

Operating Manual

BPRO3

Version 3.2

Doc.-no. 212.952/DGB 04.95

Ident-no.: 00441108330

Edition: d131 April 95

SIG Positec BERGERLAHR GmbH&Co.KG

Breslauer Str. 7
Postfach 1180

D-77901 Lahr

**Proposals
Improvements**

**BPRO3
Operating Manual**

Edition: d131 April 95
Doc. no. 212.952/DGB 04.95

Sender:

Name:

Company/department:

Address:

Telephone no.:

Please inform us, using this form, if you have discovered any errors when reading this document.

We should also appreciate any new ideas and proposals.

Proposal and/or improvements:

Table of contents

	Page
1 General description	1-1
1.1 Reference documentation	1-2
1.2 Contents of the software package	1-2
1.3 Purpose	1-2
1.4 Programming system features	1-3
1.4.1 User interface	1-5
1.4.2 The options of the pull-down menue "Project"	1-6
1.4.3 Editors	1-7
1.4.4 "Online" communication with controller	1-10
1.4.5 The "Special" menue options	1-10
1.4.6 Comments in programs	1-11
2 Installation	2-1
2.1 Scope of supply	2-1
2.2 Accessories	2-1
2.3 Requirements	2-2
2.4 Connection diagram	2-3
2.5 Setup	2-4
2.6 BPRO3 quick start-up	2-5

Table of contents

	Page
3 Operation	3-1
3.1 Starting the programming system	3-1
3.2 User interface	3-2
3.2.1 Selecting a menu option	3-3
3.2.2 Help provided by the programming system	3-5
3.2.3 General editing functions	3-8
3.2.4 Log files, repeating an editing session	3-14
3.3 The BPRO3 main menu	3-16
3.4 Project-related operations	3-17
3.4.1 Loading a project	3-18
3.4.2 Storing a project	3-18
3.4.3 Copying a project	3-19
3.4.4 The project directory	3-19
3.4.5 Task configuration	3-23
3.4.6 Controller configuration	3-28
3.4.7 Defining units of measurement	3-35
3.4.8 Project data output	3-36
3.4.9 Defining a new password	3-37
3.4.10 Access protection in BPRO3	3-38
3.5 The editors	3-41
3.5.1 Creating a block header	3-42
3.5.2 The instruction list editor (IL editor)	3-54
3.5.3 The data block type editor	3-66
3.5.4 The data block editor	3-73
3.5.5 The assignment list editor	3-76
3.6 Communication with the controller	3-82
3.6.1 The pull-down menu "Transfer"	3-83
3.6.2 Miscellaneous commands	3-84
3.6.3 Viewing the CR	3-85
3.6.4 Debugging options	3-86
3.6.5 The pull-down menu "Contr. status"	3-93
3.6.6 The pull-down menu "Special"	3-106

	Page
3.7 Special functions	3-107
3.7.1 Exiting the menu or leaving the program	3-108
3.7.2 Defining, deleting and starting macros	3-108
3.7.3 Function key assignment (F keys)	3-109
3.7.4 25-line or 43-line screen display	3-110
3.7.5 Setup	3-111
4 Block programming	4-1
4.1 Programming program blocks	4-1
4.1.1 Valid variable types and block calls	4-1
4.1.2 Creating a program block	4-2
4.2 Programming function blocks	4-4
4.2.1 Valid variable types and block calls	4-4
4.2.2 Creating a function block type	4-5
4.2.3 Declaring a function block	4-7
4.3 Programming global blocks	4-8
4.3.1 Valid variable types and block calls	4-8
4.3.2 Creating a global block	4-9
4.4 Programming functions	4-11
4.4.1 Valid variable types and block calls	4-11
4.4.2 Creating a function	4-12
4.5 Programming data blocks	4-14
4.5.1 Creating a data block	4-14
4.6 Creating an assignment list	4-18

Table of contents

5	Data backup	Page 5-1
6	Error messages	6-1
6.1	Errors stored in error memory	6-2
7	Appendix	7-1
7.1	Glossary	7-1
7.2	Abbreviations	7-5
8	Index	8-1

1 General description

Control programs
Programming device
Network

The BPRO3 programming system has been designed for creating, testing and documenting control programs for BERGER LAHR Series 300 controllers (e.g. WDP5-318).

An IBM PC/AT 386 or 100% compatible computer is used as the programming device.

With two serial interfaces, the BNET network (a special BERGER LAHR network type) can be used for programming and operating up to 62 BERGER LAHR Series 300 controllers from the PC. The following are some typical characteristics of BERGER LAHR Series 300 controllers:

- Programming according to the IEC 1131-3 standard
- Parallel PLC and movement function processing
- Individual movement programming due to various axis operating modes (point-to-point mode, speed mode, position following mode)
- Numerous predefined functions for movement programming



NOTE

Support for standardized serial interfaces (e.g. Profibus-DP) is in preparation.

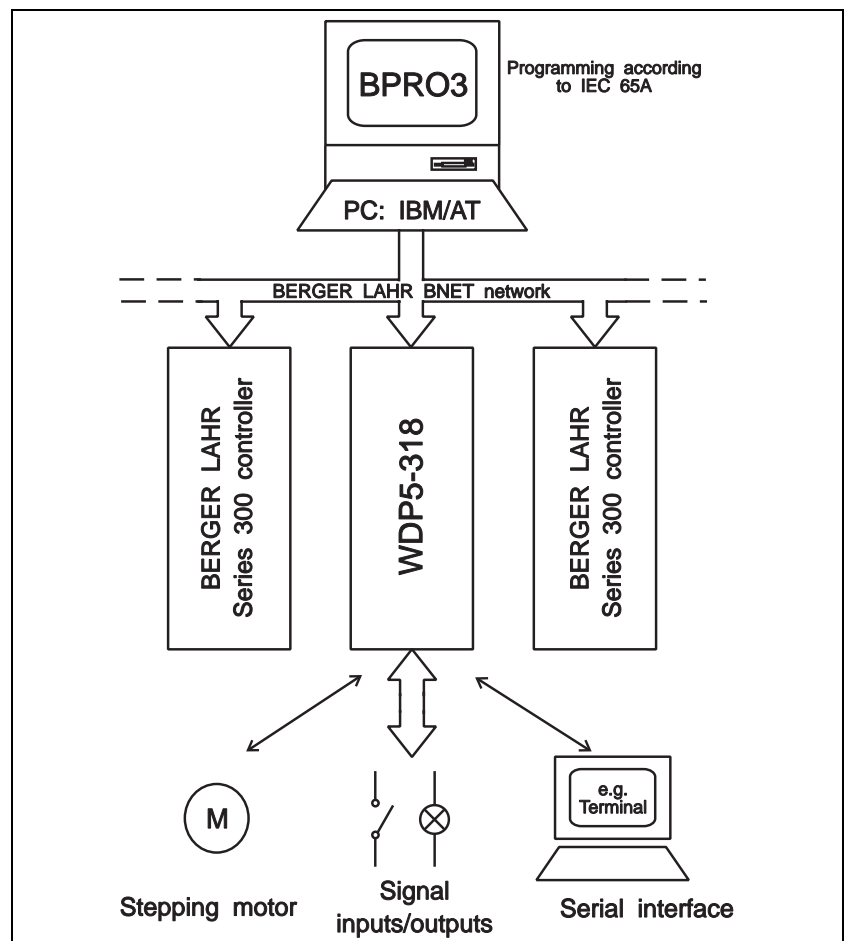


Fig. 1-1 BERGER LAHR-network with Series 300

General description

Project All data required for solving the control tasks are included in one project and can be handled in BPRO3 using the comfortable and clearly structured menu system. The data, which comprise program code, variables, configuration data, etc. are entered into the PC using editors which are specifically designed for their respective tasks.

1.1 Reference documentation

The BPRO3 Operating Manual contains information on installation and operation of the BPRO3 programming system and the various editors.

The BPRO3 Programming Manual contains all information required for developing a control program.

The BPRO3 Library documentation describes the user library. The user library contains blocks which facilitate programming.

The controller manual contains controller-specific information.

1.2 Contents of the software package

The software package consists of the following items:

- BPRO3 programming system
- A sample project (EXAMPLE1) and a user library (LIBRARY) to facilitate programming
- A "READ.ME" file which contains the latest information on the documentation and a description of the sample project.

1.3 Purpose

The BPRO3 programming system has been designed for creating, testing and documenting control programs for BERGER LAHR Series 300 controllers (e.g. WDP5-318).

1.4 Programming system features

The BPRO3 software package facilitates solutions for complex control tasks by means of a comfortable menu system which offers the following features:

- Project management
- Adjustment of the software configuration to the controller
- Controller programming
- Adding comments to project data
- Uploading/downloading programs to/from the controller
- Testing programs and debugging in online mode

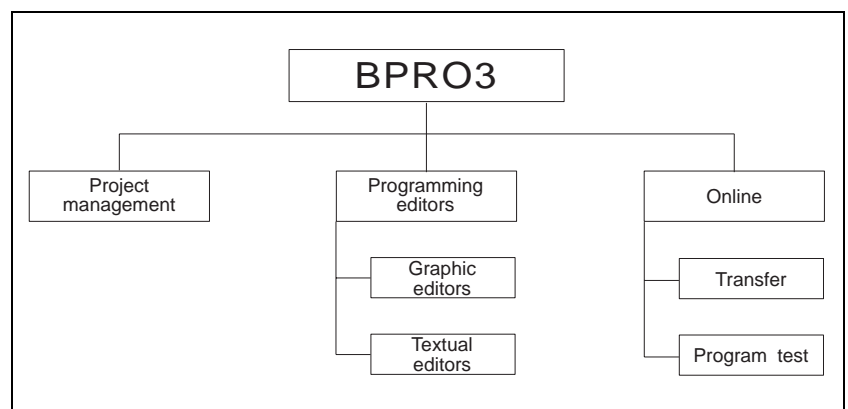


Fig. 1-2 Main programming system features

A project comprises all data required for defining the control tasks for the Series 300 controller. The data are stored in project-related databases.

Project data

A project can include the following data elements:

- Blocks (POU: program organization unit)
 - Program blocks (PRG)
 - Function blocks (FB)
 - Global blocks (GLB)
 - Functions (FUN)
 - Data blocks (DB)
- Task assignment
- Controller configuration
- Assignment list
- Blocks from the standard and controller libraries



NOTE

For a detailed description of these project data elements, see chapter 1 of the programming manual.

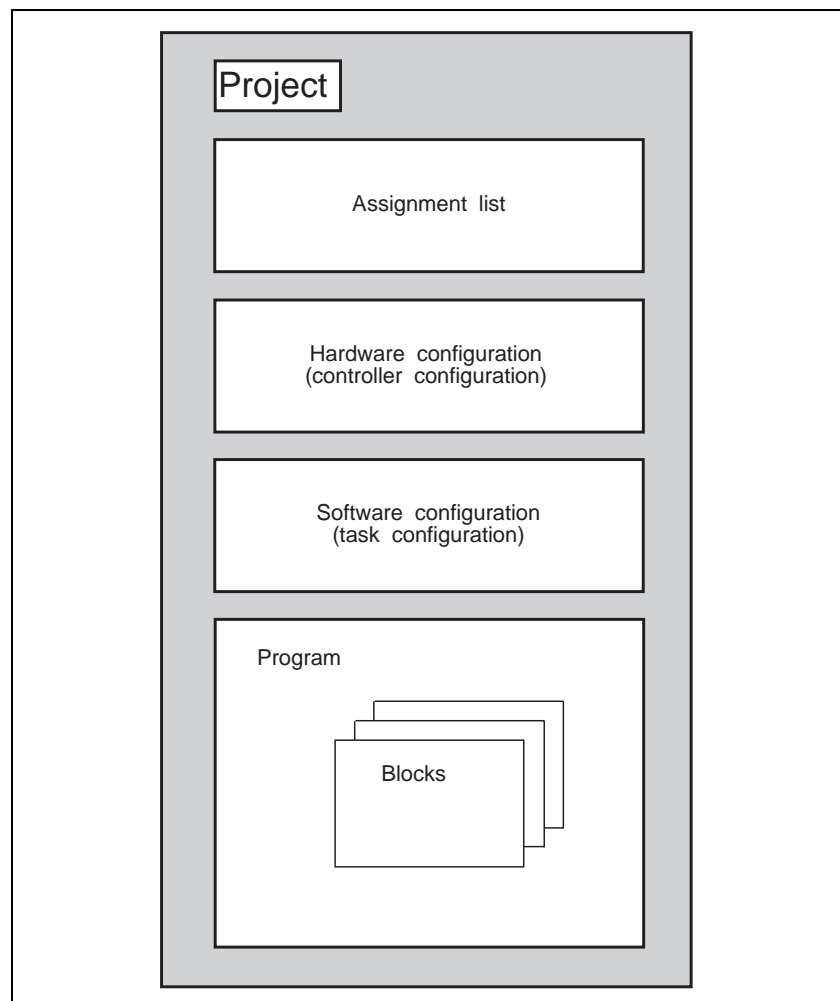


Fig. 1-3 Project data

The following sections describe the essential features of the programming system.

1.4.1 User interface

After program start the BPRO3 main menu is displayed. The various pull-down menus are available when pressing <F2>. The BPRO3 main menu is the point of departure for all other menu levels.

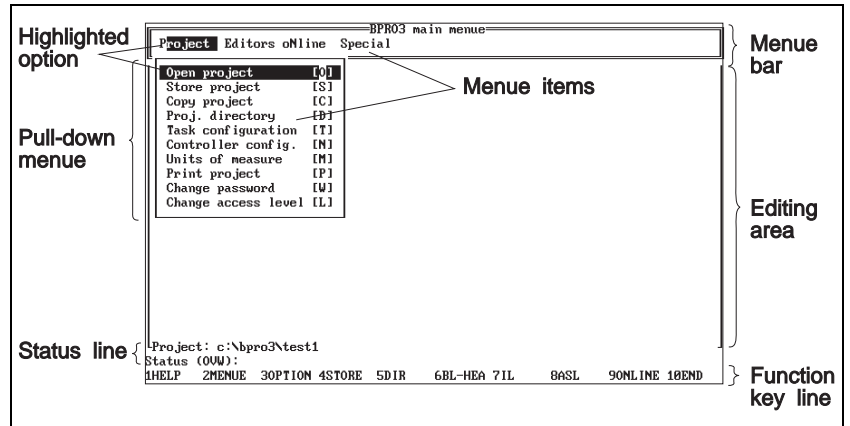


Fig. 1-4 User interface

The screen is consistently structured into four areas in the entire menu system:

- Menu area
- Editing area
- Status information on objects (status line)
- Function key line

The contents of these four areas is uniformly structured on all menu levels and in all editors.

Menu area

The menus are displayed in the top section of the screen. You can open or close all menus by pressing <F2>. A menu option of the menu bar may be an executable function or a headword for an associated pull-down menu.

Editing area

The central portion of the screen is an area enclosed by a frame which is used by the various editors according to their individual functionality.

Status information on objects

The line above the function key line is the status line. In this line, information on project data is displayed; this depends on the cursor position. The type of information displayed depends on the editor you are currently using. For example:

- Comment and address of a variable
- Task status in online mode
- Comment of a block.

Function key line

The bottom line of the screen shows the function key assignment. The function key combination assignments with the <Alt>, <Shift> and <Strg> keys are displayed when you press the <Alt>, <Shift> or <Strg> key, respectively, for approximately 2 seconds.

The function keys can be freely assigned menu options, with the exception of <F1>, <F2> and <F3>. These keys are reserved for the help function, for opening the pull-down menus and for the option list function.

1.4.2 The options of the pull-down menu "Project"

The pull-down menu "Project" includes options which affect the project as a whole.

It provides options for loading, storing and copying projects, on software and hardware configuration, for printing the project on a printer or into a file as well as access permission functions.

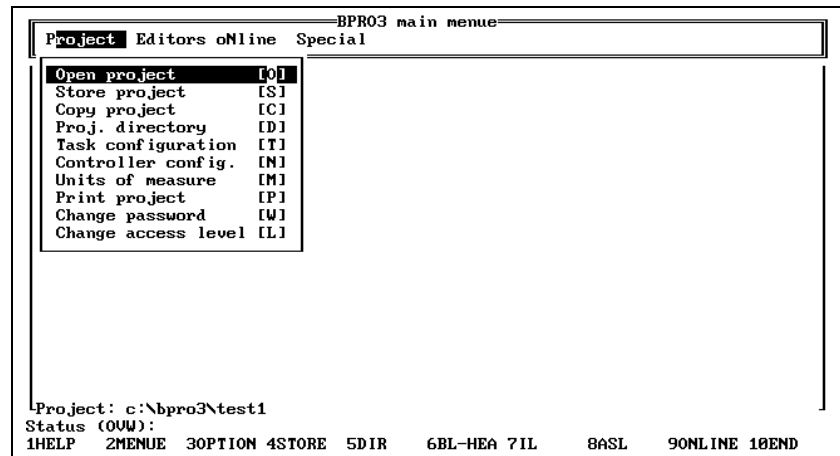


Fig. 1-5 The pull-down menu "Project"

The menu option "Proj. directory" can be used for displaying all the blocks which form part of a project. In addition, you can display two block libraries with predefined blocks which you can use for programming:

- Standard library* – The standard library contains blocks described in the IEC 1131-3 standard.
- Controller library* – The controller library contains controller-specific blocks for movement programming, interface programming, error handling, etc.



NOTE

For a description of the libraries, refer to the BPRO3 Programming Manual.

1.4.3 Editors

This pull-down menu provides access to the various editors required for creating a project. They are used for entering the program code and for organizing the program and data structure. Another editor function is the assignment of names to inputs, outputs and flags.

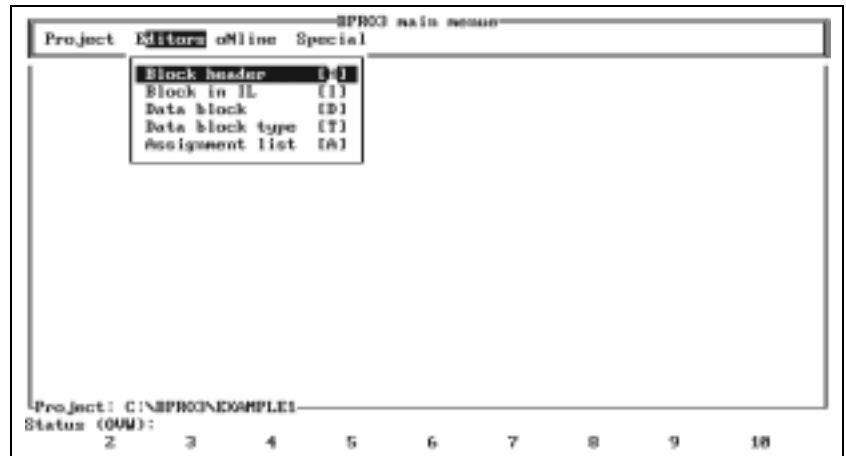


Fig. 1-6 The pull-down menu "Editors"

Block

A block (program block, function block, global block, function) is always composed of a block header and a block body. Figure 1-7 illustrates the various editors to be used for creating these individual elements.

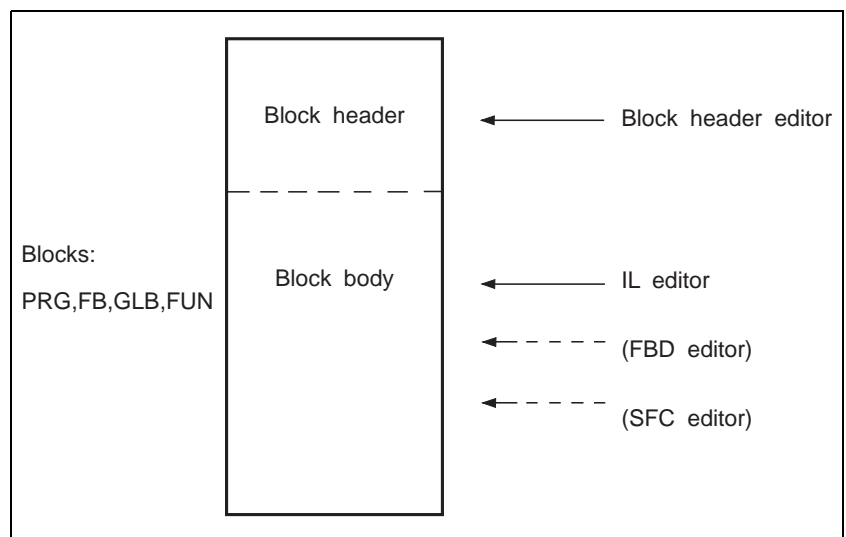


Fig. 1-7 Block structure

The block header contains block-specific information. The block body contains the actual program instructions. Globally addressable data structures can be created using the DB editor and the DB type editor. The assignment list editor is used for assigning symbolic names to inputs, outputs and flags.

Block header editor

The block header editor allows you to create, read and modify a block header. The block header describes the call interface of a block and contains additional information on the block and the variables used.

The call interface is described by:

- Block name
- Block type
- Input variables
- Output variables
- Input and output variables.

The following additional information is stored in the block header:

- Programmer's name
- A comment on the block
- Status information
- Access level
- The format to be used for loading the block into the controller
- Variable and function block declarations

Sequential function chart, function block diagram, and instruction list editors (SFC, FBD, IL)

When programming the controller, the control tasks are described in a suitable programming language.

Editors The following editors are available for entering program code into the programming system:

- Sequential function chart editor (SFC editor)
- Function block diagram editor (FBD editor)
- Instruction list editor (IL editor)

These editors are designed to describe the control tasks in graphic or textual representation.

Programming languages

The sequential function chart language SFC and the function block diagram language FBD are programming languages with graphic representation while instruction list IL is a language with textual representation (see figure 1-8). Programming is governed by the guidelines of the IEC 1131-3 standard.



NOTE

The SFC and FBD editor options will be implemented in a later release of the programming system.

The program code entered via an editor is compiled subsequently into a language which can be processed by the controller.

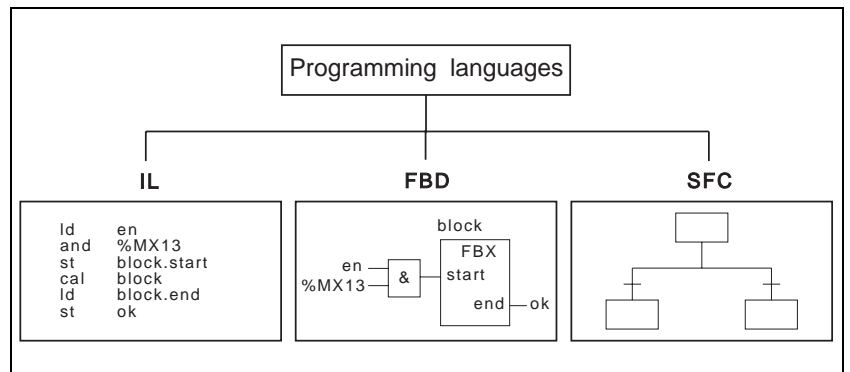


Fig. 1-8 Programming languages

Data block and data block type editors (DB and DB type editors)

The DB editor is designed for creating data blocks. Data blocks are used for storing variables which can be globally accessed.

The DB type editor is designed to define the structure of the data block.

Assignment list editor

The assignment list editor (AL editor) allows you to assign arbitrary names and various data types to inputs, outputs and flags. In addition, initial values and comments can be specified.

The data elements created in this way are global variables.

Example:

START_1 %IX3.9 START_1 is assigned to input bit 3.9
 START_2 MW7 START_2 is assigned to flag word 7

When this assignment has been made, the input can be addressed with either START_1 or %IX3.9. The effect is the same. The same principle applies to the flag word MW7 and START_2.

General description

1.4.4 “Online” communication with controller

The options of the “Online” menu are designed for controlling the communication link to the controller. The controller and the programming system exchange programs, data and commands through this link. A number of menu options are provided for communication with the controller. Some of these menu options are also available in the editors accessed via the pull-down menu “Debug”. The most important functions of “Online” are:

- Loading a program or program parts into or from the controller
- Testing a program
- Debugging a program

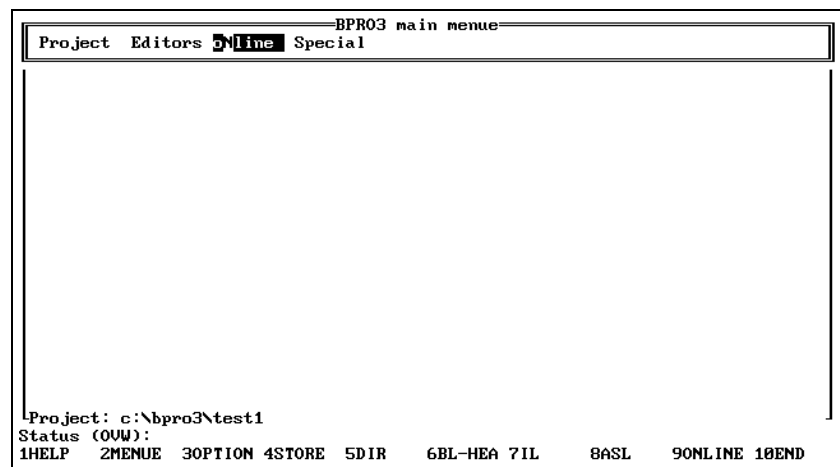


Fig. 1-9 The menu option “Online”

1.4.5 The “Special” menu options

The pull-down menu “Special” provides options for exiting the program, creating macros, assigning function keys, toggling between 25-line and 43-line display modes and selecting various setup parameters.

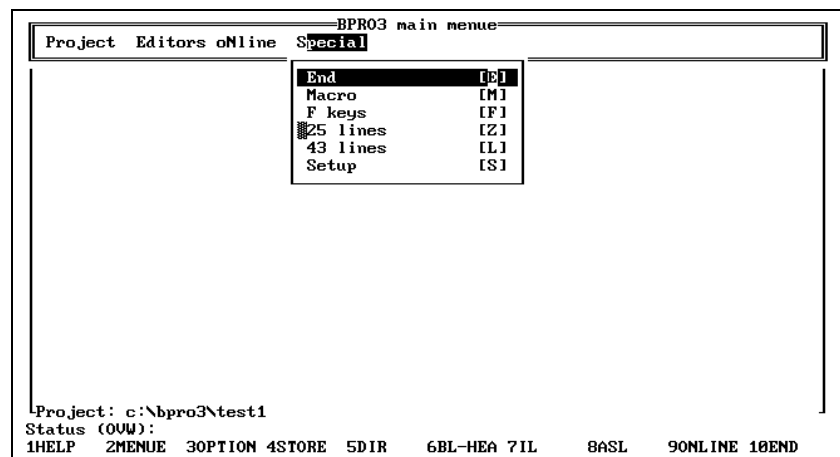


Fig. 1-10 The pul-down menu “Special”

- 1.4.6 Comments in programs** A large variety of individual project data (e.g. blocks, variables, assignments in assignment lists) can be provided with comments. For this purpose, the built-in text editor is used. Comments can be entered on two levels: a 40-character comment title which is entered directly from one of the editors described above, and a comment of any length which is displayed only when the comment editor is called.

General description

2 Installation

2.1 Scope of supply

The delivery must be checked for completeness.

The scope of supply comprises one folder with:

Qty.	Designation
4	Diskettes (two 3½" and two 5¼" diskettes)
1	BPRO3 Operating Manual
1	BPRO3 Programming Manual
1	BPRO3 Library

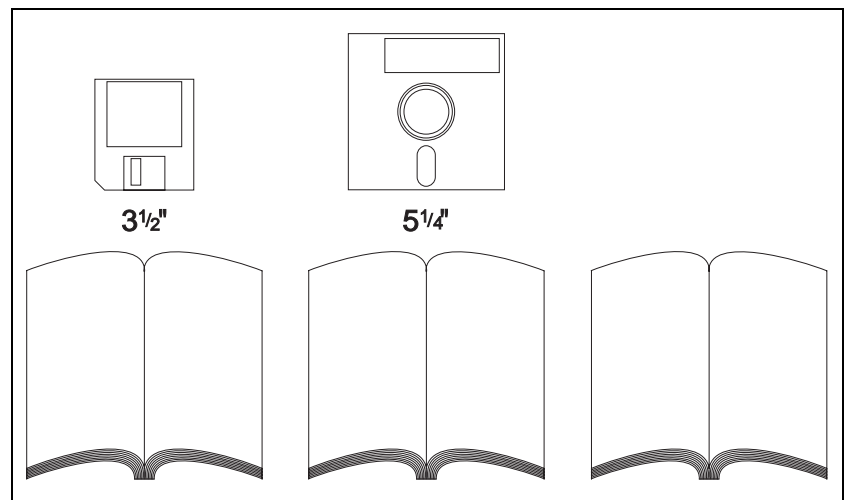


Fig. 2-1 Scope of supply

2.2 Accessories

The following accessories (for a description, see controller manual) are available on request:

Designation	Order number
Interface cable, male/female	See sales documentation
Interface cable, male/male	
Interface converter MP 923 (RS 485/RS 232)	
Interface distributor MP 924	

2.3 Requirements

The following requirements must be met in order to use BPRO3 on a PC:

Hardware requirements

- IBM PC/AT 386 or compatible
- 640 Kb RAM min.
- At least 1 floppy disk drive, 5¹/₄" 1.2 Mb or 3¹/₂" 1.44 Mb
- Hard disk with a minimum free storage capacity of 6 Mb
- Serial interface RS 232 (V 24) or RS 422 configured as COM1 (I/O address 3F8, IRQ 4) or COM2 (I/O address 2F8, IRQ 3)
- One of the following graphics adapters with a compatible display screen must be installed:
 - EGA graphics adapter
 - VGA graphics adapter
- Keyboard
- Interface cable for communication with the controllers through the network

Software requirements

- DOS operating system version 3.3 or higher
- The "FILES" command in the "CONFIG.SYS" file must be set to 25 or higher (see DOS manual).
- BPRO3 software package

Interface configuration

- The serial interface of the PC is automatically set up by the BPRO3 programming system.



NOTE

Further information, e.g. on the required memory capacity or processing speed optimization, is contained in the "readme.exe" file on program diskette 1.

2.4 Connection diagram

Communication between the PC and the controller is effected by serial data transmission. Figure 2-2 shows how to connect the controller to the PC (see chapters 2 and 6 of the controller manual).



ATTENTION

When wiring, take into account whether the controller is provided with an RS 485 interface (female connector) or an RS 232 interface (male connector).

If the controller is provided with an RS 485 interface and the PC with an RS 232 interface, an interface converter (e.g. MP 923) must be used.

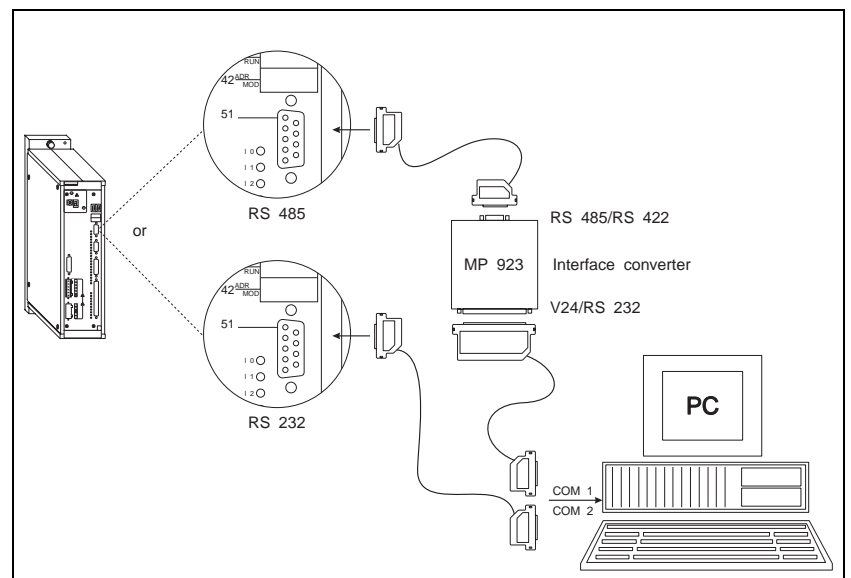


Fig. 2-2 Connection



NOTE

With an RS 485 interface, the MP 924 interface distributor can be used for implementing a network (see controller manual). A maximum of 62 controllers can be programmed via two serial interfaces on the PC.



NOTE

With an RS 232 interface, networking is not possible.



NOTE

Interface wiring is described in chapters 2 and 6 of the controller manual.

2.5 Setup

To install BPRO3 on the PC proceed as follows:

1. Create backup copies of all diskettes included in the package (for instructions on copying diskettes, refer to your DOS manual).



NOTE

Use the backup copies for installing BPRO3 as described below.

2. Insert program diskette no. 1 into the diskette drive.
3. Change to the drive which contains the program diskette by entering, for example:

A: <↵>



NOTE

Before starting the installation program, you should first read the information contained in the "readme.bat" file on program diskette 1 (type "readme"). The "readme.bat" file contains important information on:

- *Installation*
- *Operation*
- *Common problems*
- *Sample projects*
- *Notes concerning previous program versions*

4. Start the installation program by entering:
INSTALL



NOTE

If you do not specify drive and directory, the installation program uses drive C: and the \BPRO3 directory by default.

5. Follow the instructions of the installation program displayed on the screen.

Installing a BPRO3 update version

The procedure for installing a BPRO3 update is the same as described above.



NOTE

At the end of the update installation process, you will have to insert diskette 1 of your original registered BPRO3 version.

2.6 BPRO3 quick start-up

The following steps allow you to quickly use the BPRO3 programming system.

Wiring controller and PC

1. Wire the controller; see controller manual (e.g. WDP5-318), chapter on "Installation".



ATTENTION

Do not yet couple any mechanical load to the motor!

2. Connect the controller to the PC; see BPRO3 Operating Manual, chapter on "Cabling".
3. Switch on the controller and the PC. A "01" appears in the seven-segment display of the controller (application mode).

Installing BPRO3 on a PC

4. Install BPRO3 on the PC; see BPRO3 Operating Manual, chapter on "Setup".
5. Start BPRO3 as follows: Change to the BPRO3 directory and enter "BPRO3".
6. Compare the BPRO3 software version number (see initial screen) with the programming manual version (see title page).



NOTE

The programming manual version number must be equal to or greater than the software version number to ensure that all programming features are described in the manual.

Loading a project into the controller

7. Load the EXAMPLE1 project into the controller:
 - Press <F2> to display the menu; select "Online".
 - Login is effected automatically. The electrical link to the controller is established. (Defaults: controller address = 1, serial interface = COM1). Acknowledge any system messages by pressing the return key.
 - Press <F2> to display the menu; select "Transfer/Download project". The EXAMPLE1 project is transferred to the controller.



NOTE

The sample project EXAMPLE1 moves the motor shaft by one revolution to the right, activates output 2 of the controller, moves the motor shaft by one revolution to the left and deactivates output 2. The program then waits for one second and restarts the cycle.

The LIBRARY project contains a user library with many useful blocks for programming (see thumb index section 3).

- Starting the controller*
8. Press and hold the RUN/STOP key on the controller for at least 2 seconds in RUN position. The program sequence starts. The right-hand point on the seven-segment display lights up (RUN status).



ATTENTION

The motor shaft rotates, alternating in a clockwise and counter clockwise direction.

- Stopping the controller*
9. Press the RUN/STOP key on the controller to select STOP. The program sequence stops. A "01" appears in the seven-segment display of the controller (application mode).

- Modifying the project*
10. To modify the EXAMPLE1 project:
 - Press <F2> to display the menu; select "Special/End" (alternatively, you can press <F10>).
 - Press <F2> to display the menu; select "Editors/Block in IL".
 - Press <F3> to select the block name "SEQUENCE_PRG" and confirm the selection.
 - In the SEQUENCE_PRG block, replace the command line "ld 1000" by "ld 2000" and the command line "ld -1000" by "ld -2000" (this will make the motor shaft move by two revolutions in clockwise and counterclockwise direction). Any modification is automatically stored in temporary storage.
 - Press <F2> to display the menu; select "Special/End" (alternatively, you can press <F10>).

- Updating a project on the controller*
11. To load the modified project into the controller:
 - Press <F2> to display the menu; select "Online".
 - Press <F2> to display the menu; select "Transfer/Update project". The modification in the project is transferred to the controller.



NOTE

A block loaded into the controller can be viewed with the programming system. This means that the contents of the current result (CR) is displayed along with the program instructions.

- Viewing program execution*
- By default, the view feature of the programming system is enabled after "Login". Viewing can be enabled or disabled in the IL editor with the option "Contr. status/View on/off".*

To view a block, you have to specify the name of the corresponding task (e.g. for viewing the block "SEQUENCE_PRG", specify the task name "SEQUENCE").

While viewing, the controller is in DEBUG status (right-hand point of seven-segment display flashes). The DEBUG status can only be disabled from the programming system by selecting "Debug/Reset controller".

3 Operation

3.1 Starting the programming system

Before starting the program, the following requirements must be met:

- The BPRO3 software package must be properly installed on the hard disk (see chapter 2).
- PC and controller(s) must be properly interconnected (see chapter 2.4).
- The device address on the controller must be set (see controller manual).



NOTE

Program development can also be carried out when there is no connection to the controller (off-line programming). Wiring is only necessary when communication with the controller is required.

To start the program, carry out the following steps:

1. Select the drive where BPRO3 is installed, e.g.:
C: <↵>
2. Change to the BPRO3 directory, e.g.:
CD C:\BPRO3 <↵>
3. Call the program:
BPRO3 <↵>
(for a monochrome screen:
BPRO3 MONO <↵>)

The following initial screen is displayed, which also shows the serial number:



Fig. 3-1 Initial screen with serial number

3.2 User interface

This chapter describes general operating functions of the BPRO3 programming system. These operating functions are applicable to all BPRO3 modules. Figure 3-2 shows the various screen areas of BPRO3 (see also chapter 1) and the most important control keys.

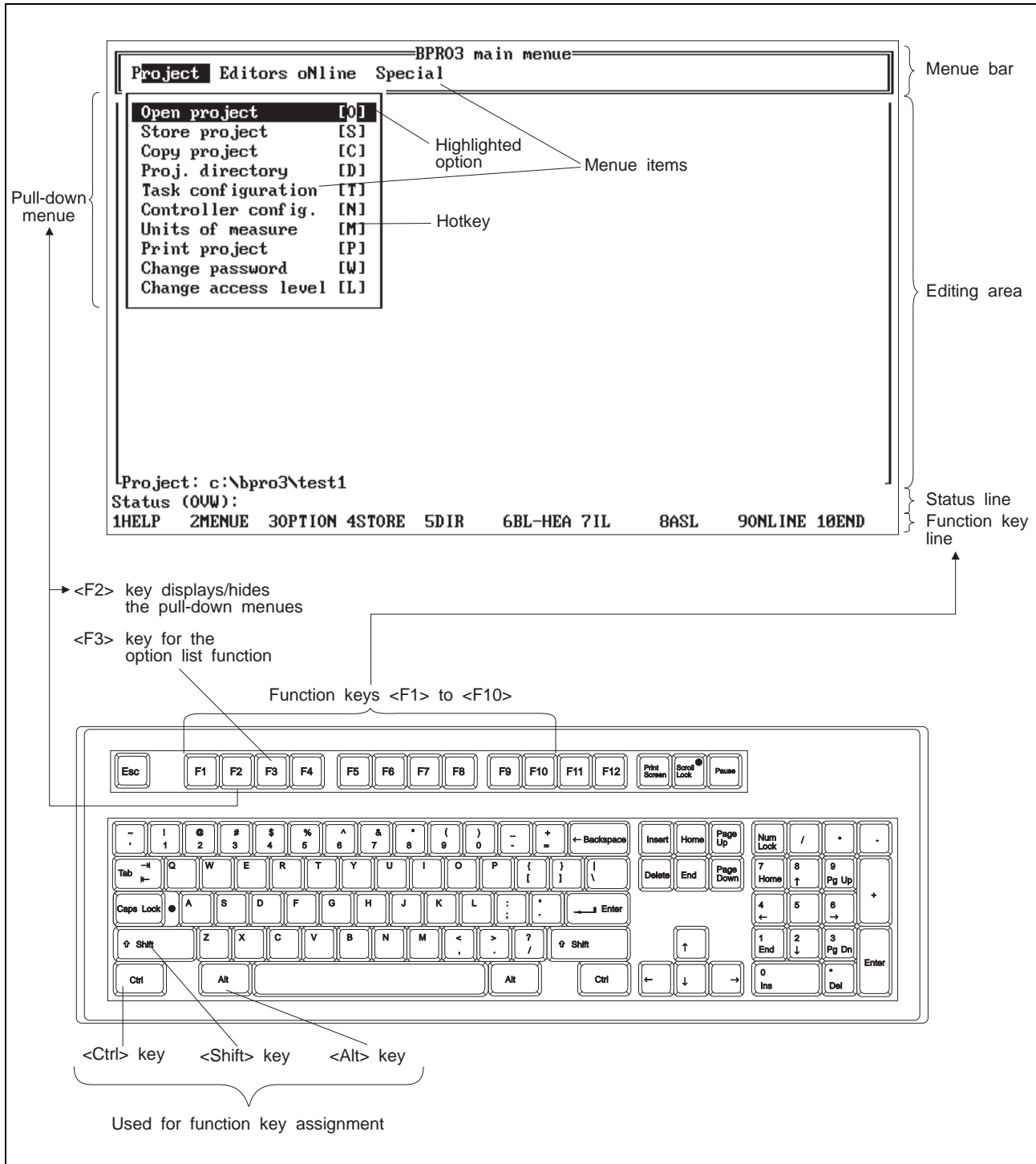


Fig. 3-2 BPRO3 user interface

- 3.2.1 Selecting a menu option** You can open or close a pull-down menu by pressing <F2> (see fig. 3-2). The adjacent pull-down menus open automatically when you move the highlight with the cursor keys (<←> and <→>). The highlight marks the current menu option by inverse representation (see fig. 3-2).

