
 6       JMP       S2
 7       JMP       S3
 :
 :
```

Program in CP80:

```
           0        LDAA      R0000
           3        JSR       SSW
3 Byte     6        NOP
           7        END
           9        JMP       S1
          12        JMP       LAB1
          15        JMP       LAB1
          18        JMP       S2
          21        JMP       S3
          22  LAB1  RTS
           :
           :
```

CHAPTER 5 - STATEMENT LIST PROGRAMMING

3. STL EDITOR STRUCTURE (Pages 5-4 and 5-5)

a) Status line:

NOTE:

A label must begin with a letter and is up to four characters long.

The number of labels is restricted to max. 1499. When labels are deleted (e.g. by editing a program line and deleting a label field), they remain in the PROgramming SYStem memory. This memory area only becomes free after a Run-Test (RNT) or a RUN is executed.

7.2.2. DEBUG (Page 5-32)

Command: ***DEBUG* [↵]**

Function: The debug menu is called. The program is not sent to the PLC-CPU.

If anything is wrong with the PLC connection (programming device interface defect, online cable not connected etc.) the following corresponding messages can be displayed:

```
E052  MISSING ONLINE INTERFACE CARD
E056  PLC DISCONNECTED
E057  PLC ERROR
```

PROGRAMMING SYSTEM CHANGES FROM VERSION 5.40 TO 5.50

1. REQUIREMENTS

To install the PROgramming SYStem you need at least 540 KBytes (552.960 bytes) of free space within your 640 Kbyte DOS memory to run the software without problems.

2. COMMAND CHANGES

2.1. RANGE DEFINITION

The range definitions (FROM, UP_TO, #) for the following commands have been changed to:

- CHAN <Range>
- FIND <Range>
- FINDL <Range>
- FINDP <Range>
- COPY <Range> <Destination>
- MOVE <Range> <Destination>
- CLEAR BLOCK <Range> (was DELETE BLOCK <Range>)
- SAVE <FNAME> MACRO <Range>
- LOAD MACRO <FNAME> <Destination>

A range to be searched for, copied or moved are now only defined with the FROM and UP_TO commands. The following are no longer possible.

- **#** (Length of the range in lines)
- **current cursor position**

  Example: A search with FIND (FINDL, FINDP) for the range from the current cursor position to line 3000:

  FIND UP_TO 3000 [↵]   is no longer possible

  In its place the full range must be defined:

  FIND FROM 1200 UP_TO 3000 [↵]

The softkey line allocation has been adjusted slightly to compensate for the changes made.

With the commands MOVE, COPY and LOAD MACRO the destination <Destination> with TO <LN/LB/ADDR> must be entered. The destination is not determined by the current cursor position.

2.2. LOADING, SAVING, ...

Saving Macros

If saving a macro from an STL, SYMB or TAB the desired editor **no longer must** be entered with the command.

OLD: `SAVE <FNAME> MACRO STL/SYMB/TAB <Range>`

NEW: `SAVE <FNAME> MACRO <Range>`

The macro is now saved from the editor (STL, SYMB or TAB) which is currently active.

Saving a part of a TAB or SYMB in ASCII format

If saving a TAB or a SYMB part in ASCII format the desired editor **no longer must** be entered with the command.

OLD: `SAVE <FNAME> ASCII SYMB/TAB <Range>`

NEW: `SAVE <FNAME> ASCII <Range>`

The corresponding part to be saved comes from the active editor.

Loading Macros

If loading a macro from an STL, SYMB or a TAB the desired editor **no longer must** be entered with the command.

OLD: `LOAD MACRO <FNAME> STL/SYMB/TAB <Destination>`

NEW: `LOAD MACRO <FNAME> <Destination>`

The macro is now loaded from the editor (STL, SYMB or TAB) which is currently active.

Loading a part of a TAB or SYMB in ASCII format

If loading parts of a TAB or SYMB the desired editor **must** be entered with the command.

```
LOAD ASCII SYMB <FNAME> [↵]
LOAD ASCII TAB <TNAME> <FNAME> [↵]
```

Display Directory

The output of one file type (display of files with a certain extension) is no longer possible. The following entries are possible:

DIR [↵]	Default directory display
DIR <NAME> [↵]	Display the directory with the name entered

NOTE:

To display files with a certain extension the LOAD command can be used.
If, for example, all program files in the default directory should be displayed the following command is entered.

```
LOAD PROG [↵]
```

Copying, Renaming and Deleting Files

The `COPY DISK`, `RENAME` and `DELETE` commands have been changed. Instead of the file type (PROG, SYMB, TAB, ...) the file name together with the extension is now entered.

```
OLD: COPY DISK <File type> <FNAME1> TO <FNAME2> [↵]
NEW: COPY DISK <FNAME1.EXT> <FNAME2.EXT> [↵]

OLD: RENAME <File type> <FNAME1> TO <FNAME2> [↵]
NEW: RENAME <FNAME1.EXT> <FNAME2.EXT> [↵]

OLD: DELETE <File type> <FNAME> [↵]
NEW: DELETE <FNAME.EXT> [↵]
```

2.3. PRINTING

Printing and listing programs or program parts

The following commands are no longer available:

```
PRINT PROG <Range>          LIST PROG <Range>
PRINT SYMB <Range>          LIST SYMB <Range>
PRINT TAB <Range>           LIST TAB <Range>
PRINT LAB                   LIST LAB
XREF
```

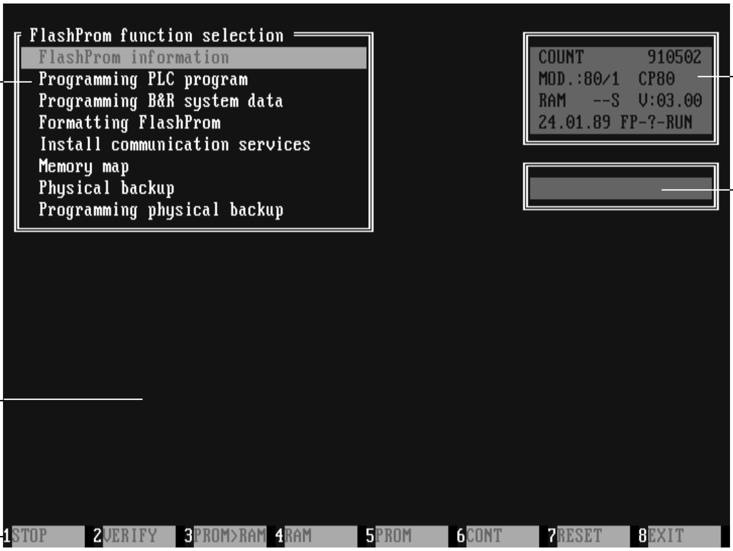
Only selections possible from the print menu are available (can be called with `DMEN`, `PRINT` or `LIST`).

NEW FUNCTIONS for the PROgramming SYStem Version 5.50

The FlashPROM Menu

A	B	C	PG1000	CP80	PG-PC	PC80
●	●	●		●		●

When the function ***EEPROM* [↵]** is selected in the command line of the PROgramming SYStem, the following menu is displayed if a FlashPROM is utilized:



```
FlashProm function selection
 FlashProm information
 Programming PLC program
 Programming B&R system data
 Formatting FlashProm
 Install communication services
 Memory map
 Physical backup
 Programming physical backup
```

```
COUNT        910502
MOD.:80/1  CP80
RAM   --S  V:03.00
24.01.89 FP-?-RUN
```

```
1STOP  2VERIFY  3PROM>RAM 4RAM  5PROM  6CONT  7RESET  8EXIT
```

PLC status window

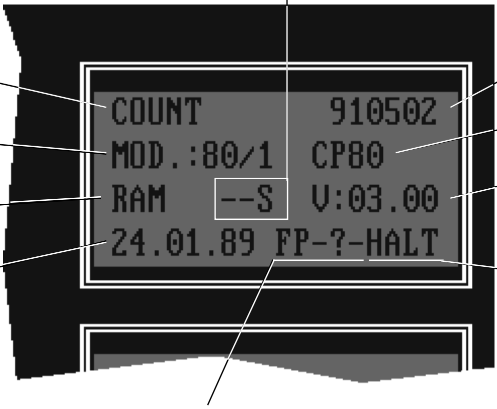
Selection window for FlashPROM functions

Programming status

Messages to the user

Softkey line

PLC Status Window



```
COUNT        910502
MOD.:80/1  CP80
RAM   --S  V:03.00
24.01.89 FP-?-HALT
```

Additional information: A = Arithmetic processor
S = Serial interface

Program name

Module series

Program run in RAM/PROM

CPU operating system version date

Program date

Module type

CPU operating system version number

Processor status: HALT/RUN

PROM type

Programming Status

This window shows the type of FlashPROM access while functions are being executed. The following messages may be displayed:

```
READ:     xx xx xx
FORMAT:   xx xx xx
PROGRAM:  xx xx xx
```

Address in FlashPROM changes constantly while functions are being completed.

Function Key Assignments

[F1 STOP] This function stops the program in the controller. This is indicated in the PLC status window with “HALT”.

The controller status does not change. Outputs are not reset; the current status remains.

CAUTION: Outputs, which control motors for example, remain set.

[F2 VERIFY] Compares programs in PROM and RAM.

If both programs are identical, the following is displayed:

```
W247  EEPROM Verify OK !
```

If the programs are not identical, this error message is displayed:

```
E167  Verify NOT OK !  (WP,NOT EMPTY, DEFECT)?
```

[F3 PROM>RAM] Copies program from PROM to RAM.

The controller must be switched to PROM, otherwise this error message is displayed:

```
E170  COPY PROM TO RAM ONLY IF PLC IN PROM MODE
```

[F4 RAM] Switches PLC to RAM (Switching occurs in the END routine). Program in RAM is run.

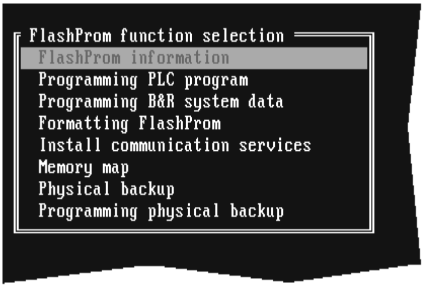
[F5 PROM] Switches PLC to PROM (Switching occurs in the END routine). Program in PROM is run.

[F6 CONT] If the PLC CPU is in HALT mode, the program is continued in the PLC.

[F7 RESET] The PLC is halted and reset to a defined base setting. All outputs are cleared.

[F8 EXIT] or **[Esc]** Exits the FlashPROM menu.

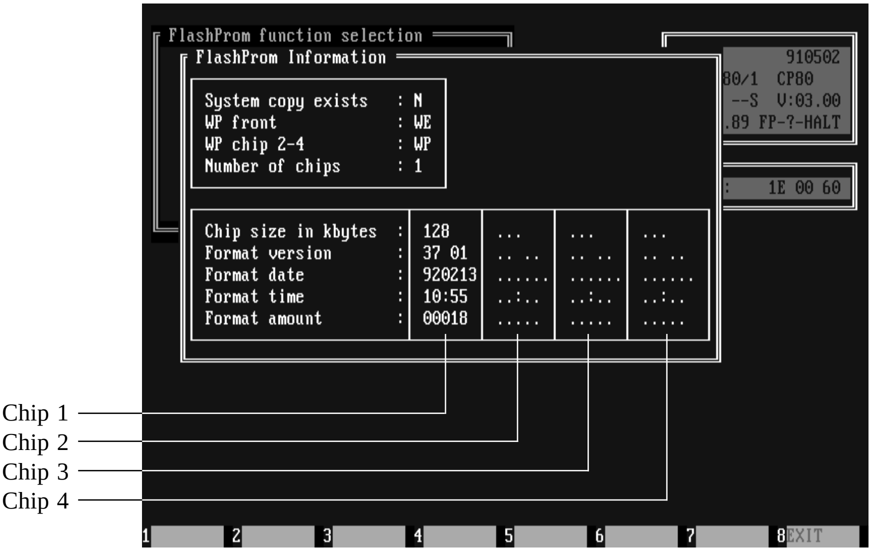
Function Selection Window


FlashProm function selection
FlashProm information
Programming PLC program
Programming B&R system data
Formatting FlashProm
Install communication services
Memory map
Physical backup
Programming physical backup

Select a function with the [❖] keys, then confirm it with [↵], which activates the function.

FlashPROM Information

When you select this function, a window which provides information on the FlashPROM appears.



