# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction</td>
</tr>
<tr>
<td>1.1</td>
<td>Overview</td>
</tr>
<tr>
<td>1.1.1</td>
<td>Program creation</td>
</tr>
<tr>
<td>1.2</td>
<td>Program structuring</td>
</tr>
<tr>
<td>1.2.1</td>
<td>Non-modular programming</td>
</tr>
<tr>
<td>1.2.2</td>
<td>Depiction with 907 PC 33</td>
</tr>
<tr>
<td>1.2.3</td>
<td>Modular programming</td>
</tr>
<tr>
<td>1.2.4</td>
<td>Depiction with 907 PC 33</td>
</tr>
<tr>
<td>1.3</td>
<td>Program input</td>
</tr>
<tr>
<td>1.4</td>
<td>Program test</td>
</tr>
<tr>
<td>1.5</td>
<td>Program structuring with 907 PC 33</td>
</tr>
<tr>
<td>1.5.1</td>
<td>Absolute and symbolic programming</td>
</tr>
<tr>
<td>1.5.2</td>
<td>Function Block diagram/ladder diagram (FBD/LD)</td>
</tr>
<tr>
<td>1.5.3</td>
<td>Extended instruction list (Ext. IL)</td>
</tr>
<tr>
<td>1.5.4</td>
<td>Standard features</td>
</tr>
<tr>
<td>1.6</td>
<td>Program test with 907 PC 33</td>
</tr>
<tr>
<td>1.7</td>
<td>Multiuser program creation</td>
</tr>
<tr>
<td>1.8</td>
<td>Top down program creation</td>
</tr>
<tr>
<td>1.8.1</td>
<td>Bottom up program creation</td>
</tr>
<tr>
<td>1.9</td>
<td>Programming environment</td>
</tr>
<tr>
<td>1.9.1</td>
<td>Programming unit (PC)</td>
</tr>
<tr>
<td>1.9.2</td>
<td>Data Backup</td>
</tr>
<tr>
<td>1.9.3</td>
<td>MS-DOS file CONFIG.SYS</td>
</tr>
<tr>
<td>1.9.4</td>
<td>Mouse Operation</td>
</tr>
<tr>
<td>2</td>
<td>Installation</td>
</tr>
<tr>
<td>2.1</td>
<td>907 PC 33 files</td>
</tr>
<tr>
<td>2.2</td>
<td>Reserve files</td>
</tr>
<tr>
<td>2.3</td>
<td>Auxiliary files</td>
</tr>
<tr>
<td>2.4</td>
<td>Subdirectory structure</td>
</tr>
<tr>
<td>2.5</td>
<td>Password protection</td>
</tr>
<tr>
<td>2.6</td>
<td>Password editor</td>
</tr>
<tr>
<td>2.7</td>
<td>Starting 907 PC 33</td>
</tr>
<tr>
<td>3</td>
<td>Data Backup</td>
</tr>
<tr>
<td>3.1</td>
<td>Principle</td>
</tr>
<tr>
<td>3.2</td>
<td>Saving data on diskette</td>
</tr>
<tr>
<td>3.3</td>
<td>Copy project</td>
</tr>
<tr>
<td>4</td>
<td>907 PC 33 menu interface</td>
</tr>
<tr>
<td>5</td>
<td>Project date</td>
</tr>
<tr>
<td>5.1</td>
<td>Introduction</td>
</tr>
<tr>
<td>5.2</td>
<td>Description of the fields</td>
</tr>
<tr>
<td>6</td>
<td>Configuration</td>
</tr>
<tr>
<td>6.1</td>
<td>Data area editor</td>
</tr>
<tr>
<td>6.2</td>
<td>Terminal operation</td>
</tr>
<tr>
<td>6.3</td>
<td>Color configuration</td>
</tr>
<tr>
<td>7</td>
<td>not used</td>
</tr>
<tr>
<td>8</td>
<td>Field editor</td>
</tr>
<tr>
<td>8.1</td>
<td>Commands</td>
</tr>
<tr>
<td>8.2</td>
<td>Number conversion</td>
</tr>
<tr>
<td>8.3</td>
<td>File overview</td>
</tr>
<tr>
<td>9</td>
<td>Modularization</td>
</tr>
<tr>
<td>9.1</td>
<td>Introduction</td>
</tr>
<tr>
<td>9.2</td>
<td>Basic elements of modularization</td>
</tr>
<tr>
<td>9.3</td>
<td>907 PC 33 menu interfaces</td>
</tr>
<tr>
<td>9.4</td>
<td>Subdirectory structure</td>
</tr>
<tr>
<td>9.5</td>
<td>Procedure for creating a modularized project</td>
</tr>
<tr>
<td>9.5.1</td>
<td>Creating a new modularized project</td>
</tr>
<tr>
<td>9.5.2</td>
<td>Dividing an excessively large program module</td>
</tr>
<tr>
<td>9.5.3</td>
<td>Modularization of a project because of too many variables</td>
</tr>
<tr>
<td>9.5.4</td>
<td>Modularization of a project because of too large PLC program</td>
</tr>
<tr>
<td>9.6</td>
<td>Modularization editor</td>
</tr>
<tr>
<td>9.7</td>
<td>Change module</td>
</tr>
<tr>
<td>9.8</td>
<td>Variable editor</td>
</tr>
<tr>
<td>9.9</td>
<td>Reserve configuration/module configuration</td>
</tr>
<tr>
<td>10</td>
<td>FBD/LD editor</td>
</tr>
<tr>
<td>10.1</td>
<td>Explanation of terms</td>
</tr>
<tr>
<td>10.2</td>
<td>Keys</td>
</tr>
<tr>
<td>10.3</td>
<td>Menu control</td>
</tr>
<tr>
<td>10.4</td>
<td>Management</td>
</tr>
</tbody>
</table>