For a more detailed description of the display screen areas illustrated in figure 3-2, refer to chapter 1.



NOTE

Disabled menu options are marked with a "■" symbol.

There are several methods for selecting a menu option:

- Moving with the cursor keys and pressing <↵>
- Using the function keys
- Pressing the hotkey
- Using the "expert shortcuts"
- Executing macros.

The effect is always the same, regardless of the method used for selecting the menu option.

Selecting a menu option using the cursor keys and <↵>

1. Press <F2> to display the menu.
2. Use the cursor keys <←> and <→> to move the menu bar highlight to the desired menu option.
3. If the selected option on the menu bar has a pull-down menu assigned, move the highlight to the desired menu option using the cursor keys <↑> and <↓>.
4. Press the <↵> key.

Selecting a menu option using the function keys

The function is selected directly without opening a pull-down menu first. There are also function key combinations such as <Ctrl>-<F2>. These are used as follows:

1. Press and hold the <Ctrl> key.
2. Press and release the <F2> key.
3. Release the <Ctrl> key.

You can freely assign menu options to function keys. The function keys can also be used in a combination with the <Shift>, <Alt> and <Strg> keys, which results in a maximum number of 40 function key assignments. You can design the function key assignments individually for each menu.

The following keys are exceptions:

- <F1>: Reserved for help information.
- <F2>: Reserved for displaying/hiding the menus.
- <F3>: Reserved for the option list function.

**NOTE**

The key assignment of these keys cannot be changed.

Selecting a menu option using the hotkey

Each menu option has one character which is emphasized by a different colour on a colour screen or by inverse representation on a monochrome screen. The key corresponding to this character is called the hotkey. The pull-down menus show the hotkeys behind the menu options in square brackets.

A menu option can be selected directly by pressing the hotkey. This reduces the time required for selecting a menu option.

Selecting a menu option using the expert shortcut

The so-called expert shortcut is a means of directly selecting menu options by hotkey combinations.

This is achieved by pressing the menu bar hotkey in a combination with the <Alt> key:

1. Press and hold the <Alt> key.
2. Press and release the hotkey corresponding to the menu bar option.
The selected pull-down menu is displayed.
3. Release the <Alt> key.
4. Press and release the hotkey corresponding to the menu option.

Step 4 is not applicable if there is no pull-down menu defined for a menu option.

Automatic selection of menu options using macros

You can use a macro to automatically select several menu options. For details, refer to chapter 3.7.2.

3.2.2 Help provided by the programming system

The programming system provides help on two levels:

- <F1> Help information; gives information on the currently selected menu option.
- <F3> Option list function; gives information on possible input options or displays an option list window which can be used for entering data.

Calling help information with <F1>

You can access help information anytime by pressing <F1>. A window is displayed which contains context-sensitive help information. This means that the type of information displayed depends on your current situation (e.g. the editor loaded, the menu option selected or, for an error message, the error type). The help text contains the following information:

- The purpose of the editor loaded or the menu option selected
- Information on how to use the editor or menu option
- Effects and results of the function.

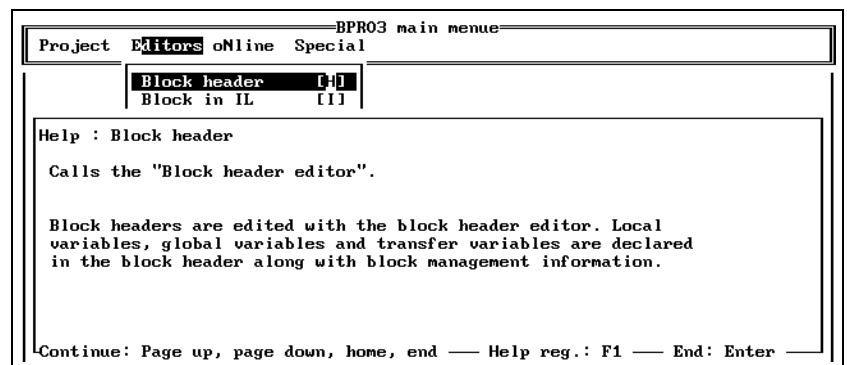


Fig. 3-3 Help information

Operating functions available while help window is open:

Key	Description
<Page Up> and <Page Down>	Scroll help text by page
<Home>	Go to first page
<End>	Go to last page
<F1>	Call a register of all help texts available for the current menu
<Esc>	Close the help window

Calling the option list function and option list window with <F3>

You can call the option list function from any input situation. The field cursor must be positioned on an input field in order to be able to call the option list function.

The option list function is available for inputs required by the programming system. If possible, the programming system offers you a selection of various input options. Whenever this type of input selection is not possible, a window is displayed which describes the valid input. The option list function is called by pressing <F3>.

Figure 3-4 illustrates the option list function for entering an identifier (symbolic name). A general description shows the input format for the identifier.

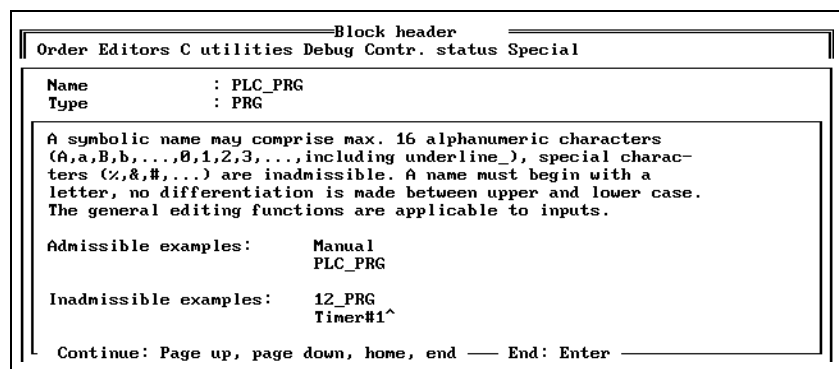


Fig. 3-4 Option list function

Figure 3-5 illustrates an option list window for selecting block names.

Selecting an input option:

1. Move the highlight to the desired input option, using the <↑>, <↓>, <Page Up> or <Page Down> key.
2. Press the <↵> key. The BPRO3 program copies the selected input option into the input field.

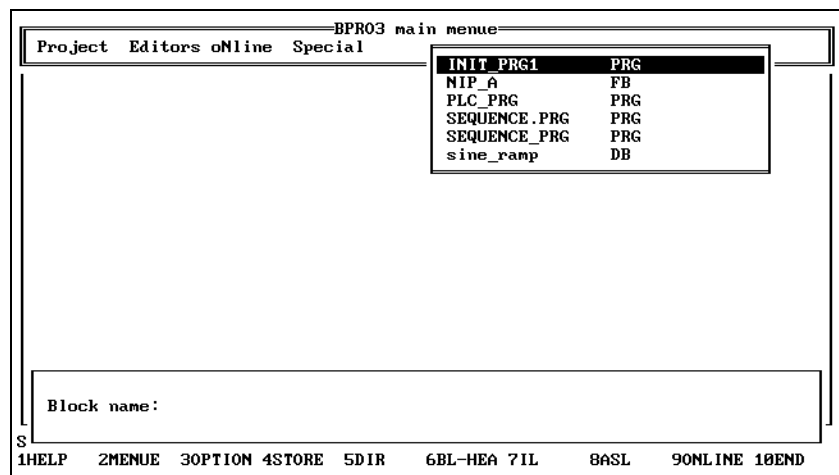


Fig. 3-5 Option list window



NOTE

The number of listed input options can be limited.

Limiting the number of input options listed

Figure 3-6 illustrates a list which is limited to those options beginning with the letter "I":

1. Press the <I> key (the result is shown in fig. 3-6).
2. Pressing the spacebar cancels the limitation.

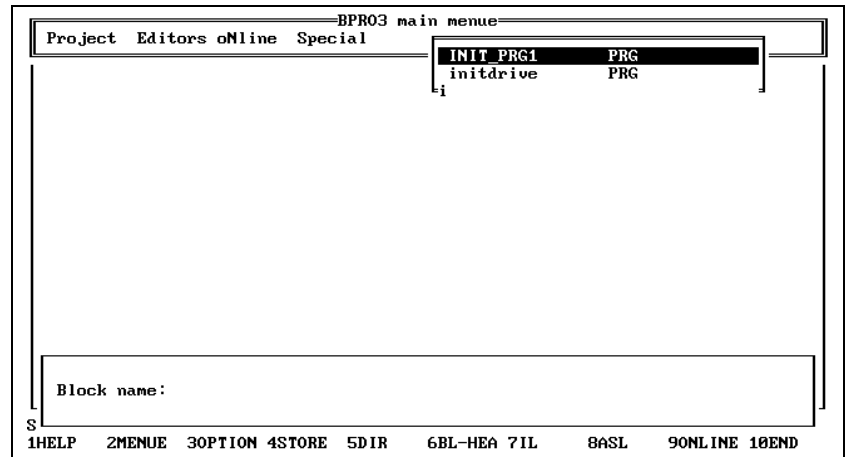


Fig. 3-6 Limited list

3.2.3 General editing functions This chapter describes the general editing functions which are valid for all editors.

3.2.3.1 Cursor types

When loading an editor, a field cursor is displayed in the topmost line of the editing area. A field cursor is characterized by an inverted editable field in the editing area (fig. 3-7). In an editable field, a character cursor is displayed in addition (fig. 3-7).

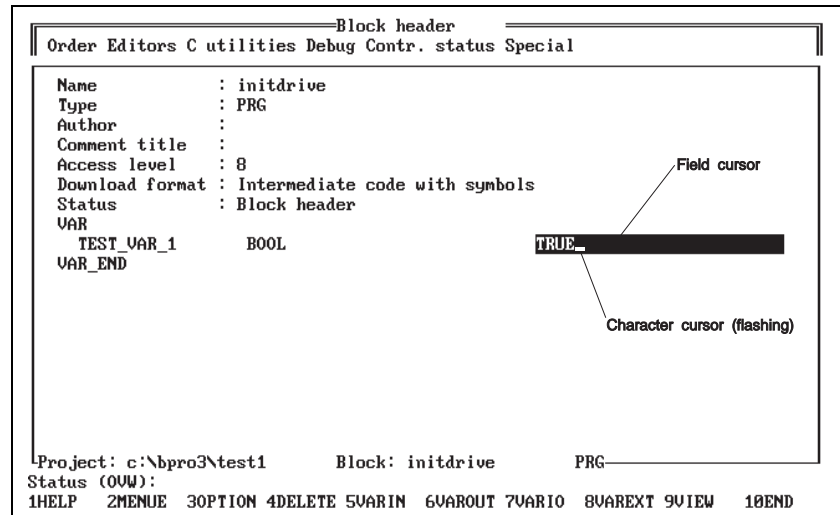


Fig. 3-7 Field and character cursor

Controlling the field cursor


Key	Description
<↑>	One field up
<↓>	One field down
<Tab>	One field to the right
<Shift> and <Tab>	One field to the left
<Page Up>	Field cursor scrolls back by one page
<Page Down>	Field cursor scrolls forward by one page

Controlling the character cursor

Key	Description
<←>	One character to the left
<→>	One character to the right
<Home>	Character cursor moves to first character in the field
<End>	Character cursor moves to last character in the field

3.2.3.2 Editing a field

The following keys can be used for editing within a field (see also "Controlling the character cursor):

Key	Description
<<-> (Backspace)	Deletes the character to the left of the cursor
	Deletes the character at the cursor position
<Ins>	Toggles between insert and overwrite mode.  NOTE <i>The operand input field can be extended by pressing <Ins> at the end of the line.</i>
All character keys (a, b, c, ... 1, 2, 3, ...)	The characters are entered into the input field.

Entering information into a field

1. Position the field cursor in the field to be edited, using the above mentioned keys.
2. Type the appropriate string and correct it, if necessary, using the above mentioned keys.



NOTE

The option list function can be called in any input situation (see chapter 3.2.2).

3. Accept the input by pressing either <Tab>, <Shift> and <Tab>, <↑>, <↓> or <↵>.

Overwriting or inserting characters

The <Ins> key is used for toggling between insert and overwrite mode. The selected mode is displayed in the status line: "INS" for insert mode and "OWR" for overwrite mode.

Inserting lines

1. Move the field cursor to the appropriate line.

Fig. 3-8 Cursor positioning

```

Status          : Block header
VAR
  TEST_VAR_1    BOOL          TRUE
  OUTPUT_1      BOOL          FALSE
VAR_END
    
```

2. Press the <↵> key. A line is inserted above the current line if the field cursor is on the first input field; otherwise it is inserted below the current line.


Fig. 3-9 Line inserted

```

VAR
  TEST_VAR_1    BOOL          TRUE
  OUTPUT_1      BOOL          FALSE
VAR_END
    
```

Exiting a field after editing

To exit a field after editing, press any of the following keys:

Key	Description
<↵>	Accept the input or change. At the same time, a blank line is inserted.
<↑>	Accept the input or change. The field cursor moves to the previous field (up).
<↓>	Accept the input or change. The field cursor moves to the next field (down).
<Tab>	Accept the input or change. The field cursor moves to the next field on the right.
<Shift> and <Tab>	Accept the input or change. The field cursor moves to the next field on the left.
<Esc>	Discard the input or change.
	 ATTENTION <i>Any input or change will be lost!</i>

3.2.3.3 The comment editor

A large variety of individual project data (e.g. blocks, variables, assignments in assignment lists) can be provided with comments. For this purpose, the built-in text editor is used. Comments can be entered on two levels: A comment title which must be entered in the appropriate editor, and a comment text of any length which is only displayed in the comment editor. Figure 3-10 shows the comment title in the block header editor.

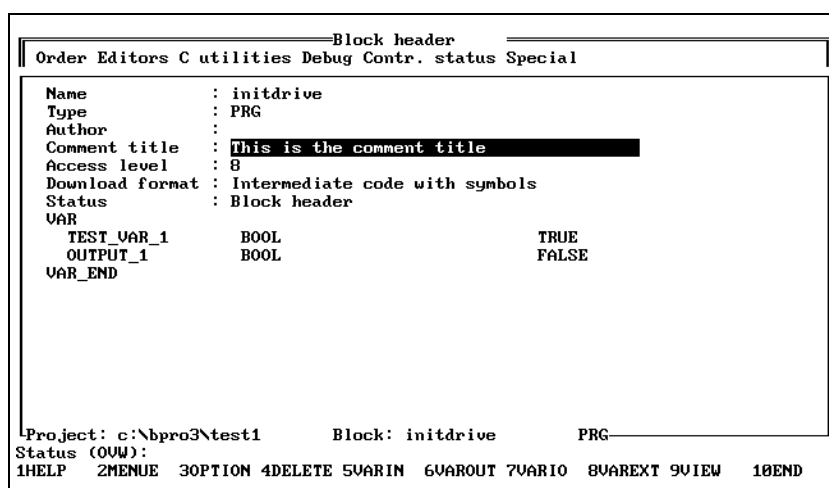


Fig. 3-10 Comment title displayed in block header editor

Calling the comment editor

1. Move the field cursor to the comment to be edited.
2. Select the menu option "Order/Comment". The comment editor is displayed.

Block orders in the comment editor

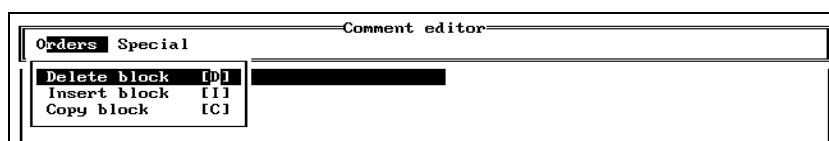


Fig. 3-11 Block commands in comment editor

Menu option	Description
Delete block	Delete a block in the comment text
Insert block	Insert a block in the comment text
Copy block	Copy a block in the comment text



NOTE

For more information, see "Block processing" in chapter 3.2.3.4.

The “Special” pull-down menu in the comment editor

Fig. 3-12 The pull-down menu “Special”



Menu option	Description
End	Exiting the comment editor
Macro	Operations with macros; see chapter 3.7.2.
F keys	Function key assignment; see chapter 3.7.3.



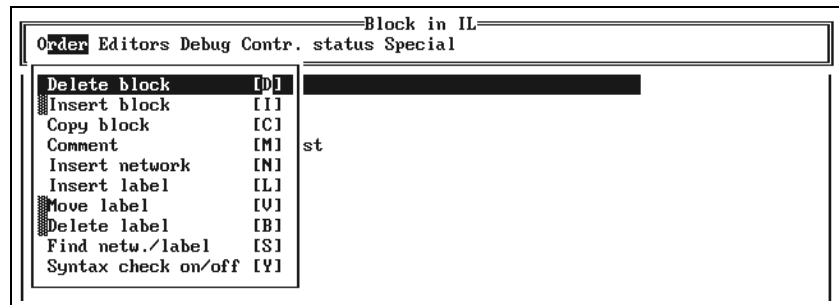
NOTE

For more information, see “Special functions” in chapter 3.7.

3.2.3.4 Block processing

In the various editors (e.g. IL editor), the pull-down menu “Order” is used for deleting, inserting or copying blocks.

Fig. 3-13 The pull-down menu “Order”



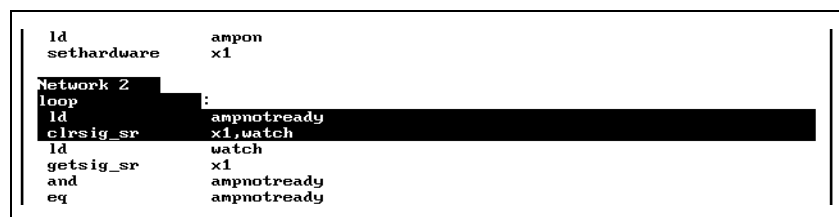
Deleting a block

“Order/Delete block”

A marked block is deleted from the editing area and written to a buffer (file on hard disk).

1. Move the field cursor to the start of the block to be deleted.
2. Select the menu option “Order/Delete block”.
3. Use the <↓> key to move the field cursor to the end of the block to be deleted. The block marked during this process is displayed inversely.
4. Press the <↵> key to delete the marked block.

Fig. 3-14 Deleting a block



5. In the lower screen area a window is displayed which prompts you to enter the name of the file in which the block is to be stored. The default setting is "STANDARD". Enter a different name, if necessary.
6. Press the <↵> key. If a block already exists with the same name, the program prompts you to confirm overwriting the old block.

**NOTE**

For the standard buffer, no confirmation prompt is output.

```

Network 2      Wait for power drive ready
loop          :
ld            ampnotready
cirsig_sr     x1,watch
ld            watch
getsig_sr     x1
and           ampnotready
eq            ampnotready
jmpc         loop

Network 3      Calculate acceleration
ld            200

Please enter filename : STANDARD
S
1HELP  2MENU  3OPTION 4BL-DEL 5BL-INS 6BL-COP 7NW-INS 8LA-INS 9VIEW 10END

```

Fig. 3-15 Entering a buffer name

You can cancel this procedure by pressing <Esc>.

**NOTE**

After executing "Order/Delete block" the block data in the editing area are lost. They are only kept in the buffer then.

**NOTE**

The option list function (<F3>) displays an option list window with the existing block names.

Inserting a block from the buffer

"Order/Insert block"

The menu option "Order/Insert block" can be used for inserting blocks written to the buffer by "Order/Delete block" or "Order/Copy block". Proceed as follows:

1. Select the menu option "Order/Insert block".
2. Specify the buffer name (default: "STANDARD"). You can call the option list function (<F3>) to display a list of all available blocks.
3. Press the <↵> key. The block is inserted before the cursor position.

Copying a block

"Order/Copy block"

Copying a block is almost identical to deleting a block, with one exception: The marked block remains in the editing area when copied.

3.2.3.5 Defining symbolic names Symbolic names improve the readability of a program. The following program elements can be identified with symbolic names:

- Variables
- Labels
- Blocks

A symbolic name for a variable can have a maximum length of 16 alphanumeric characters (A, a, B, b, ..., 0, 1, 2, 3, ...) and include an underline character (_). Special characters (% , & , # , ...) are not allowed. Names must have initial capitals, and they are case-sensitive. The general editing functions (see chapter 3.2.3) can be used when entering symbolic names.

Valid examples: INP_1
 inp_1
 Lamp_ON

Invalid examples: 12_Output, (Name starts with a number)
 status#1, (Name includes a special character).

**3.2.4 Log files,
repeating an
editing session**

In log files, all keystrokes made during an editing session are stored. This allows you to automatically repeat an interrupted editing session (e.g. due to power failure). There are up to 2 log files: "BPRO3TRC.JO1" and "BPRO3TRC.JO2". You can set the number from 0 to 2. They are stored in the "BLOCK" directory (see setup) on the project path (see also chapter 3.4.1). The time interval for storing the keystrokes and the number of log files are defined with "Special/Setup". The log file can be renamed on the operating system level to be reused for later editing sessions.



ATTENTION

A log file can only be executed properly if the current state of the project is identical with the state of the project at the time of the last BPRO3 start. This is not the case if the project was stored during the last editing session.

To use log files, proceed as follows:

1. Backup the current state of the project, for example:
`COPY C:\BPRO3\PROJECT1 C:\BACKUP`
2. Start BPRO3 and work as usual (all keystrokes are stored in the log file "BPRO3TRC.JO1").
3. Reestablish the previous state of the project:
`COPY C:\BACKUP C:\BPRO3\PROJECT1`
4. Copy the log file to the BPRO3 directory:
`COPY C:\BPRO3\BLOCK\BPRO3TRC.JO1 C:\BPRO3`
5. Execute the log file:
`CD C:\BPRO3`
`BPRO3 -r BPRO3TRC.JO1`

Execution of a log file can be aborted by pressing <Esc>.



NOTE

If an error occurs during execution of the log file, BPRO3 interrupts its execution until the error message is acknowledged by pressing <↓>.

Saving the log files

The keystrokes are first written to temporary storage in RAM. They are saved to the hard disk when:

- the preset time as defined in "Special/Setup" has elapsed or
- the temporary storage has reached a size of 1 Kb.

Number of log files

In "Special/Setup", you can set the number of log files to 2. In this case the programming system stores the keystrokes in two files and ensures that only one of the two files is open at a time. This means that at least one usable log file will be available in any case even if a power failure should occur while saving a log file.

3.3 The BPRO3 main menu

Figure 3-16 shows an overview of the BPRO3 main menu. The main menu is the topmost level of the menu system. From the BPRO3 main menu, you can access all the editors, menus and functions available in the programming system.

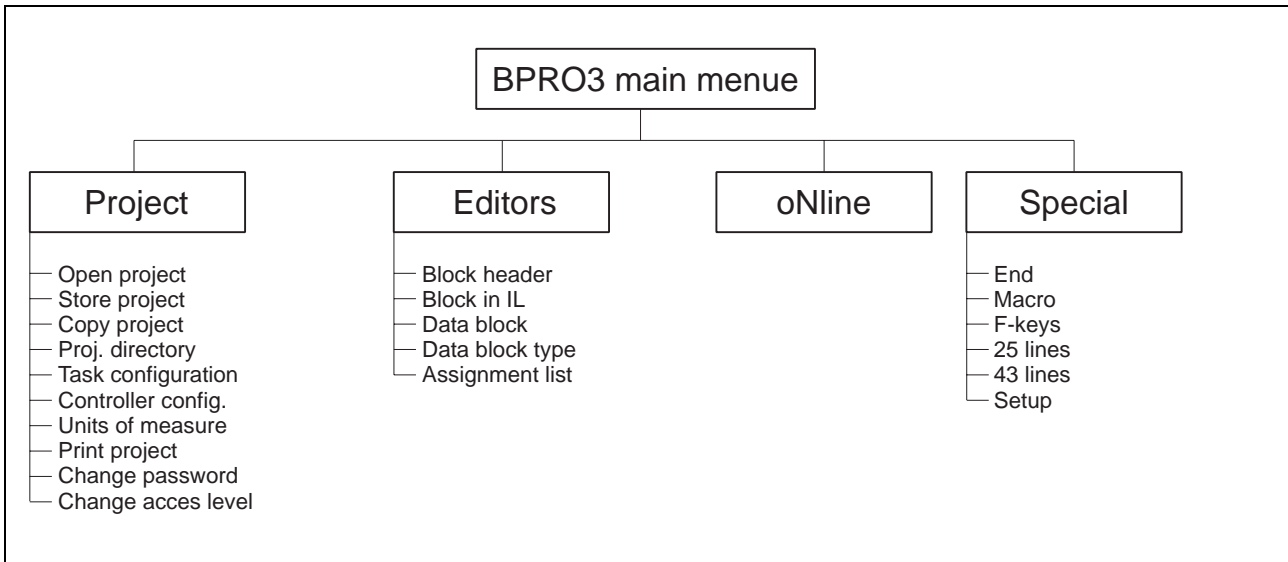


Fig. 3-16 BPRO3 main menu



NOTE

Figure 3-16 serves as an overview for chapters 3.4 to 3.7.

3.4 Project-related operations

“Project” This pull-down menu contains options for managing the project as a whole. A project comprises all data required for solving a specific control task (see chapter 1).

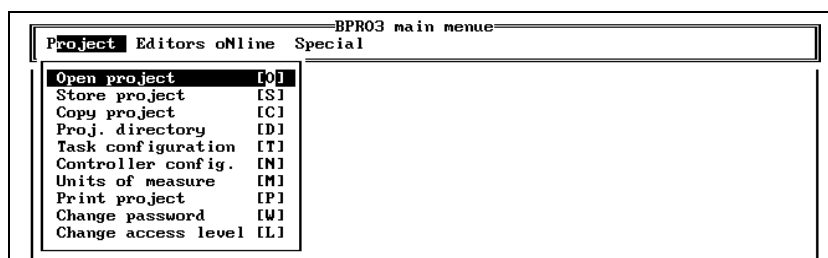


Fig. 3-17 Project operations

Menu option	Description	Reference
Open project	Open a project.	See chapter 3.4.1
Store project	Store the current project.	See chapter 3.4.2
Copy project	Copy a project.	See chapter 3.4.3
Proj. directory	List all blocks included in a project, list all block libraries, read the controller directory, and other functions.	See chapter 3.4.4
Task configuration	Open the task configuration.	See chapter 3.4.5
Controller config.	Open the controller configuration.	See chapter 3.4.6
Units of measure	Enter user-defined units of measurement	See chapter 3.4.7
Print project	Select the project data to be printed, set printing parameters and print project data.	See chapter 3.4.8
Change password	Change the password.	See chapter 3.4.9
Change access level	Change the access level.	See chapter 3.4.10

3.4.1 Loading a project

"Project/Open project"

The menu option "Project/Open project" is designed for creating a new project or loading an existing one.

When BPRO3 is started, it automatically loads the project set with "Special/Setup/Project path". When you start the programming system for the first time, it automatically loads the sample project EXAMPLE1 (see chapter 2.6 for a description). The programming system then creates further directories and files; see figure 3-18.



NOTE

The LIBRARY project contains a user library with many useful blocks for programming (see thumb index section 3).

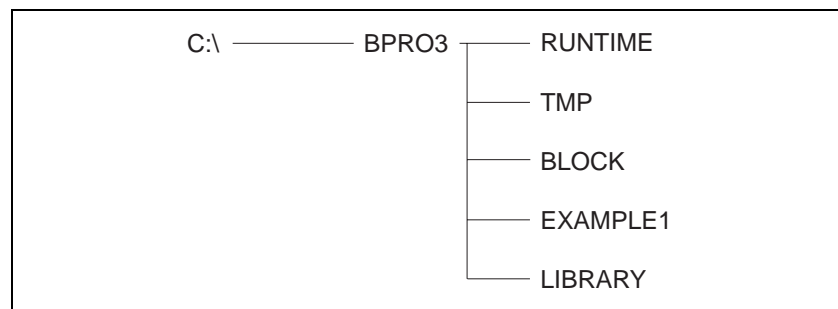


Fig. 3-18 DOS directory structure

When loading an existing project, the programming system creates a copy of all project data. All changes and inputs which are entered until the next saving process are written to this project copy by the programming system. The copy is stored in the project directory TMP.

The original project data are not overwritten by BPRO3 until "Project/Store project" is executed. This gives you the possibility to reset the project to the previous state by reloading.

When you exit the programming system without saving, the original project data are also retained.



ATTENTION

Any changes and new inputs will be lost if you exit BPRO3 without executing "Store project", since the programming system deletes the project copies.



ATTENTION

If an editing session is interrupted by a severe failure (e.g. power failure), the current copy of the project data may be unusable for BPRO3. The program will recognize such an error when opening a project and will prompt you whether to use the latest backup copy.

3.4.2 Storing a project

"Project/Store project"

As described in chapter 3.4.1, "Loading a project", the programming system creates backup copies where changes and new inputs are entered. When storing, BPRO3 overwrites the original project data with the data from the backup copies.



ATTENTION

When a project has been stored, the last editing session cannot be automatically repeated any more (see chapter 3.2.4, "Log files").

3.4.3 Copying a project

“Project/Copy project”

The menu option “Copy project” can be used for copying a project to the current project path. The current project will be overwritten. Thus, if you want to create a copy of a project, you first have to create a new project or project path using the menu option “Project/Open project”. Then use “Project/Copy project” to copy the desired project to the new project path.

3.4.4 The project directory

“Project/Proj. directory”

You can access the project directory from the menu option “Project/Proj. directory”. A menu is displayed which provides options for the following purposes:

- Output a list containing the blocks of the project
- List the controller library and the standard library
- Sort blocks by name and date
- Various commands such as Delete, Find, Rename, etc.
- Online functions

Figure 3-19 gives an overview of the project directory menu structure.

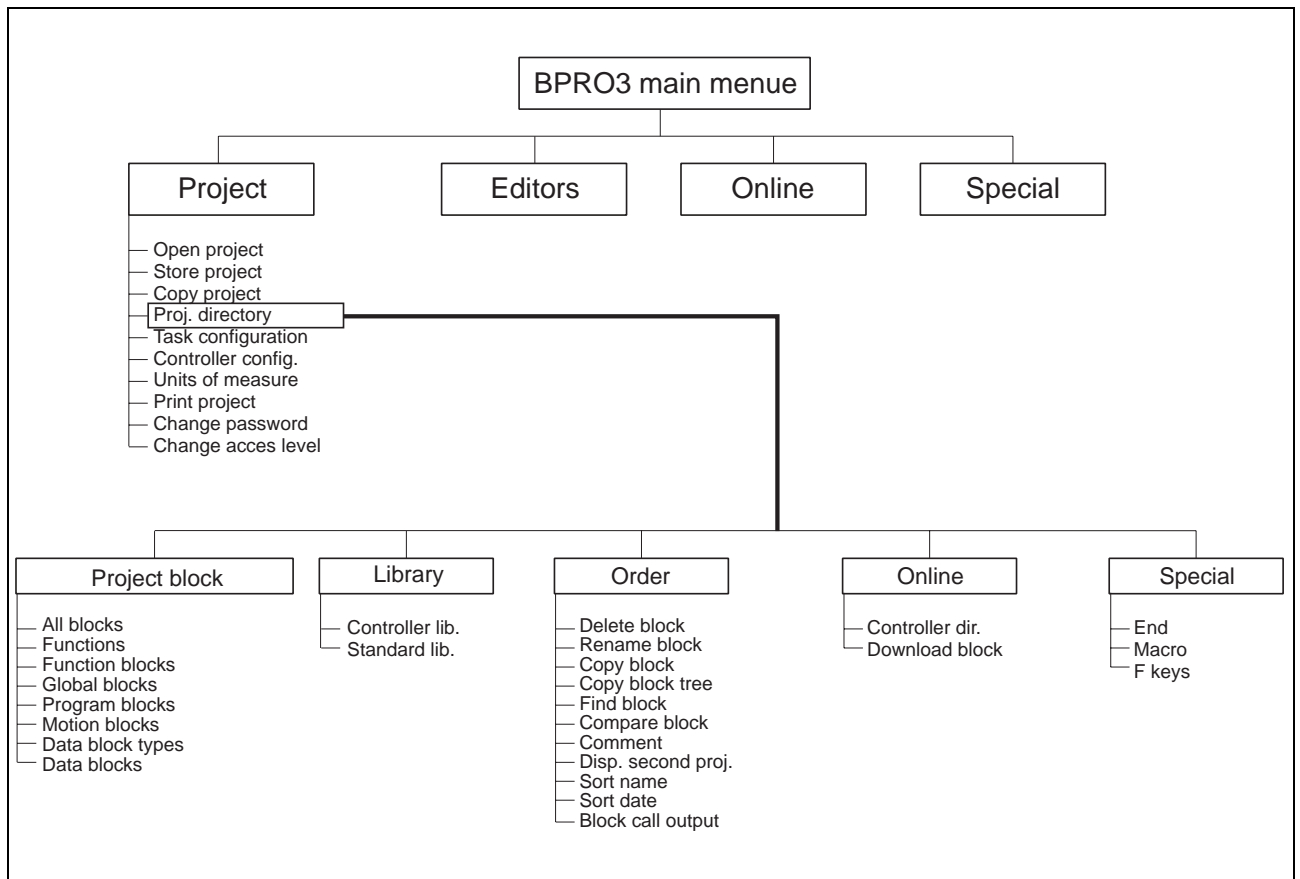


Fig. 3-19 Project directory menu

3.4.4.1 Selecting the block types to be listed

When opening the project directory it shows a list of all available blocks. The options of the pull-down menu "Project block" enable you to limit the list to certain types of blocks.

"Project/Proj. directory/
Project block"

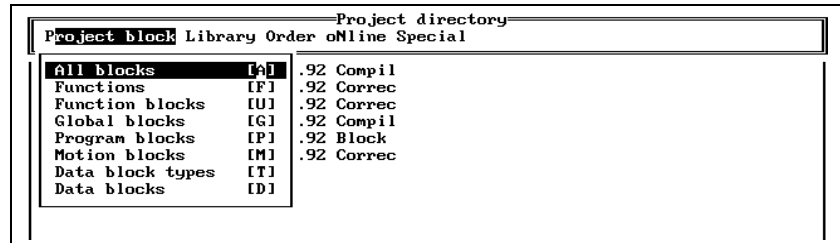


Fig. 3-20 Project blocks

Menu option	Description
All blocks	List all blocks available in the project.
Functions	List all functions available in the project.
Function blocks	List all function blocks available in the project.
Global blocks	List all global blocks available in the project.
Program blocks	List all program blocks available in the project.
Motion blocks	List all motion blocks available in the project.
Data block types	List all data block types available in the project.
Data blocks	List all data blocks available in the project.

3.4.4.2 Listing the block libraries

The BPRO3 program package includes two block libraries created by the manufacturer (see programming manual, chapter 7).

"Project/Proj. directory/
Library"

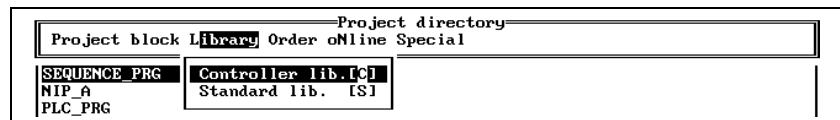


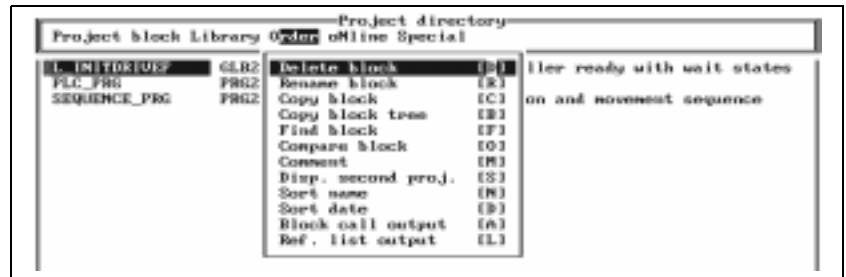
Fig. 3-21 Block libraries

Menu option	Description
Controller lib.	List the controller library.
Standard lib.	List the standard library.

3.4.4.3 Miscellaneous commands

“Project/Proj. directory/
Order”

Fig. 3-22 Miscellaneous
commands




Menu option	Description
Delete block	Delete the block marked by the cursor.
Rename block	Rename the block marked by the cursor.
Copy block	Duplicate a block from the current project or copy a block from another project.
Copy block tree	Copy the selected block together with all dependent blocks from another project; the dependent blocks are copied first.
Find block	When entering the name of the block, the field cursor moves to the block. If the block does not exist, an error message is out put.
Compare block	Compare two blocks.
Comment	When you first select this menu option, the comment title for the block can be entered. When you select this option for the second time, the comment editor is called.
Disp. second proj.	Display a second project.
Sort name	Sort the list by name.
Sort date	Sort the list by date.
Block call output	Output the call tree of the block marked by the cursor.
Ref. list output	Output the reference list of the block marked by the cursor.

3.4.4.4 Online functions

“Project/Proj. directory/
Online”

Fig. 3-23 Online functions



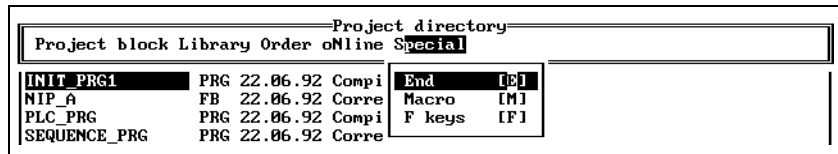
Menu option	Description
Controller dir.	All blocks stored in the controller are compared to the blocks of the current BPRO3 project and displayed.  NOTE Login must have been performed (see chapter 3.6.2).
Download block	Load the block marked by the field cursor into the controller.

3.4.4.5 The pull-down menu “Special”

“Project/Proj. directory/
Special”

Fig. 3-24 “Special” pull-down menu

The options of this pull-down menu are described in chapter 3.7.



Menu option	Description
End	Exit the BPRO3 main menu.
Macro	Operations with macros.
F keys	Function key assignment.

3.4.5 Task configuration

In the task configuration, the blocks are assigned to the individual tasks. Figure 3-25 shows the menu structure of the task configuration editor.