```
FlashProm function selection
 FlashProm Information
  System copy exists   : N
  WP front             : WE
  WP chip 2-4          : WP
  Number of chips      : 1

  Chip size in kbytes  : 128    ...    ...    ...
  Format version       : 37 01  .. ..  .. ..  .. ..
  Format date          : 920213 ...... ...... ......
  Format time          : 10:55  ..:..  ..:..  ..:..
  Format amount        : 00018  .....  .....  .....
```

```
          910502
80/1   CP80
 --S   V:03.00
.89 FP-?-HALT

:    1E 00 60
```

Function keys:

[F8 EXIT] or **[Esc]** Closes the window and returns you to the FlashPROM menu.

Info window:

`System copy exists:`	**`Y`**	The system module including the PLC program has been programmed in the FlashPROM (chip 1).
	`N`	Chip 1 has not been programmed. (But there may be system or user data on the blocks that are not reserved for the PLC program.)
`WP front:`	**`WP`**	The WE/WP switch on the front of the FlashPROM module is set to WE (= write-protected).
	`WE`	The WE/WP on the front of the FlashPROM module is set to WE (=write-enabled).
`WP chip 2-4:`	**`WE`** or **`WP`**	With a jumper on the FlashPROM module you can select whether chips 2 - 4 are always set to WE, or if they should match the WP front switch.
`Number of chips:`	**`n`**	Number of memory blocks in the FlashPROM module (1 or 3; 1 ⇒ FP128, 3 ⇒ FP384).

The following information is given for each chip.

Chip size in Kbytes:	**xxx**	Chip storage capacity (e.g. 128 Kbytes)
Format version:	**xx xx**	System module version number
Format date:	**xxxxxx**	Date of format received from real-time clock in the CPU
Format time:	**xx:xx**	Time of format received from real-time clock in the CPU
Format amount:	**xxxxx**	Indicates how many times the chip has been formatted (e.g. 038 = the chip has been formatted 38 times)

Programming PLC Program

The current PLC program in the CPU RAM is to be programmed to chip 1 of the FlashPROM along with the system module.

If you select this function with the [❖] keys and activate it with [↵], the CPU must be in the following mode:

- the CPU must be in HALT mode
- the CPU must be SET to RAM

If one of these conditions is not met, one of the following error messages is displayed in the message line:

```
E249  Function only possible if PLC in HALT mode !
E222  Function only possible in RAM mode !
```

If chip 1 of the FlashPROM has just been formatted, the PLC program and the system module are programmed immediately. Otherwise, the entire chip is reformatted (approx. 1 minute).

If system or user data is stored on the chip, however, the following question appears in the message line:

```
W302  FlashPROM also used otherwise ! SAVE (Y/N/ESC)?
```

[Esc] The function is aborted and you return to the FlashPROM menu.

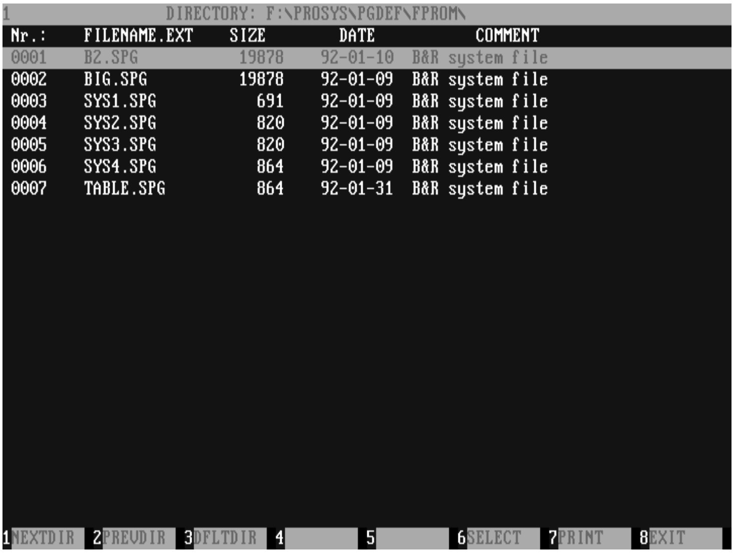
[N] Chip 1 is formatted, the PLC program and the system module are programmed.

Caution: User and system data is deleted!

[Y]
- All user and system data found on chip 1 is saved in a tempory file on the hard disk of the PG.
- Chip 1 is formatted.
- The PLC program and the system module are programmed.
- The saved user and system data is programmed again.

Programming B&R System Data [1)]

When this function is selected, a list of all B&R system data files which are located in the default directory appears for you to choose from.



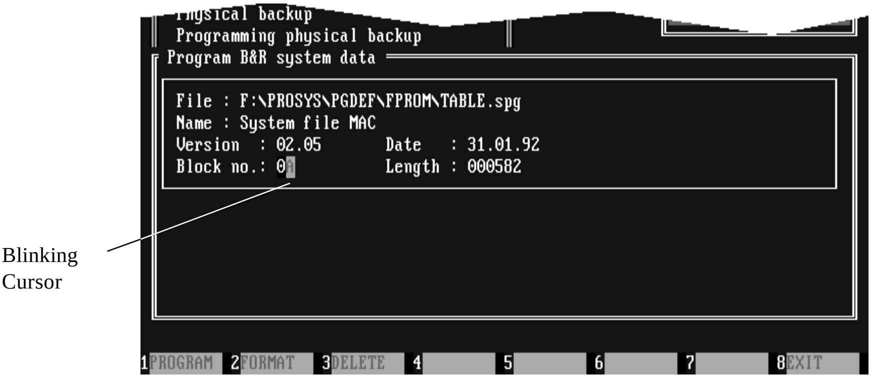
```
1                  DIRECTORY: F:\PROSYS\PGDEF\FPROM\
Nr.:     FILENAME.EXT     SIZE        DATE         COMMENT
0001     B2.SPG           19878     92-01-10   B&R system file
0002     BIG.SPG          19878     92-01-09   B&R system file
0003     SYS1.SPG           691     92-01-09   B&R system file
0004     SYS2.SPG           820     92-01-09   B&R system file
0005     SYS3.SPG           820     92-01-09   B&R system file
0006     SYS4.SPG           864     92-01-09   B&R system file
0007     TABLE.SPG          864     92-01-31   B&R system file
```

Key	FUNCTION
[F1 NEXTDIR]	If one of the directories is marked (directory - indicated by <DIR>), it will be opened. The contents of this directory will be displayed. This function corresponds to the DOS command “`cd <directory name>`”.
[F2 PREVDIR]	The previous directory is returned to and its contents are displayed. [F2 PREVDIR] corresponds to the DOS command “`cd ..`”.
[F3 DEFDIR]	The current directory is defined as the default directory of the current drive.
[F6 SELECT] **[↵]**	The system file chosen with the [❖] keys is selected.
[F7 PRINT]	The directory displayed on the screen is printed out.
[F8 EXIT] **[Esc]**	Exit returns you to the FlashPROM menu.

[1)] B&R system data is offered by B&R for various applications. (e.g. boot table for the MAC1).

When you select a system file with [F6 SELECT] or [↵], a window is displayed for entering the FlashPROM block number (default value from the system file has already been entered). Various information is given which is read from the header of the system file. The block number indicates the target address in the FlashPROM where the system file is to be programmed.

The default value of the block number can be changed.



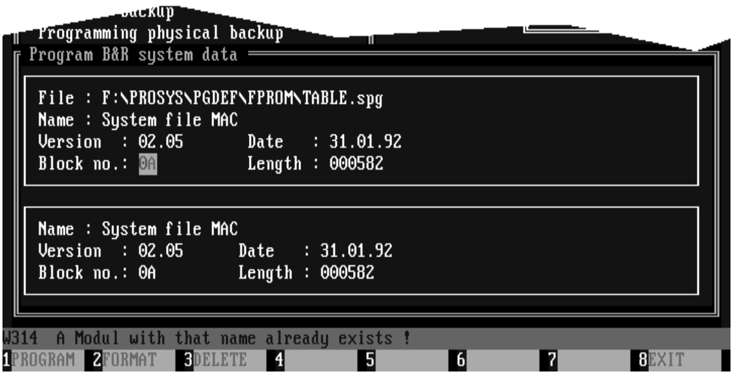
```
 ysical backup
 Programming physical backup
Program B&R system data

 File : F:\PROSYS\PGDEF\FPROM\TABLE.spg
 Name : System file MAC
 Version  : 02.05       Date   : 31.01.92
 Block no.: 0A          Length : 000582
```

Blinking Cursor

[F1 PROGRAM]

1) If the necessary blocks are free starting with the indicated block number, the selected system file is programmed in the FlashPROM as module and entered in the module directory. In this case, the keys [F2 FORMAT] and [F3 DELETE] are not active. If they are pressed, the appropriate message will appear in the message line.

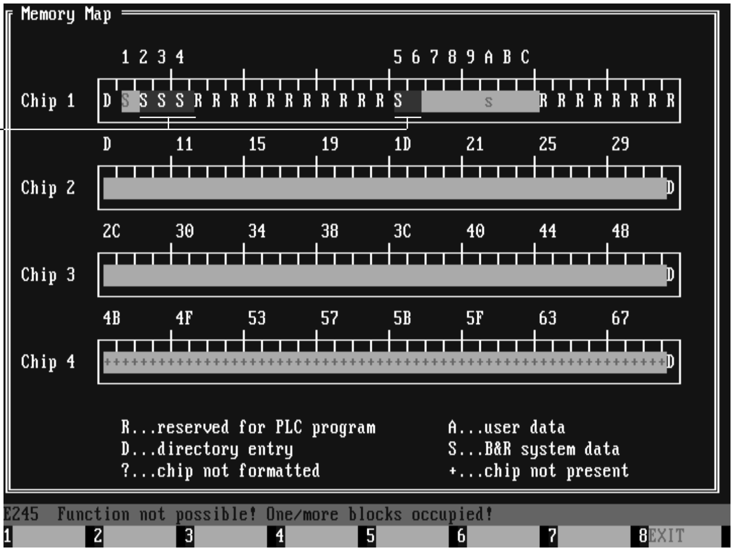
2) If a module of the same name already exists on the FlashPROM (with the same or different version number), it is displayed in an additional window under the first one. The appropriate message is displayed in the message line:



```
 backup
 Programming physical backup
Program B&R system data

 File : F:\PROSYS\PGDEF\FPROM\TABLE.spg
 Name : System file MAC
 Version  : 02.05       Date   : 31.01.92
 Block no.: 0A          Length : 000582

 Name : System file MAC
 Version  : 02.05      Date   : 31.01.92
 Block no.: 0A         Length : 000582

W314  A Modul with that name already exists !
```

In this case, the function key [F3 DELETE] is active.

3) If the necessary blocks are not free starting with the indicated block number, this is shown in the message line and a FlashPROM memory map is displayed. The overlapping areas are marked in this map.



R...reserved for PLC program
D...directory entry
?...chip not formatted
A...user data
S...B&R system data
+...chip not present

E245 Function not possible! One/more blocks occupied!

Overlapping Area

The memory map can be exited with [F8 EXIT] or [Esc]. You now have the following options:

1) You can enter the number of a block that is free and program it to the FlashPROM with [F1_PROGRAM].
2) You can format the necessary chips with the function [F2 FORMAT].

[F2 FORMAT]

The necessary chips are formatted.

[F3 DELETE]

If there is a module with the same name as the one to be programmed to the FlashPROM, the module that is already there can be deleted from the module directory with this function. The blocks occupied by the deleted module are not free, however; they can only be used again after the chip has been reformatted. These modules are distinguished by lower-case letters in the memory map (S => s, U => u).

[F8 EXIT] or **[Esc]**

Returns you to the FlashPROM menu.

Formatting FlashPROM

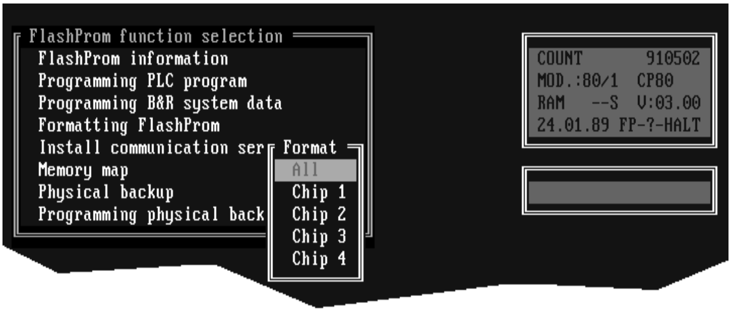
When you select this function, a window appears where you can select whether all chips or individual chips are to be formatted (deleted).

If the CPU is in RUN mode, the following error message is displayed:

`E249  Function only possible if PLC in HALT mode !`

HALT mode is selected for the CPU with [F1 STOP] or [F7 RESET] !

If the CPU is in HALT mode, the following window is opened:



Select a chip by moving the cursor up and down with the key [↑] and [↓]. Start formatting the chips by confirming your choice with [↵]. (One chip takes approximately 50 seconds.)

If a chip is not present, this is shown in the message line:

`E230  Chip not present !`

If the CPU is in PROM mode, the following message is displayed:

`E222  Function only possible in RAM mode !`

If the CPU is in RAM mode, the chip selected is formatted.

Install Communication Services

The communication services module is usually transferred to the PLC and installed in the CPU RAM in RUN mode. If there is operating trouble (power down, run-down accumulator, or no battery in power supply unit), this module can be destroyed. With this function, the FlashPROM communication services are installed without the need of a preceding RUN.

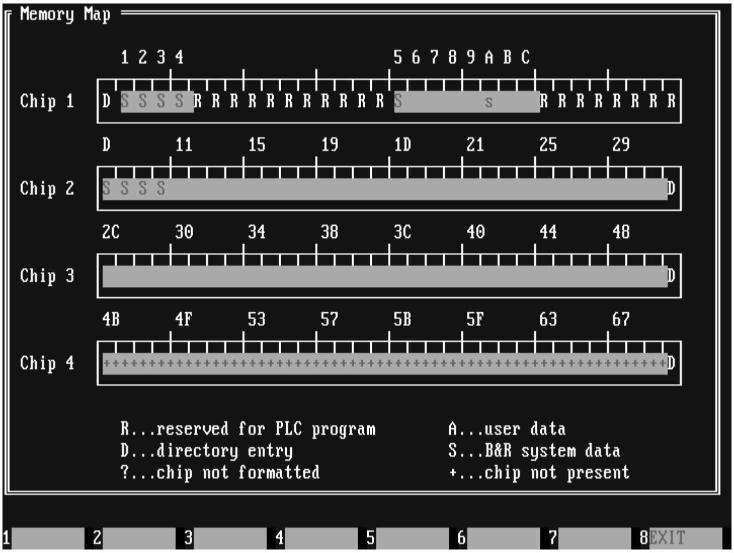
Memory Map

This function gives you information on FlashPROM distribution. It offers an overview of all occupied blocks in the PROM (1 block => 4 Kbytes). Each block contains a one-character code for the type of occupancy:

R .... reserved for PLC program and system module
D .... directory (data is stored here for all blocks on a chip and their occupancy)
? .... chip not formatted
U .... user data [1)]
u .... "deleted" user data (entry in directory deleted)
S .... B&R system data
s .... "deleted" B&R system data (entry in directory deleted)
+ .... chip not present

> **Note:** Blocks with "deleted" user or B&R system data can be programmed only after the chip has been reformatted.

The memory map has the following format:

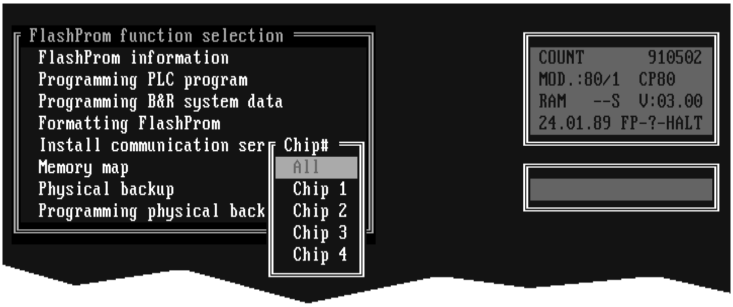