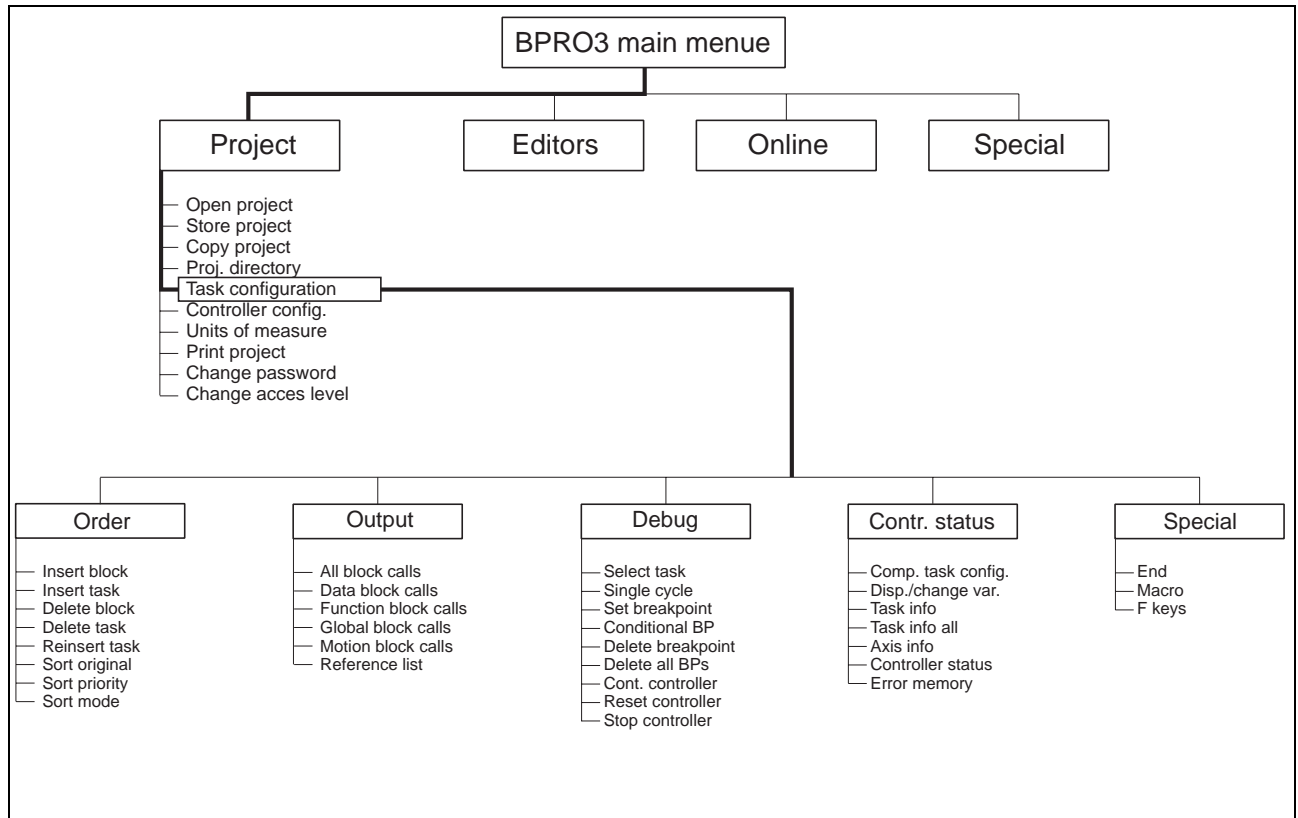


Fig. 3-25 Task configuration menu structure

The task configuration is structured according to five columns which contain information on the tasks within the project.

Explanations on the columns

Name:	The name of the task.
Event:	The execution of the blocks assigned to a task on the controller depends on the occurrence of an event (logical address on input). In place of a logical address, a symbol from the assignment list or a TRUE status for an event which is always true may be present.
Interval:	The blocks assigned to the task are always executed after the time entered here.
Prio:	If several tasks are ready for processing at the same time, those with the higher priority are processed first.
Blocks:	The names of the program blocks assigned to the task.



NOTE

The "Name", "Event", "Interval", "Prio" and "Blocks" columns are pre-defined in the current release and cannot be changed.

3.4.5.1 The pull-down menu "Order"

"Project/Task configuration/
Order"

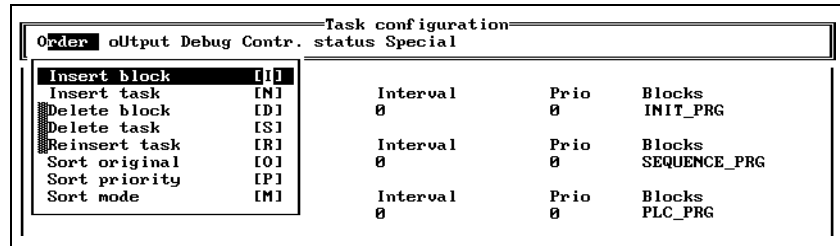



Fig. 3-26 The pull-down menu "Order"

Menu option	Description
Insert block	Assign a block to the task marked by the field cursor.
Insert task	Insert a new task. <div style="text-align: center;">  <p>ATTENTION Some Series 300 controllers can only be operated with tasks generated automatically by BPRO3, e.g. WDP5-318. Additional tasks are not supported by these controllers.</p> </div>
Delete block	Delete the block marked by the field cursor.
Delete task	Delete the task marked by the field cursor.
Reinsert task	Reinsert a previously deleted task.
Sort original	Sort the task list as it was entered originally.
Sort priority	Sort the task list by priority.
Sort mode	Sort the task list by mode.

Assigning a block to a task

"Project/Task configuration/
Order/Insert block"

To assign a block, proceed as follows:

1. Move the cursor to any field of the task.
2. Press <F2> to open the menu.
3. Select "Order/Insert block". BPRO3 moves the field cursor to the block column and inserts a new field for the block name.
4. Enter the block name or select it using the option list function (press <F3>).



NOTE

If a block with the name entered does not yet exist, the programming system prompts you whether to create a new block. If you confirm by pressing <Y>, the block header editor is called.

During program run, the controller processes the blocks in the sequence defined here.

Inserting a new task

“Project/Task configuration/
Order/Insert task”

There are three different tasks (for a description, see BPRO3 Programming Manual):

- INIT task (Name: INIT)
- SEQUENCE task (Name: SEQUENCE)
- PLC task (Name: PLC)

Further tasks can be inserted.



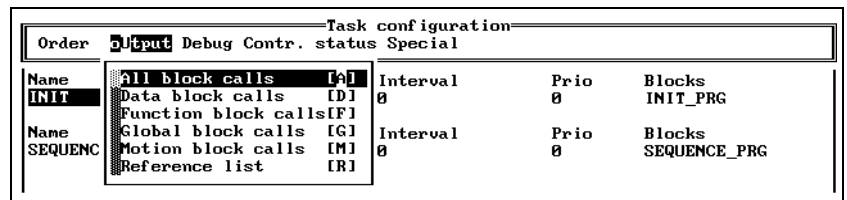
ATTENTION

Certain Series 300 controllers can only work with the tasks generated automatically by BPRO3; this is valid, for example, for the WDP5-318 controller. Additional tasks are not supported by these controllers.

3.4.5.2 Output block calls and a reference list

“Project/Task configuration/
Output”

Fig. 3-27 The pull-down menue “Output”



Menue option	Description
All block calls	Output all block calls of the configured blocks.
Data block calls	Output all data block calls of the configured blocks.
Function block calls	Output all function block calls of the configured blocks.
Global block calls	Output all global block calls of the configured blocks.
Motion block calls	Output all motion block calls of the configured blocks.
Reference list	Output reference list.

3.4.5.3 Online functions

This pull-down menu offers various options for program testing (debugging).

“Project/Task configuration/
Debug”

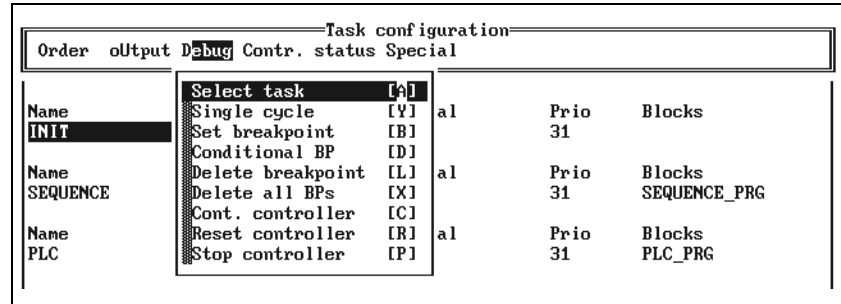


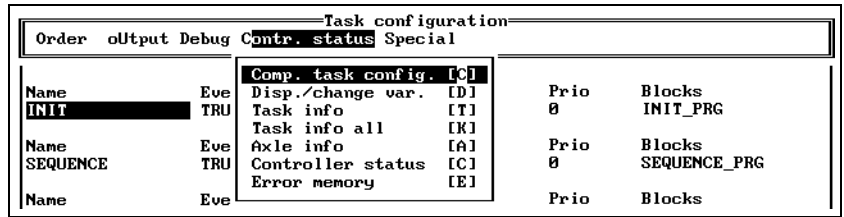
Fig. 3-28 The pull-down menue “Debug”

Menue option	Description
Select task	Select task for Step, Step over, Single cycle, Set breakpoint, Conditional breakpoint, Stop task, Reset task, and Continue task. If you omit this menue option, BPRO3 will prompt you for the name of the task whenever any of the above mentioned options is called.
Single cycle	Execute a program in single cycles.
Set breakpoint	Set a breakpoint. Program execution stops at the beginning of the block.
Conditional BP	Set a conditional breakpoint.
Delete breakpoint	Delete individual breakpoints.
Delete all BPs	Delete all breakpoints.
Cont. controller	Start the controller.
Reset controller	Reset the controller.
Stop controller	Stop the controller.

3.4.5.4 The pull-down menu
“Contr. status”

“Project/Task configuration/
Contr. status”

Fig. 3-29 The pull-down menu
“Contr. status”

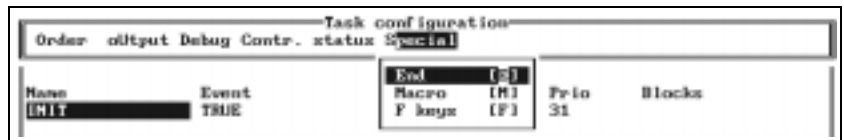


Menu option	Description
Comp. task config.	Compare the controller task configuration with the current programming system task configuration.
Disp./change var.	Display and change global variables in the controller.
Task info	Shows the status, the attribute and the block calls of a task.
Task info all	Shows the status, the attribute and the block calls of all tasks.
Axis info	Output information on the current axis status.
Controller status	Output information on the current controller status.
Error memory	Displays the controller error memory.

3.4.5.5 The pull-down menu
“Special”

“Project/Task configuration/
Special”

Fig. 3-30 The pull-down menu
“Special”



Menu option	Description
End	Exit the task configuration.
Macro	Operations with macros.
F keys	Function key assignment.

The options of this pull-down menu are described in chapter 3.7.

3.4.6 Controller configuration

The controller configuration defines the following characteristics of the controller:

“Project/Controller config.”

- Number and type of tasks and programmable blocks
- Flag area allocated
- Number and addresses of digital inputs and outputs
- Addressable interfaces and axes, communication and position detection
- Available library blocks
- Constants for parameter setting for the library blocks

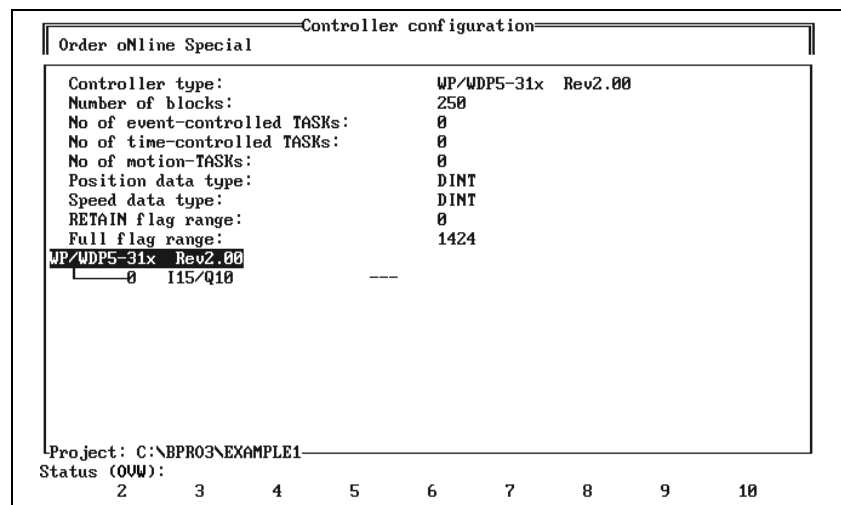


Fig. 3-31 Controller configuration



NOTE

Before you can work with the editors in a new project you must define a controller configuration.



NOTE

Projects created with BPRO3 versions up to V3.11 can only be processed after reentering a controller configuration.

New project

When you create a new project using the menu option “Project/Open project” in the BPRO3 main menu, the menu system automatically calls the controller configuration and an option list with the available controller types. The individual controller types are described in chapter 3.4.6.1. When having selected the controller type, the controller configuration can be adjusted to the controller in BPRO3.

Editing

The controller configuration editor comprises two editing areas:

- General information
- Controller tree

General information

The general information area contains the maximum values for tasks and the blocks supported by the selected controller type. In addition, the size of the flag area and the RETAIN flag area which is included therein are displayed.

**ATTENTION**

After entering or changing the controller configuration (when exiting the controller configuration editor), system constants for parameter setting of the library blocks are generated. These occupy some space in the upper section of the flag area and therefore reduce the available flag range. In the assignment list, flag addresses within the range occupied by the system constants cannot be used. The system constants occupy approx. 400 flag words.

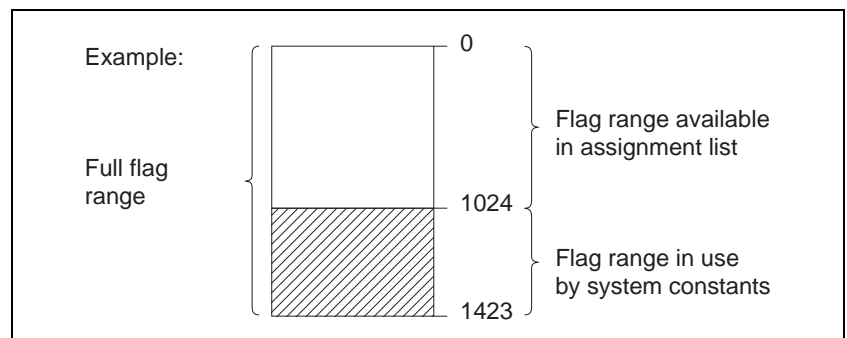


Fig. 3-32 Flag area for system constants

Controller tree

The controller tree shows the available input/output modules (interfaces) for the selected controller type. Input/output extensions can be installed on controllers with Rev3.0 or higher (refer to revision level ID on the type plate of the controller).

Input/output extensions can be addressed by entering input/output module designations for module slots 1 to 10 (see chapter "Addressing inputs/outputs"). To do this, move the cursor to the corresponding input field, open the option list by pressing <F3> and select the appropriate designation.

**NOTE**

When leaving the controller configuration screen, the project is set to the specified controller type. This process may take a few minutes.

3.4.6.1 Available controller types The following controller types are available:

- WP/WDP5-31X Rev2.00
Single-axis stepping motor controller with or without integral power controller (see WP-311 and WDP5-318 controller manual).
- WPM-311.004 Rev2.00
Four-axis stepping motor controller without integral power controller (see WPM-311.004 controller manual).
- WP/WDP3-33X Rev2.00
Single-axis AC servomotor controller with integral power controller (see WDP3-33X controller manual).
- WP/WDP5-31X Rev3.00
Similar to Rev2.0 units, however, can be extended using field bus systems. The axis-related inputs I16 to I20 can also be addressed using the input word 0.1 (see chapter “Addressing inputs/outputs”).
@%IW0.1 = @%IW1
- WPM-311.004 Rev3.00
Similar to Rev2.0 units, however, can be extended using field bus systems. The axis-related inputs (limp, limn, ref, stop and trig) of the four axes can also be addressed using the input words 0.2 and 0.3 (see chapter “Addressing inputs/outputs”).
@%IW2 = @%IW0.2
@%IW3 = @%IW0.3
- WP/WDP3-33X Rev3.00
Similar to Rev2.0 units, however, can be extended using field bus systems. The axis-related inputs I16 to I20 can also be addressed using the input word 0.1 (see chapter “Addressing inputs/outputs”).



ATTENTION

Series 300 controllers of version 3.02 or higher can address additional inputs and outputs via the process image, as compared to earlier controller versions. As a prerequisite, the connected input/output extensions must be specified in the controller configuration.

3.4.6.2 Addressing inputs/outputs

In the controller tree, the available inputs and outputs appear as module designations such as "I15/I5/Q10". This module has 2 input words with 15 bits in the first word, 5 bits in the second word and 1 output word with 10 bits.

The address, and thus the format in BPRO3 for addressing these inputs and outputs, can be determined as follows:

- In the controller tree, a number appears next to the module designation (e.g. "I15/I5/Q10"). This is the module number of the input/output designation.
- The first word of an input/output module is assigned word number 0, the second word is assigned word number 1 accordingly.
- Input/output modules connected via Interbus-S start with module number 1. The first input/output module in the Interbus-S ring is assigned to the first configured input/output module. The second input/output module in the Interbus-S ring is assigned to the second configured module, and so on.
- Input/output modules of the central unit are assigned module number 0 (see chapter 1.4.2.5, "Inputs/outputs" in the programming manual).

Addressing inputs/outputs with Rev.2.00 single-axis unit

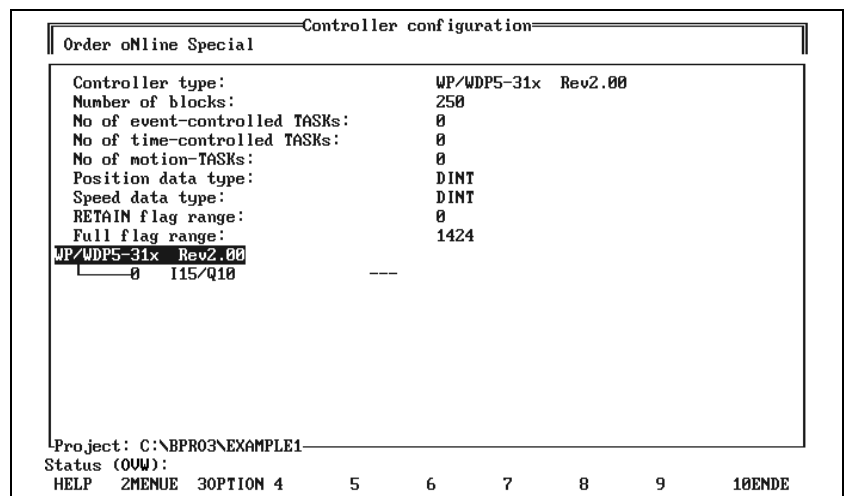


Fig. 3-33 Single-axis unit Rev.2.00

Module		I/O bits	I/O word
I15/Q10-1	I15	%IX0.0 to %IX0.14 @IX0.0 to @IX0.14	%IW0 @IW0
	Q10	%QX0.0 to %QX0.9	%Q0 @Q0

Addressing inputs/outputs with Rev.3.00 single-axis unit

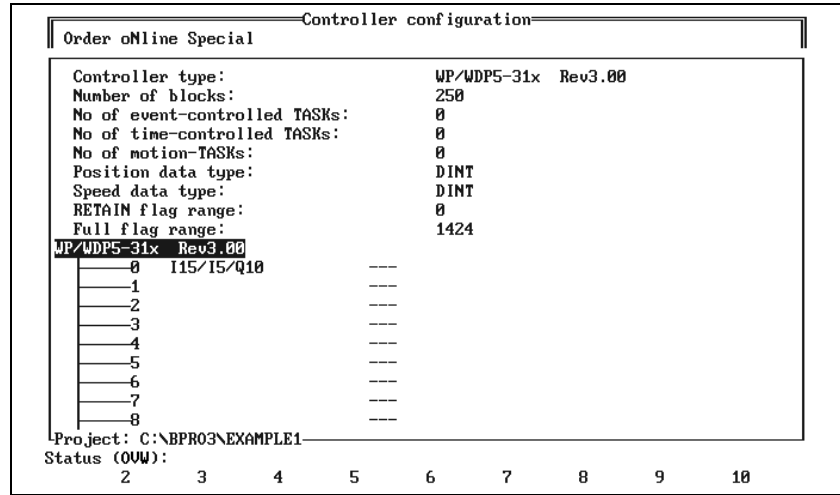


Fig. 3-34 Single-axis unit Rev.3.00

Module		I/O bits	I/O word
I15/I5/Q10-1	I15	%IX0.0 to %IX0.14 @IX0.0 to @IX0.14	%IW0 @IW0
	I5	%IX0.16 to %IX0.20 @IX0.16 to @IX0.20	%IW1 @IW1
	Q10	%QX0.0 to %QX0.9	%Q0 @Q0

Addressing inputs/outputs with Rev.3.00 four-axis unit

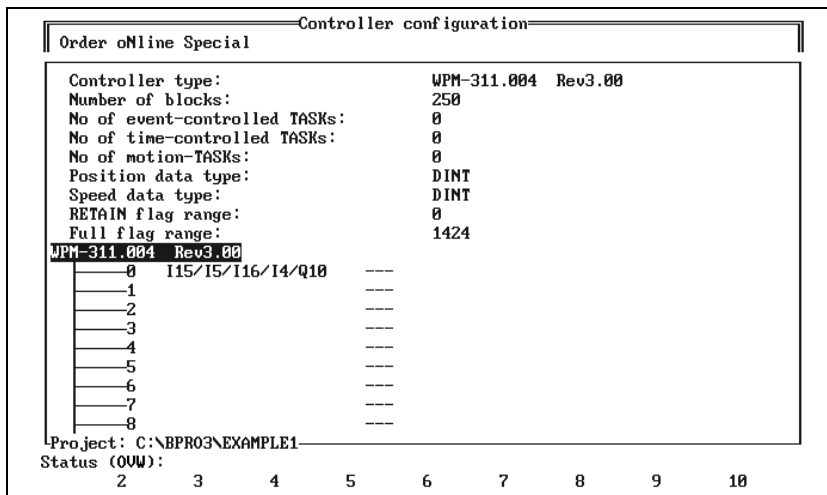


Fig. 3-35 Four-axis unit Rev.3.00

Module		I/O bits	I/O word
I15/I5/I16/I4/Q10	I15	%IX0.0 to %IX0.14 @IX0.0 to @IX0.14	%IW0 @IW0
	I5	%IX0.16 to %IX0.20 @IX0.16 to @IX0.20	%IW1 @IW1
	I16	%IX0.32 to %IX0.47 @IX0.32 to @IX0.47	%IW2 @IW2
	I4	%IX0.48 to %IX0.51 @IX0.48 to @IX0.51	%IW3 @IW3
	Q10	%QX0.0 to %QX0.9	%Q0 @Q0

Addressing inputs/outputs with MP 926 input/output module



NOTE

Only possible for units with revision level 3.00 and higher.

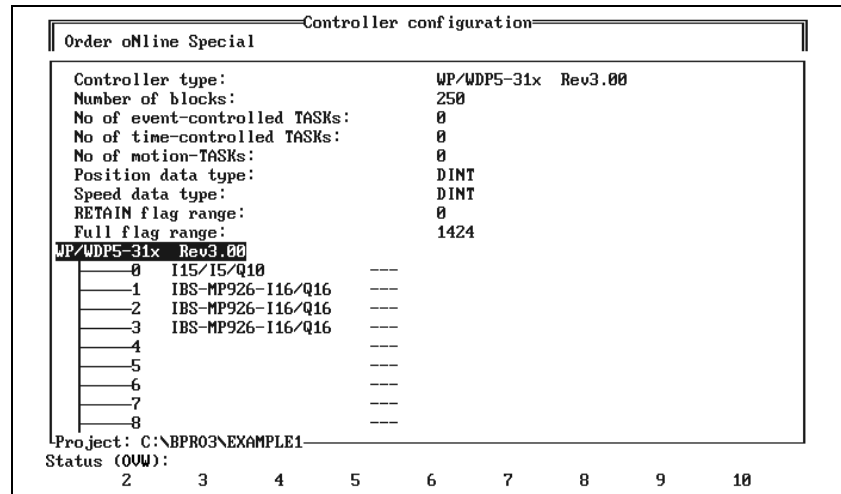


Fig. 3-36 Input/output modul MP 926

Module		I/O bits	I/O word
1st MP 926	I16	%IX1.0 to %IX1.15 @IX1.0 to @IX1.15	%IW1.0 @IW1.0
	I16/Q16-1	Q16	%QX1.0 to %QX1.15 %Q1.0 @Q1.0
2nd MP 926	I16	%IX2.0 to %IX2.15 @IX2.0 to @IX2.15	%IW2.0 @IW2.0
	I16/Q16-1	Q16	%QX2.0 to %QX2.15 %Q2.0 @Q2.0
3rd MP 926	I16	%IX3.0 to %IX3.15 @IX3.0 to @IX3.15	%IW3.0 @IW3.0
	I16/Q16-1	Q16	%QX3.0 to %QX3.15 %Q3.0 @Q3.0
...
nth MP 926	I16	%IXn.0 to %IXn.15 @IXn.0 to @IXn.15	%IWn.0 @IWn.0
	I16/Q16-1	Q16	%QXn.0 to %QXn.15 %Qn.0 @Qn.0

3.4.7 Defining units of measurement

“Project/Units of measure”

Constants can be assigned symbolic names. BPRO3 handles these symbolic names like multipliers or units of measurement.

Example: 1000 mm, 0.78 rad, 70 inch

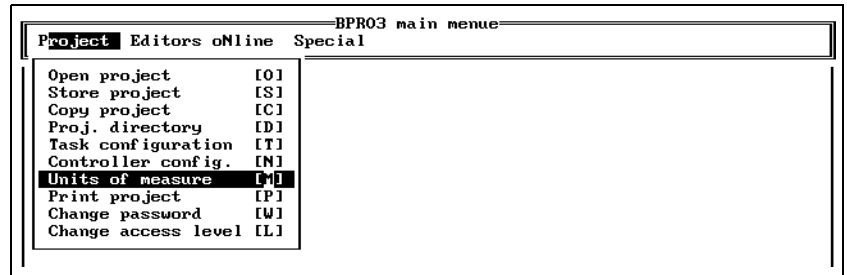


Fig. 3-37 Defining units of measurement

Prior to this, the symbolic name (called “Unit of measure” in BPRO3) must be assigned a numerical value (called “Evaluation” in BPRO3). When you select the menu option “Project/Units of measure”, an editor is called which can be used for making this assignment. When entering symbolic names and numerical values, the usual editing functions are available.

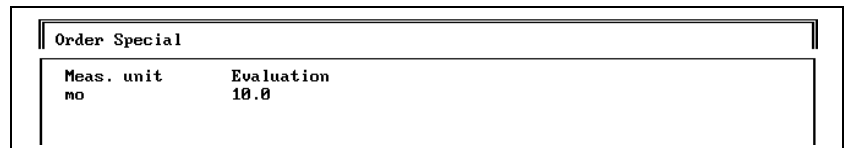


Fig. 3-38 Assigning numerical values

If the symbolic name is entered next to a constant according to figure 3-38, the value of the constant is determined as follows:

Example: 1000 mm

Value of entered constant	x	Unit of measurement	=	Value of constant
1000	x	10.0	=	10000



ATTENTION

If decimal places occur as a result of the multiplication, they are not rounded off, but truncated.

The pull-down menu “Order” provides the block commands “Order/Insert” and “Order/Delete” (see chapter 3.2.3.4). Return to the BPRO3 main menu by selecting the menu option “Special/End”.

3.4.8 Project data output

The project data can be output on a printer or into a file.

“Project/Print project”

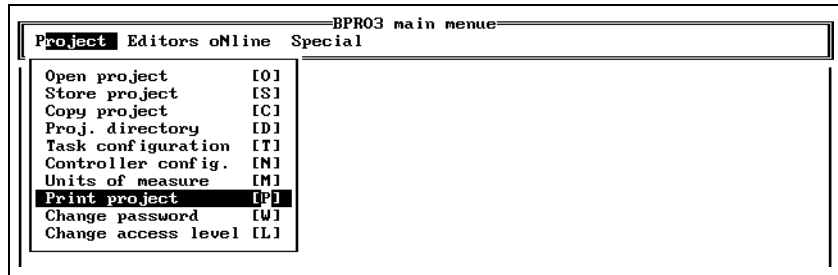


Fig. 3-39 Printing project data

The settings of the various options define the data to be output.

1. Use the cursor keys to move the field cursor to the option to be set and press the <F3> key or select Yes/No.
2. Make the setting using the cursor keys and <↓>.

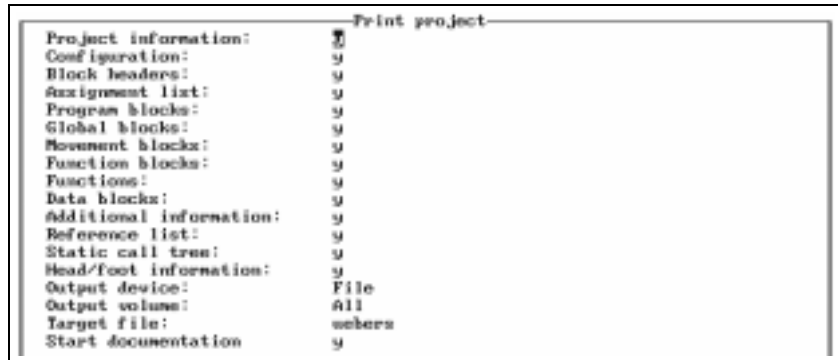


Fig. 3-40 Setting options

3. After having set all options, move the field cursor to “Start documentation” and press <↓>.

A special procedure is valid for the head/foot information: If it is set to “Yes”, and you press the <↓> key, another input window is displayed for entering the three-line foot information (the header is generated by the system and cannot be changed).

Leave the “Print project” screen by pressing <Esc>.

3.4.9 Defining a new password You can use the “Project/Change password” option to define a new password for the current access level.

“Project/Change password”

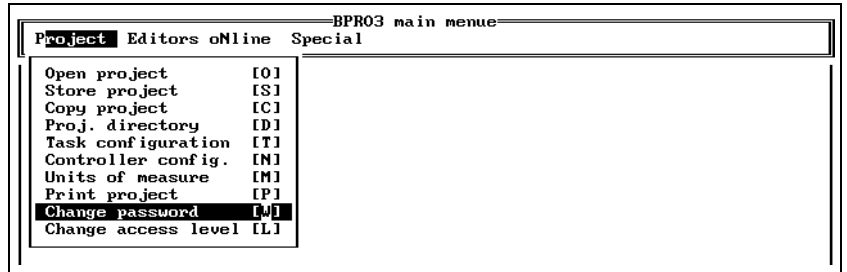


Fig. 3-41 Selecting “Project/Change password”

When you select this menu option, an input field is displayed. The programming system assumes that the password refers to the current access level (see also chapter 3.4.10). The previous password valid for the current access level is overwritten.



Fig. 3-42 Entering a new password

3.4.10 Access protection in BPRO3

“Project/Change access level”

The password protection feature of the programming system (access protection) enables you to specify access permissions for blocks. The access permissions are structured hierarchically into 8 levels. Level 8 has the highest, level 1 the lowest access rights. This means that with access level 8 you have unrestricted access to all blocks created on lower levels. On the other hand, with level 1 you cannot access any block of a higher level. Levels 2 to 8 is assigned a password each (fig. 3-43). Unrestricted access is possible on the lowest level.

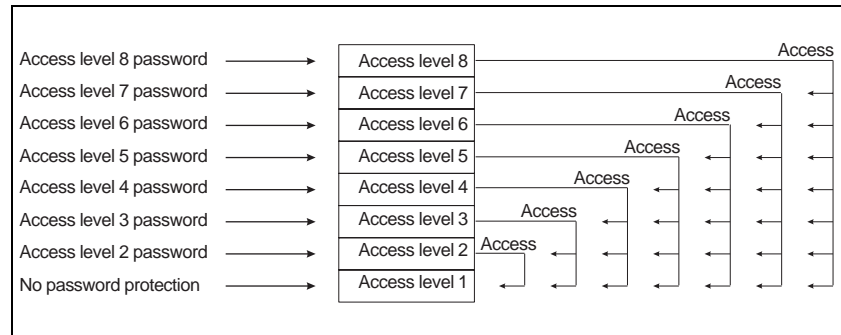


Fig. 3-43 Access levels

The passwords are stored together with the project. Different access rights can be defined for each project.

Effects of access protection

In the following table, the activities which can be restricted by access protection are marked “X”. All other activities are always permitted.

X	Calling an editor for the individual objects of a project.
X	Deleting an object in the programming system and on the controller.
	Downloading an object to the controller.
X	Documenting an object.
	Comparing blocks on the controller and in the programming system.
	Compiling and linking blocks.
	Uploading a block from the controller.
	Programming blocks.

3.4.10.1 Password prompt

The programming system prompts you for a password in the following situations:

- When starting the programming system
- When setting a new access level
- When loading a project
- When copying a project.

When you enter your password, the programming system automatically sets the access level corresponding to your password.



NOTE

When a password has not yet been defined for a project, press the <↓> key in response to the password prompt. This selects the highest access level without password. For a new project, this is level 8, with the highest access permissions.

3.4.10.2 Defining passwords for the individual access levels

A new project is assigned access level 8. Before you can select a different access level, you must first define one or more passwords:

1. Select "Project/Change password" to define the password for the current access level.
2. Select the menu option "Project/Change access level".

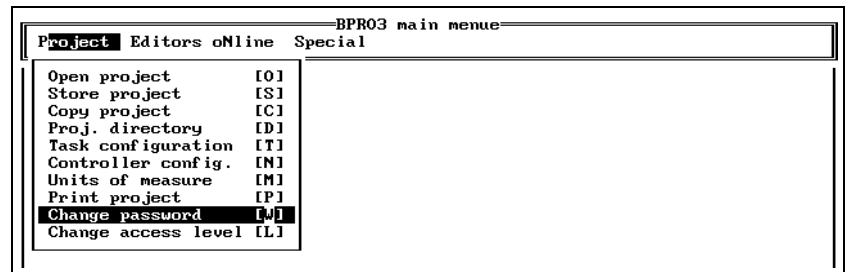


Fig. 3-44 Selecting "Project/Change password"

3. In the password input field, press the <↓> key. BPRO3 sets the highest level without password protection, which is one level below the current level in this case.

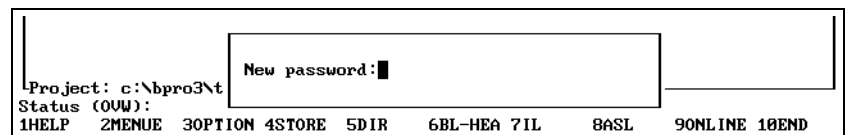


Fig. 3-45 Password input field

4. Repeat steps 1 to 3 until the required number of passwords (max. 7) has been defined.

3.4.10.3 Setting the access level

"Project/Change access level"

1. Select the menu option "Project/Change access level".

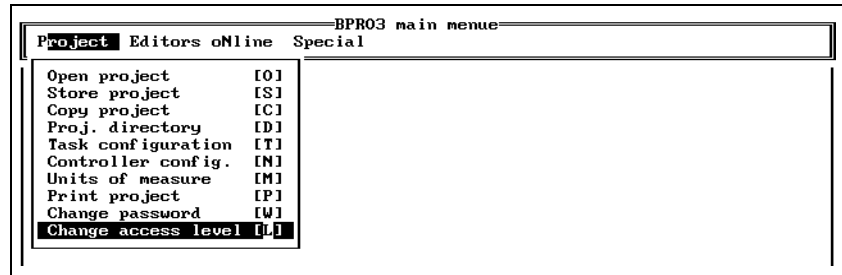


Fig. 3-46 Selecting "Project/Change access level"

2. Enter the password for the desired access level and press the <↵> key. This activates the corresponding access level.

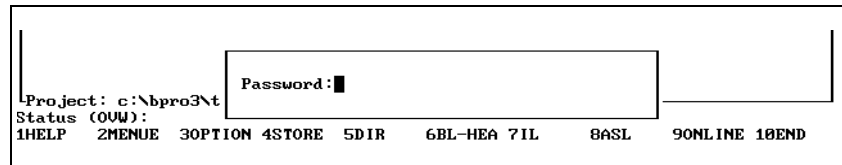


Fig. 3-47 Entering a password



NOTE

If you do not define a password, BPRO3 selects the highest level without password by default.

3.5 The editors

This pull-down menu contains the editors for block programming; see also chapter 4.

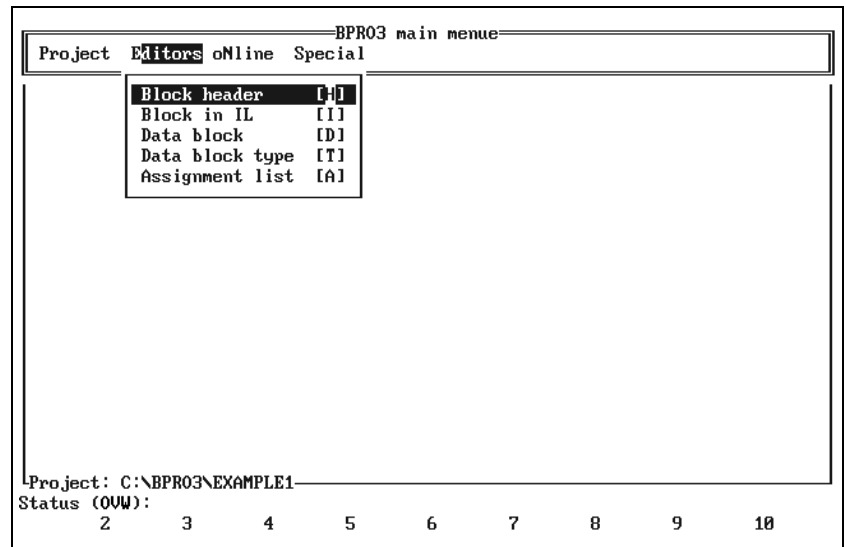


Fig. 3-48 Editors

Menu option	Description	Reference
Block header	Create block header.	See chapter 3.5.1
Block in IL	Create block body using the IL programming language.	See chapter 3.5.2
Data block	Create data block (from data block type).	See chapter 3.5.4
Data block type	Create data block type.	See chapter 3.5.3
Assignment list	Create assignment list.	See chapter 3.5.5

3.5.1 Creating a block header

"Editors/Block header"

With this editor, you can create a block header (for a description of the block header, see chapter 1 in the operating manual and in the programming manual).

Figure 3-49 shows the menu structure of the block header editor. The block header editor is called from the BPRO3 main menu via the pull-down menue "Editors".

If you want to load a block which does not exist yet, you first have to create a block header. The programming system then automatically calls the block header editor.

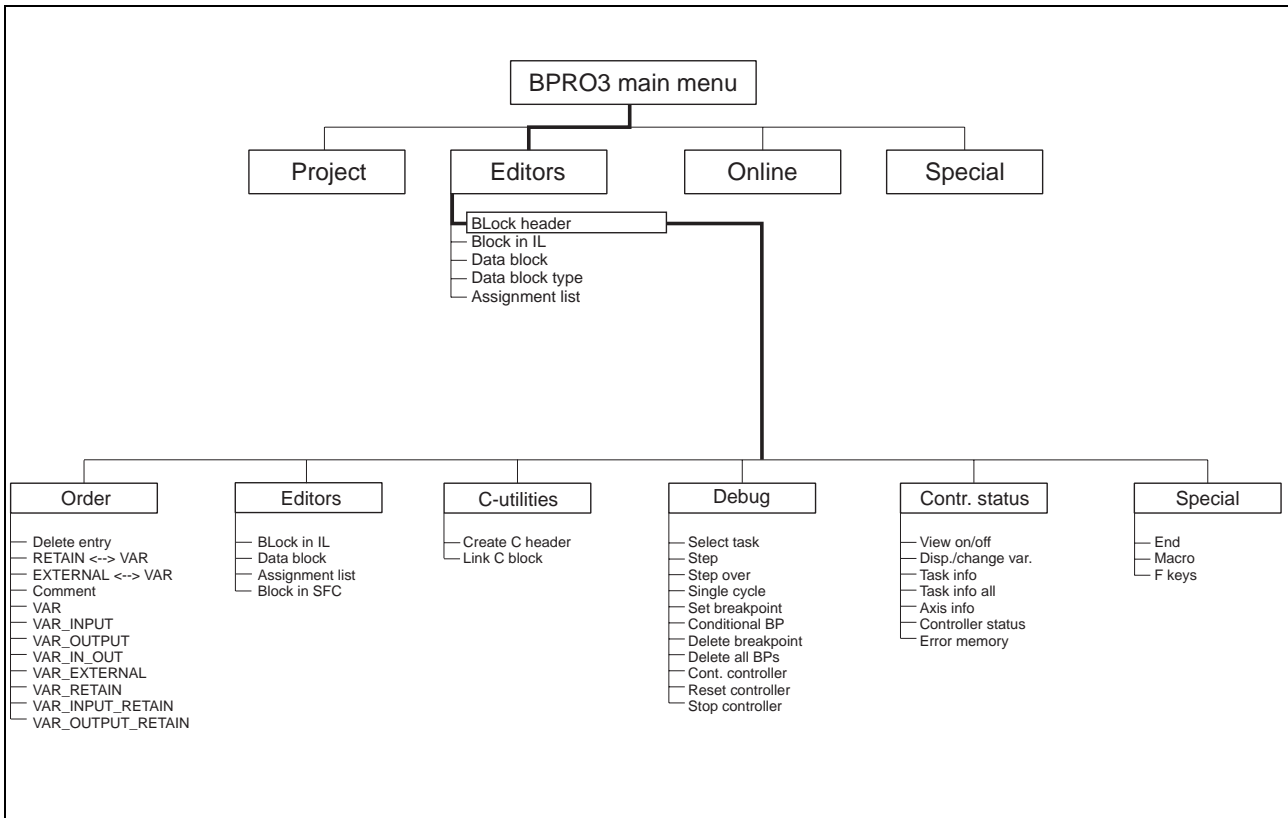


Fig. 3-49 Block header editor menu tree

When calling the block header editor, a block name must be entered.



NOTE

You can use the option list function (<F3>) for selecting an existing block header.

Figure 3-50 shows the block header editor after entering a new block name.

You can now make the entries describing the call interface and other information.

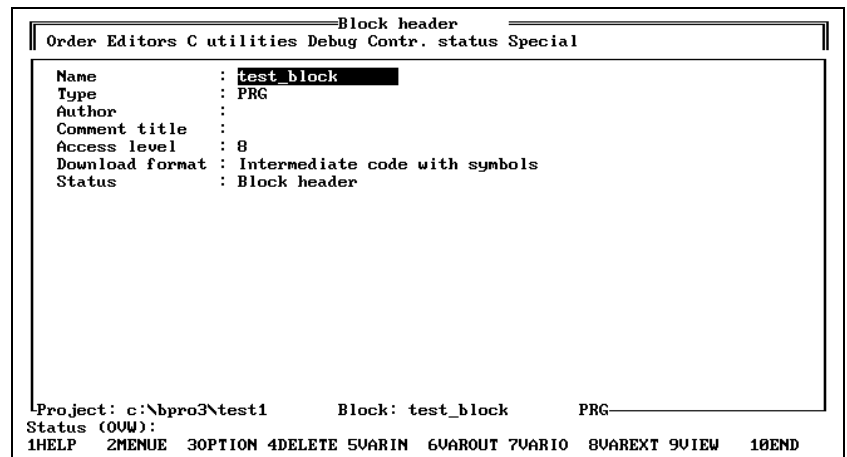


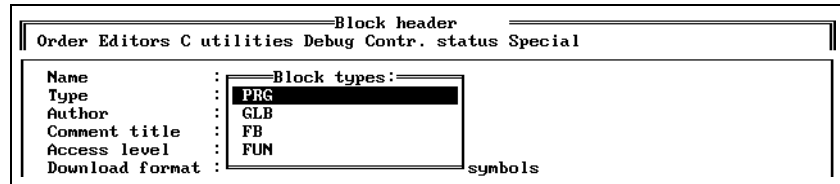
Fig. 3-50 The block header editor

Menu option	Description
Name	Block name
Type	Block type
Author	Name of the programmer. This name can have up to 20 characters.
Comment title	Comment title for the block. A comment title can have up to 40 characters (see also "Comment editor" in chapter 3.2.3.3).
Access level	The access level for the block is displayed. A new block is automatically assigned the access level currently set in the programming system. The access level of the block can be changed at this point, however, you can only set a lower level than the one displayed.
Download format	The download format of the block.
Status	Information on the programming status of the block.

Selecting the block type

1. Move the field cursor to the input field "Type" and press the <F3> key.
2. Select the block type using the cursor keys and <↵>.

Fig. 3-51 Selecting the block type



Block type	Description
PRG	Program block in IL.
GLB	Global block in IL.
FB	Function block in IL.
FUN	Function in IL.



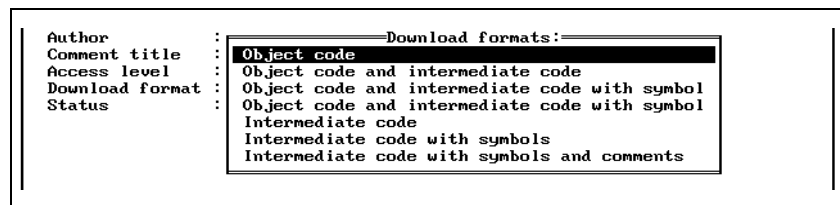
NOTE

The block types are described in the programming manual, chapter 1.

Selecting the download format

1. Move the field cursor to the input field "Download format" and press the <F3> key.
2. Select the download format using the cursor keys and <↵>.

Fig. 3-52 Download formats



Description of the object code:

Object code

Instructions which have been compiled into object code can be executed directly by the controller. Object code instructions are faster in execution than pseudo-code (intermediate code) instructions.

Description of the pseudo-code (intermediate code):

Pseudo-code

Instructions loaded as pseudo-code into the controller can be decompiled into source code (not yet implemented), which means they can be re-edited (this is not possible with plain object code). Pseudo-code instructions must be interpreted before being executed on the controller. To test (debug) a program or block, the instructions must be loaded in pseudo-code.

**NOTE**

To test (debug) a program, the blocks must be loaded into the controller in pseudo-code (not in object code).

Download format	Description
Object code	<p>The block is loaded into the controller in plain object code.</p> <p> NOTE The instructions cannot be decompiled into source code since there is no pseudo-code available.</p>
Object code and intermediate code	The block is loaded into the controller in object and pseudo-code. The controller executes the instructions in object code. The instructions can be decompiled into source code.
Object code and intermediate code with symbols	The block is loaded into the controller in object and pseudo-code. In addition, the pseudo-code contains symbols such as block names, variable names, etc.
Object code and intermediate code with symbols and comments	The block is loaded into the controller in object and pseudo-code. In addition, the pseudo-code contains symbols such as block names, variable names, etc. as well as the comments of the block.
Intermediate code	The block is transferred in plain pseudo-code.
Intermediate code with symbols	The block is transferred only in pseudo-code with symbols such as block names, variable names, etc.
Intermediate code with symbols and comments	The block is transferred only in pseudo-code with symbols such as block names, variable names, etc. as well as the comments of the block.

Information on the programming status of the block

The status information is generated automatically by the system. It can take the following status conditions:

Message	Description
Correct syntax	This is the status of a block when you exit the IL editor and a syntax check is carried out successfully.
Error in block body	The instructions in the block body contain errors.
Unchecked	This status is valid for blocks after having been copied from another project.
Object code (Obj)	This status means that the block has a syntactically correct pseudo-code and object code format.
Block header	The block has not been programmed yet. Only a block header was created.
Compiled	A correctly compiled block.

The output parameter for functions (“VAR_OUTPUT”)

For functions, BPRO3 inserts the additional output parameter VAR_OUTPUT. The type of the output parameter can be modified, if necessary.



NOTE
This is only valid for functions.

```

Block header
Order Editors C utilities Debug Contr. status Special
Name      : FUN_BLOCK_1
Type      : FUN
Author    :
Comment title :
Access level : 8
Download format : Intermediate code with symbols
Status    : Block header
VAR_OUTPUT
FUN_BLOCK_1      INT
VAR_END
Output variable
    
```

Fig. 3-53 Output variable of a function

3.5.1.1 The pull-down menu "Order"

This pull-down menu is used for inserting sections, defining variables and declaring function blocks.

For a more detailed description of the variables, refer to the programming manual, chapter 1.

"Editors/Block header/
Order"

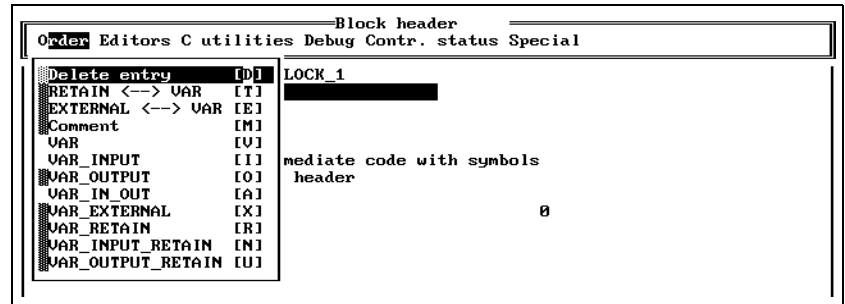


Fig. 3-54 The pull-down menu
"Order"



NOTE

Invalid variable declarations (e.g. VAR_IN_OUT) are marked with the symbol "■".




Menu option	Description
Delete entry	Delete the variable declaration marked by the field cursor.  NOTE <i>If the variables exist in the block body, they cannot be deleted.</i>
RETAIN <—> VAR	Convert RETAIN variable to VAR and vice versa.
EXTERNAL <—> VAR	Convert EXTERNAL variable to VAR and vice versa.
Comment	Call the comment editor. The field cursor must be positioned on the comment title.
VAR	For local variables and for function block declarations.
VAR_INPUT	For local variables to accept a value from the calling block.
VAR_OUTPUT	For output variables to pass a value to the calling block.  NOTE <i>For functions, the output variable (“VAR_OUTPUT”) is automatically entered into the block header.</i>
VAR_IN_OUT	For variables the values of which are accepted from the calling block, modified internally and returned to the calling block.
VAR_EXTERNAL	This section is designed for declaring the global variables to be accessed by the block.  NOTE <i>The variables must have been entered into the assignment list.</i>
VAR_RETAIN	The variables with the “_RETAIN” extension have the same characteristics as the variables of the same name without “_RETAIN” described above. The extension merely means that the variable values are retained after switching off the controller.
VAR_INPUT_RETAIN	
VAR_OUTPUT_RETAIN	

Figure 3-55 shows a section for local retentive variables (VAR_RETAIN). This section is designed for entering the variable name, the variable type and the initial value.

```

Block header
Order Editors C utilities Debug Contr. status Special
Name      : FUN_BLOCK
Type      : PRG
Author    :
Comment title : testblock
Access level : 8
Download format : Intermediate code with symbols
Status    : Block header
VAR_RETAIN
TEST_VAR_1  BOOL      0
VAR_END
    
```

Fig. 3-55 Output variable for functions



NOTE

When defining names you should ensure that a local variable is not assigned the name of a global variable since this would impede access to the global variable.

Defining a variable to be retentive or non-retentive

“Editors/Block header/
Order/RETAIN <—> VAR”

Move the field cursor to the appropriate variable and select the menu option “Order/RETAIN <—> VAR”.

```

Block header
Order Editors C utilities Debug Contr. status Special
Delete entry  IDJ  LOCK
RETAIN <—> VAR  IJI
EXTERNAL <—> VAR  IJL
Comment      IMJ  lock
VAR          IUJ
VAR_INPUT    IJI  mediate code with symbols
VAR_OUTPUT   IOJ  header
VAR_IN_OUT   IAJ
VAR_EXTERNAL IXJ
VAR_RETAIN   IRI  TRUE
VAR_INPUT_RETAIN INI  TRUE
VAR_OUTPUT_RETAIN IUI  TRUE
TEST_VAR_6   BOOL  FALSE
TEST_VAR_7   BOOL  TRUE
VAR_END
    
```

Fig. 3-56 Selecting the RETAIN option

The contents of retentive variables (RETAIN variables) is retained after switching off the controller.

The variable is moved to the appropriate section. The previous state can be restored by selecting “Order/RETAIN <—> VAR” again.

```

VAR
TEST_VAR_1    BOOL  TRUE
TEST_VAR_2    BOOL  TRUE
TEST_VAR_3    BOOL  TRUE
TEST_VAR_4    BOOL  TRUE
TEST_VAR_5    BOOL  FALSE
TEST_VAR_6    BOOL  FALSE
VAR_END
VAR_RETAIN
TEST_VAR_7    BOOL  TRUE
VAR_END
    
```

Fig. 3-57 Effect of RETAIN

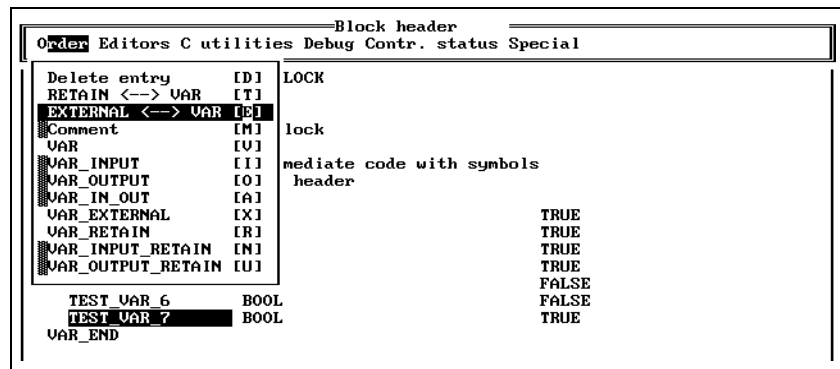
Defining a variable to be external or local

*“Editors/Block header/
Order/EXTERNAL <—> VAR”*

This menu option is used for declaring global variables in a block. It converts a variable of type VAR to EXTERNAL and vice versa.

Proceed as follows:

1. Declare the variable in the VAR section first, for example, TEST_VAR_7.
2. Move the field cursor to TEST_VAR_7 and select the menu option “Order/EXTERNAL <—> VAR”.



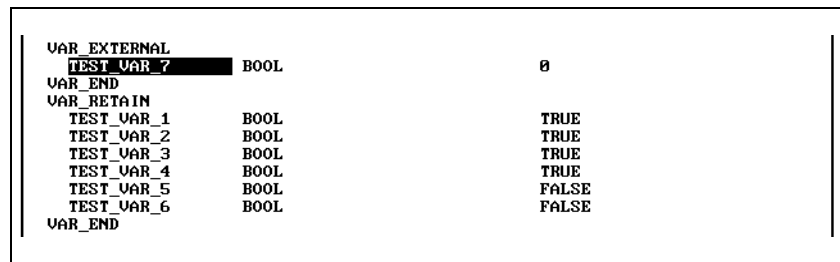
*Fig. 3-58 Selecting
“EXTERNAL <--> VAR”*

The variable is automatically moved to the section of external variables (VAR_EXTERNAL - VAR_END). If the section VAR_EXTERNAL - VAR_END is not yet defined, it will be created.



NOTE

External variables must be entered in the assignment list.



*Fig. 3-59 Variable sections
after selecting “Order/
EXTERNAL <--> VAR”*

Selecting the menu option “Order/EXTERNAL <—> VAR” a second time reverts the variable to a local variable, i.e. the EXTERNAL type is converted to VAR type.

3.5.1.2 Calling another editor

When you select one of the menu options shown in figure 3-60, the programming system calls the corresponding editor.

"Editors/Block header/Editors"



Fig. 3-60 Calling another editor

Menu option	Description
Block in IL	Change to the IL editor.
Data block	Change to the data block editor.
Assignment list	Change to the assignment list editor.

3.5.1.3 "C" program components



NOTE

The functions of the pull-down menu "C utilities" are not implemented in the current release.

"Editors/Block header/C utilities"

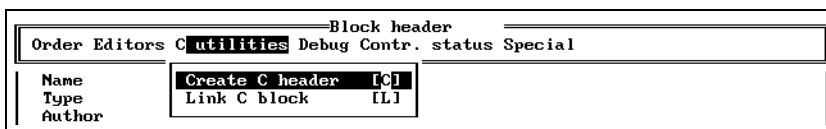


Fig. 3-61 Linking a C program

3.5.1.4 Debugging functions

This pull-down menu offers various options for program testing.

"Editors/Block header/Debug"

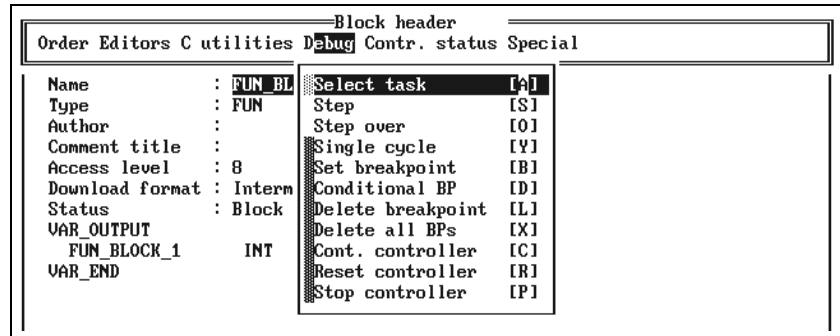


Fig. 3-62 Debugging functions

Menu option	Description
Select task	Select task for Step, Step over, Single cycle, Set breakpoint, Conditional breakpoint, Stop task, Reset task, and Continue task. If you omit this menu option, BPRO3 will prompt you for the name of the task whenever any of the above mentioned options is called.
Step	Process a single IL command line. After program start, or after controller reset, processing begins with the first line of the INIT task. After controller stop, the controller processes the program line following the one at which it was stopped.
Step over	This menu option is the same as the "Step" menu option, with one exception: Block calls are skipped.
Single cycle	Execute a program in single cycles.
Set breakpoint	Set a breakpoint. Program execution stops at the beginning of the block.
Conditional BP	Set a conditional breakpoint.
Delete breakpoint	Delete individual breakpoints.
Delete all BPs	Delete all breakpoints.
Cont. controller	Start the controller.
Reset controller	Reset the controller.
Stop controller	Stop the controller.



NOTE

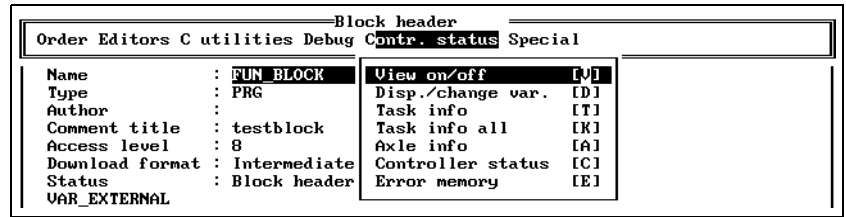
For a more detailed description of this pull-down menu, refer to chapter 3.6.4.

3.5.1.5 The pull-down menu “Contr. status”

“Editors/Block header/
Contr. status”

Fig. 3-63 The pull-down menu “Contr. status”

This pull-down menu is designed for indicating various controller states.



Menu option	Description
View on/off	Read current values from controller and display them.
Disp./change var.	Display and change global variables in the controller.
Task info	Shows the status, the attribute and the block calls of a task.
Task info all	Shows the status, the attribute and the block calls of all tasks.
Axis info	Output information on the current axis status.
Controller status	Output information on the current controller status.
Error memory	Displays the controller error memory.



NOTE

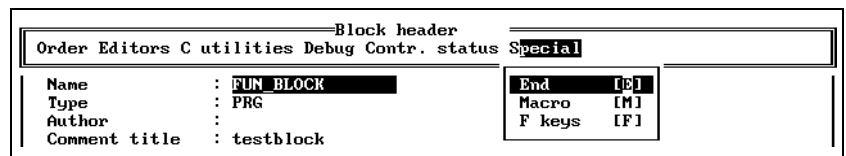
For a more detailed description of this pull-down menu, refer to chapter 3.6.5.

3.5.1.6 The pull-down menu “Special”

“Editors/Block header/Special”

Fig. 3-64 The pull-down menu “Special”

The options of the pull-down menu “Special” are explained in chapter 3.7.



Menu option	Description
End	Exit the BPRO3 main menu.
Macro	Operations with macros.
F keys	Function key assignment.

3.5.2 The instruction list editor (IL editor)

3.5.2.1 Menue structure of IL editor

Figure 3-65 shows the path from the BPRO3 main menu to the IL editor and a summary of the IL editor menus.

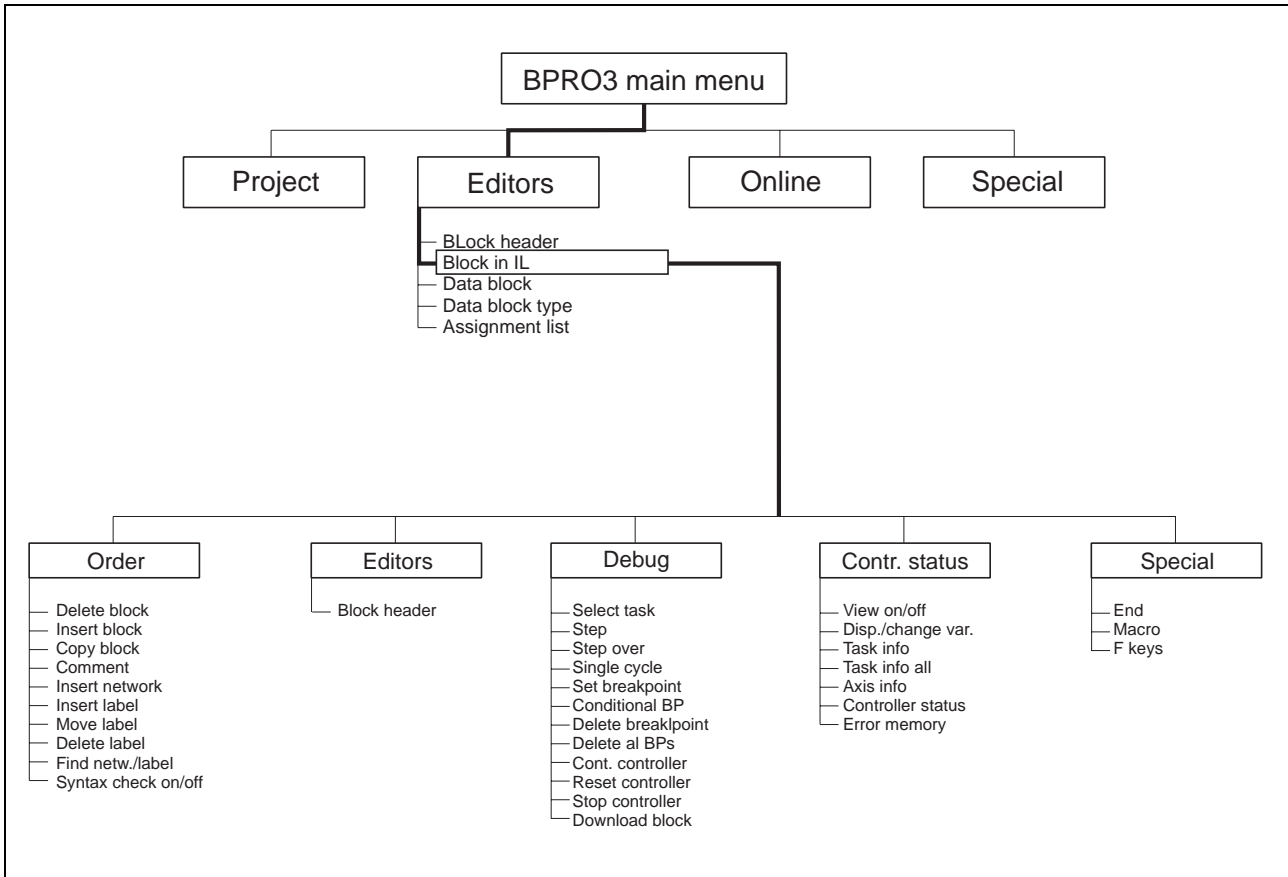


Fig. 3-65 IL editor menue tree

3.5.2.2 Calling the IL editor

The IL editor is the editor for programming in instruction list format. The IL editor is called from the BPRO3 main menu.

“Editors/Block in IL”

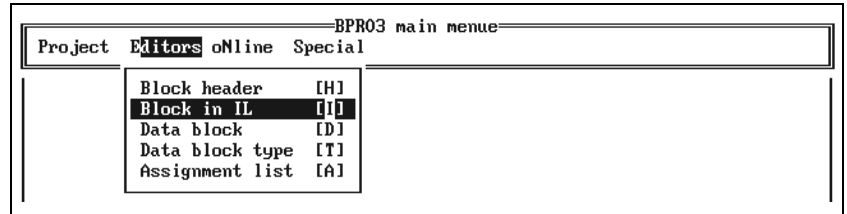


Fig. 3-66 Starting the IL editor

After selecting this menu option you have to enter a block name or select a name using the option list function. If the specified block does not exist, BPRO3 prompts you to confirm that a new block should be created and calls the block header editor (see chapter 3.5.1).

3.5.2.3 User interface of the IL editor

The IL editor is structured according to the screen areas described in chapter 3.2. The menu can be displayed by pressing the <F2> key. In the bottom line of the editing area, the current project name and the name of the block currently loaded into the editor are displayed.

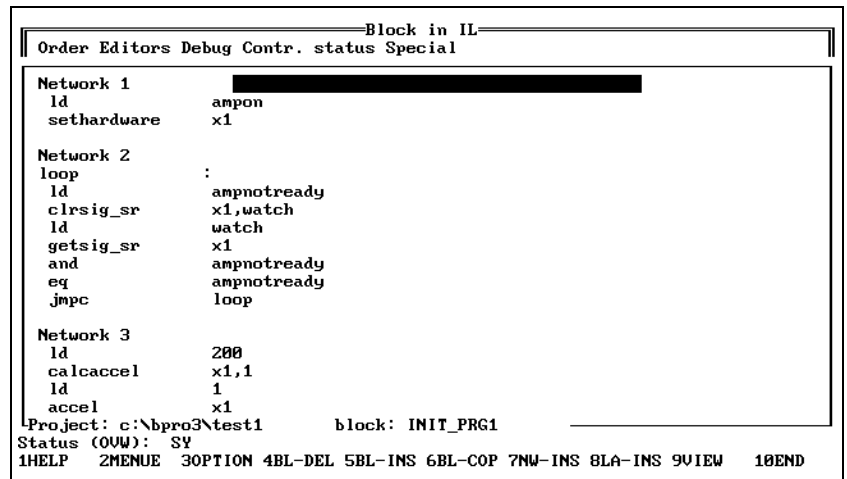


Fig. 3-67 IL editor screen

3.5.2.4 Editing a network



NOTE

The usual editing functions are available (see chapter 3.2.3)

A network begins with the designation "Network" and a number generated automatically by the programming system. To the right, a comment title can be input for the network.

Below this section, the lines are divided into two fields. The left-hand field is provided for the operator, the right-hand field for the operand or the operand list. Note that you first have to enter the operator before entering the operand or operand list.



NOTE

When the field cursor is on the operator field, the option list function (<F3>) can be used to display a list of all operators.

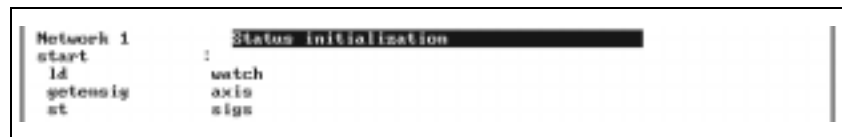


Fig. 3-68 Editing an IL network

When the field cursor is positioned on the operand field and the operator "cal" is contained in the operator field, the option list function (<F3>) can be used to call a list of the available function and global blocks. If a function block or global block was specified, the option list function (<F3>) gives a description of the declared I/O variables of the block.



NOTE

With the "Syntax check" active, a syntactically correct program must be entered (to disable the syntax check, select "Order/Syntax check on/off"). If this is not done, BPRO3 outputs an error message and disables the syntax check in response to a prompt. The menu options are still available.



ATTENTION

There are checks on line level and checks among different lines which are not carried out until the network is exited (type checks: to check the compatibility of the operand types).



NOTE

For guidelines on designing the program code, refer to the programming manual.

3.5.2.5 The pull-down menu “Order”

“Editors/Block in IL/
Order”

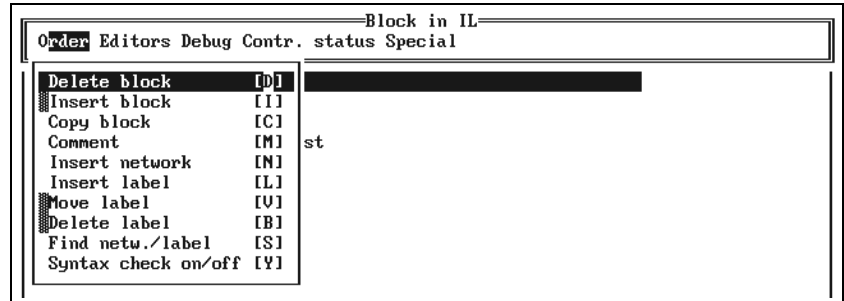




Fig. 3-69 The pull-down menu
“Order”

Menu option	Description
Delete block	Delete a block from the editing area. The block is stored in a file in the directory specified under “Special/Setup/IL export path” (see also chapter 3.7.5).
Insert block	Reinsert the blocks stored by “Delete block” or “Copy block” (see also chapter 3.2.3.4). You can display an option list with the option list function.  NOTE This submenu option can also be used for inserting network lists created with a standard text editor.
Copy block	Copy a block (see also chapter 3.2.3.4).  NOTE When copying networks, labels are not included in the copy.
Comment	Call the comment editor (see also chapter 3.2.3.3). The field cursor must be positioned on the comment title.
Insert network	Insert a new network.
Insert label	Insert a label in the network marked by the field cursor. Labels can also be entered directly.
Move label	Move the label marked by the field cursor.
Delete label	Delete the label marked by the field cursor.
Find netw./label	Find a certain network or label.
Syntax check on/off	Enable/disable the syntax check.

Block commands

For usage, see chapter 3.2.3.4.

**NOTE**

Labels cannot be deleted as long as they are referenced in a jump instruction.

Insert network

*“Editors/Block in IL/
Order/Insert network”*

A new network is inserted in the network list in front of the network marked by the cursor. The numbers of the inserted network and the subsequent networks are dynamically adapted.

**NOTE**

The maximum number of 256 networks must not be exceeded.

Insert label

*“Editors/Block in IL/
Order/Insert label”*

The menu option “Order/Insert label” is designed for inserting labels into the network marked by the cursor. A label is a target point for jump operations from other networks.

**NOTE**

You can also enter a label directly.

Example:

```
Network 1 Both signals active?
ld          MAN_POSITIVE
and         MAN_NEGATIVE
jmpc       STOP
.
.
.
Network 5 Stop movement
STOP:
ld          drive
stop       x1
ret
```

A conditional jump operation is programmed in network 1. If both “MAN_POSITIVE” and “MAN_NEGATIVE” are active, a jump operation is to be performed to the STOP label. At this point, the movement is stopped.

Insert a label as follows:

1. Move the field cursor to the appropriate network and select the menu option "Order/Insert label".

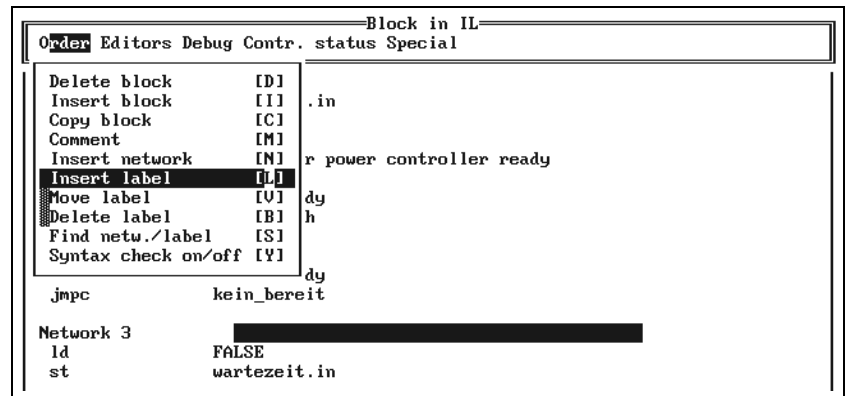


Fig. 3-70 Inserting a label

2. Enter the name of the label.



Fig. 3-71 Entering a label name

"Editors/Block in IL/
Order/Move label"

Moving a label

1. Move the field cursor to the label to be moved and select the menu option "Order/Move label".

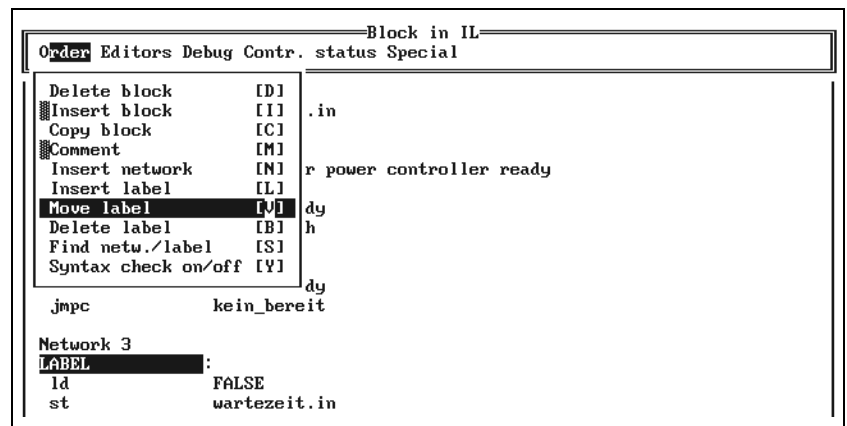
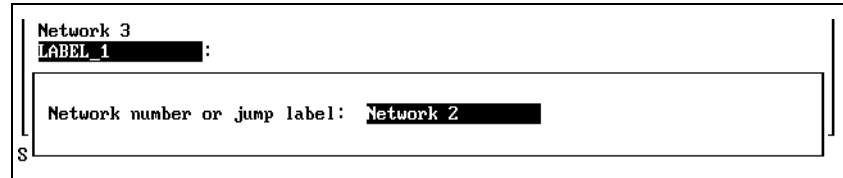


Fig. 3-72 Moving a label

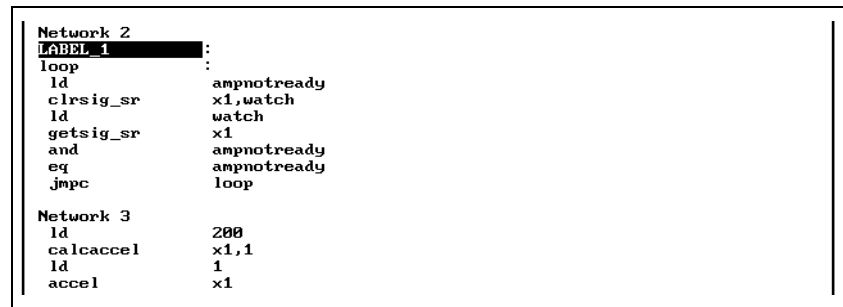
2. Enter the network number, e.g. "Network 2", or the label and press the <↵> key.

Fig. 3-73 Entering a network number or label



BPRO3 moves the label from network 4 to network 2.

Fig. 3-74 Label moved



"Editors/Block in IL/
Order/Delete label"

Deleting a label

Delete the label marked by the field cursor.



NOTE

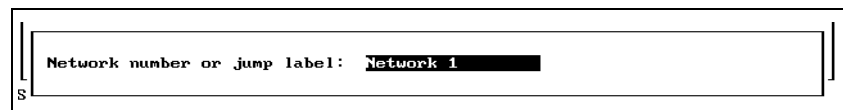
All jump instructions from other networks to the specified label must be removed previously.

"Editors/Block in IL/
Order/Find netw./label"

Finding a label or network

1. Select the menu option "Order/Find netw./label".
2. Enter a network number or label. Default is network 1.

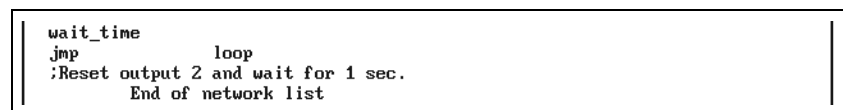
Fig. 3-75 Finding a network or label



Comment lines

A semicolon as the first character in the operator field marks a comment line.

Fig. 3-76 Comment lines



Enabling/disabling syntax checking

*“Editors/Block in IL/
Order/Syntax check on/off”*

This menu option can be used to enable or disable the syntax check. The status line indicates the current setting. When the syntax check is enabled, the entered program text is checked for correct syntax.

Status line	Description
SY-FREE	Syntax check is disabled.
SY	Syntax check is enabled.

When the syntax check is enabled, a syntax check is performed with the following actions:

- When leaving a field
- When leaving a line
- When importing blocks

The data entered are checked for:

- Correct operators (see programming manual)
- Correct operands or operand lists (see programming manual)
- When editing, a type check is carried out with the operands or operand lists upon leaving the network.



NOTE

With the syntax check disabled you can write programs almost without restriction. A few checks, however, are still carried out. For example:

- *Check for correct symbolic names.*
- *Check for references to other networks when deleting labels. A label cannot be deleted if it is still referenced in other networks.*



NOTE

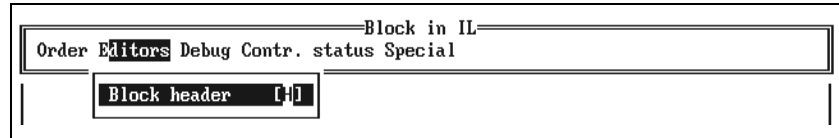
When the syntax check is disabled, the block cannot be loaded into the controller. The block must therefore be checked first by enabling the syntax check.

3.5.2.6 Changing directly to another editor

This menu option can be used for changing directly from the IL editor to another editor.

“Editors/Block in IL/Editors”

Fig. 3-77 The pull-down menu “Editors”



Changing to the block header editor

“Editors/Block in IL/Editors/Block header”

You can use “Editors/Block header” to change directly to the block header editor.



NOTE

When the syntax check is enabled, a syntax check is carried out upon leaving the IL editor.

3.5.2.7 Debugging options

“Editors/Block in IL/
Debug”

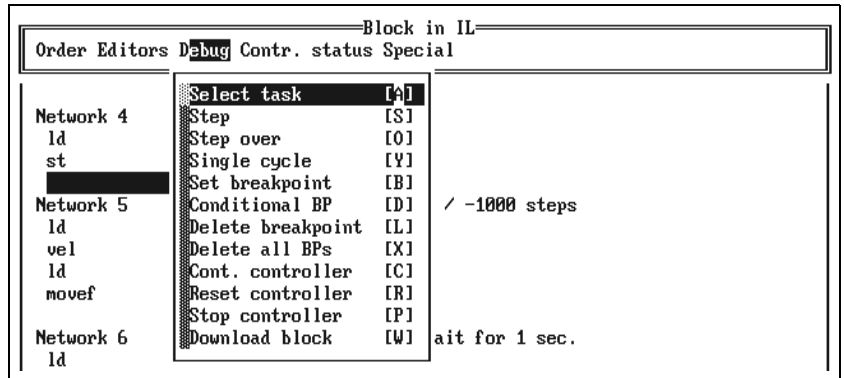


Fig. 3-78 The pull-down menu “Debug”

Menu option	Description
Select task	Select task for Step, Step over, Single cycle, Set breakpoint, Conditional breakpoint, Stop task, Reset task, and Continue task. If you omit this menu option, BPRO3 will prompt you for the name of the task whenever any of the above mentioned options is called.
Step	Process a single IL command line. After program start, or after controller reset, processing begins with the first line of the INIT task. After controller stop, the controller processes the program line following the one at which it was stopped.
Step over	This menu option is the same as the “Step” menu option, with one exception: Block calls are skipped.
Single cycle	Execute a program in single cycles.
Set breakpoint	Set a breakpoint in a program line.
Conditional BP	Set a conditional breakpoint.
Delete breakpoint	Delete individual breakpoints.
Delete all BPs	Delete all breakpoints.
Cont. controller	Start the controller.
Reset controller	Reset the controller.
Stop controller	Stop the controller.
Download block	Load a block into the controller.
	<p>NOTE The controller must have stop status.</p>



NOTE

The remaining menu options are the same as with the “Online” menu described in chapter 3.6.4.

3.5.2.8 The pull-down menue “Contr. status”

This pull-down menue is designed for indicating various controller states.