```
Memory Map

          1 2 3 4                             5 6 7 8 9 A B C
Chip 1    D S S S S R R R R R R R R R R R S        s        R R R R R R R R

          D       11      15      19      1D      21      25      29
Chip 2    S S S S                                                        D

          2C      30      34      38      3C      40      44      48
Chip 3                                                                   D

          4B      4F      53      57      5B      5F      63      67
Chip 4    ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++D

          R...reserved for PLC program          A...user data
          D...directory entry                   S...B&R system data
          ?...chip not formatted                +...chip not present
```

[1)] Not currently available. This function is being developed.

Physical Backup

All occupied blocks of one or more chips are saved. The user can make one of the following selections in the window.

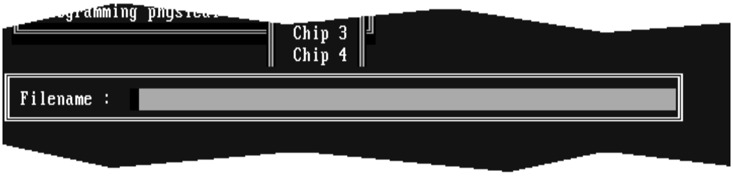


```
COUNT       910502
MOD.:80/1  CP80
RAM   --S  V:03.00
24.01.89 FP-?-HALT
```

With keys [↑] and [↓], either all chips or an individual chip is selected. The selection is confirmed with [↵]. If the selected chip does not exist, an error message is displayed in the message line:

```
E230  Chip not present !
```

If the selected chip is present, a window is opened where a file name without extension must be entered.



The physical backup is stored under this name in the default directory with the extension “.PHY”. Any path name with drive code can also be entered in front of the file name (e.g: A:\BACKUP\CHIP1).

After the file name is confirmed with [↵], the backup will be done.

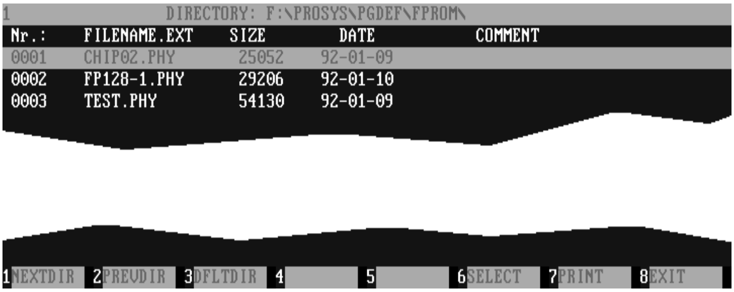
If there are no occupied blocks on the chip, the following message is displayed:

```
E244  Chip not defined !
```

Only data with the proper module header is saved (system and user data).
The PLC program is not saved with a physical backup.

Programming Physical Backup

When this function is activated (CPU must be in HALT mode), all files with the extension "`.PHY`" are listed. The desired backup file can be selected with the [↑] and [↓] keys.



```
1                  DIRECTORY: F:\PROSYS\PGDEF\FPROM\
Nr.:    FILENAME.EXT    SIZE       DATE          COMMENT
0001    CHIP02.PHY       25052    92-01-09
0002    FP128-1.PHY      29206    92-01-10
0003    TEST.PHY         54130    92-01-09

1NEXTDIR 2PREVDIR 3DFLTDIR 4        5        6SELECT  7PRINT   8EXIT
```

Key	FUNCTION
[F1 NEXTDIR]	If one of the directories is marked (directory - indicated by <DIR>), it will be opened. The contents of this directory will be displayed. This function corresponds to the DOS command "`cd <directory name>`".
[F2 PREVDIR]	The previous directory is returned to and its contents are displayed. [F2 PREVDIR] corresponds to the DOS command "`cd ..`".
[F3 DEFDIR]	The current directory is defined as the default directory of the current drive.
[F6 SELECT] **[↵]**	The backup file chosen with the [❖] keys is selected.
[F7 PRINT]	The directory displayed on the screen is printed out.
[F8 EXIT] **[Esc]**	Exit returns you to the FlashPROM menu.

All blocks saved in the backup file are programmed again in their original positions in the FlashPROM. For this reason, all required blocks must be free. Otherwise, the function is interrupted with an error message.

`E245  Function not possible ! One/more blocks occupied !`

If blocks that should be saved on a chip which is not present on the PROM module are saved in the backup file, this is recognized, and the function is interrupted with the following error message:

`E250  Restore - required chip not present !`

APPENDIX G

PROGRAMMING SYSTEM UPDATES VERSION 5.60

Contents:

1. PROGRAM COMPARISON

A	B	C	PG1000	CP80	PG-PC	PC80
		●	●	●	●	●

1.1 GENERAL INFORMATION

Quite often a project must be developed in segments by different people or in fact by several groups of people e.g. visualization by one group and the logic by another. In the initial stages of operation the software can be modified again and again. These problems which arise can lead to a degree of uncertainty in the exact state of development. In this case it is necessary to compare programs or sub-programs.

Another case for the use of program comparisons within an application could be the absence of for the valid documented version. This can be caused if, for example, changes are made to the structure without updating the documented version or if the current software is lost because of a diskette failure. Now, what is left is a program without documentation (LOAD from PLC) and an old documented version. With the new COMPARE command these two programs can be checked for differences which can be easily changed.

1.2. COMPARISON PROCEDURES

In principle a program in window one can be compared with a program in window two, which means that no PLC is required for the procedure. If the last valid program is available in the PLC it must be loaded from the PLC to one of the windows with the LOAD PLC command. The second program is loaded in the other window. By entering the COMPCFG command now the COMPARE command configuration is set. In most cases the standard settings are correct.

By all means the report function and **auto mode** should be switched on the **first** time that COMPARE is run. This causes the entire program to be searched for differences which are then reported. This report of the differences can either be printed out through the serial or parallel interface or they can be saved in a file. From these the amount of difference between the two programs can be recognized immediately.

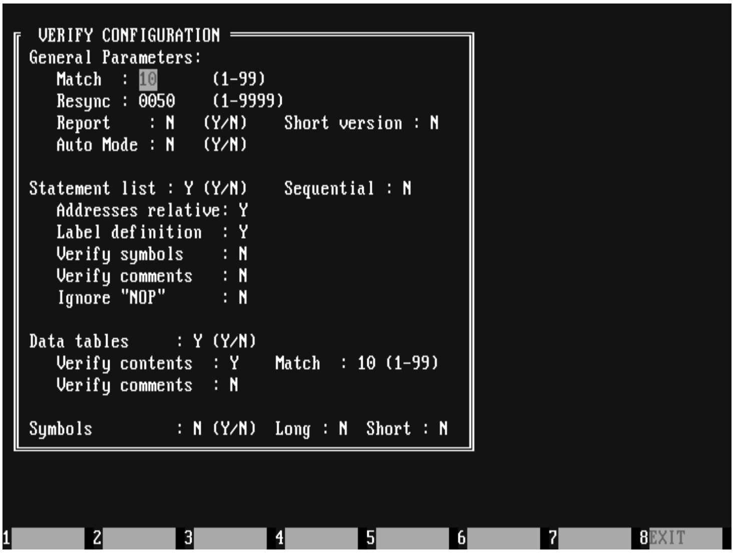
The second time COMPARE is run, either changes can be made directly line for line by going through the report or auto mode and the report function can be switched off and the routine can be restarted. When auto mode is off the routine will stop every time there is a difference and the corresponding corrections can then be made. By entering the COMPCONT command the routine is continued.

If individual lines in a window should be copied to the other window the "Mark" function is very helpful. Of course it is possible to call other commands between steps in the COMPARE routine such as e.g. "LOAD MACRO" etc.

1.3. CHANGE COMPARE CONFIGURATION

Command: ***COMPCFG [↵]***

Function: Calls a menu to modify the COMPARE configuration.



```
VERIFY CONFIGURATION
General Parameters:
   Match  : 10       (1-99)
   Resync : 0050     (1-9999)
   Report    : N   (Y/N)    Short version : N
   Auto Mode : N   (Y/N)

Statement list : Y (Y/N)    Sequential : N
   Addresses relative: Y
   Label definition  : Y
   Verify symbols    : N
   Verify comments   : N
   Ignore "NOP"      : N

Data tables     : Y (Y/N)
   Verify contents  : Y    Match  : 10 (1-99)
   Verify comments  : N

Symbols         : N (Y/N)  Long : N  Short : N
```

General Parameters

Match:

If the COMPARE operation is stopped because of a difference between the two programs it can be restarted with the COMPCONT command. After the COMPARE operation is continued the PROgramming SYStem searches for lines in both programs which are identical. This procedure is known as synchronization. COMPARE is however only continued now if the amount of lines defined for **Match** are identical, otherwise the function is interrupted with an error message.
A more precise explanation is given in the description of the COMPCONT command.

Resync:

If, during the synchronization, within the number of lines defined for **Resync** no block of identical lines (defined in **Match**) is found the function is broken with an error message.
This procedure is explained in more detail under the COMPCONT command description.

Report:

A report of differences can be made with the PROgramming SYStem by executing the COMPARE function. The report function can be switched **on** or **off** from this position in the menu with **Y / N**. The user can also define here with **Y / N** whether the report should be in short form. This report can be either printed through one of the interface ports with a setting in the print menu or put into a file.

Auto Mode:

If **Auto Mode** is switched on the COMPARE function will not be interrupted when it comes to a difference in program structure. This mode is only useful when used in connection with the report function!

Statement List

Entire statement lists can either be compared with one another or not by selecting either **Y** or **N** which turns the statement list comparison parameter **on** or **off**.

Sequential:

The behaviour of synchronization after the COMPCONT command can be influenced with this option.

y ***Sequential Synchronization:*** After the COMPCONT command is entered and if those lines according to the definitions for **Match** and **Resync** are found to be identical the following lines are searched. This search procedure is explained in detail in the COMPCONT command description.

N ***Label synchronization:*** After the COMPCONT command is entered, the FIXED-window is searched for the next label. When it is found the PROgramming SYStem searches the second window for the same label and the compare is started again. This method is used if an STL translation of a LAD is in different locations within the two STLs (when a LAD is changed the old STL translation is deleted and the new translation is added to the end of the STL).

Addresses relative:

Labels (for branch instructions etc.) can be assessed either absolute or relative with the COMPARE function.

Example:

(Window)	Lines to be compared:	Relative	Absolute
(1)	`0010      JSR     LABX 2764`	Program is the same	Program is different
(2)	`0024      JSR     LABX 1400`		

Label definition:

Label definition verification:

Example:

(Window)	Lines to be compared:	No	Yes
(1)	`0010 DEF1 LDAA    R 0200`	Program is the same	Program is different
(2)	`0017      LDAA    R 0200`		

Verify symbols:

Symbols to addresses can be verified

Example:

(Window)	Lines to be compared:	No	Yes
(1)	0010 DEF1 LDAA R 0200 Motor on	Program is the same	Program is different
(2)	0017 LDAA R 0200 Motor 1 on		

Verify comments:

Verification of comments or comment lines

Example:

(Window)	Lines to be compared:	No	Yes
(1)	0010 JSR INIT Initialize	Program is the same	Program is different
(2)	0017 JSR INIT Init-Up		

Ignore "NOP":

NOP lines are ignored

Example:

(Window)	Lines to be compared:	No	Yes
(1)	0101 LDAA R 0020 0102 NOP 0103 = O 010	Program is different	Program is the same
(2)	0176 LDAA R 0020 0177 = O 010		

Tables

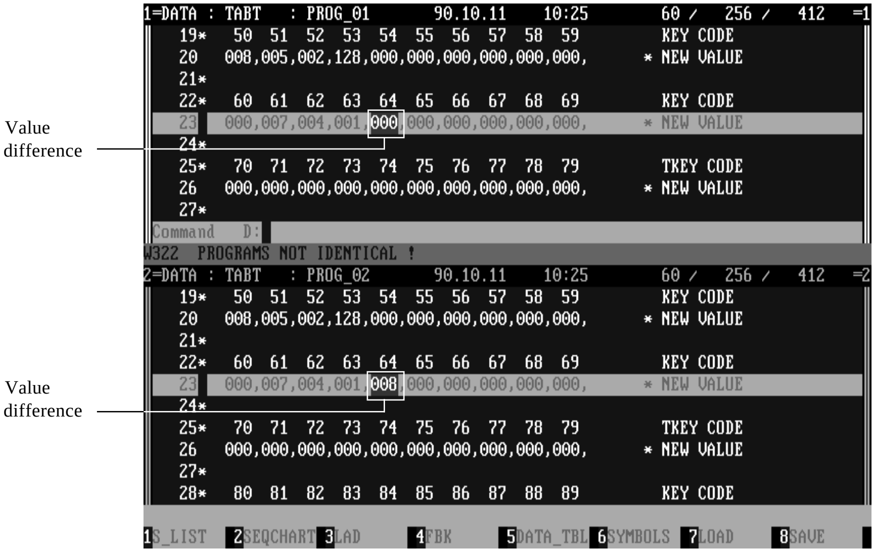
The table comparison is switched **on** or **off** by selecting **Y or N.**

Verify contents:

N Only the source lines in a table are compared.
Y Only the table's binary contents are compared byte for byte.

Match:

This value is only used in comparing the binary contents of two tables. If a difference is found while comparing the tables the COMPARE function is broken with an error message. The **Match** value is used to synchronize the continuation of the COMPARE function with the COMPCONT operation command.
If a value difference is found within the contents of the two tables all table entries with this value are displayed in red (blinking with an B/W or LCD monitor). E.g.:



```
1=DATA : TABT   : PROG_01       90.10.11    10:25       60 /   256 /   412   =1
    19*   50  51  52  53  54  55  56  57  58  59          KEY CODE
    20   008,005,002,128,000,000,000,000,000,000,       * NEW VALUE
    21*
    22*   60  61  62  63  64  65  66  67  68  69          KEY CODE
    23   000,007,004,001,000,000,000,000,000,000,       * NEW VALUE
    24*
    25*   70  71  72  73  74  75  76  77  78  79          TKEY CODE
    26   000,000,000,000,000,000,000,000,000,000,       * NEW VALUE
    27*
Command   D:
W322  PROGRAMS NOT IDENTICAL !
2=DATA : TABT   : PROG_02       90.10.11    10:25       60 /   256 /   412   =2
    19*   50  51  52  53  54  55  56  57  58  59          KEY CODE
    20   008,005,002,128,000,000,000,000,000,000,       * NEW VALUE
    21*
    22*   60  61  62  63  64  65  66  67  68  69          KEY CODE
    23   000,007,004,001,008,000,000,000,000,000,       * NEW VALUE
    24*
    25*   70  71  72  73  74  75  76  77  78  79          TKEY CODE
    26   000,000,000,000,000,000,000,000,000,000,       * NEW VALUE
    27*
    28*   80  81  82  83  84  85  86  87  88  89          KEY CODE
```

Value difference

Value difference

Verify comments:

N Comments and comment lines are ignored.
Y Comments and comment lines are verified for uniformity.

This parameter has no relevance in connection with the "verify contents" parameter.

Symbols

Symbol comparison is switched **on** or **off** with this function by selecting **Y** or **N.**

Long (STL-SYMB):

N STL-SYMBs are not checked.
Y STL-SYMBs are verified for uniformity.

Short (LAD-SYMB):

N LAD-SYMBs are not checked.
Y LAD-SYMBs are verified for uniformity.

Exiting the Menu

Leaving the COMPARE menu is done by pressing either the [Esc] key or the [F8 EXIT] key. When one of these keys is pressed the following message is displayed in the message line:

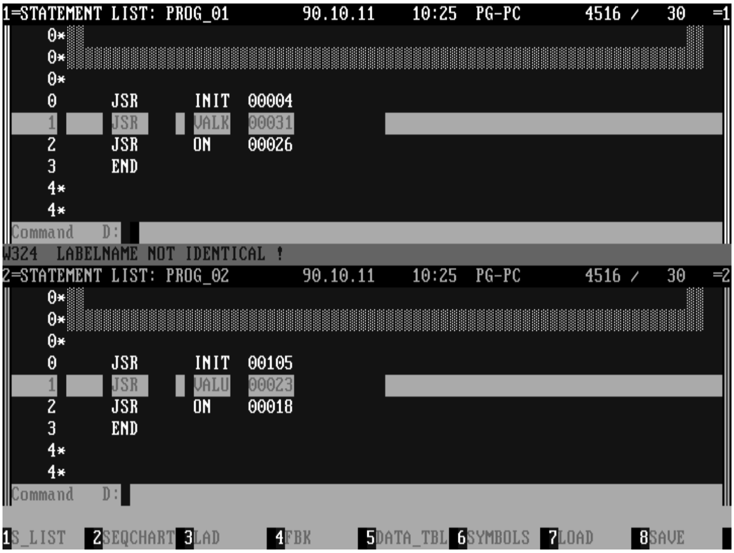
```
W300  WRITE DATA ? ( Y / N )
```

N   Configuration changes are not saved!
Y   Configuration changes are saved to a file!

1.4. STARTING THE COMPARISON

Command: ***COMPARE [↵]***

Function: Overall program comparison is executed according to the parameters set in the COMPARE configuration menu. After the command is entered both windows are opened. The window arrangement which can be set with the WINDOW operation command is ignored (this is replaced with the arrangement set for the COMPARE command). The screen is set up as follows:



```
1=STATEMENT LIST: PROG_01        90.10.11    10:25  PG-PC       4516 /   30   =1
     0*
     0*
     0*
     0      JSR      INIT  00004
     1      JSR      VALK  00031
     2      JSR      ON    00026
     3      END
     4*
     4*
Command    D:
W324  LABELNAME NOT IDENTICAL !
2=STATEMENT LIST: PROG_02        90.10.11    10:25  PG-PC       4516 /   30   =2
     0*
     0*
     0*
     0      JSR      INIT  00105
     1      JSR      VALU  00023
     2      JSR      ON    00018
     3      END
     4*
     4*
Command    D:

1S_LIST  2SEQCHART 3LAD      4FBK      5DATA_TBL 6SYMBOLS 7LOAD     8SAVE
```

Program lines (STL, TAB and SYMB) are compared and checked for uniformity in both windows. If a difference between the two is found the COMPARE function is broken with an error message and the editors for both windows are in this line. Any one of the following messages can be displayed:

```
W321  PROGRAMS ARE IDENTICAL !
W322  PROGRAMS NOT IDENTICAL !
W323  LABELDEFINITION NOT IDENTICAL !
W324  LABELNAME NOT IDENTICAL !
W325  ABSOLUTE ADDRESS NOT IDENTICAL !
W326  SYMBOLS NOT IDENTICAL !
W327  COMMENTS NOT IDENTICAL !
W329  NO PROGRAM LOADED !
W330  NO DATA TABLE EXISTS !
W331  DATA TABLE NAME NOT IDENTICAL !
```

```
        W337  COMPARE ONLY FOR IDENTICAL CPU TYPES !
```

If **Auto mode** is switched on the next identical line is found and the COMPARE routine is continued. If a synchronization is not possible the COMPARE routine is broken with the following message.

```
        W338  AUTO_MODE VERIFY - PROGRAMS NOT IDENTICAL !
```

Compare Sequence

The compare is carried out in the following sequence:

1. **Statement lists**
   The STL editor is positioned in line 0 in both windows. The program is now compared line for line.

2. **Tables**
   The TAB editor is positioned to the first table following the alphabetical sequence. The compare is then run for the names of the tables in alphabetical order. If a name is different the corresponding error message is displayed. Otherwise the comparison is done according to the COMPARE configuration.

3. **Symbols**
   Both windows are positioned with the SYMB editor at R 0000. The synchronization for SYMB compare is done by addresses only, i.e. only SYMBs (long or short - see configuration) with the same address are compared.

Depending on the compare configuration one or more editors can be ignored in the compare routine. If, for example, the statement list verification is switched off in the configuration menu compare immediately starts the routine in the tables.

Intervention

The compare routine can basically be interrupted for other tasks at any time during most procedures with the [Esc] key:

e.g.: Changing, deleting or inserting a line.
Changing, deleting or inserting a block.
Loading a macro etc.
Compare-configuration change

1.5. CONTINUE COMPARE

Command: ***COMPCONT [↵]***

Function: Overall program comparison is continued according to the parameters set in the COMPARE configuration menu. The user can optionally position the editor (STL, SYMB, TAB) in both windows and subsequently continue the compare function.

> **Attention:**
>
> Both windows must have the same editor or the following error message is displayed:
>
> ```
> W335  Continue - EDITOR NOT IDENTICAL !
> ```

After the command is entered both windows are opened. The window arrangement which can be set with the WINDOW operation command is ignored (this is replaced with the arrangement set for the COMPARE command).

The screen is set up the same as with the COMPARE command.

Synchronization

If the lines where the cursors in both windows are positioned on are not the same the PROgramming SYStem automatically does a synchronization, i.e. both windows are searched for the next identical line.

The parameters set in the configuration menu differentiate between *sequential synchronization* and *Label synchronization.*

The window in which the COMPARE command is given in is designated as the **FIXED window**.

Sequential synchronization

The **Match** and **Resync** parameters are used for this type of synchronization for which values can be set by the user. Both programs are compared from the **current position** up to the **current position + Resync**. Within this amount of lines a **Match**-amount of lines must be identical in both programs for the compare routine to continue. When two identical blocks are found the synchronization is successful and the following message is displayed:

```
W328  MATCH CONDITION SATISFIED  !
```

If the synchronization is not successful i.e. no block is found within the defined values set for **Resync** and **Match** the following message is displayed:

```
W342  SYNCHRONIZATION NOT POSSIBLE !
```

Label synchronization

In this type of synchronization the FIXED window is searched from the cursor position for the next label. This label is then searched for in the second window. When it is found the compare routine is started.

If no more labels exist in the FIXED window the following message is displayed:

```
W343  NO MORE LABELDEFINITIONS IN WINDOW !
```

If a label exists only in the FIXED window and not in the second window the compare routine is broken with the following message:

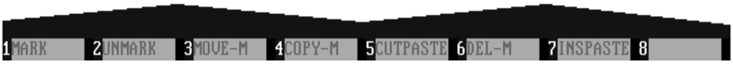
```
W339  LABELSYNCHRONIZATION NOT POSSIBLE !
```

2. MARKING FUNCTIONS

A	B	C	PG1000	CP80	PG-PC	PC80
●	●	●	●	●	●	●

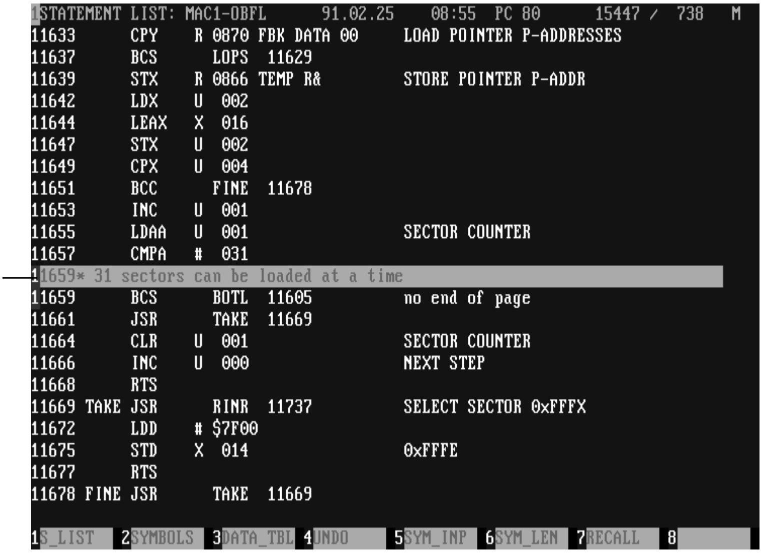
Ranges within an STL, TAB or SYMB editor can be marked with this function. Marked areas can be deleted, copied, moved or placed in temporary memory or inserted from temporary memory.

When the cursor is in an input line of an STL, TAB or SYMB editor the softkey line can be switched with the [F9] function key:



[F9] + [F1 MARK] - Marking an Area
[F9] + [F2 UNMARK] - Unmarking an Area

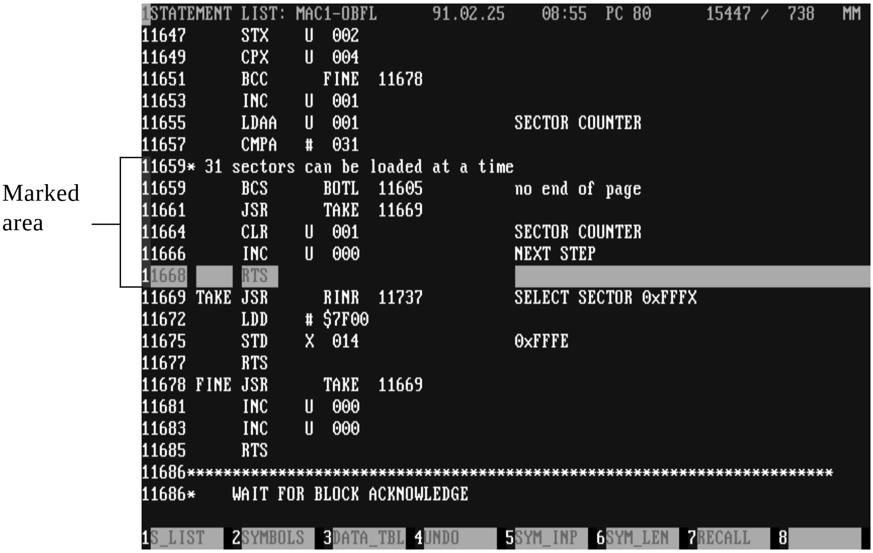
These function keys are used to mark a specified area within an editor. The start or end of an area can defined first. The cursor is placed at the start or the end of the area to be marked and with the [F9] + [F1 MARK] function keys it is marked. E.g.:



```
1STATEMENT LIST: MAC1-OBFL      91.02.25    08:55  PC 80      15447 /  738   M
11633      CPY    R 0870 FBK DATA 00     LOAD POINTER P-ADDRESSES
11637      BCS      LOPS  11629
11639      STX    R 0866 TEMP R&         STORE POINTER P-ADDR
11642      LDX    U  002
11644      LEAX   X  016
11647      STX    U  002
11649      CPX    U  004
11651      BCC      FINE  11678
11653      INC    U  001
11655      LDAA   U  001                 SECTOR COUNTER
11657      CMPA   #  031
11659* 31 sectors can be loaded at a time
11659      BCS      BOTL  11605          no end of page
11661      JSR      TAKE  11669
11664      CLR    U  001                 SECTOR COUNTER
11666      INC    U  000                 NEXT STEP
11668      RTS
11669 TAKE JSR      RINR  11737          SELECT SECTOR 0xFFFX
11672      LDD    # $7F00
11675      STD    X  014                 0xFFFE
11677      RTS
11678 FINE JSR      TAKE  11669
```

Line is marked

The cursor is then placed at the other end of the area (start or end depending on which was previously marked) and with [F9] + [F1 MARK] the entire area is marked. The area is marked on the left side with a bar. E.g.:



```
1STATEMENT LIST: MAC1-OBFL      91.02.25    08:55  PC 80      15447 /  738   MM
11647      STX    U  002
11649      CPX    U  004
11651      BCC      FINE  11678
11653      INC    U  001
11655      LDAA   U  001                    SECTOR COUNTER
11657      CMPA   #  031
11659* 31 sectors can be loaded at a time
11659      BCS      BOTL  11605             no end of page
11661      JSR      TAKE  11669
11664      CLR    U  001                    SECTOR COUNTER
11666      INC    U  000                    NEXT STEP
11668      RTS
11669 TAKE JSR      RINR  11737             SELECT SECTOR 0xFFFX
11672      LDD    # $7F00
11675      STD    X  014                    0xFFFE
11677      RTS
11678 FINE JSR      TAKE  11669
11681      INC    U  000
11683      INC    U  000
11685      RTS
11686***********************************************************************
11686*    WAIT FOR BLOCK ACKNOWLEDGE

1S_LIST  2SYMBOLS  3DATA_TBL 4UNDO     5SYM_INP  6SYM_LEN  7RECALL   8
```

If the mark function is called while a marked area exists the following error message is displayed:

E257 RANGE ALREADY MARKED !

In this case the [F9] + [F2 UNMARK] function keys must be used to unmark the area before a new area can be marked.

[F9] + [F3 MOVE-M] - Moving a Marked Area

A marked area is moved from the previous position to the current cursor position. If the cursor is inside the marked area or in a protected line in an STL the following message is displayed:

E256 INVALID TARGET POSITION FOR COPY, MOVE !

[F9] + [F4 COPY-M] - Copying a Marked Area

A marked area is copied to the current cursor position with this function. If the cursor is in a marked area of a protected line within an STL the following message is displayed:

E256 INVALID TARGET POSITION FOR COPY, MOVE !

[F9] + [F5 CUTPASTE] - Copy to Temporary Memory

A marked area is temporarily stored in the directory which the PROgramming SYStem was installed in with the file name BURPASTE.CFG. The file remains in memory after the STL program is deleted with NEW, after loading another program and after exiting the PROgramming SYStem. I.e. the area stored in temporary memory can be inserted into another window or into a newly loaded program.

[F9] + [F6 DEL-M] - Deleting a Marked Area

A marked area in an editor is deleted with this function.

[F9] + [F7 INSPASTE] - Inserting from Temporary Memory

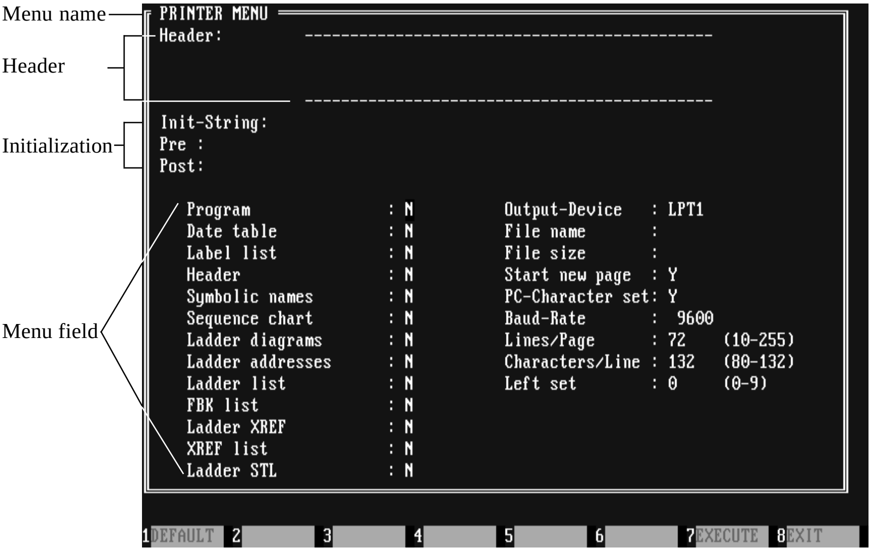
An area which was stored in temporary memory can be retrieved and inserted in the current cursor position with this function. If the cursor is within a marked range or a protected line of an STL the following message is displayed:

```
E256  INVALID TARGET POSITION FOR COPY, MOVE !
```

3. PRINT MENU MODIFICATIONS

A	B	C	PG1000	CP80	PG-PC	PC80
●	●	●	●	●	●	●

After calling the print menu with PMEN, PRINT or LIST the following menu is displayed:



```
PRINTER MENU
Header:        -------------------------------------------

               -------------------------------------------
Init-String:
Pre :
Post:

   Program              : N        Output-Device    : LPT1
   Date table           : N        File name        :
   Label list           : N        File size        :
   Header               : N        Start new page   : Y
   Symbolic names       : N        PC-Character set: Y
   Sequence chart       : N        Baud-Rate        :  9600
   Ladder diagrams      : N        Lines/Page       : 72    (10-255)
   Ladder addresses     : N        Characters/Line  : 132   (80-132)
   Ladder list          : N        Left set         : 0     (0-9)
   FBK list             : N
   Ladder XREF          : N
   XREF list            : N
   Ladder STL           : N

1DEFAULT 2        3        4        5        6        7EXECUTE 8EXIT
```

Header: Four lines of optional text can be entered in this text field. The text entered here is printed on every page as a "header". The header is saved with the corresponding program which means that a different header can be defined for every program.

Initialization: A pre and post string can be defined respectively for printer initialization. The pre-string is sent before a print entry and the post-string is sent after. The following characters can be entered in both lines:

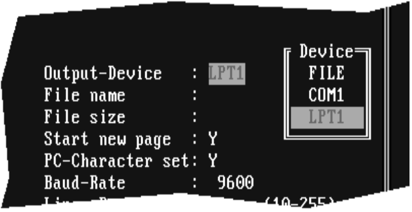
Entry	ASCII-Value		Entry	ASCII-Value	
Syntax	dec.	hex.	Syntax	dec	hex.
<NUL>	0	0	^Q	17	11
^A	1	1	^R	18	12
^B	2	2	^S	19	13
^C	3	2	^T	20	14
^D	4	2	^U	21	15
^E	5	2	^V	22	16
^F	6	2	^W	23	17
^G	7	2	^X	24	18
^H	8	2	^Y	25	19
^I	9	2	^Z	26	1A
^J	10	A	<ESC>	27	1B
^K	11	B	<FS>	28	1C
^L	12	C	<GS>	29	1D
^M	13	D	<RS>	30	1E
^N	14	E	<US>	31	1F
^O	15	F	<DEL>	127	7F
^P	16	10			

Menu field: The print menu is the same in all three expansion levels (A, B and C). Only those commands which are allowed for the corresponding expansion level can be selected however.

This section deals only with the modifications made to the menu field:

- Output device
- Baud - rate
- File name
- Left margin
- File size

Output device: By placing the cursor on this menu point and pressing the [↵] key a selection window is displayed from which the following outputs can be selected:



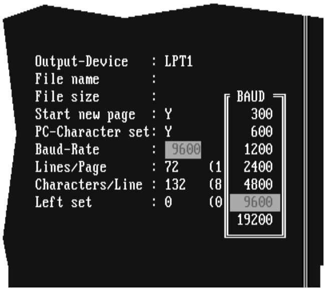
```
Output-Device   : LPT1
File name       :
File size       :
Start new page  : Y
PC-Character set: Y
Baud-Rate       :  9600
```

File name: A maximum 12 character (including extension) file name can be entered in this field. If FILE is chosen as the output the PRINT command sends the data to the file created. If PRINT is executed more than once the output is added to the end of the previous output with the same file name. If no drive name of path is given the file is placed in the default directory.

Environment variable B&R$PRINTFILE=<file name> has been removed!

File size: Displays the current file size in bytes.

Baud - rate: By placing the cursor on this menu point and pressing the [↵] key a selection window is displayed from which the desired baud rates can be set (for COM1 serial interface).



```
Output-Device   : LPT1
File name       :
File size       :
Start new page  : Y
PC-Character set: Y
Baud-Rate       : 9600
Lines/Page      : 72     (1
Characters/Line : 132    (8
Left set        : 0      (0
```

Left margin: The left margin is set with the menu point. A number between 0 and 9 can be entered. If for example, 5 is selected a left margin of five blank characters is set for the start of each line.

Form feed: Version 5.50 and previous versions of the PROgramming SYStem utilized the Line feed in place of form feed. Version 5.60 now can send the ASCII character <FF> as a replacement (Form Feed, hex.: $0C, dec.: 12).

Characters/line: 80 to 132.

Lines/page: 10 to 255.

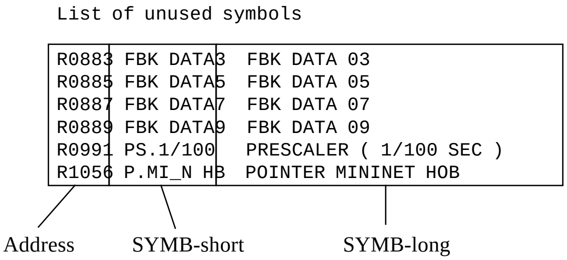
Cross reference list: All unused SYMBs which are defined in the SYMB editor are now printed as well. The list of unused symbols is printed within the frame of the normal cross reference list which means that the list can not be singled out in the print menu.

A	B	C	PG1000	CP80	PG-PC	PC80
●	●	●	●	●	●	●

The cross reference list now consists of three elements:

1) Cross reference list (see “Chapter 9 PRINTING”)
2) Label reference list (see “Chapter 9 PRINTING”)
3) List of unused symbols

When printed the third element looks like the following example:

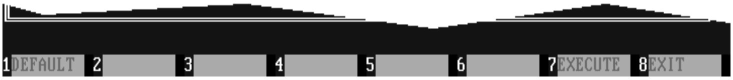


```
List of unused symbols

R0883 FBK DATA3  FBK DATA 03
R0885 FBK DATA5  FBK DATA 05
R0887 FBK DATA7  FBK DATA 07
R0889 FBK DATA9  FBK DATA 09
R0991 PS.1/100   PRESCALER ( 1/100 SEC )
R1056 P.MI_N HB  POINTER MININET HOB
```

Address
SYMB-short
SYMB-long

Print Menu Softkey Line

The softkey line is laid out as follows in the print menu:



[F1 DEFAULT] The stored default settings are loaded. Any changes which may have been made are lost.

[F7 EXECUTE] The print out is started according to the previously set parameters.

[F8 EXIT] or [Esc] Exits the print menu. After pressing one of these keys the following message is displayed:

```
W300  WRITE DATA ? ( Y / N )
```

N Configuration changes are ignored!
Y Configuration changes are made and stored in a file!

4. PLC MEMORY LOCATION UP / DOWNLOAD

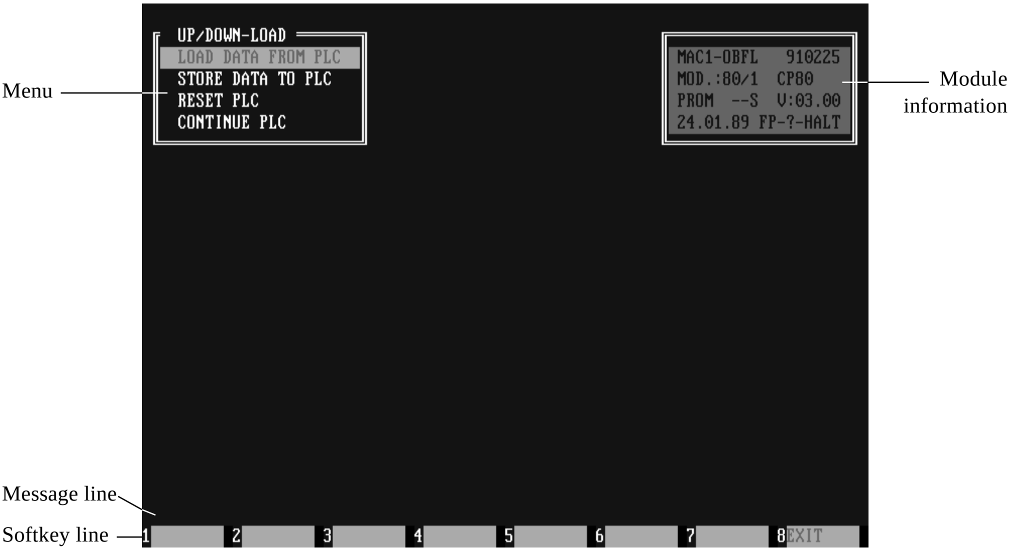
4.1. GENERAL

All flags and registers can be loaded in a file and saved to disk (upload) with this command. It is also possible to reload the saved data back to the PLC (download). A safety device has been created with these options to enable a security copy of PLC data on disk which can be loaded back to the PLC if lost for any reason.

4.1. MENU CALL

Command: ***UPDL [↵]***

Function: Displays the menu for use:



```
MAC1-OBFL   910225
MOD.:80/1  CP80
PROM  --S  V:03.00
24.01.89 FP-?-HALT
```

Menu points are selected by moving the highlight bar in the menu up and down with the [↑] and [↓] keys. By pressing [↵] the menu point selected is called.

[Esc] or [F8 EXIT] is pressed to exit the UP-/DOWN LOAD menu!

LOAD DATA FROM PLC

After this menu point is called all files in the default directory with the extension *.UPL are displayed. The function keys can be used to select another directory or to change the default directory.

By pressing the [↑] and [↓] keys an existing file can be selected which should be overwritten by the new PLC data. An optional file name or extension can be entered by means of the keyboard. By pressing the [↵] key now all flags and registers are loaded from the PLC and saved in the defined file.

If the data has been loaded from a PP40 MEM or PP60 MEM the following message is displayed in the message line:

```
W348  MEM FROM PLC <PLC RESET>  (Y/N) ?
```

Y If the PP is in RUN mode it is put into RESET state and the 64 Kbyte memory and registers and flags are all loaded and saved to the defined file.

N The PP is not put into RESET state. Only flags and registers are loaded and saved to the defined file.

STORE DATA TO PLC

After this menu point is called all files in the default directory with the extension *.UPL are displayed. The function keys can be used to select another directory or to change the default directory.

By pressing the [↑] and [↓] keys a file to be loaded to the PLC can be selected. A file name (without extension) can be entered by means of the keyboard. By pressing the [↵] key now all flags and registers are loaded from the defined file to the PLC.

If the data has been loaded to a PP40 MEM or PP60 MEM the following message is displayed in the message line:

```
W350  MEM TO PLC <PLC RESET>  (Y/N) ?
```

Y If the PP is in RUN mode it is put into RESET state and the 64 Kbyte memory and registers and flags are all loaded with the data saved in the defined file.

N The PP is not put into RESET state. Only the flags and registers are loaded with the data from the file.

The CPU type and program identification are saved with the data. If these values do not correspond to the type of CPU used or with the currently running program the following message appears in the message line:

`W351  CPU-TYPE/PROGRAM-ID   DIFFERENT - CONTINUE (Y/N) ?`

Y Data is transferred to the PLC

N The function is broken and the data is not transferred.

RESET PLC

The PLC is put into RESET state. Before the function is executed the following message appears in the message line:

`W349  PLC RESET (Y/N) ?`

Y The control is stopped and is set to a **defined** base state.

- All outputs are set to logic “0”
- R0000 to R0019 are cleared
- F 000 to F 499 are cleared
- All program registers are cleared

N The control operating state is not changed

CONTINUE PLC

If the control is in HALT or RESET state the program is started new or restarted with this option.

5. ADDITIONAL CHANGES

A	B	C	PG1000	CP80	PG-PC	PC80
●	●	●	●	●	●	●

5.1. START PARAMETER STDSYM (see Chapter 3 INTRODUCTION)

Because of the modifications made for EMS -SYMB management, file extensions must now be entered with the file name for the STDSYM start parameter.

Example: `STDSYM=fname.sym`
or `STDSYM=fname.esm`

5.2. TWO WINDOW OPERATION

The status line is displayed in both windows now.

APPENDIX H

PROGRAMMING SYSTEM UPDATES VERSION 5.70

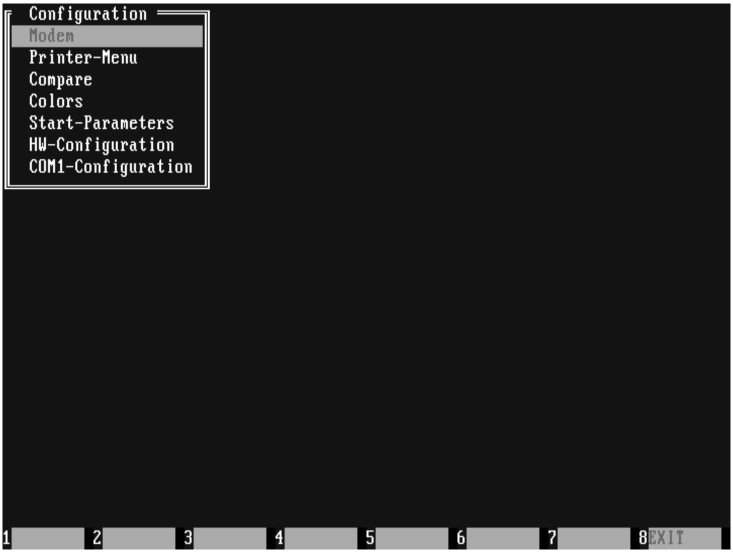
Contents:

1. CONFIGURATION MENU

Most of the parameters for version 5.7 of the PROgramming SYStem can be entered or changed through on-screen menus. The new version has eliminated many operational commands and all of the start parameters which in previous versions had to be entered before the actual start of the PROgramming SYStem.

Command: **`CONFIG [↵]`**

Function: Configuration menu call.



Entries are now selected from the configuration menu by means of the [↑] and [↓] cursor keys. The selected option can then be called with the [↵] key.

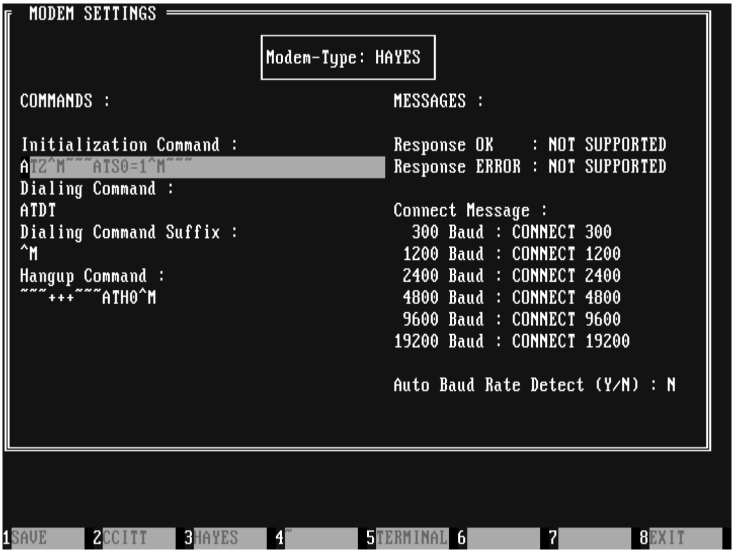
The following menus can be called from this configuration menu:

- Modem configuration
- Printer menu
- Compare configuration
- Color settings for the PROgramming SYStem
- Start parameters
- Hardware configuration
- COM1 configuration

The configuration menu is exited with **[F8 EXIT]** or **[Esc]**.

1.1. MODEM CONFIGURATION

After selecting “**`Modem`**” from the configuration menu, the parameters which are used to change the modem configuration are displayed on the screen.



```
MODEM SETTINGS

                         Modem-Type: HAYES

COMMANDS :                                MESSAGES :

Initialization Command :                  Response OK    : NOT SUPPORTED
ATZ^M~~~ATS0=1^M~~~                       Response ERROR : NOT SUPPORTED
Dialing Command :
ATDT                                      Connect Message :
Dialing Command Suffix :                    300 Baud : CONNECT 300
^M                                         1200 Baud : CONNECT 1200
Hangup Command :                           2400 Baud : CONNECT 2400
~~~+++~~~ATH0^M                            4800 Baud : CONNECT 4800
                                           9600 Baud : CONNECT 9600
                                          19200 Baud : CONNECT 19200

                                          Auto Baud Rate Detect (Y/N) : N
```

Modem settings are entered in this menu the same as with previous versions of the PROgramming SYStem.

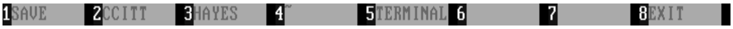
The only differences are as follows:

- The COM1 interface is now configured from a different menu (COM1 configuration)!
- If “N” (No) is entered for “**`Auto Baud Rate Detect (Y/N)`**” it is interpreted differently than before:

  Before version 5.70: The PROgramming SYStem waits for a connect message from the modem and compares this message with the message given under “**connection recognized**”. The connection is only made if the message from the modem and one from the PROgramming SYStem coincide.

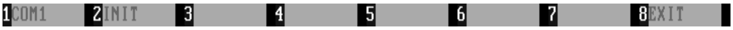
  From version 5.70: The PROgramming SYStem waits for a message from the modem. In this case only the character chain “**`CONNECT`**” is waited for. The baud rate which is included in the connect message has no influence. Therefore the data exchange can occur over a telephone line at any non-standard baud rate (e.g. 14 4000 baud). The baud rate between the PC and the modem corresponds to the value entered in the COM1 configuration.
  This mode of operation is only possible with modems which can communicate with the opposite station at a different baud rate than with the PC whereas the PC ↔ modem baud rate remains unchanged.

- The function key assignment has be slightly modified:



Function keys	Assignment
[F1 SAVE]	Saves the given modem configuration
[F2 CCITT]	Identical with [F1 CCITT] of the previous versions of the PROgramming SYStem
[F3 HAYES]	Identical with [F2 HAYES] of the previous versions of the PROgramming SYStem
	HAYES is the standard modem configuration setting after the PROgramming SYStem is installed!
[F4 ~]	Identical with previous versions of the PROgramming SYStem
[F5 TERMINAL]	Identical with previous versions of the PROgramming SYStem
[F8 EXIT] or [Esc]	Identical with previous versions of the PROgramming SYStem

- The terminal emulation function line has been changed as follows:



These function keys can only be called in combination with the [⇑] key:

Function keys	Assignment
[Shift] + [F1 COM1]	COM1 - configuration menu call
[Shift] + [F2 INIT]	Terminal emulation initialization (character definition, cursor sizes, display format to default value, clear screen, cursor home)
[Shift] + [F8 EXIT]	Exit terminal emulation

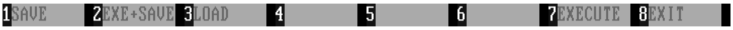
- The **MODEM** operation command has been replaced by the call in the configuration menu!

1.2. PRINTER MENU

After selecting the "**`Printer menu`**" option from the configuration menu the printer menu is called. The printer menu for version 5.70 is different from the previous version in the following ways:

- The baud rate configuration for the COM1 interface in no longer performed from this menu (now done under the COM1 configuration menu)!

- There has been some minor function key changes:



The function of the function keys is sometimes different depending on the call (from the configuration menu or operational commands PRINT or LIST):

Function keys	Call from configuration menu	Call with "PRINT"	Call with "LIST"
[F1 SAVE]	Saves the set printer configuration		
[F2 EXE+SAVE]	Saves the set printer configuration and exits the printer menu	Saves the set printer configuration and starts the print job	Saves the set printer configuration and starts the screen output
[F3 LOAD]	Loads the saved printer configuration		
[F7 EXECUTE]	Exits the printer menu and takes over the given changes as the current settings.		
		Then the printing job is started	Then the screen output is started
[F8 EXIT] or [Esc]	Exits the printer menu. If changes have been made a request as to whether the data should be saved or not is made. Regardless of the user's response, the changes made are used as the current settings.		

- The **DMEN** operational command has been replaced by the call in the configuration menu!

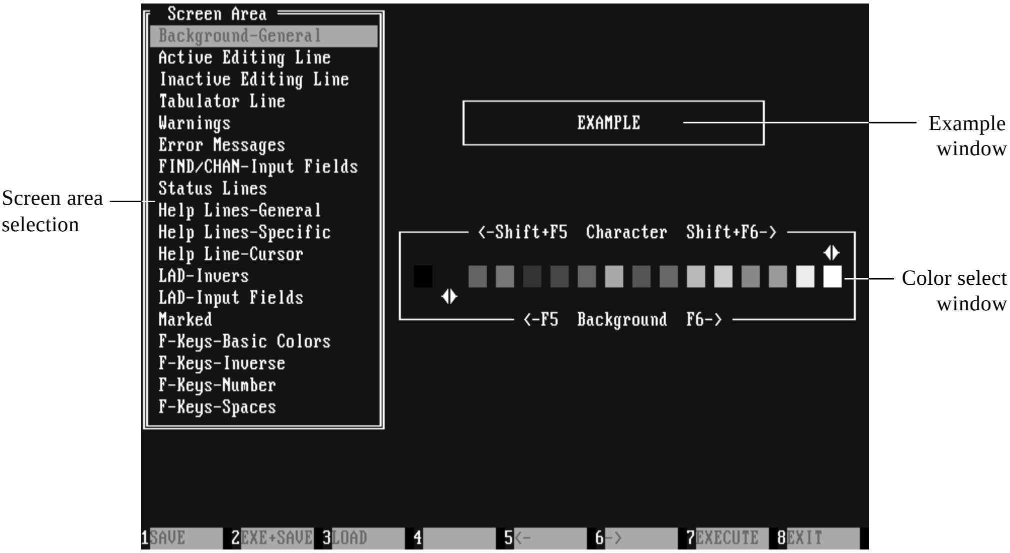
1.3. COMPARE CONFIGURATION

After selecting the "**`Compare`**" option from the configuration menu, the compare configuration is called. This menu in version 5.70 is identical with the 5.60 version described in appendix G.

- The **COMPCFG** command has been replaced by the call from the configuration menu!

1.4. COLOR SETTINGS

After selecting the “`Colors`” option from the configuration menu, the menu in which the color settings for the PROgramming SYStem screen color settings can be changed is called:



The area of the PROgramming SYStem screen for which the colors should be changed is selected through the **Screen area** selection menu. You can see what the color will look like by watching the **Example window**.

❍ The functions keys are assigned as follows:

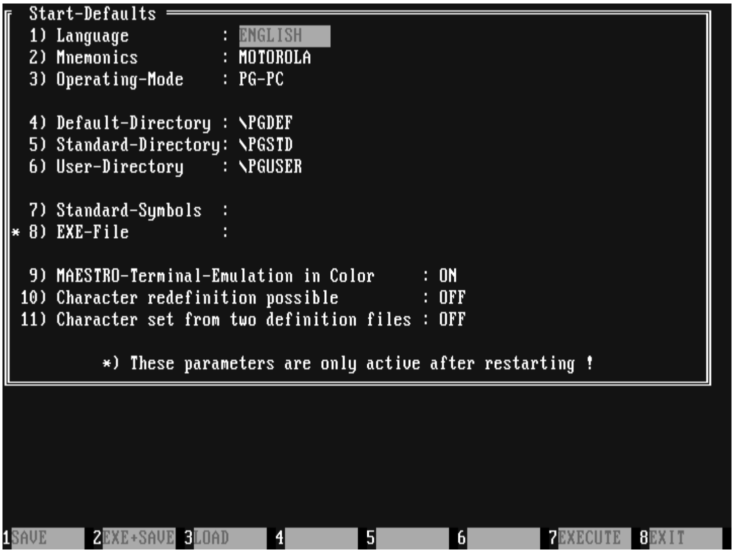
Function keys	Assignment
[F1 SAVE]	Saves the selected colors.
[F2 EXE+SAVE]	Saves the selected colors and exits the menu.
[F3 LOAD]	Loads the saved color settings.
[F5 <-]	Background color changes
[F6 ->]	
[Shift] + [F5 <-]	Character color changes
[Shift] + [F6 ->]	
[F7 EXECUTE]	Activates the set values for the enire PROgramming SYStem without saving any changes.
[F8 EXIT] or [Esc]	Exits the menu. If any changes are made, a request as to whether the data should be saved or not is made. Regardless of the user's response the changes which are made are used as the current settings.