“Editors/Block in IL/
Contr. status”

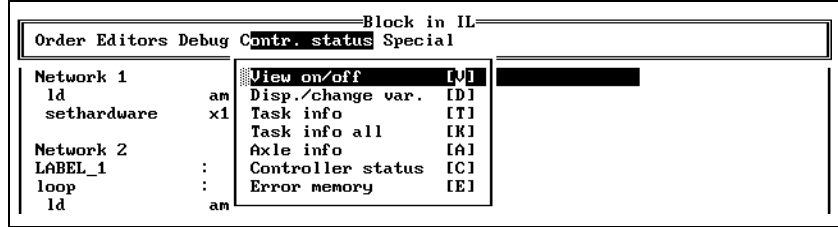


Fig. 3-79 The pull-down menue “Contr. status”

Menue option	Description
View on/off	Display the CR contents during program execution.
Disp./change var.	Display and change global variables in the controller.
Task info	Shows the status, the attribute and the block calls of a task.
Task info all	Shows the status, the attribute and the block calls of all tasks.
Axis info	Output information on the current axis status.
Controller status	Output information on the current controller status.
Error memory	Displays the controller error memory.



NOTE

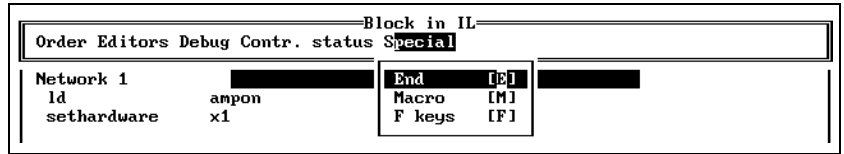
For a more detailed description of this pull-down menue, refer to chapter 3.6.5.

3.5.2.9 The pull-down menu "Special"

The options of the pull-down menu "Special" are explained in chapter 3.7.

*"Editors/Block in IL/
Special"*

Fig. 3-80 The pull-down menu "Special"



Menu option	Description
End	Exit the IL main menu.
Macro	Operations with macros.
F keys	Function key assignment.



NOTE

When exiting the IL editor, the syntax of the block is only checked if the syntax check is enabled.

3.5.3 The data block type editor The data block type editor is designed for creating data block structures and defining the variable types occurring in them. Limit values and initial values can be specified for the variables.

Figure 3-81 shows the menu structure of the data block type editor.

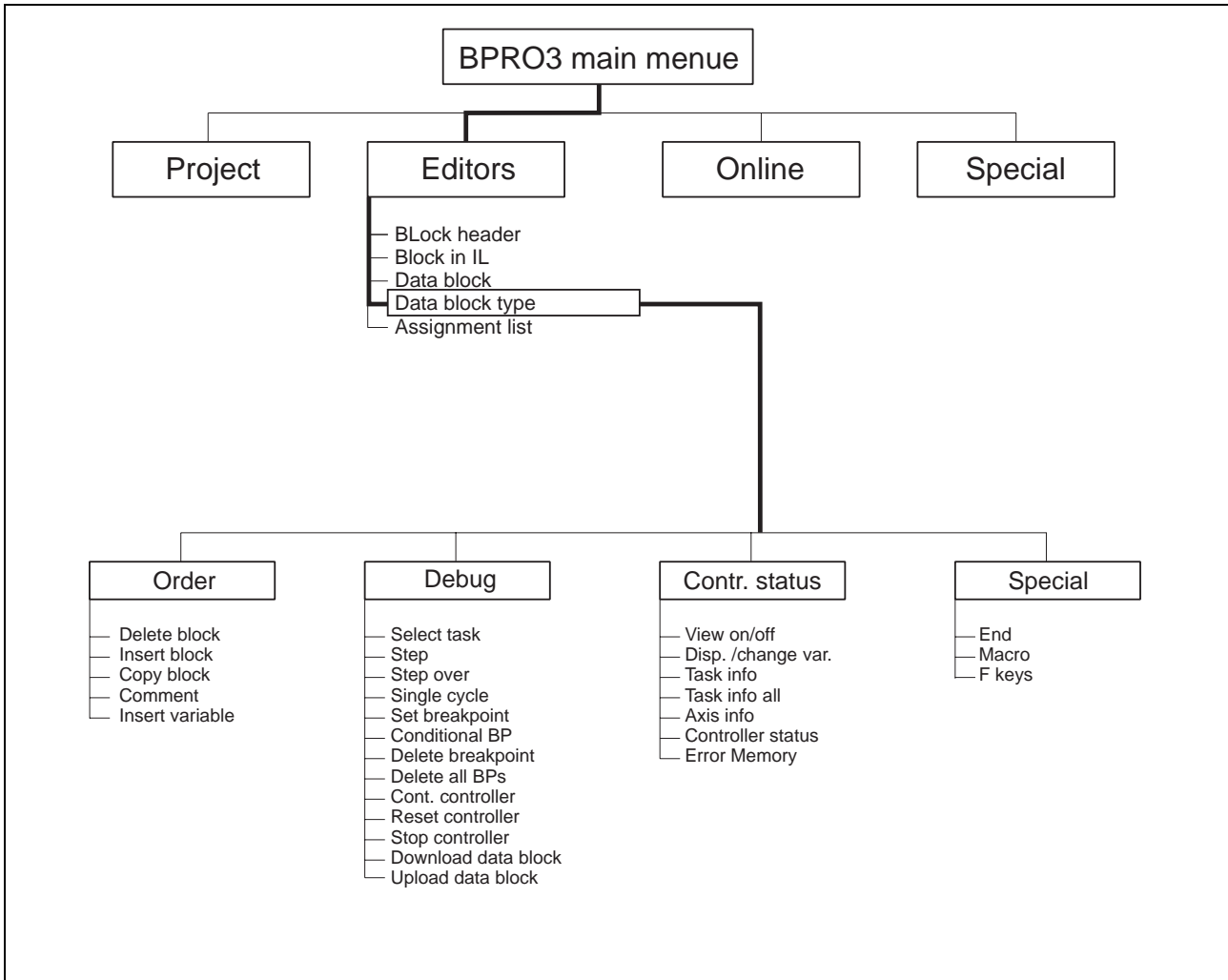


Fig. 3-81 DB type editor menu tree

The structures created with the data block type editor can be used for creating multiple data blocks (see data block editor, chapter 3.5.4).

When creating a data block type, you merely define the structure for data blocks to be created using the data block editor. The data block editor allocates space for the variables in the data block.

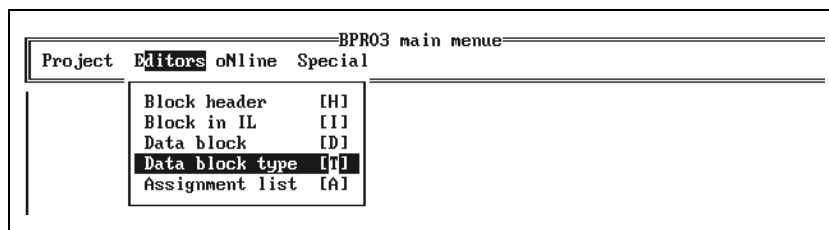


NOTE

A maximum of 256 data block types (data block structures) can be defined within one project.

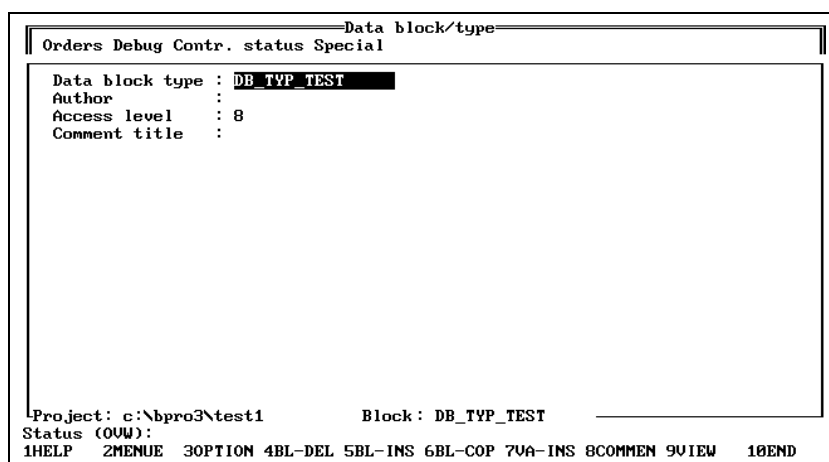
The data block type editor is called from the BPRO3 main menu via the pull-down menu "Editors". The program prompts you for the name of the data block type to be selected. If you want to create a new data block type, the program prompts you to confirm this.

Fig. 3-82 Calling the data block type editor



When the data block type editor is called and a new name entered, the editor appears on the screen as illustrated in figure 3-83.

Fig. 3-83 The DB type editor



NOTE

In the DB type editor, the editing functions described in chapter 3.2.3 are available.

The following information is displayed in the upper section of the editing area:

Designation	Description
Data block type	Indicates the name of the block type.
Author	A 20-character field for the programmer's (author's) name.
Access level	The access level for this data block type is displayed. A new block is automatically assigned the access level currently set in the programming system. The access level of the data block type can be changed at this point, however, only a lower level than the one displayed can be set.
Comment title	A 40-character input field for the title of the comment assigned to the data block type (see also "Comment editor" in chapter 3.2.3.3).

3.5.3.1 Miscellaneous commands The pull-down menu "Orders" provides various block commands and options for inserting comment titles or comments for variables as well as inserting a section for defining the data structure.

"Editors/Data block type/
Orders"

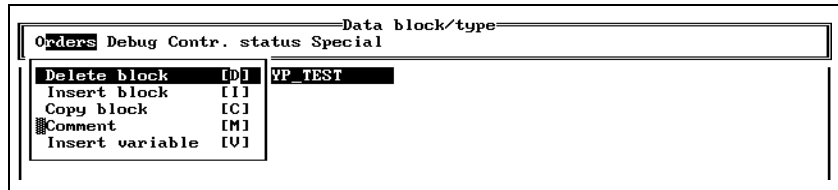


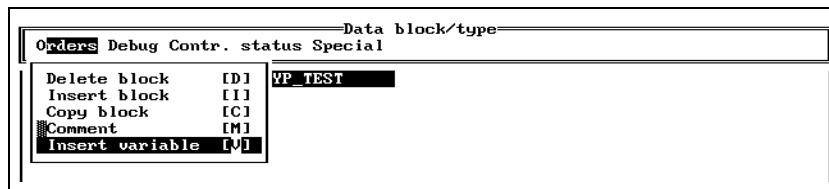
Fig. 3-84 The pull-down menu "Orders"

Menu option	Description
Delete block	Delete a block (see chapter 3.2.3.4).
Insert block	Insert a block (see chapter 3.2.3.4).
Copy block	Copy a block (see chapter 3.2.3.4).
Comment	Insert a comment for the variable marked by the field cursor. If the field cursor is positioned on a variable which already has a comment title, BPRO3 calls the comment editor (see also chapter 3.2.3.3).
Insert variable	Insert a section for defining the data structure.

The menu option “Insert variable”

*“Editors/Data block type/
Orders/Insert variable”*

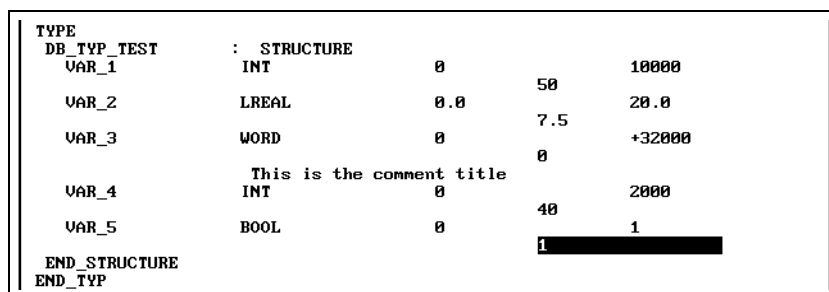
Before entering a data structure, the menu option “Orders/Insert variable” must be used for defining a section for it.



*Fig. 3-85 Selecting
“Orders/Insert variable”*

Enter the structure into the five input fields provided.

1. Enter the name of the variable into the first field, using a maximum of 16 characters.
2. Enter the type of the variable into the second field, or select it using the option list function.
3. Specify the minimum limit value for the variable in the third field.
4. Specify the maximum limit value for the variable in the fourth field.
5. The fifth field is provided for entering an optional initial value for the variable. Whenever a data block is created from the DB type defined here, the variable is automatically assigned this initial value. The initial value must be within the range defined by the limit values.



*Fig. 3-86 Data structure
definition section*

If you select the menu option “Orders/Insert variable” again, BPRO3 inserts an empty line for the next variable.

3.5.3.2 Online functions

This pull-down menu offers various options for program testing (debugging).

“Editors/Data block type/
Debug”

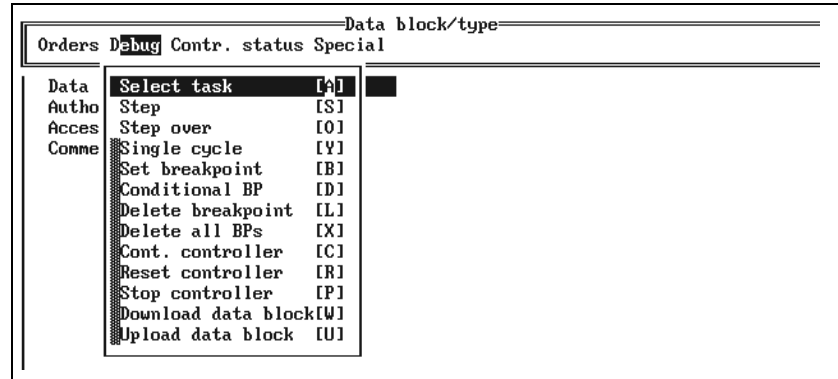


Fig. 3-87 The pull-down menu “Debug”

Menu option	Description
Select task	Select task for Step, Step over, Single cycle, Set breakpoint, Conditional breakpoint. If you omit this menu option, BPRO3 will prompt you for the name of the task whenever any of the above mentioned options is called.
Step	Process a single IL command line. After program start, or after controller reset, processing begins with the first line of the INIT task. After controller stop, the controller processes the program line following the one at which it was stopped.
Step over	This menu option is the same as the “Step” menu option, with one exception: Block calls are skipped.
Single cycle	Execute a program in single cycles.
Set breakpoint	Set a breakpoint. Program execution stops at the beginning of the block.
Conditional BP	Set a conditional breakpoint.
Delete breakpoint	Delete individual breakpoints.
Delete all BPs	Delete all breakpoints.
Cont. controller	Start the controller.
Reset controller	Reset the controller.
Stop controller	Stop the controller.
Download data block	Load a data block into the controller.
Upload data block	Load a data block from the controller into the computer.

“Editors/Data block type/
Debug/Step”

Executing single steps with “Debug/Step”

“Debug/Step” allows you to execute a program in single steps, i.e. program execution stops after each instruction. It is recommended to enable “viewing” for this function so that BPRO3 reads the current variable values from the controller and displays them on the screen.



ATTENTION

By default, outputs and axes are reset automatically at program stop. This feature can be disabled with the “Debug” option (see programming manual). However, the following effects must always be taken into account:

Outputs and axes are no longer controlled by the controller after a program stop. This means that, for example, motors continue to run without control, which may result in injury or damage!

“Editors/Data block type/
Debug/Step over”

Executing single steps with “Debug/Step over”

This option is identical with the menu option “Debug/Step”, with one exception: Block calls with the operator “cal” are skipped.



ATTENTION

By default, outputs and axes are reset automatically at program stop. This feature can be disabled with the “Debug” option (see programming manual). However, the following effects must always be taken into account:

Outputs and axes are no longer controlled by the controller after a program stop. This means that, for example, motors continue to run without control, which may result in injury or damage!



NOTE

The other menu options are described in the pull-down menu “Debug” in chapter 3.6.4.

3.5.3.3 The pull-down menu "Contr. status"

This pull-down menu is designed for indicating various controller states.

"Editors/Data block type/Contr. status"

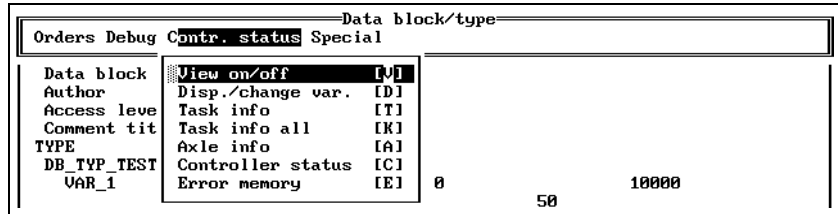


Fig. 3-88 The pull-down menu "Contr. status"

Menu option	Description
View on/off	Read current values from controller and display them.
Disp./change var.	Display and change data from the process image, direct I/O, flags and variables of the data blocks.
Task info	Shows the status, the attribute and the block calls of a task.
Task info all	Shows the status, the attribute and the block calls of all tasks.
Axis info	Output information on the current axis status.
Controller status	Output information on the current controller status.
Error memory	Displays the controller error memory.



NOTE

For a more detailed description of the pull-down menu "Contr. status", refer to chapter 3.6.5.

3.5.3.4 The pull-down menu "Special"

The options of the pull-down menu "Special" are explained in chapter 3.7.

"Editors/Data block type/Special"

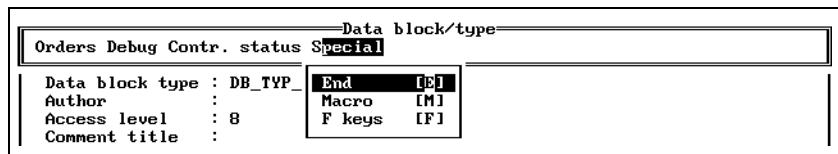


Fig. 3-89 The pull-down menu "Special"

Menu option	Description
End	Exit the data block type editor menu.
Macro	Operations with macros.
F keys	Function key assignment.

3.5.4 The data block editor

The data block editor (DB editor) is designed for creating data blocks, the structure of which has been defined with the DB type editor. The initial values and comments of the DB type are included in the data blocks. The initial values may be changed with the DB editor. Figure 3-90 shows the menu structure of the DB editor.



NOTE

The structure of the data blocks must have been created previously with the DB type editor.

You have to enter a data block name. If the name does not yet exist, you are prompted to confirm creating a new block.



NOTE

A maximum of 256 data blocks of a data block structure can be created.

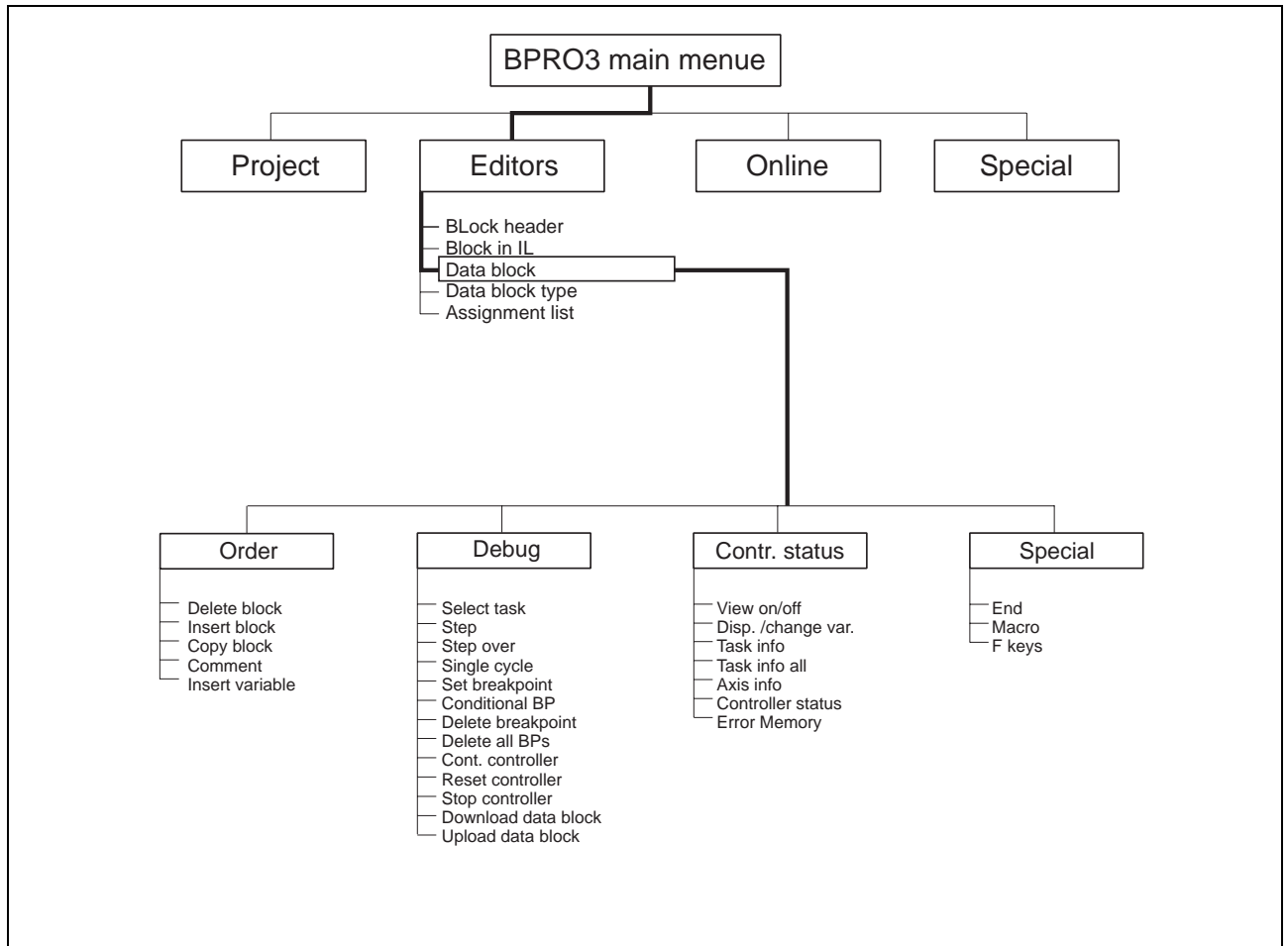


Fig. 3-90 The DB editor



NOTE

In the DB editor, the editing functions described in chapter 3.2.3 are available.

The DB editor is called from the pull-down menu "Editors/Data block". When you call the editor, you have to enter the name of the desired data block structure and the name of the data block.

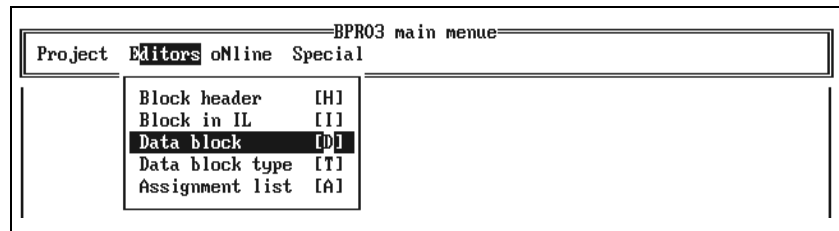


Fig. 3-91 Calling the data block editor

Figure 3-92 is an example of a DB editor screen.

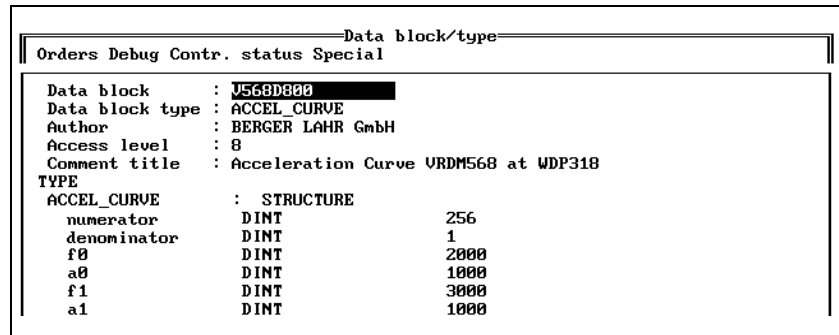


Fig. 3-92 Calling the data block editor

The following information is displayed in the upper section of the editing area:

Menu option	Description
Data block	Name of the data block.
Data block type	Indicates the name of the block type.
Author	A 20-character field for the programmer's (author's) name.
Access level	The access level for this data block is displayed. A new block is automatically assigned the access level currently set in the programming system. The access level of the data block can be changed at this point, however, only a lower level than the one displayed can be set.
Comment title	A 40-character input field for the title of the comment assigned to the data block (see also "Comment editor" in chapter 3.2.3.3).

The DB editor menu

The menu of the DB editor is similar to the menu of the DB type editor, with a few restrictions.



NOTE

The structure of the data block cannot be changed.

All other menu options are identical to those of the DB type editor; see chapter 3.5.3.

Variables

Variables are displayed in the same way as in the DB type editor, except that the limit values are not shown. There are three columns for name, type and initial value. You can change the initial values, if necessary. Figures 3-93 and 3-94 allow you to compare the display in the DB type editor and the DB editor, respectively.

```

TYPE
DB_TYP_TEST      : STRUCTURE
  VAR_1          INT           0           10000
                 50
  VAR_2          LREAL        0.0         20.0
                 7.5
  VAR_3          WORD         0           +32000
                 0
                 This is the comment title
  VAR_4          INT           0           2000
                 40
  VAR_5          BOOL         0           1
                 1
END_STRUCTURE
END_TYP
    
```

Fig. 3-93 DB type editor display

```

-----Data block/type-----
Orders Debug Contr. status Special
Data block type : ACCEL_CURVE
Author          : BERGER LAHR GmbH
Access level    : 8
Comment title   : Acceleration Curve
TYPE
ACCEL_CURVE    : STRUCTURE
  numerator     DINT          1           10000
                 256
  denominator   DINT          1           10000
                 1
  f0            DINT          1           500000
                 2000
    
```

Fig. 3-94 DB editor display

3.5.5 The assignment list editor The assignment list editor is used for assigning symbolic names to flags, inputs and outputs. These symbolic names can be used in a program in place of hardly intelligible identifiers such as %MX0.0 or %IB0.1. These symbolic names can be used in a block without VAR_EXTERNAL.

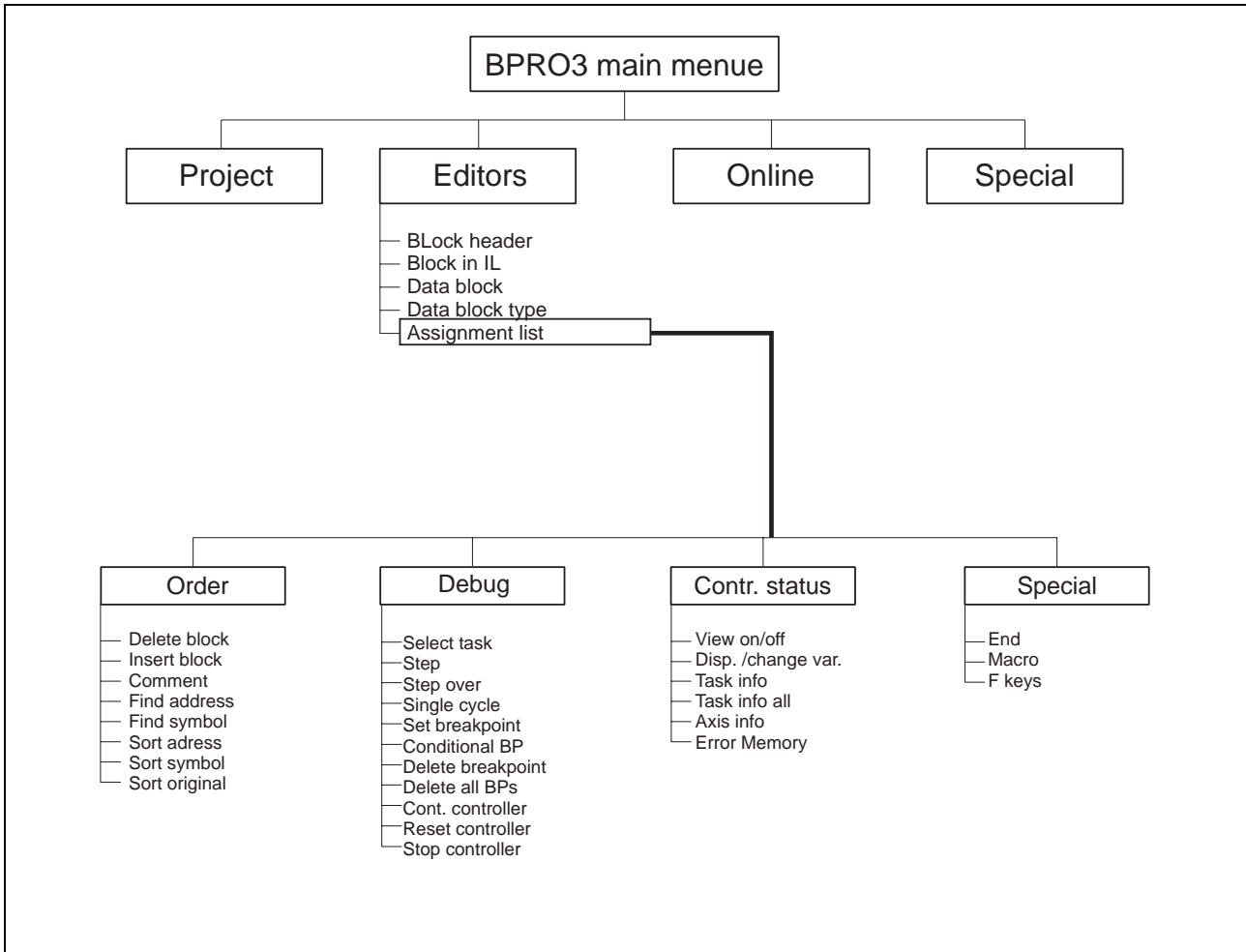


Fig. 3-95 AL editor menu tree

The AL editor consists of the sections described in chapter 1. The AL editor definition lines comprise four fields.

- The first field is provided for entering the variable name.
- The second field specifies the logical address of the variable.
- The third field specifies the type of the variable.
- The fourth field can be used for entering an optional initial value.

You can enter a comment title below these four fields.

```

Assignment list
Order Debug Contr. status Special
TEST VAR ?      %MX0.1      BOOL      0
start_input     %IX0.0      WORD      0
stop_input      %IX0.1      BOOL      0
length          %MW0       DINT      2000
frequency       %MW2       DINT      5000
x1              %MW200    WORD      16#7000
p1              %MW201    WORD      16#7000
p2              %MW202    WORD      16#7001
c2              %MW203    WORD      16#6301
plc             %MW204    WORD      16#1000
keys            %MW205    WORD      16#1100
batt           %MW206    WORD      16#1200
mathe          %MW207    WORD      16#1300
drive          %MW208    WORD      16#0000
velocity       %MW209    WORD      16#0002
position       %MW210    WORD      16#0003
actual         %MW211    WORD      16#1000
target         %MW212    WORD      16#2000
cont           %MW213    WORD      16#4000
Project: c:\bpro3\test1
Status (OUW): Number of symbols entered is : 153
1HELP 2MENU 3OPTION 4BL-DEL 5BL-INS 6A-SORT 7S-SORT 8O-SORT 9VIEW 10END

```

Fig. 3-96 The assignment list editor



ATTENTION

Changes to variables used in a project are inadmissible except in one case:

Converting the data type to a different but compatible data type, e.g. INT ⇒ DINT.

3.5.5.1 Miscellaneous commands

"Editors/Assignment list/Order"

The pull-down menu "Order" provides various commands.

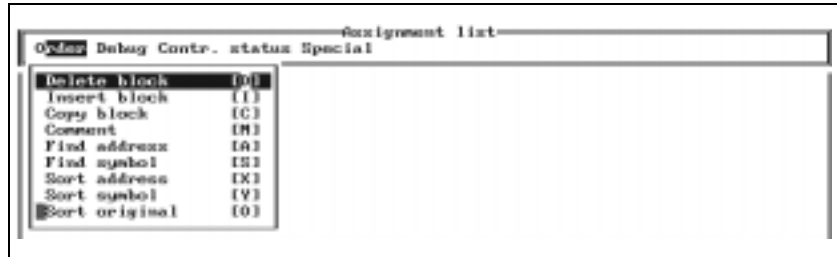




Fig. 3-97 The pull-down menu "Order"

Menu option	Description
Delete block	Delete a block (see chapter 3.2.3.4).
Insert block	Insert a block (see chapter 3.2.3.4).
Copy block	Copy a block (see chapter 3.2.3.4).
Comment	Insert a comment for the assignment marked by the field cursor. If the field cursor is positioned on an assignment which already has a comment title, BPRO3 calls the comment editor (see also chapter 3.2.3.3).
Find address	Find a specified address.
Find symbol	Find a specified symbol.
Sort address	Sort the assignment list by address.  NOTE To edit the assignment list, you have to restore the original sort order using the menu Option "Sort original".
Sort symbol	Sort the assignment list by symbol.  NOTE To edit the assignment list, you have to restore the original sort order using the menu option "Sort original".
Sort original	The assignment list is sorted in the order in which its items were originally entered (default).

Finding an address

*“Editors/Assignment list/
Order/Find address”*

1. Select the menu option “Order/Find address”.



Fig. 3-98 Selecting “Order/Find address”

2. Enter the required address and press the <↵> key. BPRO3 then searches the list for the specified address and displays it. If the address is not found in the assignment list, an error message is output.

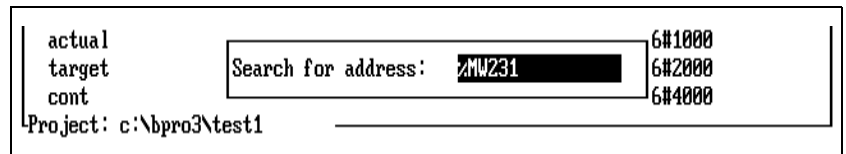


Fig. 3-99 Entering an address

Finding a symbol

*“Editors/Assignment list/
Order/Find symbol”*

1. Select the menu option “Order/Find symbol”.

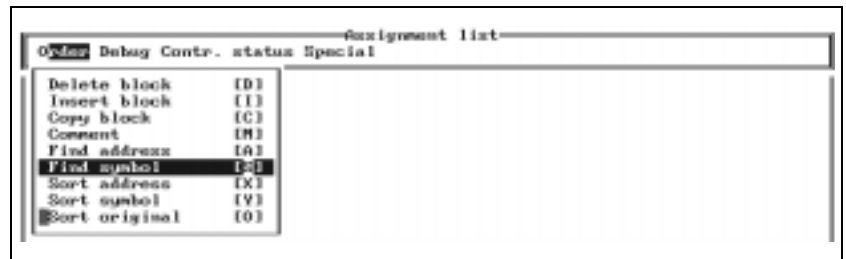


Fig. 3-100 Selecting “Order/Find symbol”

2. Enter the required symbol and press the <↵> key. BPRO3 then searches the list for the specified symbol and displays it. If the symbol is not found in the assignment list, an error message is output.

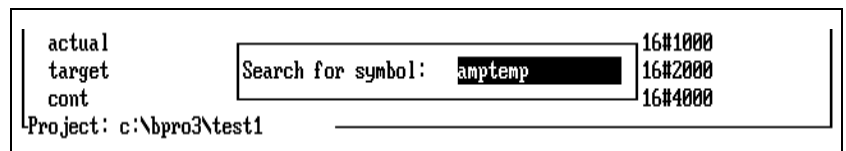


Fig. 3-101 Entering a symbol

3.5.5.2 The pull-down menu "Debug"

This pull-down menu offers various options for program testing (debugging).

"Editors/Assignment list/Debug"

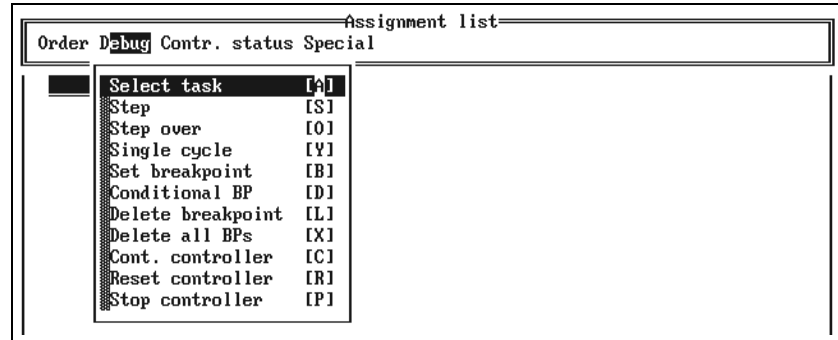


Fig. 3-102 The pull-down menu "Debug"

Menu option	Description
Select task	Select task for Step, Step over, Single cycle, Set breakpoint, Conditional breakpoint, Stop task, Reset task, and Continue task. If you omit this menu option, BPRO3 will prompt you for the name of the task whenever any of the above mentioned options is called.
Step	Process a single IL command line. After program start, or after controller reset, processing begins with the first line of the INIT task. After controller stop, the controller processes the program line following the one at which it was stopped.
Step over	This menu option is the same as the "Step" menu option, with one exception: Block calls are skipped.
Single cycle	Execute a program in single cycles.
Set breakpoint	Set a breakpoint. Program execution stops at the beginning of the block.
Conditional BP	Set a conditional breakpoint.
Delete breakpoint	Delete individual breakpoints.
Delete all BPs	Delete all breakpoints.
Cont. controller	Start the controller.
Reset controller	Reset the controller.
Stop controller	Stop the controller.



NOTE

For a more detailed description of the pull-down menu "Debug", refer to chapter 3.6.4.

3.5.5.3 The pull-down menu “Contr. status”

This pull-down menu is designed for indicating various controller states.

“Editors/Assignment list/
Contr. status”

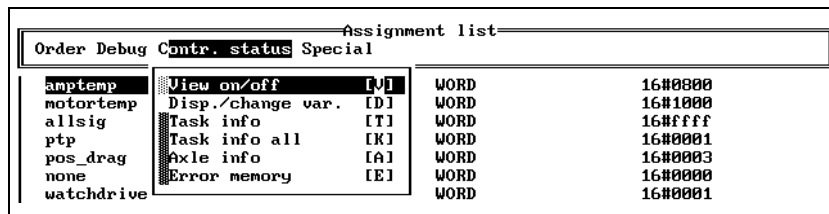


Fig. 3-103 The pull-down menu “Contr. status”

Menu option	Description
View on/off	Read current values from controller and display them.
Disp./change var.	Display and change data from the process image, direct I/O, flags and variables of the data blocks.
Task info	Shows the status, the attribute and the block calls of a task.
Task info all	Shows the status, the attribute and the block calls of all tasks.
Axis info	Output information on the current axis status.
Error memory	Displays the controller error memory.



NOTE

For a more detailed description of the pull-down menu “Contr. status”, refer to chapter 3.6.5.

3.5.5.4 The pull-down menu “Special”

The options of the pull-down menu “Special” are described in chapter 3.7.

“Editors/Assignment list/
Special”

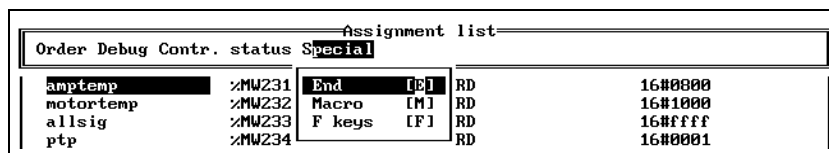


Fig. 3-104 The pull-down menu “Special”

Menu option	Description
End	Exit the assignment list editor menu.
Macro	Operations with macros.
F keys	Function key assignment.

3.6 Communication with the controller

“Online” Communication with the controller (data exchange) is handled via the “Online” menu. The “Online” menu offers the following actions:

- Connect to controller (login) or disconnect (logout)
- Transfer data from the PC to the controller (download)
- Delete project data on the controller (clear memory)
- Compile and link program components
- Monitor program execution
- Status information from controller
- Display and change variables

The following requirements must be met for communication with the controller:

- PC and controller(s) must be properly installed (see chapter 2).
- The device address on the controller must be set (see controller manual).
- The connection to the controller must be established (“Login”, see chapter 3.6.2).
- The controller must have reset status (only for upload/download commands). The controller states are described in the controller manual.



NOTE

The menu option “Special/Setup” can be used for selecting the PC interface (COM1 or COM2) to be used for communication with the controller.

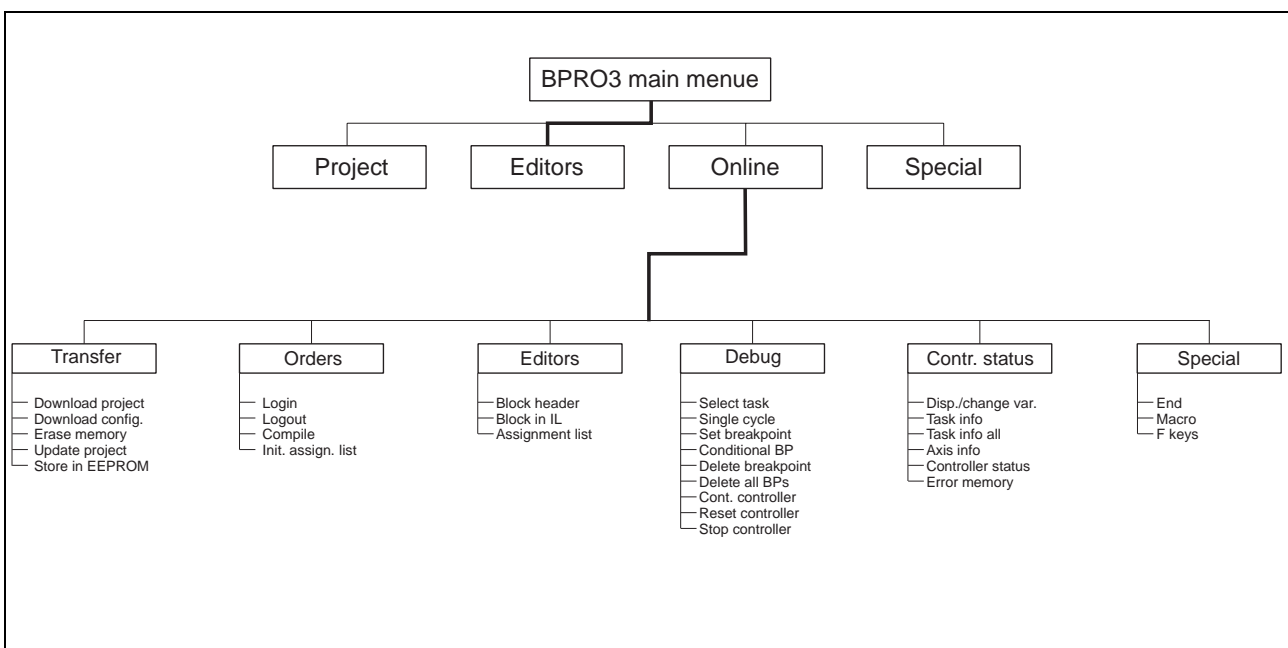


Fig. 3-105 “Online” menu tree

3.6.1 The pull-down menu “Transfer”

This menu contains options for data exchange between computer and controller.



NOTE

The connection to the controller must be established (“Login”, see chapter 3.6.2). The controller must have reset status (press stop key on controller front panel or select “Online/Debug/Reset controller”).

“Online/Transfer”

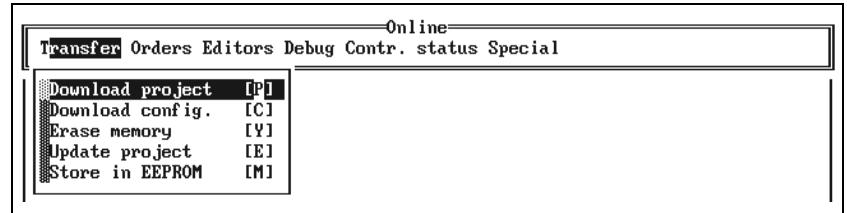




Fig. 3-106 The pull-down menu “Transfer”

Menu option	Description
Download project	Load complete project into controller.  ATTENTION <i>All data on the controller will be deleted.</i>
Download config.	Load controller configuration into controller.
Erase memory	Delete all project data on controller.
Update project	Update those blocks on the controller the state of which is not the current one.
Store in EEPROM	The project is stored in the EEPROM memory on the controller.  NOTE <i>For information on how the controller handles a project stored in EEPROM, see the appropriate controller manual.</i> <i>An especially important aspect is the handling of any differences recognized between the project version stored in the buffer RAM and the version stored in EEPROM.</i>

3.6.2 Miscellaneous commands This pull-down menu offers a selection of miscellaneous commands.

"Online/Orders"

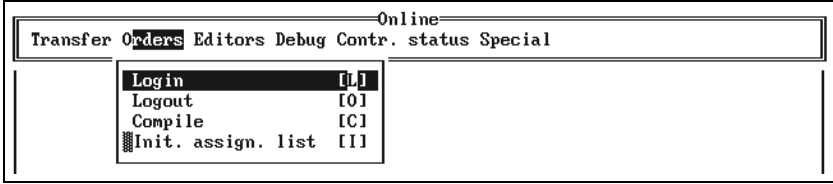


Fig. 3-107 The pull-down menu "Orders"

Menu option	Description
Login	Establish connection to controller.
Logout	Disconnect the connection to the controller.
Compile	Compile the complete project, an individual task or block.
Init. assign. list	Load the values of the assignment list into the controller and initialize.

3.6.3 Viewing the CR

These menu options are used for viewing current results and variables during controller run.

“Online/View”

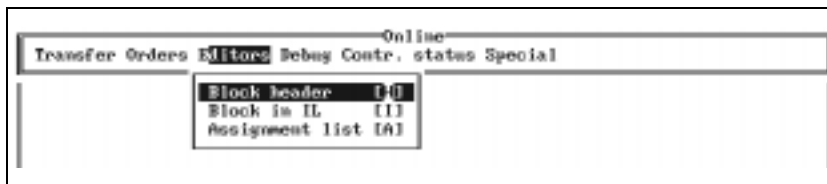


Fig. 3-108 The pull-down menu “Editors”

Menu option	Description
Block header	View the block header.
Block in IL	View the current results of a block in IL.
Assignment list	View the variables in the assignment list.

3.6.4 Debugging options

The pull-down menu “Debug” offers various options for program testing (debugging). This menu can also be called from the editors, however, with slight variations.

“Online/Debug”

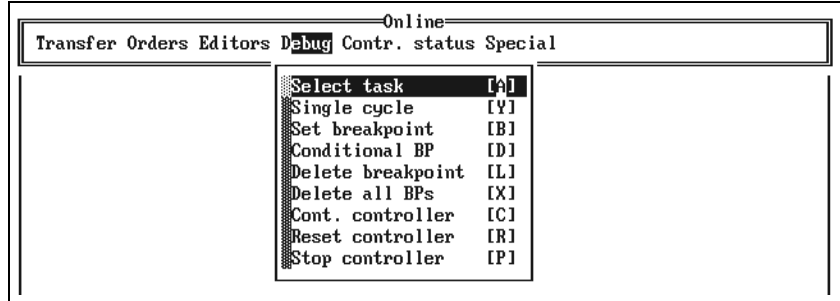


Fig. 3-109 The pull-down menu “Debug”

Menu option	Description
Select task	Select task for Step, Step over, Single cycle, Set breakpoint, Conditional breakpoint, Stop task, Reset task, and Continue task. If you omit this menu option, BPRO3 will prompt you for the name of the task whenever any of the above mentioned options is called.
Single cycle	Execute a program in single cycles.
Set breakpoint	Set a breakpoint. Program execution stops at the beginning of the block.
Conditional BP	Set a conditional breakpoint.
Delete breakpoint	Delete individual breakpoints.
Delete all BPs	Delete all breakpoints.
Cont. controller	Start the controller.
Reset controller	Reset the controller.
Stop controller	Stop the controller.

3.6.4.1 Processing a task once

“Online/Debug/Single cycle”

The “Single cycle” option is used for testing the program. A task is started and processed once. When the predefined cycle has been processed, program execution stops automatically as if a breakpoint had been set at this point.



ATTENTION

By default, outputs and axes are reset automatically at program stop. This feature can be disabled with the “Debug” option (see programming manual). However, the following effects must always be taken into account:

Outputs and axes are no longer controlled by the controller after a program stop. This means that, for example, motors continue to run without control, which may result in injury or damage!

To start a single cycle, proceed as follows:

1. Select the menu option “Single cycle”.

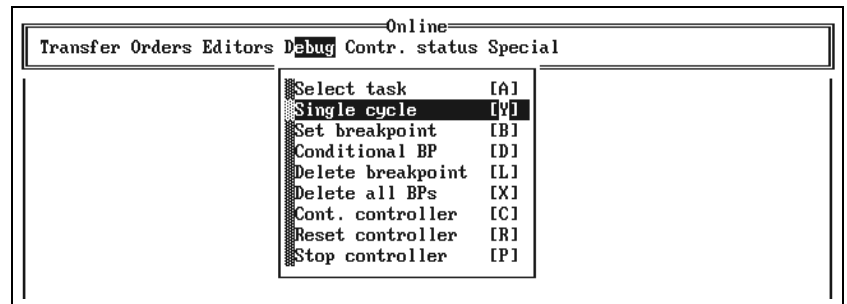


Fig. 3-110 Selecting “Debug/Single cycle”

2. Enter the task name (identifier) either directly or select it from the option list. BPRO3 then starts the single cycle on the controller and acknowledges the process with a message.

3.6.4.2 Controlled program stop

“Online/Debug/Set breakpoint”

This option is used for setting one or more breakpoints (BPs) in a program loaded into the controller in pseudo-code. The program is stopped exactly at the points defined by the breakpoints. This enables you to stop a running program at any defined point, for example, to check current results.



ATTENTION

By default, outputs and axes are reset automatically at program stop. This feature can be disabled with the “Debug” option (see programming manual). However, the following effects must always be taken into account:

Outputs and axes are no longer controlled by the controller after a program stop. This means that, for example, motors continue to run without control, which may result in injury or damage!

To set a breakpoint, proceed as follows:

1. Select the menu option “Debug/Set breakpoint”.

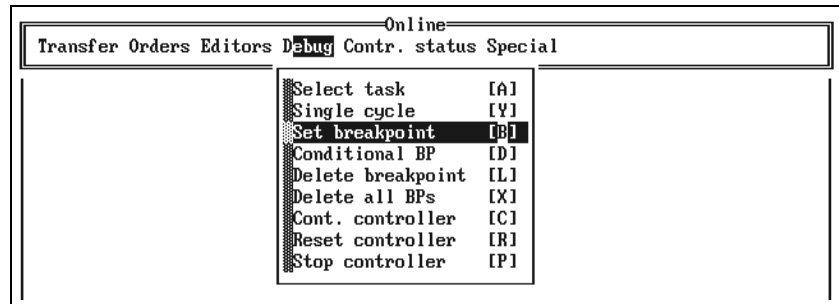


Fig. 3-111 Selecting “Debug/Set breakpoint”

2. Enter the symbolic name of the desired block either directly or by pressing <F3>. BPRO3 then sets the breakpoint at the start of the block.

3.6.4.3 Setting a conditional breakpoint

“Online/Debug/Conditional BP”



ATTENTION

By default, outputs and axes are reset automatically at program stop. This feature can be disabled with the “Debug” option (see programming manual). However, the following effects must always be taken into account:

Outputs and axes are no longer controlled by the controller after a program stop. This means that, for example, motors continue to run without control, which may result in injury or damage!

To set a conditional breakpoint, proceed as follows:

1. Select the menu option “Debug/Conditional BP”.

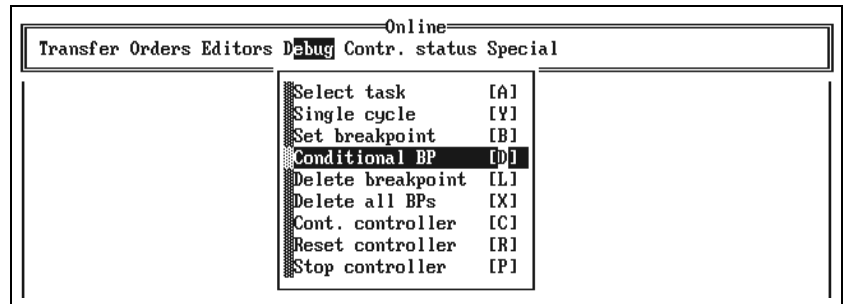


Fig. 3-112 Selecting “Debug/Conditional BP”

2. Enter the task name (identifier) either directly or select it from the option list.
3. Enter the symbolic name of the desired block either directly or select type and name of the block from the option list. The call tree is then displayed.
4. Use the cursor keys to move the field cursor to the desired point and press the <↵> key. BPRO3 then sets the breakpoint at the start of the selected block.

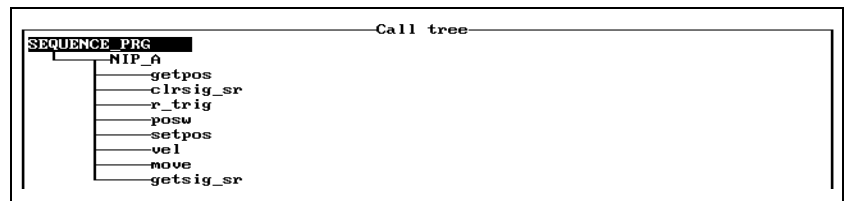


Fig. 3-113 Displaying the call tree

3.6.4.4 Deleting a breakpoint

This menu option allows you to delete individual breakpoints.

1. Select the menu option "Delete breakpoint".

*"Online/Debug/
Delete breakpoint"*

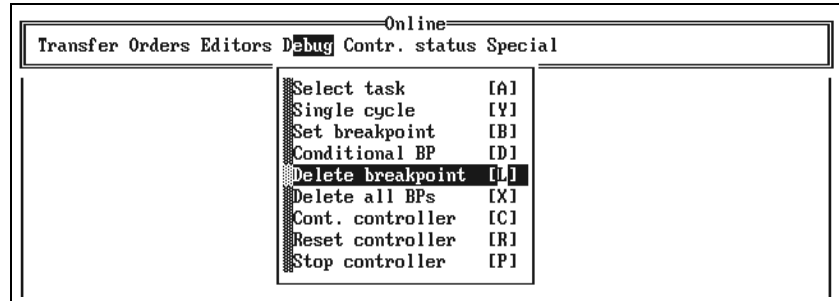


Fig. 3-114 Selecting "Debug/Delete breakpoint"

2. Enter the task name (identifier) either directly or select it from the option list.
3. Enter the symbolic name of the desired block either directly or select it from the option list by pressing <F3>. BPRO3 then removes the breakpoint from the specified block.

3.6.4.5 Deleting all breakpoints

This menu option allows you to delete all breakpoints.

1. Select the menu option "Debug/Delete all BPs".

"Online/Debug/Delete all BPs"

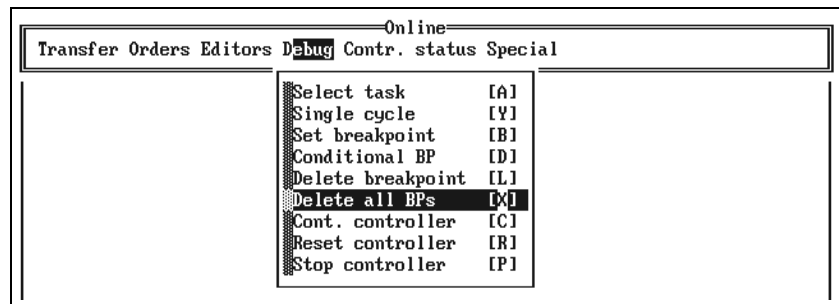


Fig. 3-115 Selecting "Debug/Delete all BPs"

2. BPRO3 then removes all breakpoints on the controller and acknowledges the process with a message.

3.6.4.6 Resuming program execution on controller

“Debug/Cont. controller” starts or resumes program execution on the controller. The program is started or resumed at the point identified by the instruction pointer.



ATTENTION

When you select “Cont. controller”, the outputs are reenabled and any interrupted motor movements continue.

“Online/Debug/Cont. controller”

Proceed as follows:

1. Select the menu option “Cont. controller”.

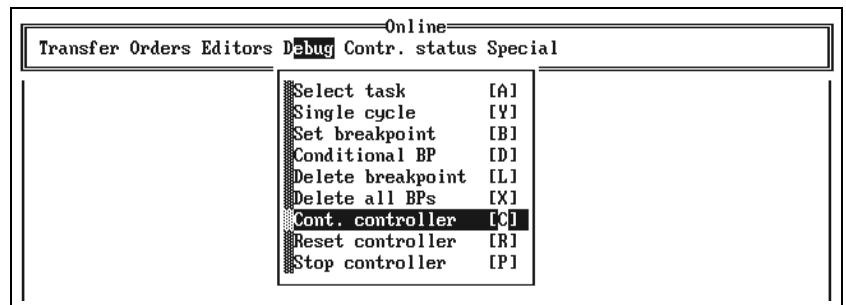


Fig. 3-116 Selecting “Cont. controller”

2. When the program prompts you with “Start controller now?”, confirm this by pressing the <Y> key (“Yes”). BPRO3 then resumes program execution on the controller and acknowledges this with a message.

The controller states are described in the appropriate controller manual.

3.6.4.7 Resetting the controller program

“Debug/Reset controller” resets the controller. The instruction pointer is reset to program start. For more information on controller reset, refer to the appropriate controller manual.

“Online/Debug/Reset controller”

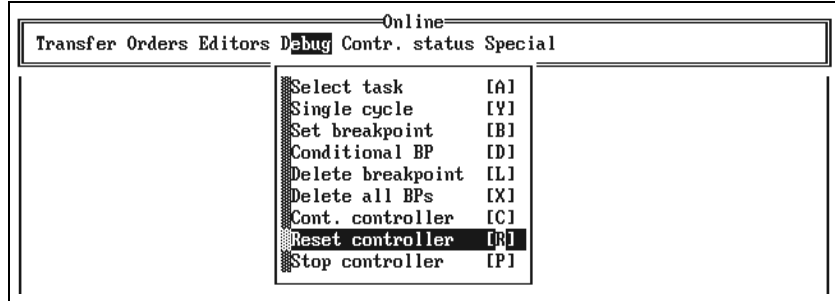


Fig. 3-117 Selecting “Reset controller”

When you select the menu option “Debug/Reset controller”, BPRO3 resets the controller and acknowledges this with a message.

3.6.4.8 Stopping program execution

“Debug/Stop controller” stops program execution. To resume execution from the same point, select “Cont. controller”.



ATTENTION

By default, outputs and axes are reset automatically at program stop. This feature can be disabled with the “Debug” option (see programming manual). However, the following effects must always be taken into account:

Outputs and axes are no longer controlled by the controller after a program stop. This means that, for example, motors continue to run without control, which may result in injury or damage!

“Online/Debug/Stop controller”

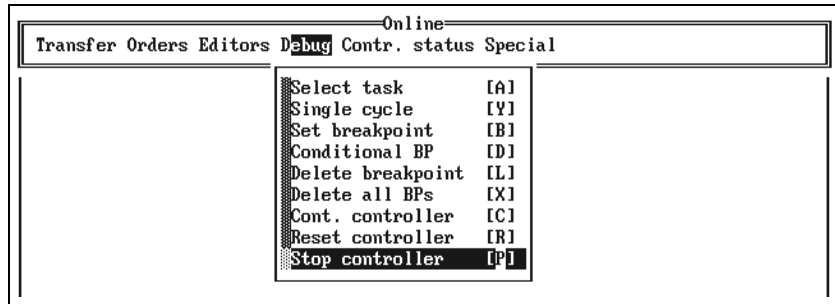


Fig. 3-118 Selecting “Stop controller”

When you select the menu option “Stop controller”, the controller stops. BPRO3 acknowledges this with a message.

3.6.5 The pull-down menu “Contr. status”

This pull-down menu is designed for indicating various controller states.



NOTE

The connection to the controller must be established (“Login”, see chapter 3.6.2).

“Online/Contr. status”

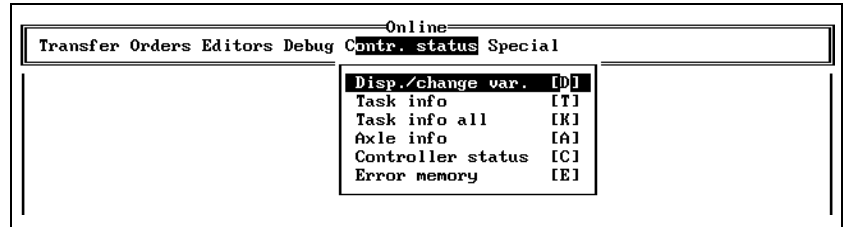


Fig. 3-119 The pull-down menu “Contr. status”

Menu option	Description
Disp./change var.	Display and change data from the process image, direct I/O, flags and variables of the data blocks.
Task info	Shows the status, the attribute and the block calls of a task.
Task info all	Shows the status, the attribute and the block calls of all tasks.
Axis info	Output information on the current axis status.
Controller status	Output information on the current controller status.
Error memory	Displays the controller error memory.

3.6.5.1 Displaying and changing variables

*“Online/Contr. status/
Disp./change var.”*

When you select the menu option “Disp./change var.”, a menu with options for displaying and changing global data on the controller is displayed. This includes data from the process image, direct I/O, flags and data block variables. Before calling this menu you must select “Login” first; see chapter 3.6.2.

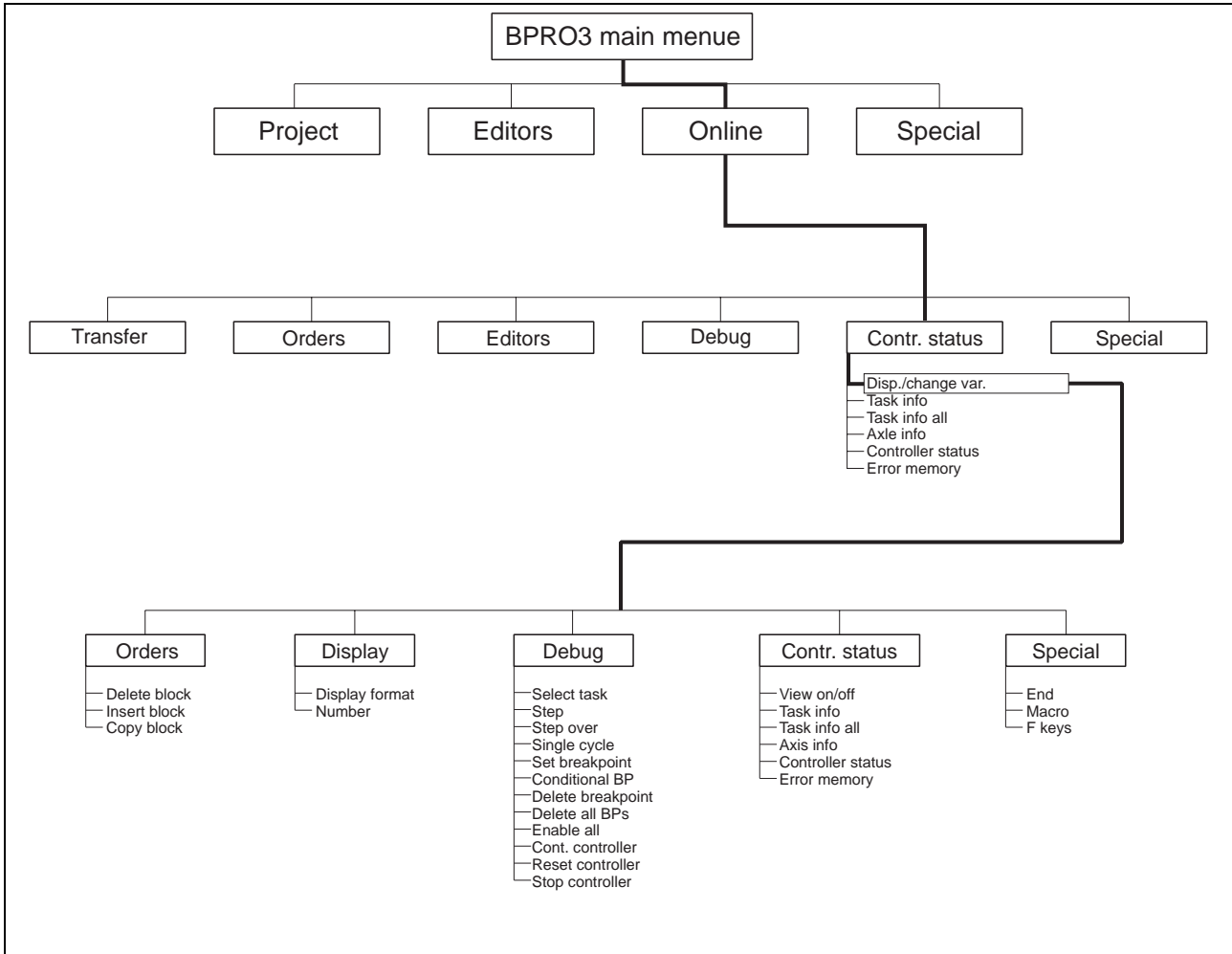


Fig. 3-120 Data editor menu tree

The user interface of the menu “Disp./change var.”

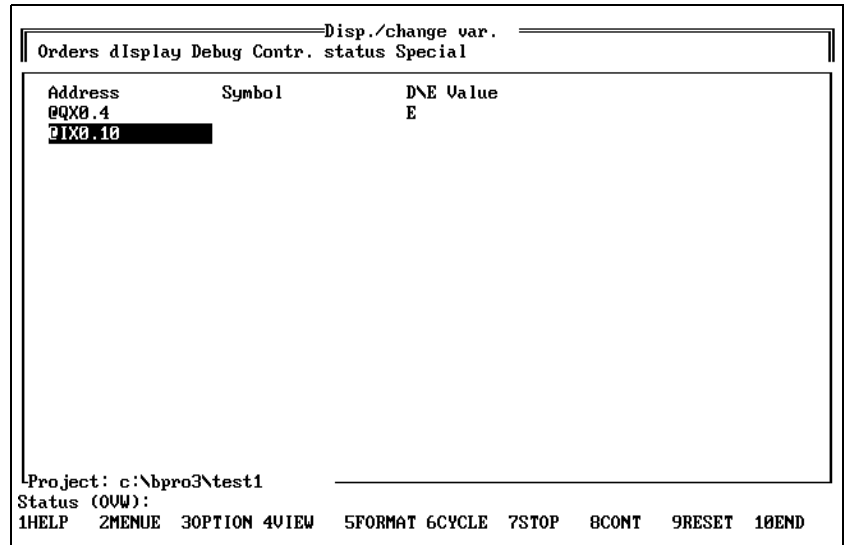




Fig. 3-121 Data editor user interface

The editing area comprises four columns:

Designation	Meaning
Address	Address of the variables.
Symbol	Symbolic name of the variables from the assignment list.
D/E	<p>You can use “D/E” to set the inputs/outputs for testing purposes as required.</p> <p> NOTE Inputs/outputs can only be set explicitly (using @I... or @Q...).</p> <p>D (Disable): “Disable” disconnects the signal interface. The signal interface input will not be interrogated, and the output is no longer controlled by the program.</p> <p>E (Enable): “Enable” means that the signal interface input is interrogated in addition, and the signal interface output is controlled by the program.</p> <p> NOTE Selecting the submenu option “Debug/Enable all” sets all inputs and outputs to “E”</p>
Value	Value assigned to a variable.



ATTENTION

If you select the menu “Disp./change var.” in the IL editor, it displays 4 lines so that the last section of the IL block is essentially visible.

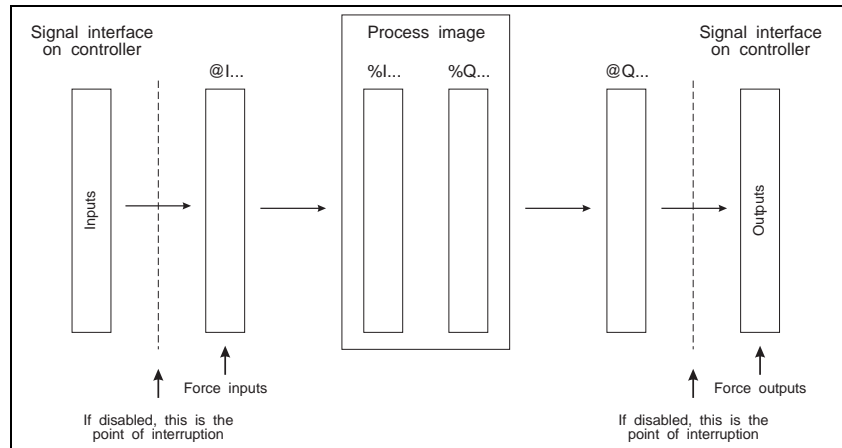


Fig. 3-122 Disabling inputs and outputs

The pull-down menu “Orders” of “Disp./change var.”

“Online/Contr. status/
Disp./change var./Orders”

The options of the pull-down menu “Orders” are described in chapter 3.2.3.4.

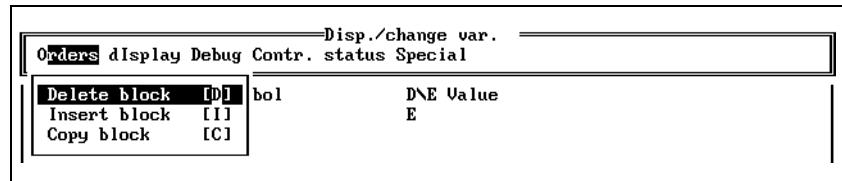


Fig. 3-123 The pull-down menu “Orders”

The pull-down menu “Display” of “Disp./change var.”

“Online/Contr. status/
Disp./change var./Display”

The pull-down menu “Display” contains options for setting the display format for the variables displayed and for listing contiguous flag ranges.

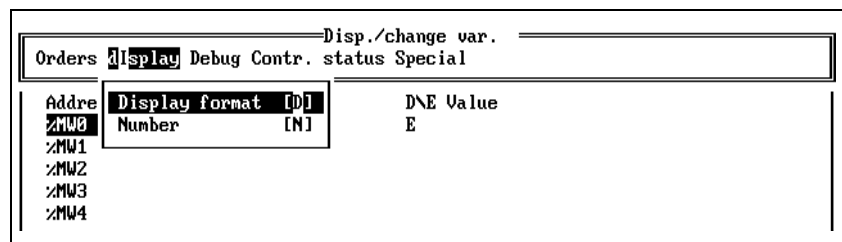


Fig. 3-124 The pull-down menu “Display”

Menu option	Description
Display format	Setting a display format for the variables. <div style="display: flex; align-items: center;"> <div style="border: 1px solid black; padding: 2px; margin-right: 5px;">i</div> <div> <p>NOTE This is only valid for the variable types BYTE, WORD and INIT.</p> </div> </div>
Number	List contiguous flag ranges. <div style="display: flex; align-items: center;"> <div style="border: 1px solid black; padding: 2px; margin-right: 5px;">i</div> <div> <p>NOTE The values are output without being formatted.</p> </div> </div>

To set a display format, proceed as follows:

1. Select the menu option "Display/Display format".
2. Use the cursor keys to mark the desired block (in the same way as when marking a block in "Orders/Delete block" in chapter 3.2.3.4) and press the <↵> key.

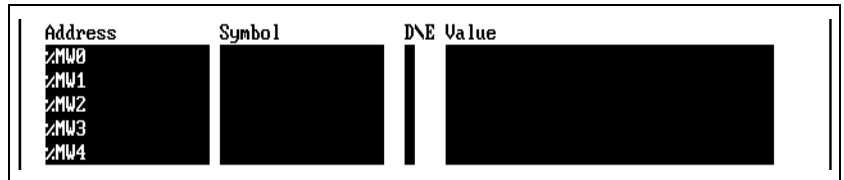


Fig. 3-125 Setting the display format

3. Select the desired display format with the cursor keys and press the <↵> key.

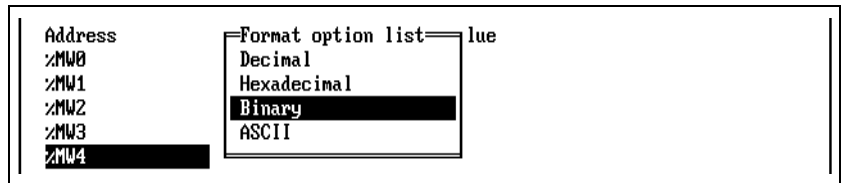


Fig. 3-126 Selecting the display format

Listing a contiguous flag area:

1. Select the menu option "Display/Number".

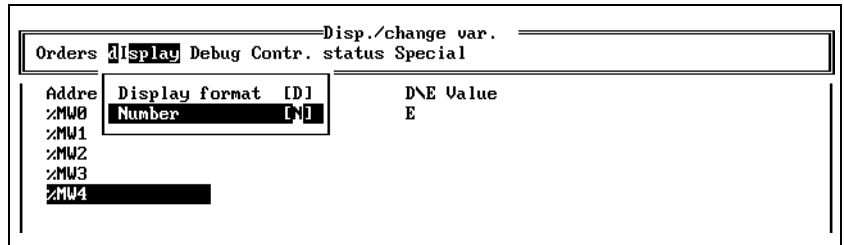


Fig. 3-127 Listing a contiguous range

2. Specify the first flag to be listed and the required number. BPRO3 then lists the specified flag range in the line below the cursor.

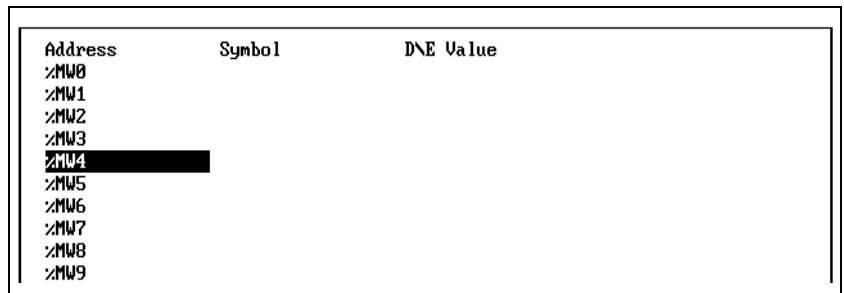


Fig. 3-128 Listed flag range

The pull-down menu “Debug” of “Disp./change var.”

*“Online/Contr. status/
Disp./change var./Debug”*

The pull-down menu “Debug” offers various options for program testing (debugging). The essential menu options have already been explained in chapter 3.6.4. At this point, you can execute the program step by step in addition. The menu option “Enable all” enables all variables.

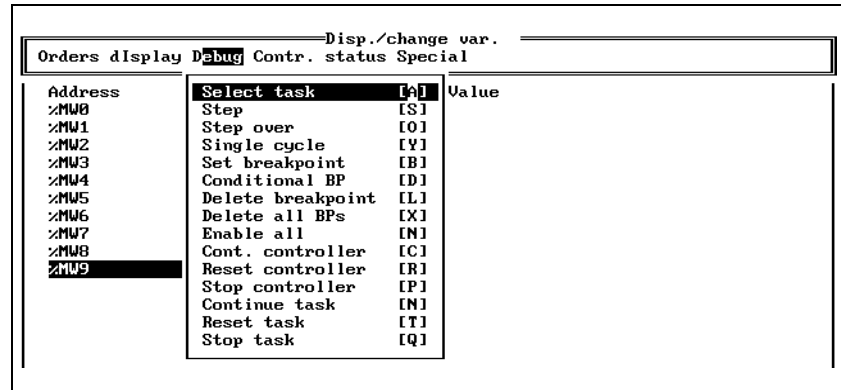


Fig. 3-129 The pull-down menu “Debug”

Menu option	Description
Select task	Select task for Step, Step over, Single cycle, Set breakpoint, Conditional breakpoint, Stop task, Reset task, and Continue task. If you omit this menu option, BPRO3 will prompt you for the name of the task whenever any of the above mentioned options is called.
Step	Process a single IL command line. After program start, or after controller reset, processing begins with the first line of the INIT task. After controller stop, the controller processes the program line following the one at which it was stopped.
Step over	This menu option is the same as the “Step” menu option, with one exception: Block calls are skipped.
Single cycle	Execute a program in single cycles.
Set breakpoint	Set a breakpoint. Program execution stops at the beginning of the block.
Conditional BP	Set a conditional breakpoint.
Delete breakpoint	Delete individual breakpoints.
Delete all BPs	Delete all breakpoints.
Enable all	Enable all variables.
Cont. controller	Start the controller.
Reset controller	Reset the controller.
Stop controller	Stop the controller.

Executing single steps with “Debug/Step”

*“Online/Contr. status/
Disp./change var./Step”*

“Debug/Step” allows you to execute a program in single steps, i.e. program execution stops after each instruction. It is recommended to enable “viewing” for this function so that BPRO3 reads the current variable values from the controller and displays them on the screen.

**ATTENTION**

By default, outputs and axes are reset automatically at program stop. This feature can be disabled with the “Debug” option (see programming manual). However, the following effects must always be taken into account:

Outputs and axes are no longer controlled by the controller after a program stop. This means that, for example, motors continue to run without control, which may result in injury or damage!

Executing single steps, skipping block calls

*“Online/Contr. status/
Disp./change var./Step over”*

The menu option “Step over” is the same as the menu option “Step” except for one difference.

Setting all variables to “Enable”

*“Online/Contr. status/
Disp./change var./Debug/
Enable all”*

Changes all variables set to “Disable” in the “D/E” column to “Enable” status.



NOTE

The remaining options of the pull-down menu “Debug” are explained in chapter 3.6.4.

The pull-down menu “Contr. status” of “Disp./change var.”

*“Online/Contr. status/
Disp./change var./Contr. status”*

This pull-down menu is designed for indicating various controller states.

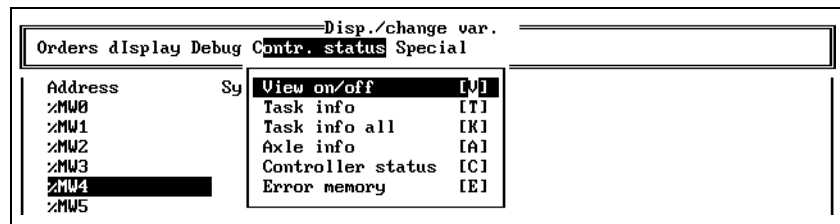


Fig. 3-130 The pull-down menu “Contr. status”

Menu option	Description
View on/off	Enable/disable displaying current controller values.
Task info	Shows the status, the attribute and the block calls of a task.
Task info all	Shows the status, the attribute and the block calls of all tasks.
Axis info	Output information on the current axis status.
Controller status	Output information on the current controller status.
Error memory	Displays the controller error memory.

To display the controller values with “View on/off”, proceed as follows:

1. Enter the controller variables to be displayed in column 1.
2. Select the menu option “Contr. status/View on/off”.
BPRO3 then reads the values from the controller and displays them in the set display format on the screen.

Address	Symbol	DNE	Value
:MW0		2#	0000000000000000
:MW1		2#	0000000000000000
:MW2		2#	0000000000000000
:MW3		2#	0000000000000000
:MW4			0
:MW5			0
:MW6			0
:MW7			0
:MW8			0
:MW9			0

Fig. 3-131 Viewing



NOTE

The remaining menu options are explained in chapter 3.6.5.

“Online/Contr. status/
Disp./change var./Special”

The pull-down menu “Special” of “Disp./change var.”

The options of the pull-down menu “Special” are explained in chapter 3.7.

Disp./change var.		
Orders	dIspay	Debug
Contr. status Special		
Address	Symbol	End [E]
:MW0		Macro [M]
:MW1		F keys [F]
:MW2		

Fig. 3-132 The pull-down menu “Special”

Menu option	Description
End	Exit the menu “Disp./change var.”.
Macro	Operations with macros.
F keys	Function key assignment.

3.6.5.2 Display task info

This menu option is designed for displaying information on a task.

“Online/Contr. status/Task info”

1. Select the menu option “Task info”.

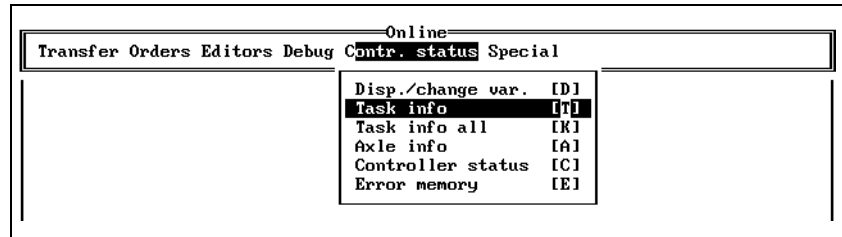


Fig. 3-133 Show task info on single task

2. Enter the task name (identifier) either directly or select it from the option list. BPRO3 then reads the relevant information from the controller.