1.5. START PARAMETERS

All parameters which, with earlier versions of the PROgramming SYStem, had to be given during the program start (start-up parameters and sometimes environment variables) can now be changed within the start parameters menu. The following parameters can be given:

- Language
- Mnemonics
- Operating mode
- Default-directory
- File name for standard symbols
- EXE file at program start
- Parameters for MAESTRO terminal emulation

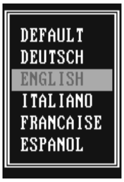
After selecting the “**`Start parameters`**” option from the configuration menu the following menu is called:



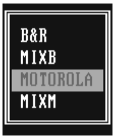
```
 Start-Defaults
  1) Language          : ENGLISH
  2) Mnemonics         : MOTOROLA
  3) Operating-Mode    : PG-PC

  4) Default-Directory : \PGDEF
  5) Standard-Directory: \PGSTD
  6) User-Directory    : \PGUSER

  7) Standard-Symbols  :
* 8) EXE-File          :

  9) MAESTRO-Terminal-Emulation in Color     : ON
 10) Character redefinition possible         : OFF
 11) Character set from two definition files : OFF

         *) These parameters are only active after restarting !
```

Entering parameters is done partially with pop-up menus and partially through keyboard entries:

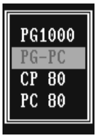
1) **Language** is used to determine the language of messages and operational commands. The desired option is selected by means of the [↑], [↓] and [↵] keys in a pop-up menu. DEFAULT corresponds with the language selected during the start of the PROgramming SYStem:



2) **Mnemonics**: For the entry or display of the STL commands there are four possibilities also chosen out of a pop-up menu:



3) The **Operating mode** is also selected by means of the pop-up menu technique:



4 to 6)
The path for the default, standard and user directories must be entered using the keyboard. The path name can also be given with the respective drive name. If for example, M:\B&R\PGDEF is entered as the default directory definition, the path in this case is only valid with drive M:. If another drive is switched to (e.g. C:) in the command line the default directory is the root of C: (C:\) unless of course another default has been defined for this specific drive.

7) Standard symbols file name

8) Name of a LEARN file, which should be executed at the start of the PROgramming SYStem.

9 to 10)
Parameters for MAESTRO terminal emulation.

These start parameters have replaced operation commands, start parameters (at the start of the PROgramming SYStem) and environment variables.

These operation commands have been replaced and no longer are used from the command line:

```
B&R       MOTOROLA    MIXB        MIXM        LANG xxxx
PC        DDIR        SDIR        UDIR
```

These (all) start-up parameters are replaced and are no longer used from the command line:

```
LANG=     MNEM=       PGMODE=     DDIR=       SDIR=
UDIR=     STDSYM=     EXE=        PAR=
```

The following environment variables are no longer used from the command line:

```
B&R$SCREEN=COL        B&R$CDEF=ON             B&R$FDEF=ON
```

Detailed information on ...

… Start-up parameters can be found in the “Introduction” chapter under *Startup Parameters of the PROgramming SYStem*.

… The parameters for MAESTRO terminal emulation can be found in “Appendix C” under *MS-DOS Environment Variables*.

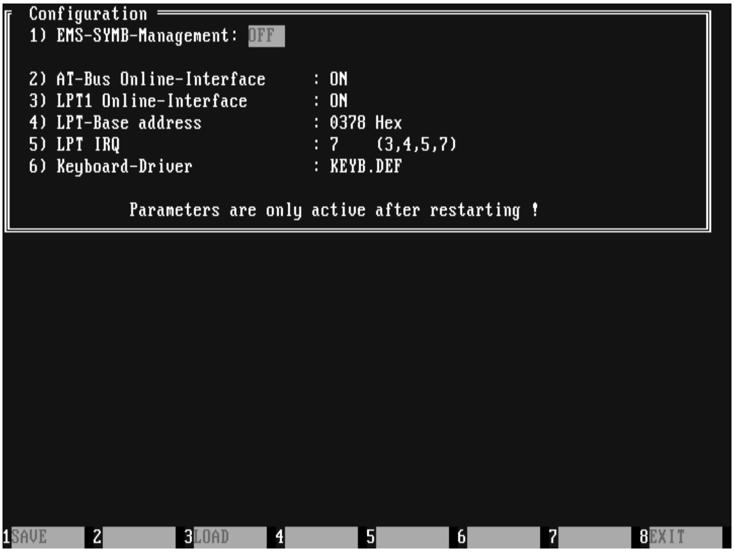
The function keys are assigned as follows:

```
1SAVE     2EXE+SAVE 3LOAD     4         5         6         7EXECUTE  8EXIT
```

Function keys	Assignment
[F1 SAVE]	Saves the entered start parameters
[F2 EXE+SAVE]	Saves the entered start parameter, exits the menu and executes the parameter which has be defined.
[F3 LOAD]	Loads the defined start parameter
[F7 EXECUTE]	Activates the set values for the entire PROgramming SYStem without saving any changes.
[F8 EXIT] or [Esc]	Exits the menu. If any changes have been made a request is made to whether the data should be saved or not. Regardless of the user's response the changes which have been made are used until the next parameter change or until a restart.

1.6. HW-CONFIGURATION

After selecting the “**`HW-Configuration`**” option from the configuration menu the following menu is called.



```
Configuration
1) EMS-SYMB-Management: OFF

2) AT-Bus Online-Interface      : ON
3) LPT1 Online-Interface        : ON
4) LPT-Base address             : 0378 Hex
5) LPT IRQ                      : 7    (3,4,5,7)
6) Keyboard-Driver              : KEYB.DEF

         Parameters are only active after restarting !

1SAVE  2      3LOAD  4      5      6      7      8EXIT
```

Entering parameters is done through the use of pop-up menus and the keyboard:

1) The **EMS SYMB-assignment** can be switched on and off.
   The desired entry (ON or OFF) is selected in the pop-up menu by means of the [↑], [↓] and [↵] keys.

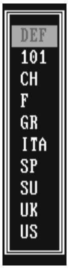
2) Switch the test for whether an **AT-Bus Online interface** is available or not, ON and OFF.

3) Switch the test for whether an **Centronics Online interface** is available or not, ON and OFF.

4) The LPTx-base address is typed in through the keyboard, if the centronics online interface should be connected to another parallel interface.

5) The interrupt number used is entered by means of the keyboard if the centronics online interface should be connected to another parallel interface.

6) Name of the keyboard driver used. This can be selected through the list of drivers shown in the pop-up menu:



```
DEF
101
CH
F
GR
ITA
SP
SU
UK
US
```

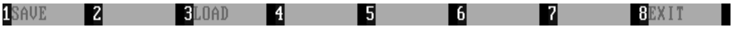
These parameters have replaced the following environment variables:

`B&R$LPT_OVER=<Base address>`
`B&R$LPT_IRQ=<IRQ>`
`B&R$KBD=<Keyboard driver>`
`B&R$EMS=ON`

Detailed information on the parameters for the MAESTRO terminal emulation can be found in “Appendix C” under *MS-DOS Environment Variables.*

> No parameters are activated until after a restart of the PROgramming SYStem!

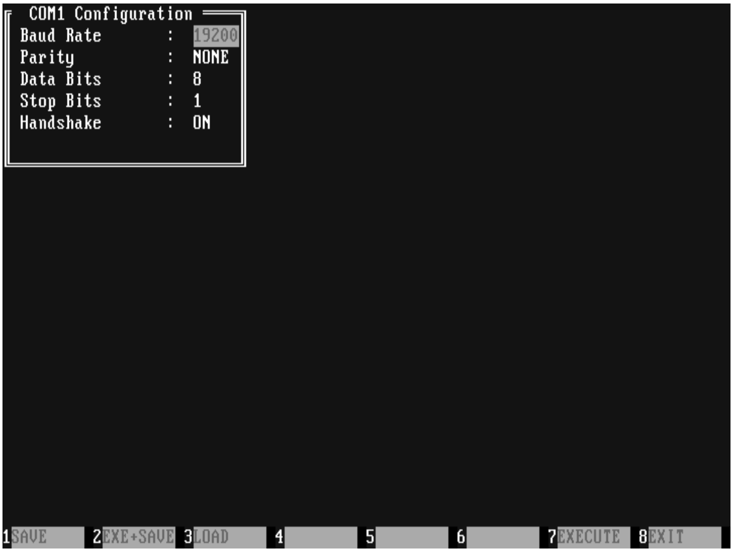
The function keys have been assigned as follows:



Function keys	Assignment
[F1 SAVE]	Saves the entered HW configuration
[F3 LOAD]	Loads the saved HW configuration
[F8 EXIT] or [Esc]	Exits the menu. If changes have been made, a request as to whether the data should be saved or not is made.

1.7. COM1 CONFIGURATION

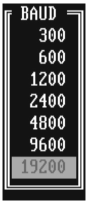
After selecting the "**`COM1-Configuration`**" option from the configuration menu the following menu is called:



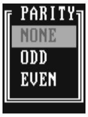
```
COM1 Configuration
Baud Rate      :  19200
Parity         :  NONE
Data Bits      :  8
Stop Bits      :  1
Handshake      :  ON
```

Parameter selection is made partially through pop-up menus and partially through keyboard entries:

- **Baud rate** is selected in the pop-up menu by means of the [↑], [↓] and [↵] keys.



- **Parity** is also selected by means of a pop-up menu.



- The number of data bits **Data bits** (5, 6, 7 or 8) is typed in through the keyboard.

- The number of **Stop Bits** is also entered through the keyboard.

- Switches the hardware **handshake** ON or OFF:

The **Handshake** parameter is ignored for the following PROgramming SYStem functions:

- For the **DIAL** and **STANDBY** operational commands, the hardware handshake is always switched off.
- For the **LOAD/SAVE PG1000** operational command, the COM1 interface operates with the following parameters independent of the configuration which has been defined:

Baud rate:	9600 Baud
Parity:	None
Data bits:	8 Bit
Stop Bits:	1 Bit
Handshake:	OFF

- The **ONL-COM1** operational command uses its own individual protocol.

Since the baud rate is defined in this menu, the **BAUD** command is no longer available from the command line.

1.8. MS-DOS ENVIRONMENT VARIABLES

Because of the introduction of the configuration menu, some environment variables are unnecessary. With PROgramming SYStem version 5.70 the following variables are still used as before:

B&R$VIRT=<Directory>	Directory for temporary files
B&R$NET=<Directory>	Directory of user specific (private) files

More information on these variables can be found in “Appendix C”.

1.9. OPERATIONAL COMMANDS

Because of the introduction of the configuration menu (called with the **CONFIG** command), many operational commands are unnecessary and have been replaced. These are shown below:

B&R **BAUD** **COMPCFG** **DDIR** **DMEN**
DRIVE **LANG xxxx** **MIXB** **MIXM** **MODEM**
MOTOROLA **PC** **SDIR** **UDIR**

2. FURTHER MODIFICATIONS

2.1. SWITCHING DRIVES

2.1.1. Command Line

Switching drives is no longer done with the **DRIVE** command, it is now done directly by entering the letter of the drive with a colon.

Example: If the desire is to switch to drive A: the following must be entered:

```
A: [↵]
```

If however, you wish to position the input cursor at a label in an STL program (e.g. **A:**) it must be entered between quotation marks.

Example: `"A:" [↵]`

2.1.2. Directory Overview

For displaying a list of the directories on a specific drive, the drive can be changed to at any time as is shown above in the first example.

Advantage:
When, for example, loading FBKs (macros, tables, ...) this has a great advantage whereas you no longer must switch to the command line to change drives.

2.2. DIRECTORY OVERVIEW

The display of file names in the directory overview (commands such as LOAD, DIR, UPDL, ...) is done alphabetically by name.

2.3. BACKUP FILES

If while saving a program (**SAVE PROG**), a file with the same name already exists, this file is renamed to <NAME.BAK>. If a file under the same name already exists as a BAK file it is deleted.

2.4. ESCAPING FROM “RUN”

By pressing the [Esc] key during the transfer to the PLC the program completes its transfer but is **not** started!

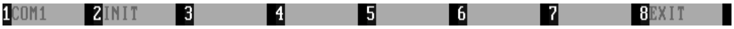
2.5. ESCAPING FROM “LOAD MACRO”

If after loading a macro, the PROgramming SYStem is found in the label exchange editor (because of allocating a label twice) the [Ctrl] + [Esc] key combination can be used to escape the procedure without accepting any changes.

2.6. TERMINAL EMULATION

With terminal emulations (**TERM** or **MAESTRO**) the baud rate can no longer be set by means of the function keys. This must be done in the **COM1** configuration menu.

The function keys now have the following allocation.