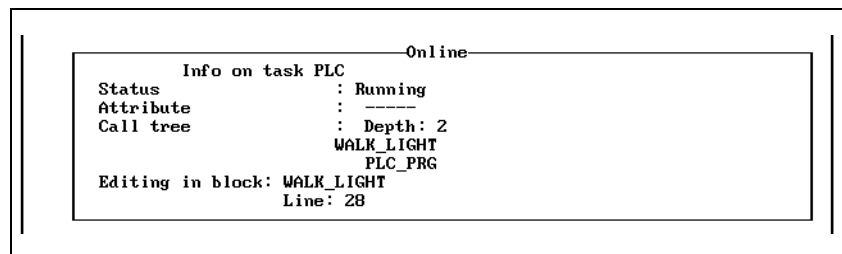


Fig. 3-134 Displaying task info

Designation	Description
Status	Shows the task status.
Attribute	Indicates the reason for the current status.
Call tree	Specifies the call depth and the block calls. BLOCK_1 is called by BLOCK_4.
Editing in block	Name of the block currently processed.
Line	Specifies the line currently processed.

3.6.5.3 Display task info of all tasks

This menu option is designed for displaying information on all tasks.

1. Select the menu option "Contr. status/Task info all".

"Online/Contr. status/Task info all"

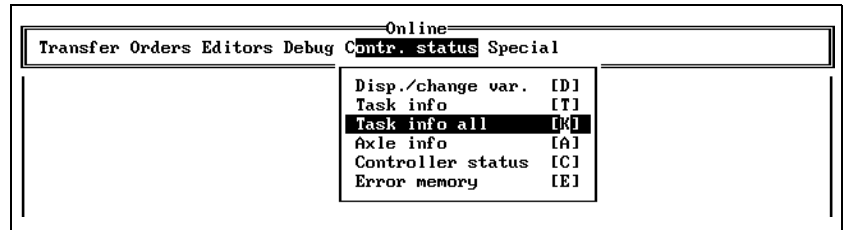


Fig. 3-135 Show task info on all tasks

2. BPRO3 shows the information on all tasks one after the other. To page through the tasks, press any key.

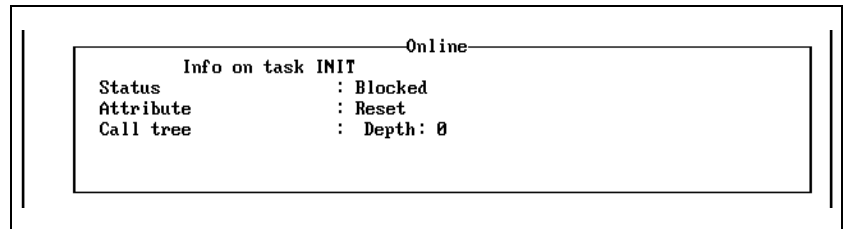


Fig. 3-136 Displaying all task infos

Designation	Description
Status	Shows the task status.
Attribute	Indicates the reason for the current status.
Call tree	Specifies the call depth and the block calls.
Editing in block	Name of the block currently processed.
Line	Specifies the line currently processed.

3.6.5.4 Display axis info

Display information on the axis status. The type of information displayed depends on the type of controller used.

“Online/Contr. status/Axis info”

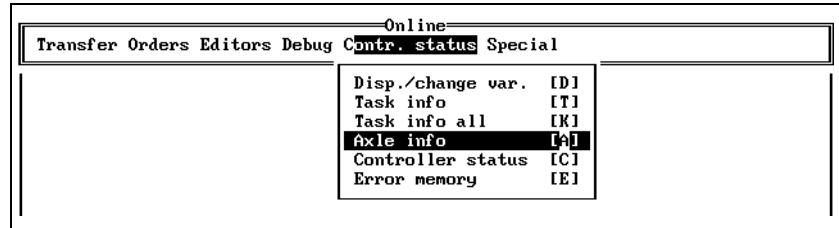


Fig. 3-137 Displaying axis info

3.6.5.5 Display controller status

This menu option can be used to display the current controller status: STOP, RESET, RUN, DEBUG or ERROR.

“Online/Contr. status/
Controller status”

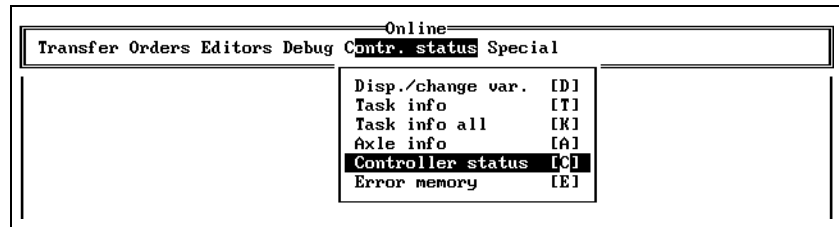


Fig. 3-138 Displaying the controller status

3.6.5.6 Display controller error memory

The controller has an error memory for storing errors occurring during processing. The menu option “Error memory” can be used for reading this error memory from the controller and displaying it on the screen.

“Online/Contr. status/
Error memory”

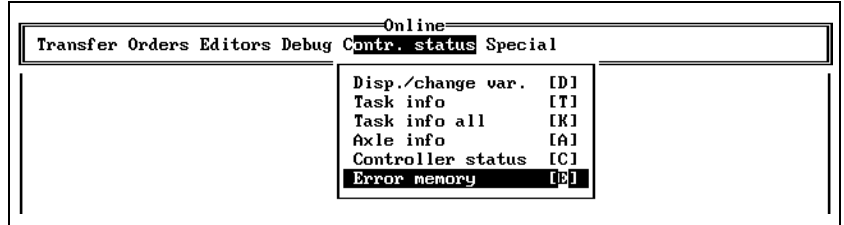


Fig. 3-139 Displaying contents of error memory

The error memory information consists of 16 entries, divided into two groups. The first eight entries contain the first eight errors, the last eight entries contain the last errors.

An error entry contains the following information in the columns displayed (from left to right):

1. Operating hours counter in seconds
2. Error class
3. Number of power-on sequences since delivery
4. Error number and error text, possibly including information on the component responsible for the error (axis, serial interface, etc.)
5. Brief additional information on the error, e.g. controller function which detected the error.

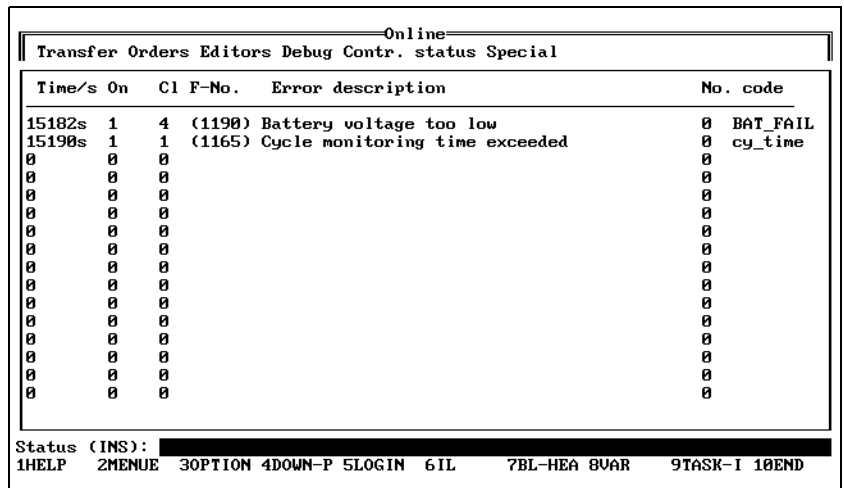


Fig. 3-140 Error memory example



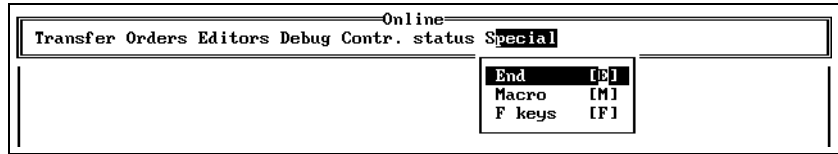
NOTE
The possible causes of errors registered in the error memory are described in chapter 6.

3.6.6 The pull-down menu "Special"

The options of the pull-down menu "Special" are explained in chapter 3.7.

"Online/Special"

Fig. 3-141 The pull-down menu "Special"



Menu option	Description
End	Exit the "Online" menu.
Macro	Operations with macros.
F keys	Function key assignment.

3.7 Special functions

“Special” The pull-down menu “Special” is available from each menu bar. It always contains the menu options “End”, “Macro” and “F keys”.

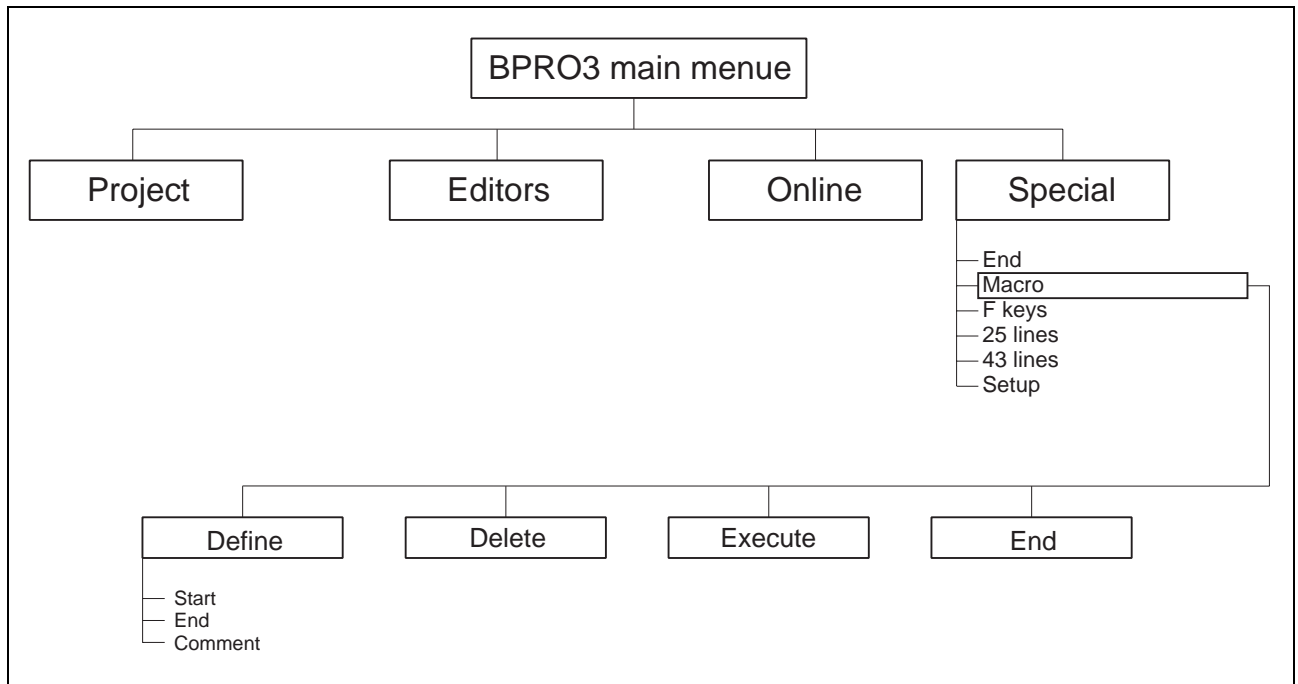
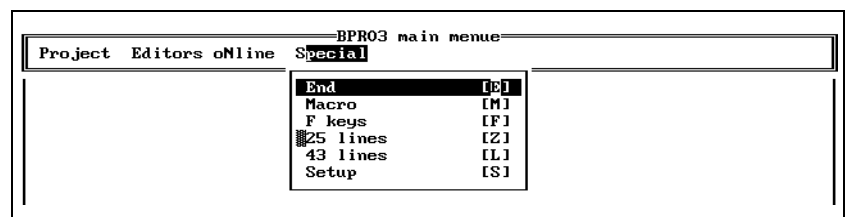


Fig. 3-142 “Special” menu tree

Fig. 3-143 The pull-down menu “Special”



Menu option	Description	Reference
End	In the main menu, selecting “End” exits BPRO3. In a submenu, e.g. “Online”, BPRO3 closes the submenu and returns to the calling menu.	See chapter 3.7.1
Macro	Operations with macros.	See chapter 3.7.2
F keys	Function key assignment.	See chapter 3.7.3
25 lines	To select 25-line screen display.	See chapter 3.7.4
43 lines	To select 43-line screen display.	See chapter 3.7.4
Setup	Various setup parameters such as interfaces and paths.	See chapter 3.7.5

3.7.1 Exiting the menu or leaving the program

“Special/End”

You can exit the program by selecting the menu option “Special/End” from the BPRO3 main menu. As you can see in the figure below, you can exit the program with (<S> key) or without (<Q> key) storing the project. You can cancel exiting by pressing <Esc> or <C>.

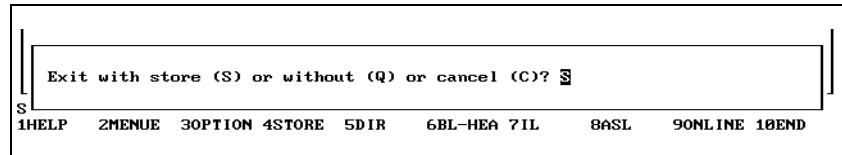


Fig. 3-144 Exiting the program

3.7.2 Defining, deleting and starting macros

“Special/Macro”

A series of keystrokes (hotkeys) is recorded in a macro. When executing the macro, BPRO3 automatically executes all keystrokes stored in the macro. This allows you to record inputs which have to be entered frequently in a macro and to run the macro later.

When you select the menu option "Macro", all existing macros created for the selected menu or editor are displayed.

The editing area is divided into two columns: The left-hand column contains the names of the macros already created, the right-hand column contains descriptive comments. The name column accepts a maximum of eight characters, the comment column a maximum of 40 characters.

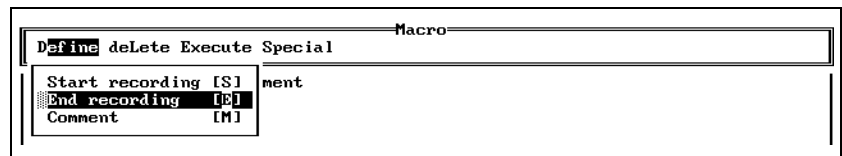



Fig. 3-145 The pull-down menue “Define”

Menu option	Description
Start recording	Start recording of keystrokes.
End recording	Stop recording of keystrokes.
Comment	Call the comment editor.
Delete	Delete the macro marked by the field cursor.
Execute	Execute the macro marked by the field cursor.
	 NOTE You can interrupt execution of a macro by pressing <Esc>.
End	Exit from the "Macro" menue.
F keys	Function key assignment.

Defining a macro

1. Select "Define/Start recording".
2. Enter a macro name and press the <↵> key.
3. Enter the desired sequence of hotkeys.
4. To stop recording of keystrokes: Select the menu option "Special/Macro". This returns you to the macro editor.
5. From the macro editor, select the menu option "Define/End recording".

Modifying a macro

Macros can be modified in one of two ways:

- Re-record a macro using the same name.
- BPRO3 stores the macros in plain text in a text file on the hard disk. This text file contains the hotkeys of the menu options and the names of the respective menu options as a comment in plain text. You can edit this text file using any editor.

3.7.3 Function key assignment (F keys)

The function keys <F1> to <F10> and combinations of function keys with the control keys <Alt>, <Shift> and <Ctrl> can be freely assigned, with three exceptions. The function key assignment is displayed in the bottom line of the screen. The function key combination assignments are displayed when you press <Alt>, <Shift> and <Ctrl>, respectively, for more than 2 seconds.

You can assign functions to the function keys using the menu option "F keys" from the pull-down menu "Special". With this option you can assign functions to the function keys of the currently loaded editor or menu by selecting from all menu options currently available. This allows you to customize the assignment to your specific requirements.

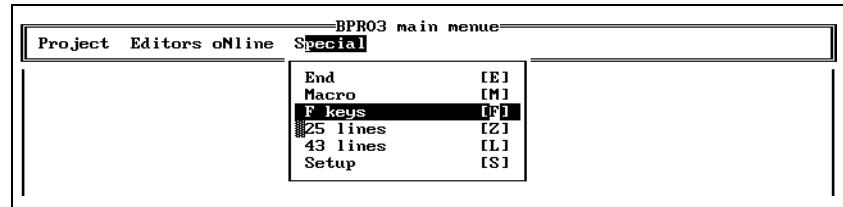
Freely definable function keys:

- <F4> to <F10>
(<F1> is reserved for help information, <F2> for opening the pull-down menus and <F3> for the option list function.)
- <Shift> - <F1> to <Shift> - <F10>
- <Alt> - <F1> to <Alt> - <F10>
- <Ctrl> - <F1> to <Ctrl> - <F10>

Assigning function keys

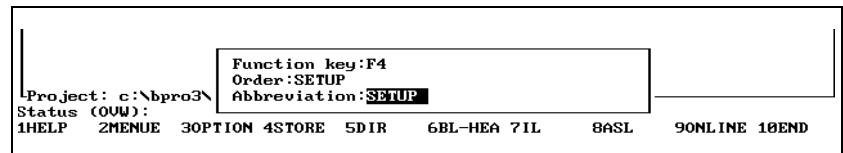
1. Select "Special/F keys".

Fig. 3-146 The pull-down menu "F-keys"



2. Press the function key or the combination of a function key with <Shift> or <Alt> that you want to assign.
3. Enter the menu option (next to "Order") or select it from the option list (<F3>) and press the <↵> key.
4. Enter a 6-character abbreviation and press the <↵> key. This abbreviation will be displayed in the bottom line as the function key assignment.

Fig. 3-147 Input for assigning a function key



3.7.4 25-line or 43-line screen display

The programming system offers two different screen display formats.

- Normal display with 25 lines on the screen
- Display with 43 lines on the screen



NOTE

This is only valid for computers with a VGA graphics card.

3.7.5 Setup

The following table describes the setup parameter settings.

```

                                Setups
-----
Audible alarm (y/n) :                               ]
Interface to controller (COM1, COM2) :             COM1
Number of log files ( 0, 1, 2 ) :                 2
Number of minutes for log file storing ( 0 ..60 ) : 1
System messages on (y/n) :                       Y
Macro path :
C:\BPRO3
IL export path :
C:\BPRO3
Project path :
c:\bpro3\test1
BACKUP of database:                               N
Download format :
Default:      Intermediate code with symbols
Data blocks:  Intermediate code
Controller config: Intermediate code
Assignment list: Intermediate code

<F3> = Option list      Help for setup only via BPRO3 main menue
                          Menue option "Special/Setups"

Status (00W):

```

Fig. 3-148 Setup parameters

Setup parameter	Description
Audible alarm	Switch the audible alarm on or off.
Interface to controller	Set the interface for communication with the controller.
Number of log files	See chapter 3.2.4.
Number of minutes for log file storing	The interval for storing the log files.
System messages on (y/n)	Enable/disable system message display.
Macro path	The path to be used by BPRO3 for storing macros.
IL export path	The path to be used for storing the blocks deleted or copied in the IL editor.
Project path	At program start, BPRO3 loads the project from this path.
BACKUP of database	Specifies whether a backup copy of the databases is stored in the "BAK" directory (on the project path).
Download format	The default download format set when creating a block in the block header editor.

You can leave the "Setup" menu at any point by pressing <Esc>.

4 Block programming

This chapter describes the various block types and the way they are created. It also describes the variables which can be declared and the block types which can be called.

4.1 Programming program blocks

Program blocks are integral program components which cannot call each other and therefore do not have any input or output variables. Program blocks are assigned to a task which determines the processing sequence of the program blocks. Data exchange between program blocks is effected via global variables such as flags or variables of data blocks.

4.1.1 Valid variable types and block calls **Valid variable types**

In a program block, only certain types of variables can be used. All other variable types (which cannot be declared in a program block) are marked with a “■” symbol.

	Designation	Description
	VAR	Local variable or function block.
■	VAR_INPUT	Input variable for accepting values from the calling block.
■	VAR_OUTPUT	Output variable for transferring values to the calling block.
■	VAR_IN_OUT	Input and output variable.
	VAR_EXTERNAL	Global variable which is identified to the PRG with the declaration.
	VAR_RETAIN	Same as VAR, but is retained after power-off.
■	VAR_INPUT_RETAIN	Same as VAR_INPUT, but is retained after power-off.
■	VAR_OUTPUT_RETAIN	Same as VAR_OUTPUT, but is retained after power-off.

Valid block calls

The following block types can be called in a program block:

- Function blocks
- Global blocks
- Functions



NOTE

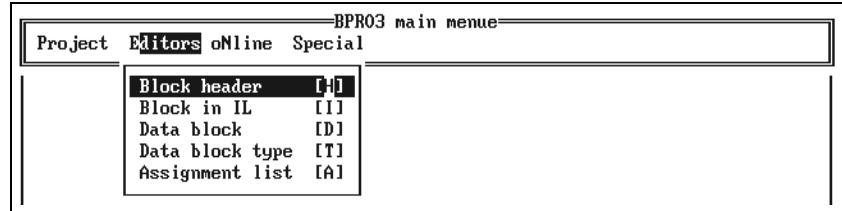
For more information, see chapter 1.2.3.3 “Block calling” in the programming manual.

Block programming

4.1.2 Creating a program block To create a new program block, you first have to define a block header. For this purpose, call the block header editor in the BPRO3 programming system:

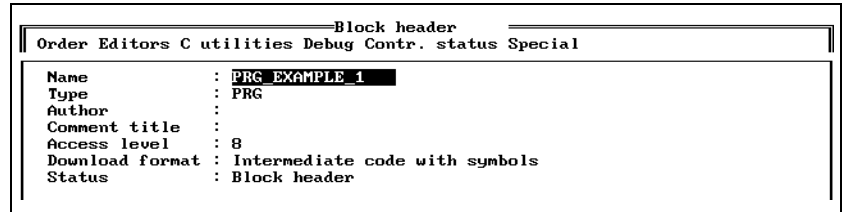
1. Select the menu option "Editors/Block header".

Fig. 4-1 Calling the block header editor



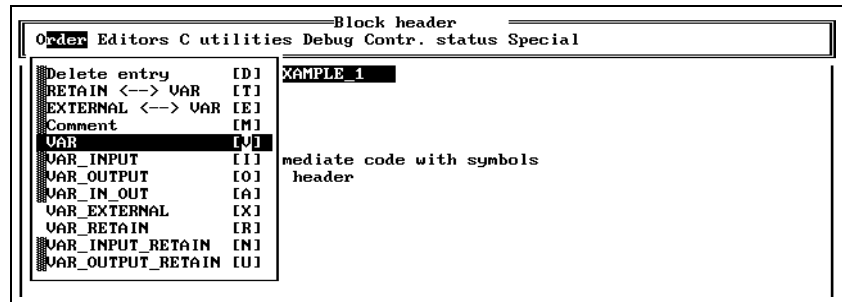
2. Enter a new block name.
3. Confirm creating a new block by pressing "Y".
4. Enter or edit general information on the block.

Fig. 4-2 Editing general block information



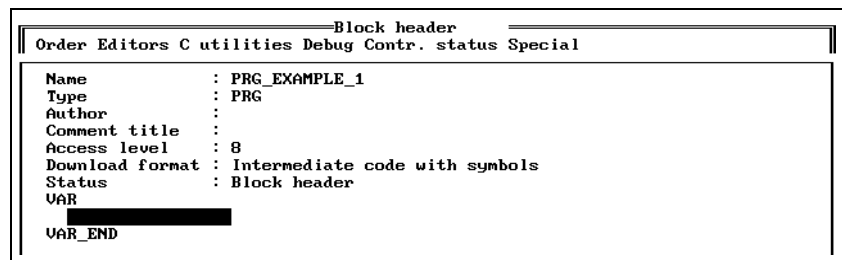
5. Use the pull-down menu "Orders" to insert the sections required for declaring variables, e.g. "VAR".

Fig. 4-3 The pull-down menu "Order"



6. Declare the variables.

Fig. 4-4 Variable declaration



7. Repeat steps 5 and 6 for each variable to be declared.

- Exit the block header editor by selecting the menu option "Special/End".
- Select the menu option "Editors/Block in IL".

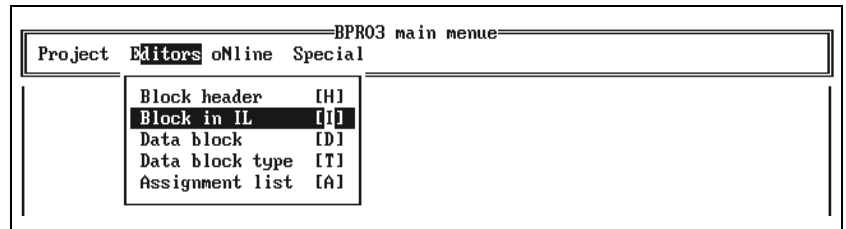


Fig. 4-5 Calling the IL editor

- Enter the program instructions for the program block.
- Exit the IL editor by selecting the menu option "Special/End".



NOTE

For more detailed information on calling and using the editors, refer to chapter 3.5.

4.2 Programming function blocks

Function blocks are program components which can be called by other blocks. They can include input and output variables. Input and output variables are used for data transfer from and to the calling block.

In addition, a function block contains internal variables which retain their values after the block call. This means that a function block has a "variable" internal status which enables the block to save status conditions and data to be used for the next call to the function block.

When creating a function block, you first have to create a function block type; a function block type cannot be used directly in a program. A function block to be called in another block must be declared in the block header of the calling block like a variable. In this declaration, memory is allocated for the variables declared in the block header and a name is assigned which is used for calling the block later. Multiple declarations are possible.

The function block can be called with the "cal" operator (see programming manual).

4.2.1 Valid variable types and block calls

Valid variable types

In a function block, all types of variables can be used.

Designation	Description
VAR	Local variable or function block.
VAR_INPUT	Input variable for accepting a value from the calling block.
VAR_OUTPUT	Output variable for transferring values to the calling block.
VAR_IN_OUT	Input and output variable.
VAR_EXTERNAL	Global variable which is identified to the PRG with the declaration.
VAR_RETAIN	Same as VAR, but is retained after power-off.
VAR_INPUT_RETAIN	Same as VAR_INPUT, but is retained after power-off.
VAR_OUTPUT_RETAIN	Same as VAR_OUTPUT, but is retained after power-off.

Valid block calls

The following block types can be called in a function block:

- Function blocks of a different function block type
- Global blocks
- Functions



NOTE

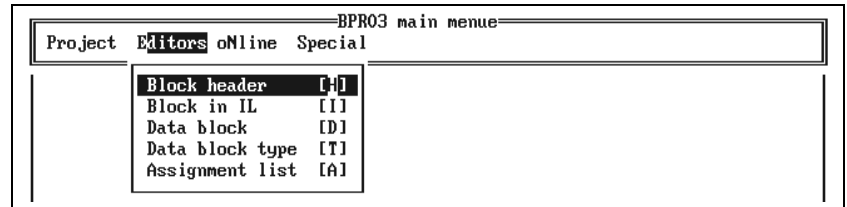
For more information, see chapter 1.2.3.3 "Block calling" in the programming manual.

4.2.2 Creating a function block type

To create a new function block, you first have to define a block header. For this purpose, call the block header editor in the BPRO3 programming system:

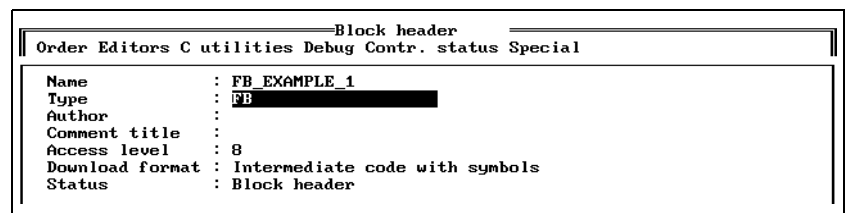
1. Select the menu option "Editors/Block header".

Fig. 4-6 Calling the block header editor



2. Enter a new block name.
3. Confirm creating a new block by pressing "Y".
4. Enter or edit general information on the block. For function blocks, the block type must be entered manually or using the option list function (<F3> key).

Fig. 4-7 Editing general block information

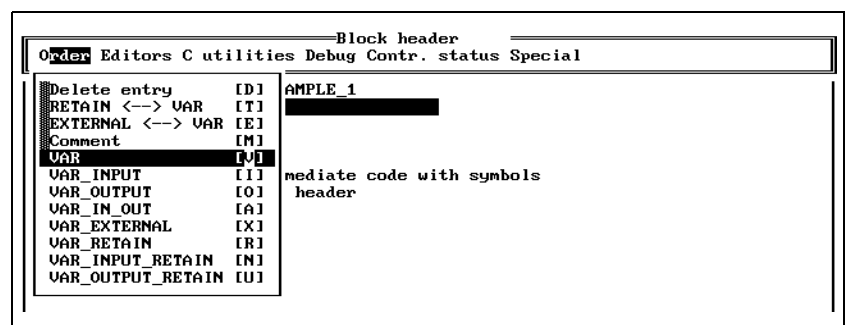


ATTENTION

This procedure creates a function block type. You must then declare the function block in the block header of the calling block.

5. Use the pull-down menu "Orders" to insert the sections required for declaring variables, e.g. "VAR".

Fig. 4-8 The pull-down menu "Order"



6. Declare the variables.

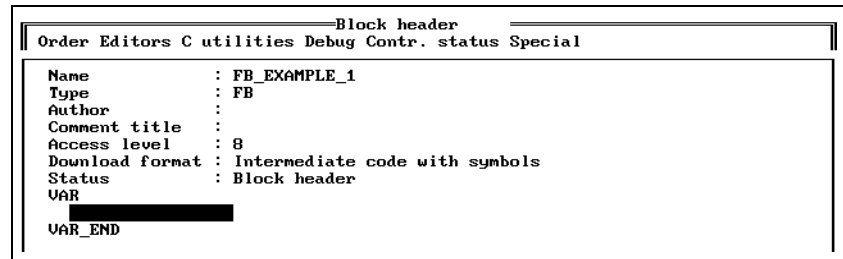


Fig. 4-9 Editing general block information

7. Repeat steps 5 and 6 for each variable to be declared.
8. Exit the block header editor by selecting the menu option "Special/End".
9. Select the menu option "Editors/Block in IL".

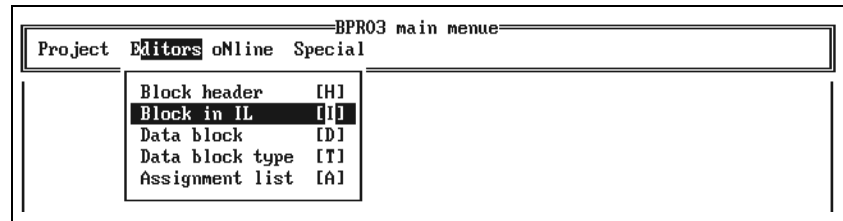


Fig. 4-10 Calling the IL editor

10. Enter the program instructions for the function block type.
11. Exit the IL editor by selecting the menu option "Special/End".



NOTE

For more detailed information on calling and using the editors, refer to chapter 3.5.

4.2.3 Declaring a function block

1. Select the menu option "Editors/Block header" and load an existing block header or create a new one.

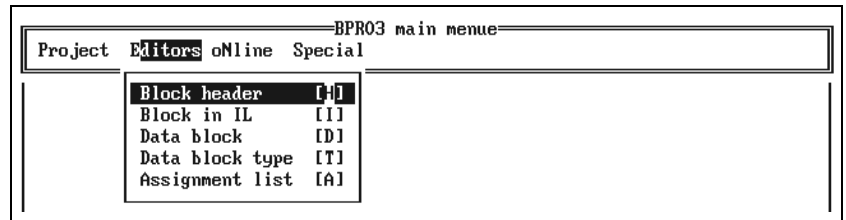


Fig. 4-11 Calling the block header editor

2. Select the menu option "Order/VAR" to insert a declaration section.

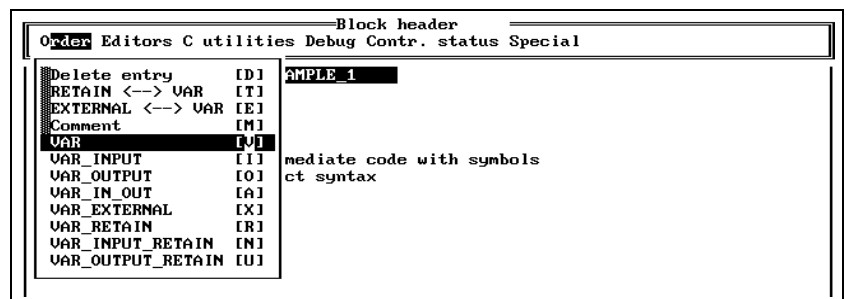


Fig. 4-12 Selecting the menu option "Order/VAR"



NOTE

If a VAR - VAR_END section already exists, you only need to insert a new line.

4. Enter the name of the function block in the first column.
5. Enter the name of the function block type in the second column.

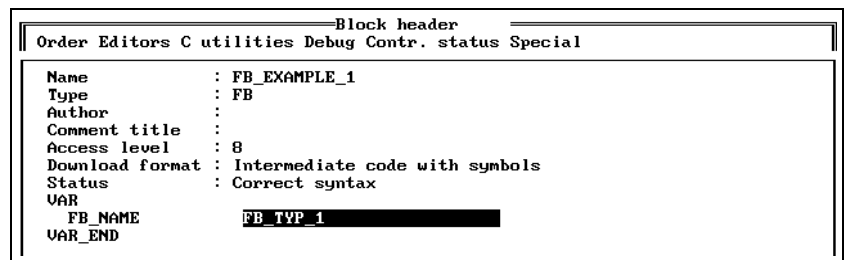


Fig. 4-13 Entering name and function block type

6. If required, insert empty lines by pressing <↵> and declare additional function blocks as necessary.
7. Exit the block header editor by selecting the menu option "Special/End".

4.3 Programming global blocks

Global blocks can be called by program blocks, function blocks and other global blocks without having to be declared previously in the block header.

The other characteristics of global blocks are the same as those of function blocks.

4.3.1 Valid variable types and block calls

Valid variable types

In a global block, all types of variables can be used.

Designation	Description
VAR	Local variable or function block.
VAR_INPUT	Input variable for accepting a value from the calling block.
VAR_OUTPUT	Output variable for transferring values to the calling block.
VAR_IN_OUT	Input and output variable.
VAR_EXTERNAL	Global variable which is identified to the PGR with the declaration.
VAR_RETAIN	Same as VAR, but is retained after power-off.
VAR_INPUT_RETAIN	Same as VAR_INPUT, but is retained after power-off.
VAR_OUTPUT_RETAIN	Same as VAR_OUTPUT, but is retained after power-off.

Valid block calls

The following block types can be called in a global block:

- Function blocks
- Global blocks
- Functions



NOTE

For more information, see chapter 1.2.3.3 "Block calling" in the programming manual.

4.3.2 Creating a global block

To create a new global block, you first have to define a block header. For this purpose, call the block header editor in the BPRO3 programming system:

1. Select the menu option "Editors/Block header".
2. Enter a new block name.

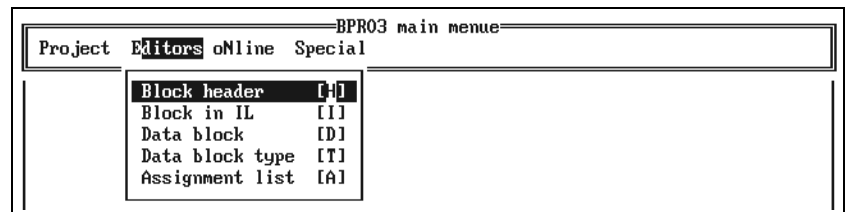


Fig. 4-14 Calling the block header editor

3. Confirm creating a new block by pressing "Y".
4. Enter or edit general information on the block.
For function blocks, the block type must be entered manually or using the option list function (<F3> key).

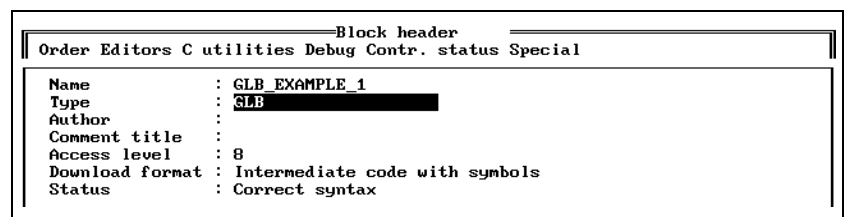


Fig. 4-15 Editing general block information

5. Use the pull-down menu "Orders" to insert the sections required for declaring variables, e.g. "VAR".

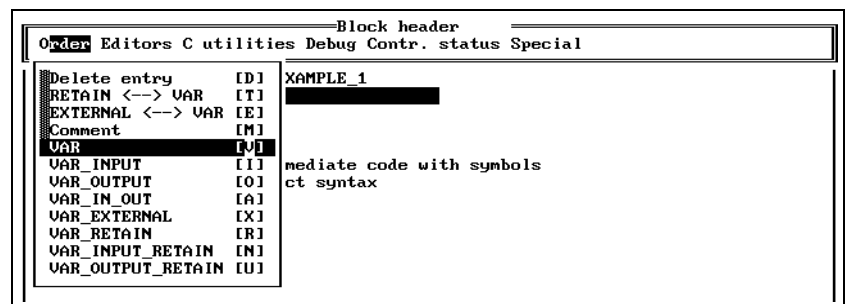


Fig. 4-16 The pull-down menu "Order"

6. Declare the variables.

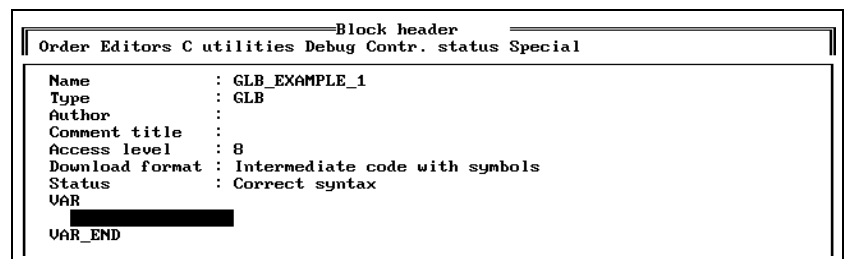


Fig. 4-17 Variable declaration

Block programming

7. Repeat steps 5 and 6 for each variable to be declared.
8. Exit the block header editor by selecting the menu option "Special/End".
9. Select the menu option "Editors/Block in IL".

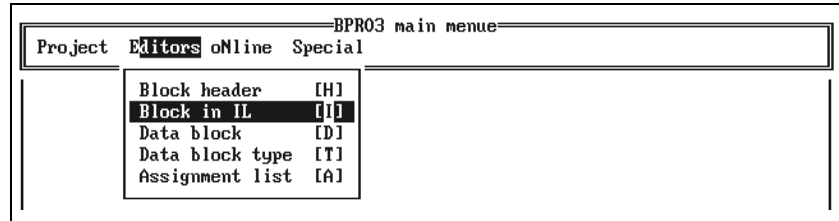


Fig. 4-18 Calling the IL editor

10. Enter the program instructions for the global block.
11. Exit the IL editor by selecting the menu option "Special/End".



NOTE

For more detailed information on calling and using the editors, refer to chapter 3.5.

4.4 Programming functions

Functions are program components which always return one single result. Their local variables are reinitialized with each call. For this reason, they cannot be used for storing status conditions from one call to the next. The same input value will always yield the same result. The result of a function is stored in the CR.

The output variable for the result is generated automatically by the programming system.

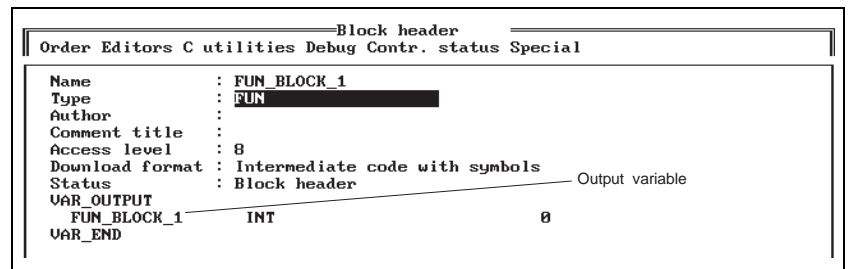


Fig. 4-19 Automatically created output variable

4.4.1 Valid variable types and block calls

Valid variable types

In a function, only certain types of variables can be used. All other variable types (which cannot be declared in a function) are marked with a "■" symbol.

	Designation	Description
	VAR	Local variable or function block.
	VAR_INPUT	Input variable for accepting a value from the calling block.
■	VAR_OUTPUT	Output variable for passing a value to the calling block.
	VAR_IN_OUT	Input and output variable.
■	VAR_EXTERNAL	Global variable which is identified to the PRG with the declaration.
■	VAR_RETAIN	Same as VAR, but is retained after power-off.
■	VAR_INPUT_RETAIN	Same as VAR_INPUT, but is retained after power-off.
■	VAR_OUTPUT_RETAIN	Same as VAR_OUTPUT, but is retained after power-off.

Valid block calls

A function cannot call any other blocks.

Block programming

4.4.2 Creating a function

To create a new function, you first have to define a block header. For this purpose, call the block header editor in the BPRO3 programming system:

1. Select the menu option "Editors/Block header".

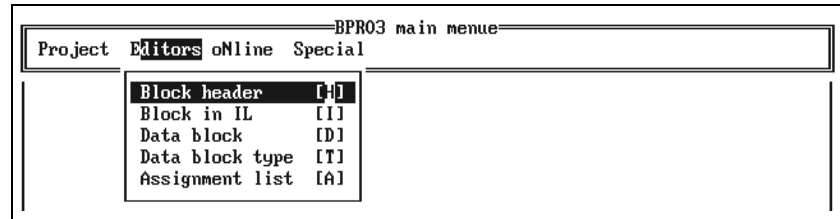


Fig. 4-20 Calling the block header editor

2. Enter a new block name.
3. Confirm creating a new block by pressing "Y".
4. Enter or edit general information on the block.
For function blocks, the block type must be entered manually or using the option list function (<F3> key).

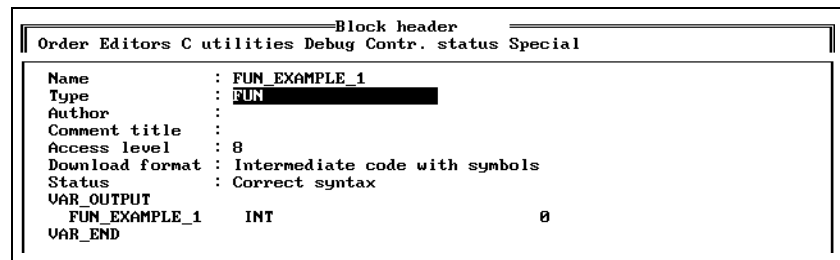


Fig. 4-21 Editing general block information

5. Use the pull-down menu "Orders" to insert the sections required for declaring variables, e.g. "VAR".

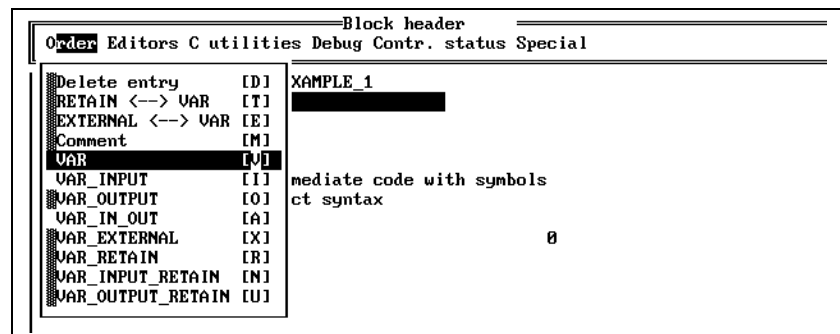


Fig. 4-22 The pull-down menu "Order"

6. Declare the variables.

```

Block header
Order Editors C utilities Debug Contr. status Special
Name      : FUN_EXAMPLE_1
Type      : FUN
Author    :
Comment title :
Access level : 8
Download format : Intermediate code with symbols
Status    : Correct syntax
VAR_OUTPUT
FUN_EXAMPLE_1 INT 0
VAR_END
VAR
VAR_END
    
```

Fig. 4-23 Variable declaration

7. Repeat steps 5 and 6 for each variable to be declared.
8. Exit the block header editor by selecting the menu option "Special/End".
9. Select the menu option "Editors/Block in IL".

```

BPRO3 main menu
Project Editors online Special
Block header [H]
Block in IL [I]
Data block [D]
Data block type [T]
Assignment list [A]
    
```

Fig. 4-24 Calling the IL editor

10. Enter the program instructions for the global block.
11. Exit the IL editor by selecting the menu option "Special/End".



NOTE

For more detailed information on calling and using the editors, refer to chapter 3.5.

4.5 Programming data blocks

Data blocks do not contain any program elements. They only contain data which can be accessed from any block (global data).

The structure of a data block is defined with the DB type editor. This structure cannot be used directly for storing data. It is designed as a template for creating data blocks using the DB editor, specifying the name of a DB type. The structure is copied from the template (DB type) and a name is assigned to the data block.

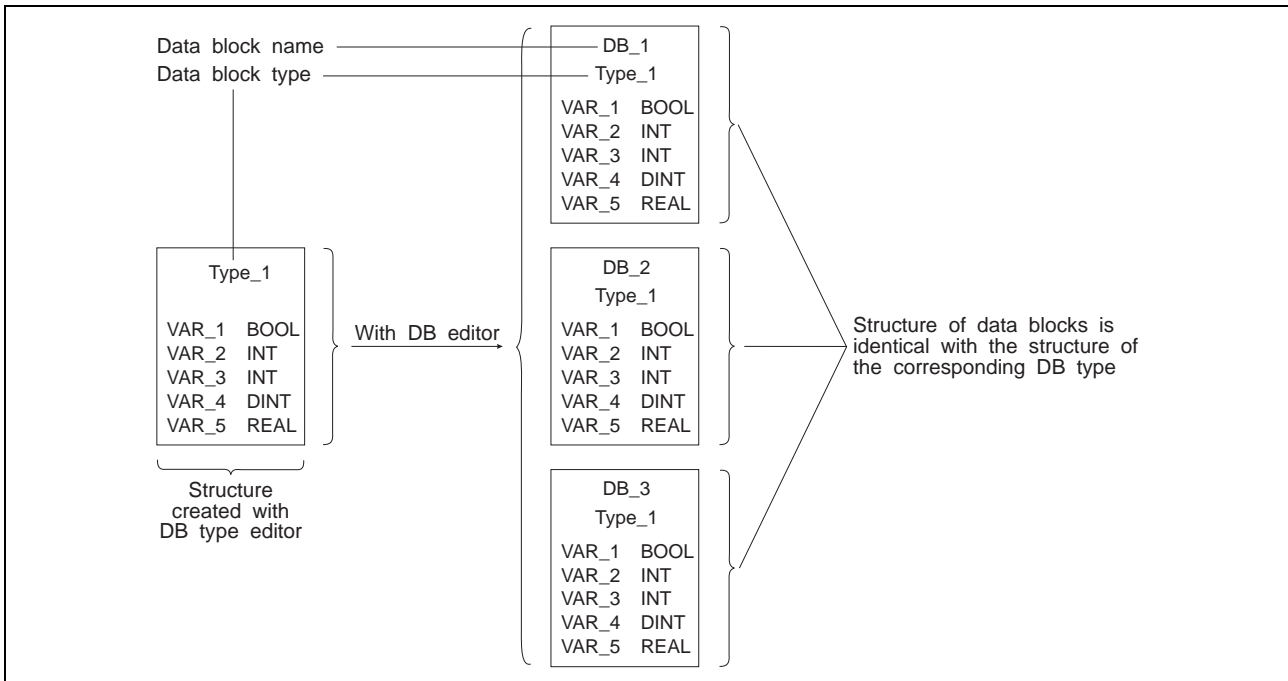


Fig. 4-25 Data block type and data block structures

4.5.1 Creating a data block

To create a new data block, you first have to define a structure using the DB type editor. For this purpose, call the DB type editor in the BPRO3 programming system:

1. Select the menu option "Editors/DB type editor".
2. Enter a new block name (for DB type block or data structure).

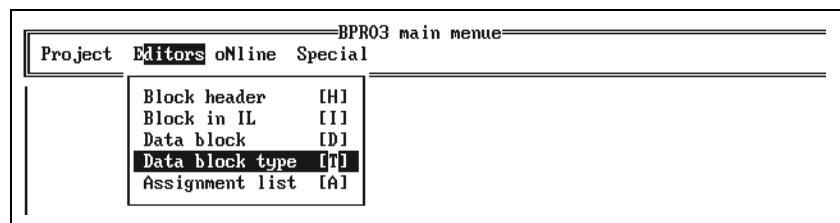


Fig. 4-26 Calling the DB type editor

3. Confirm creating a new block by pressing "Y".

4. Enter or edit general information on the block.

Fig. 4-27 Editing general block information

```

-----Data block/type-----
Orders Debug Contr. status Special
Data block type : DB_TYP_TEST_1
Author          :
Access level   : 8
Comment title  :
    
```

5. Use the pull-down menu "Order" to insert a section into the editing area for defining the data structure.

Fig. 4-28 Selecting "Insert variable"

```

-----Data block/type-----
Orders Debug Contr. status Special
Delete block   [D] VP_TEST_1
Insert block   [I]
Copy block     [C]
Comment       [M]
Insert variable [V]
    
```

6. Enter the data structure.

Fig. 4-29 Entering the data structure

```

-----Data block/type-----
Orders Debug Contr. status Special
Data block type : DB_TYP_TEST_1
Author          :
Access level   : 8
Comment title  :
TYPE
DB_TYP_TEST_1 : STRUCTURE
               INT           0           0           0
END_STRUCTURE
END_TYP
    
```

7. Move the cursor to the desired variable and select "Orders/Comment" to enter the comment title.

Fig. 4-30 Selecting "Comment"

```

-----Data block/type-----
Orders Debug Contr. status Special
Delete block   [D] VP_TEST_1
Insert block   [I]
Copy block     [C]
Comment       [M]
Insert variable [V]
STRUCTURE
TEST_VAR_1    BOOL           0           0           1
TEST_VAR_2    LREAL          0.0         23.0        1000.0
TEST_VAR_3    BOOL           0           0           1
TEST_VAR_4    INT            0           35          100
END_STRUCTURE
END_TYP
    
```

8. Enter the comment title and select "Orders/Comment" again to call the comment editor (see chapter 3.2.3.3).

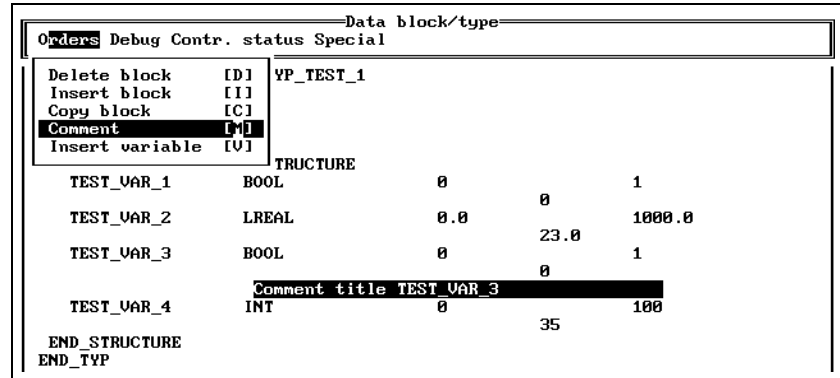


Fig. 4-31 Calling the comment editor

9. Enter a comment.
10. Exit the comment editor by selecting the menu option "Special/End".
11. Repeat steps 7 to 10 until all comments are entered.
12. Exit the DB type editor by selecting the menu option "Special/End".
13. Select the menu option "Editors/Data block".

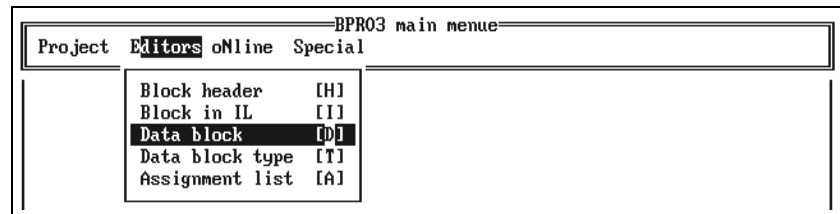


Fig. 4-32 Calling the DB editor

14. Enter the name of the DB type block.
15. Enter a new block name (for DB block).
16. Confirm creating a new block by pressing "Y". This will copy the structure of the DB type block and display it in the DB editor.
17. Enter or edit general information on the block.

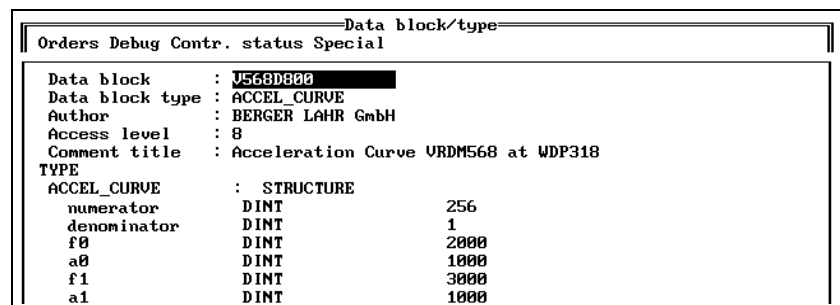


Fig. 4-33 Entering general information

18. Modify the initial values of the variables, if required.

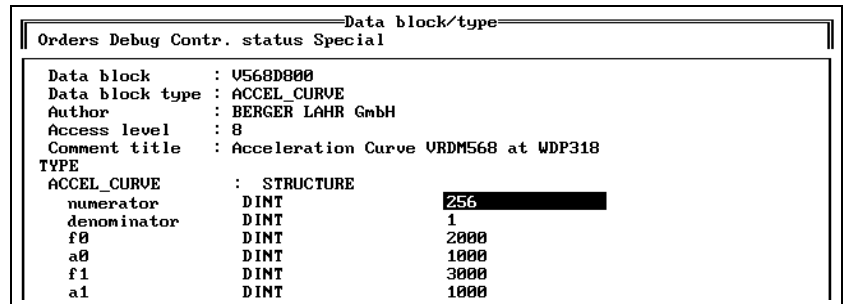


Fig. 4-34 Entering initial values

19. Exit the DB editor by selecting the menu option "Special/End".
20. You can repeat steps 8 to 14 until the required number of data blocks with the structure defined in the DB type editor has been created.



NOTE

For more detailed information on calling and using the editors, refer to chapter 3.5.

4.6 Creating an assignment list

In the assignment list editor, the inputs, outputs and flags can be assigned arbitrary names (see also chapter 1.2.2.3 in the programming manual).

To create an assignment list, proceed as follows:

1. Select the menu option "Editors/Assignment list".

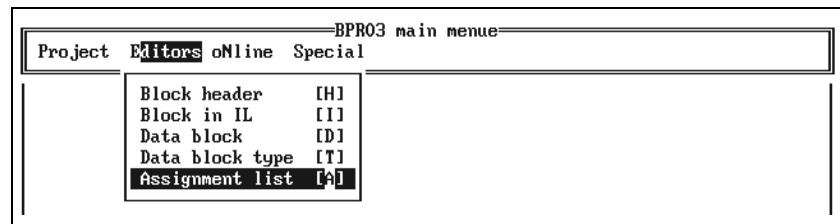


Fig. 4-35 Calling the AL editor

2. When the assignment list is displayed, move the cursor to its end.
3. Enter the symbolic name into the first column.
4. Enter the address (e.g. %MW10) into the second column.
5. Enter the data type into the third column or select it using the option list function (<F3> key).
6. Enter the initial value into the last column.

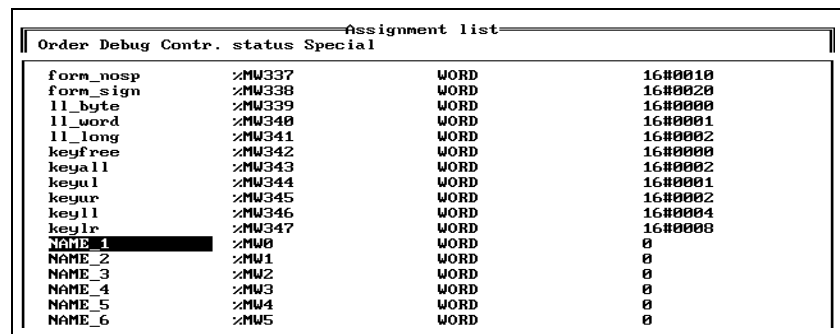


Fig. 4-36 Defining assignments

7. Select the menu option "Order/Comment".

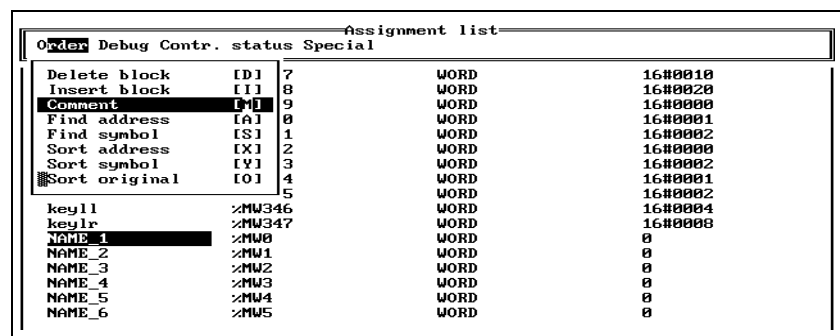


Fig. 4-37 Inserting a line for comment title

8. Enter a comment title. Selecting the menu option "Order/Comment" again will invoke the comment editor (see also "Comment editor" in chapter 3.2.3.3).

NAME_1	%MW0	WORD	0
NAME_2	%MW1	WORD	0
NAME_3	%MW2	WORD	0
NAME_4	%MW3	WORD	0
NAME_5	%MW4	WORD	0
NAME_6	%MW5	WORD	0

Fig. 4-38 Entering a comment title

9. Repeat steps 2 to 8 until all assignments have been entered.
10. Exit the assignment list editor by selecting the menu option "Special/End".

5 Data backup

You should always keep a backup copy of your project data by storing a copy of them.

For example:

COPY C:\BPRO3\TEST1*. * A:



ATTENTION

BERGER LAHR cannot be held responsible for any loss of data.

6 Error messages

BPRO3 displays error messages in plain text on the screen. To facilitate identification in case of inquiries, the errors are assigned error codes which are structured according to error categories.

Range	Description
1000 to 1999	Controller errors which are registered in the error memory; see chapter 6.1. For information on how to display the controller error memory, see chapter 3.6.5.6.
2000 to 2999	Errors occurring in "Online" mode.
3000 to 3999	Programming errors, e.g. commands which cannot be executed or syntax errors.
4000 to 4999	Internal errors related with the DOS operating system, e.g. corrupted files.



NOTE

For error messages in the range from 2000 to 4999 you can display explanatory text by pressing <F1>.

Error messages

6.1 Errors stored in error memory

The following table describes the errors which may be stored in the error memory.

All errors not listed here are system errors and should be reported to BERGER LAHR (service department), indicating the error code and the error text.

Error stored in error memory	Possible cause
1105 User block not loaded	A block is called in the program which has not been loaded into the controller or has been deleted there. Use "Download project" or "Update project" to rectify the program error.
1106 Library block not loaded	A library block is called in the program which has not been loaded into the controller or has been deleted there. Use "Download project" or "Update project" to rectify the program error.
1107 Data block not available	A data block is called in the program which has not been loaded into the controller or has been deleted there. Use "Download project" or "Update project" to rectify the program error.
1121 System error, no more memory	The available user space in memory is insufficient. Try to solve the problem by optimizing the memory using "Download project".
1122 Sys., max. number of objects reached	The maximum number of user-loadable objects is reached. This problem can usually be solved only by booting the controller.
1123 Sys. error, element cannot be read	An attempt was made to read a user object which does not exist on the controller. You should use "Download project".
1127 Application program: Division by zero	A division instruction in the application program requested division by zero. This program error can usually be located by selecting the menu option "Online/Contr. status/Task info" (current blocks).
1128 System: Invalid opcode	System error. The controller detected an invalid instruction in the application program loaded. Inform BERGER LAHR (service department) about errors of this type, stating the error code and the error text. In order to facilitate locating, analyzing and rectifying the error, it is also very important for BERGER LAHR to know the program (project) and, if possible, the block which caused the error. To determine the relevant block, select the menu option "Online/Contr. status/Task info all" in BPRO3 immediately after the error occurred and read the current block. You should also try to load such blocks in object code.

Error stored in error memory	Possible cause
1129 System: Invalid address range	<p>System error. The controller detected an invalid instruction in the application program loaded. Inform BERGER LAHR (service department) about errors of this type, stating the error code and the error text.</p> <p>In order to facilitate locating, analyzing and rectifying the error, it is also very important for BERGER LAHR to know the program (project) and, if possible, the block which caused the error. To determine the relevant block, select the menu option "Online/Contr. status/Task info all" in BPRO3 immediately after the error occurred and read the current block. You should also try to load such blocks in object code.</p>
1130 Application program: Invalid data type in instruction	<p>System error. The controller detected an invalid instruction in the application program loaded. Inform BERGER LAHR (service department) about errors of this type, stating the error code and the error text.</p> <p>In order to facilitate locating, analyzing and rectifying the error, it is also very important for BERGER LAHR to know the program (project) and, if possible, the block which caused the error. To determine the relevant block, select the menu option "Online/Contr. status/Task info all" in BPRO3 immediately after the error occurred and read the current block. You should also try to load such blocks in object code.</p>
1131 Application program: Array range exceeded	<p>An array index exceeds the predefined range in an attempt to access an array variable in the application program. This program error can usually be located by selecting the menu option "Online/Contr. status/Task info" (current blocks).</p>
1139 Resource/interface not available	<p>The resource parameter of a library function identifies an interface which does not exist. Refer to the "Code" column of the error memory output for the library function causing the error.</p>
1140 Invalid lib. function for resource	<p>The resource parameter of a library function identifies an interface for which this library function is invalid. Refer to the "Code" column of the error memory output for the library function causing the error.</p>
1141 Resource/interface not ready	<p>A resource/interface must not be addressed at this stage. This status is especially likely when debugging the application program.</p>
1142 Parameter value incorrect	<p>A library function parameter is incorrect. Refer to the "Code" column of the error memory output for the library function causing the error.</p>
1143 Pre-requisite not fulfilled	<p>A pre-requisite for execution of a command is not fulfilled. For example: A movement to an absolute position is to be performed with the actual position not defined. Refer to the corresponding command descriptions for the pre-requisites to be fulfilled.</p>
1144 Value cannot be calculated	<p>An error occurred when calculating a value. This error is particularly likely in normalizing operations.</p>
1145 Insufficient information on source	<p>The reference variable source must be correctly defined when changing to position following mode.</p>
1146 Option parameter error	<p>The option parameter is not relevant for this function.</p>

Error messages

Error stored in error memory	Possible cause
1147 Positive limit switch not defined	An attempt was made to interrogate the position of the positive limit switch using "getpos". However, there was no previous movement to this limit switch.
1148 Negative limit switch not defined	An attempt was made to interrogate the position of the negative limit switch using "getpos". However, there was no previous movement to this limit switch.
1149 LIB function only allowed at standstill	This function may only be executed at axis standstill. The axis status can be determined using the "getstate" function.
1150 Acceleration not yet defined	A movement was initiated without having defined an acceleration curve. Acceleration is defined by setting the maximum system frequencies (setvel, system).
1151 Master curve error	The master curve passed contains values which cannot be calculated.
1152 Actual position not yet defined	A reference movement has not yet been carried out, or dimensions have not yet been set.
1153 External source active	The drive is in position following mode. The command is invalid in this operating mode.
1154 Wait positioning (posw,movew) active	The command is invalid during a positioning operation with pos(d)w, pos(d)f, move(d)w, move(d)f.
1155 Drive interrupted or blocked	The command cannot be executed since the drive is interrupted or blocked (e.g. due to stop or limit switch). For more detailed information, select the BPRO3 menu option "Axis info" or use the "getstate" program function.
1156 Encoder not ready	The encoder is already used for another function. For example, it is not possible to use rotation monitoring and position following mode simultaneously with the same encoder.
1157 Calculation error in position following mode	Calculation error in position following mode. This error may occur when very great gear ratios are used.
1158 Error on processing the position list	An error occurred while processing a position list. Check the position list.
1159 Reference movement active	A reference movement is active. The command used is invalid during a reference movement.
1162 Reference movement error for /LIMP	An error occurred during the reference movement towards the positive limit switch. This error may occur due to incorrect wiring or limit switch bouncing.
1163 Reference movement error for /LIMN	An error occurred during the reference movement towards the negative limit switch. This error may occur due to incorrect wiring or limit switch bouncing.
1164 Application program: Target string variable too small	A store instruction (st) in the application program with a string variable as the target contains a string variable which is too small. This program error can usually be located by selecting the menu option "Online/Contr. status/Task info" (current blocks).
1165 Cycle monitoring time exceeded	Cycle time monitoring is enabled for the PLC task. A time value of 2 s is set by default. To solve the problem, either increase the monitoring time using "cycletime" or eliminate inadmissible program loops or WAIT commands from the PLC task.

Error stored in error memory	Possible cause
1166 Error in interface initialization param.	An initialization parameter error occurred, e.g. invalid value or invalid parameter for the current interface type.
1167 Mem. for r/t buffer not created	Error in creating the receive/transmit buffer for communication through the serial interface, e.g. due to insufficient system memory.
1168 Interface not initialized (com_init call)	Call to a manufacturer-defined function addressing the serial interface without having initialized it.
1169 Interface otherwise assigned	When accessing the interface, it was found that it is already used by another function, e.g. BERGER LAHR network.
1170 Receiving string buffer too small	The size of the string buffer available for storing the received string is too small. Enlarge the receive buffer.
1171 Transmitting string buffer too small	The current size of the string to be output exceeds the total size of the transmit buffer. Enlarge the transmit buffer.
1172 Invalid output value for display	A value to be displayed on the seven-segment display on the controller front panel is invalid. Refer to the "Code" column of the error memory output for the library function causing the error.
1176 Operating mode change only possible in STOP	Changing the operating mode is only possible in controller STOP or RESET status. The current controller status, however, is RUN or DEBUG.
1177 NW address setting only possible in STOP	Changing the network address is only possible in controller STOP or RESET status. The current controller status, however, is RUN or DEBUG.
1180 Start key pressed with application program running	The RUN key was pressed during application program execution.
1181 No application program loaded	The controller can only be started when a valid application program is loaded in the memory.
1182 Operating mode "Application mode" not set	The operating mode "Application mode" must be set on the front panel in order to be able to start the controller.
1186 Operating mode change only in STOP state	Changing the operating mode is only possible in controller STOP or RESET status. The current controller status, however, is RUN or DEBUG.
1187 System: Display value erroneous	The specified value cannot be displayed on the seven-segment display of the controller.
1188 Transmission string not converted	A combination of characters which is invalid for character conversion was found in the transmit string, e.g. \$X or \$0X.
1189 Prog. cannot be continued in ERROR state	The controller is in ERROR status. To be able to start an application program, you first have to RESET the controller using BPRO3 or the STOP key.
1190 Battery voltage too low	When this message is displayed, the battery should be replaced as soon as possible.
1191 Short-circuit at 24 V output	There is a short-circuit on one or more outputs. Check connector and wiring, switch controller off and on again.

Error messages

Error stored in error memory	Possible cause
1192 Invalid output address (@Q)	Direct output (@Q) incorrectly addressed by an application program block loaded in object code. Inform BERGER LAHR (service department) about errors of this type, stating the error code and the error text. Try to load the blocks in pseudo-code.
1193 Invalid input address (@I)	Direct input (@I) incorrectly addressed by an application program block loaded in object code. Inform BERGER LAHR (service department) about errors of this type, stating the error code and the error text. Try to load the blocks in pseudo-code.
1214 Rotation monitoring error	The initialized rotation monitoring feature triggered an alarm. Steps are lost on the drive.
1215 Encoder error (line interruption)	The encoder connector is incorrectly wired; see connector description in controller manual.
1216 Power controller ready error	The power controller does not return a ready signal. Check the power controller.
1217 Power controller overtemperature	The power controller reports overtemperature. Take steps to cool the power controller (or reduce the output current).
1218 Motor overtemperature	The motor reports overtemperature. Take steps to cool the motor (or reduce the output current).
1219 Sine: Input parameters invalid	The function was called with values which cannot be processed.
1220 Cosine: Input parameters invalid	The function was called with values which cannot be processed.
1221 ArcSine: Input parameters invalid	The function was called with values which cannot be processed.
1222 ArcCosine: Input parameters invalid	The function was called with values which cannot be processed.
1223 Square root: Negative radiant	The function was called with values which cannot be processed.
1224 Logarithm(s): Argument invalid	The function was called with values which cannot be processed.
1225 Logarithm(10): Argument invalid	The function was called with values which cannot be processed.
1226 Tangent: Input parameter invalid	The function was called with values which cannot be processed.
1227 Exponential func.: Input value too large	The function was called with values which cannot be processed.
1228 Lib. function only allowed in PTP mode	The drive is not in point-to-point mode.

Error stored in error memory	Possible cause
1229 Software limit switch error	This error may occur when the software limit switches are enabled and <ul style="list-style-type: none">- the actual position is set outside the software limit switch range;- the software limit switches are set in such a way that the actual position is in an invalid range;- the positive software limit switch value is less than the negative one;- the negative software limit switch value is greater than the positive one.
1230 Option constant LOCK for KEYS incorrect	Invalid option parameter for the "geststate keys" function.
1231 Present program not stored in EEPROM	The application programs in EEPROM and in the application program memory differ from each other.
1232 /LIMP and /LIMN are active	The function cannot be executed since the limit switches are active.
1236 Interpolation is active	The axis is in use by the interpolator. No other positioning commands may be issued to the axis during interpolation.
1249 Limit switch not enabled	When a reference movement is started towards a limit switch, the limit switch must be active (enabled).
1251 Pre-requisite for axis (axes) not fulfilled	A pre-requisite for interpolation is not fulfilled. Refer to the interpolation commands for the applicable pre-requisites.
1253 Interpolation interrupted due to axis error	One or more axes were interrupted.
1255 Incorrect field bus initialization	The initializing function "ibsm_init" received incorrect parameter values, e.g. the node list does not match the connected devices.
1257 Field bus not initialized	The field bus was not initialized by "ibsm_init".
1258 Incorrect field bus cycle	Transmission error detected on the field bus.
1260 Field bus link timeout	The master stopped operation or the ring is disrupted.
1261 Field bus cycle lost	

Error messages

7 Appendix

7.1 Glossary

Block

A block is a (sub)entity of an application program. It consists of a *block header* and a *block body*. The following block types are defined:

Program (block)

Function block

Function

Global block

Data block

Block body

The block body contains the instructions for the control program.

Block header

The block header includes information on the block itself (such as name, type, author, etc.) as well as declarations of variables and possibly function blocks.

Controller error memory

Runtime errors are written to the controller's error memory and indicated in the controller's status display. A maximum of 16 errors can be stored in the controller error memory (the first 8 and the last 8 errors which occurred).

Data blocks (DB)

A data block is a special type of block. It does not contain any *instructions* but an arbitrary number of *variables* which can be accessed from any block (*global variables*).

Data types

Each variable has a certain data type, e.g.

INT Only integer values.

REAL Numerical values with decimal places.

Drive units

Drive units are processing parameters internal to the controller; they are defined as follows:

Position-related drive units = Motor steps

Speed-related drive units = Motor steps/s/256

Acceleration-related drive units = Motor steps/s².

Flags

Flags are storage elements which can be accessed from any block. The controller has a dedicated memory area for flags; this area can be specified using the programming system (controller configuration).

Function blocks (FB)

Function blocks have the same characteristics as global blocks, however, they have to be declared in the block headers of the calling blocks, i.e. a function block type is assigned a name and allocated memory space for variables. This allows you to use several function blocks of the same type but with different names and allocated memory within a program or block.

Functions (FUN)

Functions (e.g. "sin") can be used by any block. A function can have several input variables (the first input variable is always stored in the CR), however, it returns only one result. The result is passed to the calling block via the current result (CR). The local variables of functions are reinitialized after each function call. The same input therefore always returns the same result. A function does not need to be declared and is always called with the same name.

Gear ratio

Multiplication factor for positioning operations, which is composed of a numerator and a denominator (step-down gearing or step-up gearing).

Global blocks (GLB)

Global blocks can be called from program blocks, function blocks and other global blocks. They contain input/output variables for passing data to the calling block. Global blocks can be used for storing status conditions, i.e. the local variables of global blocks are not re-initialized each time the global block is called.

Global data

Data which can be accessed from any block.

Global variables

Global variables are storage elements which can be accessed from any block.

Hardware configuration (controller configuration)

The controller configuration describes the type and the hardware components of the controller which is to execute the program. This enables the programming system to check the compatibility of program and controller.

IEC 1131-3 standard

The IEC 1131-3 standard describes:

- Program organization units (blocks)
- Control of program execution (tasks)
- Programming languages, or means of program representation (e.g. IL)
- Program data (variable declaration, data types, access paths).

IL networks

A "network" is a sequence of functionally related IL instructions. An IL program can consist of several "networks". You can freely define how many instructions are combined to form a "network".

INIT task

The INIT task controls the customer-specific default settings of the controller (e.g. output initialization, drive presettings, etc.). Program blocks assigned to the INIT task are executed once after program start. Other tasks cannot be executed until the INIT task has been processed.

Initial values

A variable must be allocated memory space and assigned an initial value. Assigning an initial value is called initialization of the variable.

Input/output variables

Input/output variables are storage elements which are used for passing data between blocks (e.g. between program blocks and function blocks). Input/output variables must be declared in the block header of the block to be called with VAR_INPUT, VAR_OUTPUT or VAR_IN_OUT.

Inputs/outputs

The controller is provided with a certain number of inputs and outputs through which sequential operations are controlled. Input/output processing can be effected simultaneously with the execution of movements.

Instruction list (IL)

Instruction list (IL) is a textual programming language for PLC programming. IL is characterized by a sequence of instructions consisting of one *operator*, one optional *modifier* and, if required for the operator, one or more *operands*.

Instruction pointer

Indicates the point in a program which is currently processed, or where processing was stopped.

Limit switch

Switch for limiting the travel and for reference movements.

Local variables

Local variables are storage elements which can be accessed only from within a block. Local variables are declared in the block header with VAR.

Logical operations

The operators “and”, “or” and “xor” can be used to perform logical operations with one or more boolean operands and the CR.

Offline mode

In this mode, the programming system is used without being linked to a controller.

Online mode

In this mode, the link to the controller is established.

Operand

All IL instructions (except “ret”) require an operand with which the operation is performed.

Operator

The operator describes the operation which is to be carried out with the *operand* and the current result (CR).

PLC task

The PLC task is called after execution of the INIT task and is executed in cycles. When the PLC task is started, the inputs are read into the process image (input/output temporary storage). After this step, the program blocks of the PLC task are processed sequentially. Subsequently the changes in the output status conditions in the process image are output and the cycle restarts.

Process image (PI)

In the process image, the inputs/outputs are temporarily stored.

Program blocks (PRG)

Program blocks are program components which can be assigned to a task. Data exchange between the program blocks of a task is effected via global variables (e.g. flags). Within a program block, function blocks, global blocks and functions can be called.

Reference movement

Motor movement towards the r.h. or l.h. limit switch or additional reference switch for setting a reference point for the system of dimensions.

SEQUENCE task

The SEQUENCE task is executed simultaneously with the *PLC task* (according to the time-slice principle), however, SEQUENCE task program blocks are processed only once. It is recommended to use SEQUENCE tasks primarily for movement programming. The process image is updated by the PLC task.

Task

A task is a program organization unit similar to the organization block of a conventional programmable logic controller. It controls the way and the time of execution of program blocks. One or more program blocks can be assigned to a task. When a task is activated, the program blocks are processed in the order of their entries in the task assignment list (from top to bottom).

User-defined units

User-defined units are processing parameters which can be freely defined by the user. The following applies:

Drive units = User-defined units x Normalizing factor

Positions can be specified in *drive units* or in user-defined units.

Speeds and *maximum accelerations* (for calculating acceleration curves) can only be specified in user-defined units.

7.2 Abbreviations

SFC	Sequential function chart (language)
IL	Instruction list (language)
BNET	BERGER LAHR network
DB	Data block
DC	Direct current
Doc. no.	Documentation number
EEPROM	Non-volatile memory
FB	Function block
FBD	Function block diagram (language)
FUN	Functions
GLB	Global block
PC	Personal computer
POU	Program organization unit
PRG	Program block
RAM	Random access memory
PLC	Programmable logic controller
AL	Assignment list

8 Index

A

Access level	3-38
Access protection	3-38
Addressing inputs/outputs	3-31 - 3-34
Alt key	3-2
Assignment list editor	1-9
Axis info	3-104

B

Backup copies	2-4
Block	1-7
Block body	1-7
Block header	1-7, 3-42, 3-62
Block header editor	1-8
Block in IL	3-55
Block list	3-19
Block processing	3-12
Block type	3-44

C

C utilities	3-51
Change access level	3-38, 3-40
Change password	3-37
Character cursor	3-8
Comments	3-11
Conditional BP	3-89
Cont. controller	3-91
Contr. status	3-53, 3-64, 3-72, 3-81, 3-93, 3-100
Control programs	1-1
Controller configuration	3-28
Controller library	1-6, 3-19
Controller status	3-104
Controller types	3-30
Copy block	3-13
Ctrl key	3-2
Cursor key	3-3

D

Data block editor	1-9
Data block type editor	1-9
Data structures	1-7
Debug	3-26, 3-52, 3-63, 3-70, 3-80, 3-86
Debug/Step	3-71
Debug/Step	3-71
Debug/Step over	3-71
Delete all BPs	3-90
Delete block	3-12
Delete breakpoint	3-90
Delete label	3-60
Device address	3-1
Disp./change var.	3-94
Download format	3-44

E

Editing area	1-5, 3-2
Editors	1-8, 3-51, 3-62
Enable all	3-100
End	3-108
Error memory	3-105
Exiting a field	3-10
Expert shortcut	3-4
EXTERNAL \longleftrightarrow VAR	3-50

F

F keys	3-109
Field cursor	3-8
Find address	3-79
Find netw./label	3-60
Find symbol	3-79
Flag area	3-29
Function block diagram editor	1-8
Function key line	1-5 - 1-6, 3-2
Function keys	3-3

H

Help information	3-5
Highlight	1-5, 3-2
Hotkey	3-2, 3-4

I

IEC 1131-3	1-1
Insert	3-9
Insert block	3-13
Insert label	3-58
Insert network	3-58
Insert task	3-25
Inserting lines	3-10
Installation	2-4
Instruction list editor	1-8
Interface cable	2-1
Interface converter	2-1
Interface distributor	2-1
Intermediate code (pseudo-code)	3-44

L

Library	3-20
Log files	3-14

M

Macro	3-108
Macro	3-4
Menu area	1-5
Menu bar	1-5, 3-2
Menu option	1-5, 3-2
Move label	3-59

N

Network	1-1
---------	-----

O

Object code	3-44
Online	3-22, 3-82, 3-84
Open project	3-18 - 3-19
Operating functions	3-2
Option list function	3-2, 3-5
Option list window	3-6
Order	3-21, 3-24, 3-47, 3-57, 3-78
Orders	3-68
Output	3-25
Output parameter	3-46
Overwrite	3-9

P

Password	3-37
Print project	3-36
Programming device	1-1
Programming languages	1-8
Programming status	3-46
Proj. directory	3-19
Project	1-2, 3-17
Project block	3-20
Project data	1-4
Project path	3-19
Pull-down menu	1-5

R

Requirements	3-1
Reset controller	3-92
RETAIN	3-48
RETAIN \longleftrightarrow VAR	3-49

S

Sequential function chart language editor	1-8
Serial interfaces	2-3
Set breakpoint	3-88
Shift key	3-2
Single cycle	3-87
Special	3-22, 3-53, 3-65, 3-72, 3-81, 3-106 - 3-107
Standard library	1-6, 3-19
Status	3-46
Status information on objects	1-5
Status line	1-5, 3-2
Step	3-99
Step over	3-100
Stop controller	3-92
Store project	3-18
Symbolic names	1-7, 3-14
Syntax check	3-61

T

Task info	3-102
Task info all	3-103
Text editor	1-11
Transfer	3-83

U

Units of measure	3-35
------------------	------

V

View	3-85
------	------