Function keys	Assignment
[Shift] + [F1 COM1]	COM1 - configuration menu call
[Shift] + [F2 INIT]	Terminal emulation initialization (character definition, cursor sizes, display format to default value, clear screen, cursor home)
[Shift] + [F8 EXIT]	Exit terminal emulation

3. OVERVIEW OF ALL OPERATIONAL COMMANDS OF THE PROGRAMMING SYSTEM VERSION 5.7

To make finding commands and knowing what to do with them a little easier, a table is provided here with all of the required PROgramming SYStem commands and the following descriptions:

<NAME> **<TNAME>** **<LNAME>**	4 character name, the first character must be a letter
<FNAME>	Max. 8 character file name without extension
<FNAME.EXT>	Max. 8 character file name with a 3 character extension
<Range>	For block operations (E.g.: COPY, MOVE, ...) a range must first be defined: **FROM <LN/LB/ADR> UP_TO <LN/LB/ADR>** The range begins with FROM <...> and ends at UP_TO <...>.
<DEST>	This is the target area for the range to be copied/moved to. **[↵]** The target will be marked by the current cursor position. **TO <LN/LB/ADR> [↵]** The target will be marked by the given data (TO <...>).
<File Type>	Determines the extension of a file name (<FNAME>)

Extension	<File Type>	Expansion level A	B	C
*.PRG	PROG	●	●	●
*.SYM	SYMB	●	●	●
*.TAB	TAB	●	●	●
*.LAD	LAD	●	●	●
*.MSY	MACRO (SYMB-Macro)	●	●	●
*.MTA	MACRO (TAB-Macro)		●	●
*.MSL	MACRO (STL-Macro)			●
*.ASY	ASCII (ASCII-SYMB)			●
*.ATA	ASCII (ASCII-Table)			●

Command	Additional parameters	Function	Page
#AD#	**<HEX-Address> [↵]**	Dump memory contents of the PLC processor (CPU or PP) starting from the given address <HEX-Address>. **Example: #AD# 3000** (Hexadecimal 3000)	----
#LEARN#	**<FNAME.EXT> [↵]**	The LEARN mode is switched on. All keys pressed afterwards are recorded and written to the <FNAME.EXT> file.	C-3
#LOFF#	**[↵]**	The LEARN mode is switched off.	C-3
CHAN	**...**	Replace / Exchange	5-20, F-11
	EXCHANGE <Range> [↵]	All program lines within the given range are examined and compared with two defined search masks. If either one of the two search masks coincide with a line in the program the line that coincides is exchanged with the other.	5-20
	EXCHAN? <Range> [↵]	This function is the same as the one above "EXCHAN" with the exception that for each exchange an acknowledgement is requested in the message line.	5-21
	REPLACE <Range> [↵]	All program lines within the given range are examined and are compared with the first of two defined search masks. If this search mask coincides with one of the lines in the program that line is then replaced by the second mask.	5-21
	REPLACE? <Range> [↵]	This function is the same as the one above "REPLACE" with the exception that for each replace an acknowledgement is requested in the message line.	5-21
	Without a range entry	If no <Range> is given, then all of the program lines, starting from the beginning are searched with the exception of the LAD translations .	
CLEAR	**...**	Deletes tables, ladder diagrams, and sequence definitions from the program memory of the PC.	
	SEQCHART <NAME> [↵]	The sequence definition with the name <NAME> is deleted.	4-12
	BLOCK <Range> [↵]	Deletes STL, SECQ or TAB ranges.	F-11
	LAD <LNAME> [↵]	The ladder diagram with the name <LNAME> is deleted.	4-42
	DATA_TBL <TNAME> [↵]	The table with the name <TNAME> is deleted.	7-12
COMPARE	**[↵]**	Starts the COMPARE procedure.	G-9
COMPCONT	**[↵]**	Continues the COMPARE procedure.	G-11
CONFIG	**[↵]**	Calls the configuration menu.	H-3
COPY	**<Range> <Dest>**	Copies STL, SYMB or TAB ranges and inserts them to certain locations (targets).	F-11, 5-28, 6-8, 7-8
	DISK <FNAME1.EXT> TO <FNAME2.EXT> [↵]	The <FNAME1.EXT> is copied and is saved as the name <FNAME2.EXT>.	F-13
CP40	**[↵]**	Switches from CP80 to PG1000 mode or from PC80 to PG-PC mode	3-11

CP80	**[↵]**	Switches from PG1000 to CP80 mode or from PG-PC to PC80 mode.	3-11
DATA_TBL	**[↵]**	Calls the DATA_TBL editor and the last processed table.	7-3
	<TNAME> [↵]	Calls an existing DATA_TBL or one which is to be defined with the given name <TNAME>.	
DATE	**<Program date> [↵]**	The current program date is replaced by <Program date>.	3-25
DELETE	**<FNAME.EXT> [↵]**	Deletes the file <FNAME.EXT>.	F-13
DIR	**<PATH> [↵]**	Displays directory contents of <PATH> (Possibly with drive name entry). Without entering a <PATH> the contents of the default directory are displayed.	F-13
DOS	**[↵]**	Calls the MS-DOS command interpreter. With the EXIT command you return to the PROgramming SYStem.	3-27
	"Command" [↵]	Executes the MS-DOS command "Command" and automatically returns to the PROgramming SYStem.	
<Lname:>	**[↵]**	The current drive is changed to <Lname:>.	H-13
DUMPS	**<FNAME> [↵]**	A PLC program is saved under a MOTOROLA S-record file under the name FNAME.S.	C-13
EEPROM	**[↵]**	Calls the EEPROM menu.	5-50, F-15
END	**[↵]**	Ends the PROgramming SYStem.	3-20
EXE	**<FNAME.EXT> [↵]**	The LEARN file <FNAME.EXT> is executed.	C-4
FBK	**<NAME> [↵]**	Accesses the FBK definition editor in order to enable the definition of the FBK with the given name <NAME>.	10-8
FIND	**<Range> [↵]**	Searches for STL program lines which coincide with the given search mask. If no <Range> is given, the STL entire program is searched.	5-25, F-11
FINDL	**<Range> [↵]**	All program lines which coincide with the given search mask are listed on the screen.	5-27, F-11
FINDP	**<Range> [↵]**	All program lines which coincide with the given search mask are printed.	5-27, F-11
FORMAT	**<Drivename:> [↵]**	Formats the diskette in the given drive.	8-22
FREE	**<ADR/SYMB> [↵]**	Searches for addresses without symbols starting from address <ADR/SYMB>. If no start address is given the search starts at R000.	6-9

GLOBCHAN	**...**	Global exchange/replace through all addresses in the entire STL program including the LAD translations.	5-24
	EXCHANGE [↵]	All program lines are examined and compared with two given search masks. If one of the search masks coincide with a program line, the line is exchanged with the other search mask.	
	REPLACE [↵]	All program lines are examined and compared with the first given search mask. If this first mask coincides with a program line, the line is replaced with the second mask.	
LAD	**[↵]**	Calls the last processed LAD	4-13
	<LNAME> [↵]	Calls the LAD with the name <LNAME>.	
LDMO	**[↵]**	Calls the last LAD debugger and enters debug mode.	4-44
	<LNAME> [↵]	Calls the LAD debug (LAD with the name <LNAME>).	
LARGE	**[↵]**	Defines the program memory to 4766 program lines.	3-26
LIST	**[↵]**	Calls the printer menu where it can be defined what should be output on the screen.	G-16 (F-13), 9-3
LOAD	**...**	Loads programs or parts of programs	8-9
	ASCII SYMBOLS <FNAME> [↵]	Loads a SYMB which has been saved in ASCII format.	8-16, F-12
	ASCII DATA_TBL <FNAME> [↵]	Loads a TAB which has been saved in ASCII format.	8-16, F-12
	EMSSYMBOLS <FNAME> [↵]	The SYMB file <FNAME> is loaded to the EMS memory (must be switched on). All SYMBs, which exist before loading, are deleted.	8-15
	LAD <FNAME> TO <LNAME> [↵]	The ladder diagram <FNAME> is loaded and entered to the program that exists with the name <LNAME>.	8-14
	SYMB <FNAME> [↵]	The SYMB file <FNAME> is loaded. All SYMBs which exist before the load are deleted.	8-11
	MACRO <FNAME> <Dest> [↵]	The STL, SYMB or TAB macro <FNAME> is loaded and bound to the existing STL/SYMB/TAB in the given location <LN/LB/ADR>.	F-12, 8-12
	PG1000 [↵]	The PROgramming SYStem switches to PG1000 mode and waits for the PG1000 programming device to transmit a program through the serial interface to the PC.	3-22
	PLC [↵]	The program which is in the RAM of the PLC is loaded into the memory of the PC.	5-65
	PROG <FNAME> [↵]	The program <FNAME> is loaded. The program which exists in memory is cleared.	8-9
	DATA_TBL <TNAME> <FNAME> [↵]	The table <FNAME> is loaded. The loaded table is saved as name <TNAME>.	8-11
MAESTRO	**[↵]**	Emulation of a PROVIT 500/600 with external ASCII keyboard.	C-9
MOVE	**<Range> <Dest> [↵]**	Moves an STL, SYMB or TAB range.	F-11, 5-30, 6-8, 7-8

NAME	<Program name> [↵]	The current program name is replaced by <Program name>.	3-24
NEW	[↵]	Deletes the program in the current window from the memory of the PC.	3-25
ONL	<Interface> [↵]	Selection of the online interface which should be used for programming a PLC: B&R0 ... Selection of the installed B&R online interface. LPT1 ... Selection of the centronics online interface. COM1 ... Selection of the modem attached to the COM1 serial interface.	3-26
P-VERS	[↵]	Displays information about the program stored in the PLC.	5-54
PRINT	[↵]	Calls the printer menu which is used to define what will be printed. By means of a function key the print job can be started from this menu.	9-3, G-16, H-6
RENAME	<FNAME1.EXT> TO <FNAME2.EXT> [↵]	Renames the file FNAME1.EXT to FNAME2.EXT.	F-13
RNT	[↵]	The program in the current window is checked for missing labels and for exceeding the maximum branch distance.	5-49
RUN	[↵]	The program is transferred to the RAM of the PLC and started. The STL debugger is called.	4-11, 5-31
RUN P	[↵]	This instruction is the same as the RUN command with the exception of the program being protected while it is transferred to the PLC. I.e.: The program can no longer be loaded from the PLC with LOAD PLC.	5-49
S_LIST	[↵]	Used for calling the STL editor.	5-4
	<LN/LB> [↵]	The input line is set to the given STL line <LN/LB>.	5-4
SAVE	[↵]	The entire program is saved with the name it was defined with using the SAVE or LOAD command.	8-5
	<FNAME> ASCII <Range/TNAME> [↵]	Saves a SYMB range or a table in ASCII format.	F-12, 8-8
	<FNAME> LAD <LNAME> [↵]	The ladder diagram <LNAME> is saved.	8-6
	<FNAME> SYMB [↵]	All defined symbols are saved.	8-6
	<FNAME> MACRO <Range> [↵]	The given STL, SYMB or TAB range is saved as a macro.	F-12, 8-7
	PG1000 <Mnemonics> [↵]	Transfers a program to the PG1000. For <Mnemonics> there are two possibilities : • INTER (MOTOROLA-Mnemonics) • GERMAN (B&R Mnemonics)	3-23
	<FNAME> PROG [↵]	The entire program is saved.	8-6
	<FNAME> DATA_TBL <TNAME> [↵]	The table <TNAME> is saved.	8-6
SEQCHART	[↵]	Used for calling the SEQC editor and the last processed SEQC.	4-5
	<NAME> [↵]	Used to call an existing SEQC or to define a new SEQC with the name given <NAME>.	4-5

SMALL	[↵]	Defines the program memory at 2719 program lines.	3-26
STANDBY	[↵]	Calls the standby menu. The PROgramming SYStem waits for a call from a remote PLC.	11-35
STAT	[↵]	Calls the STL debugger.	5-32
SYMBOLS	[↵]	Calls the SYMB editor.	6-4
	<ADR> [↵]	The input line is set to the corresponding address.	
TERM	[↵]	The PC can be used an entry terminal for operating a PROVIT 900/1000.	C-6
UPDL	[↵]	Calls the Up-/Download menu.	G-20
WINDOW	[↵]	Calls the window definition menu.	3-18
XFER	[↵]	Transfers a program to the PLC without interrupting the program running in the PROM of the PLC.	5-67

4. PROGRAMMING SYSTEM - MAXIMUM VALUES

Many values must be considered while working with the PROgramming SYStem:

	Number	**Length**
STL-lines (including comment lines)	32767	
Labels	1500 [1]	
SYMB without EMS	2000	
SYMB with EMS	8000	
Tables	= Number of existing Labels [1]	16 KByte or 44 KByte
LAD	200	50 lines
FBK	200	50 lines

[1] Since tables are headed with a label, the number of tables is limited by the number of labels.

The following prerequisites must be cared for before installing the PROgramming SYStem:

- The PC must be equipped with MS-DOS version 3.31 or higher.
- There must be at least 3 Mbytes of memory free on the hard disk for the installation.
- Operation of the PROgramming SYStem requires a maximum of 5 Mbytes (min. 300 KByte) memory on the hard disk (for temporary files).
- For starting the PROgramming SYStem program a minimum of 570 496 bytes (557.2 Kbytes) main memory are required. In some cases, memory resident programs and device drivers must be removed.

5. BRADOL ONLINE/MODEM CONVERTER

5.1. GENERAL INFORMATION

The BRADOL provides another possibility for the online connection to a PLC. The ONLINE/ MODEM CONVERTER is identical in almost all aspects with the Modem Online-Interface (MONL) with the following exceptions:

- The ONLINE/MODEM CONVERTER has no user program memory.
- It sits in an external housing and is not put in as a replacement for user memory in the CPU or PP.
- It has its own power supply (Primary: 220/110 VAC or 8 - 30 VDC)
- Contrary to the MONL the ONLINE/MODEM CONVERTER does not have to be installed next to the PLC. This provides another possibility for PCs which cannot be equipped with a B&R online interface or have problems with the parallel interface (centronics online interfaces).
- With the ONLINE/MODEM CONVERTER any limitations in selecting user memory have been eliminated (there are only two versions of the Modem Online-Interface).

These are additional features of the Modem Online-Interface:

- External diagnosis via modem
- Networking capabilities (up to 16 PLC processors can be programmed centrally and online with the programming device through twisted pair cable)

The software operation of the *Online/Modem Converter* is identical with the *User Program Memory Module with Modem Interface*.

For operation, at least PROgramming SYStem version 5.4 is required!

5.2. MODEL NUMBER - DESCRIPTION

Online/Modem Converter; for connecting HAYES modems, serial RS232 and RS485 interfaces (electrically isolated), node number switch, Status-LEDs	**BRADOL-0**