907 PC 33/General Part/Issued: 11.98

0-1
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.5</td>
<td>Management (Searching)</td>
</tr>
<tr>
<td>10.6</td>
<td>Layouts</td>
</tr>
<tr>
<td>10.7</td>
<td>Connection elements (CEs)</td>
</tr>
<tr>
<td>10.8</td>
<td>Moving CE connections</td>
</tr>
<tr>
<td>10.9</td>
<td>Inverting CE connections</td>
</tr>
<tr>
<td>10.10</td>
<td>Duplicating CE connections</td>
</tr>
<tr>
<td>10.11</td>
<td>Deleting CE connections</td>
</tr>
<tr>
<td>10.12</td>
<td>Deleting CE connections</td>
</tr>
<tr>
<td>10.13</td>
<td>Deleting CE connections</td>
</tr>
<tr>
<td>10.14</td>
<td>Deleting/swapping CE</td>
</tr>
<tr>
<td>10.15</td>
<td>Setting initial contact</td>
</tr>
<tr>
<td>10.16</td>
<td>Inserting serial contact</td>
</tr>
<tr>
<td>10.17</td>
<td>Inserting parallel contact</td>
</tr>
<tr>
<td>10.18</td>
<td>Deleting contacts</td>
</tr>
<tr>
<td>10.19</td>
<td>Setting output contact</td>
</tr>
<tr>
<td>10.20</td>
<td>Vertical contact junctions</td>
</tr>
<tr>
<td>10.21</td>
<td>Inverting contacts</td>
</tr>
<tr>
<td>10.22</td>
<td>Display of variables</td>
</tr>
<tr>
<td>10.23</td>
<td>Entering variables</td>
</tr>
<tr>
<td>10.24</td>
<td>Direct CE Output</td>
</tr>
<tr>
<td>10.25</td>
<td>Creating connections</td>
</tr>
<tr>
<td>10.26</td>
<td>Deleting connections</td>
</tr>
<tr>
<td>10.27</td>
<td>Display junctions</td>
</tr>
<tr>
<td>10.28</td>
<td>Inserting/deleting lines</td>
</tr>
<tr>
<td>10.29</td>
<td>Inserting/deleting columns</td>
</tr>
<tr>
<td>10.30</td>
<td>Comments/Control commands</td>
</tr>
<tr>
<td>10.31</td>
<td>Searching</td>
</tr>
<tr>
<td>10.32</td>
<td>Search and replace</td>
</tr>
<tr>
<td>10.33</td>
<td>Dividing/linking segment plans</td>
</tr>
<tr>
<td>10.34</td>
<td>Handling blocks</td>
</tr>
<tr>
<td>10.35</td>
<td>Plausibility check</td>
</tr>
<tr>
<td>10.36</td>
<td>Error file</td>
</tr>
<tr>
<td>10.37</td>
<td>Compiler</td>
</tr>
<tr>
<td>10.38</td>
<td>Displaying the compiled IL</td>
</tr>
<tr>
<td>10.39</td>
<td>Saving project</td>
</tr>
</tbody>
</table>

### 11 Extended IL editor

| 11.1   | Introduction | 11-1 |
| 11.2   | Keys | 11-1 |
| 11.3   | Menu | 11-4 |
| 11.4   | Management | 11-4 |
| 11.5   | Management (Searching) | 11-7 |
| 11.6   | Layouts | 11-7 |
| 11.7   | Fields | 11-9 |
| 11.8   | Commands | 11-10 |
| 11.9   | Comments | 11-11 |
| 11.10  | Connection elements | 11-11 |
| 11.11  | Plausibility check | 11-12 |
| 11.12  | Duplicating CE connections | 11-13 |
| 11.13  | Inverting CE connections | 11-13 |
| 11.14  | Deleting CE connections | 11-14 |
| 11.15  | Deleting/Swapping CE | 11-14 |
| 11.16  | Searching | 11-15 |
| 11.17  | Search and replace | 11-16 |
| 11.18  | Splitting/linking segment plans | 11-17 |
| 11.19  | Block processing | 11-18 |
| 11.20  | Error file | 11-19 |
| 11.21  | Compiler | 11-19 |
| 11.22  | Saving data | 11-21 |

### 12 Variable editor

| 12.1   | Principle of operation | 12-1 |
| 12.2   | Input fields/Commands | 12-2 |
| 12.3   | Special commands | 12-2 |
| 12.4   | Single line variable editor | 12-4 |
| 12.5   | Variable export | 12-4 |
| 12.6   | Variable import | 12-5 |
| 12.7   | Importing data from CAD/CAE- systems | 12-7 |

### 13 Text constant editor

| 13.1   | Introduction | 13-1 |

### 14 Text editor

| 14.1   | Introduction | 14-1 |

### 15 Library

| 15.1   | Introduction | 15-1 |
| 15.2   | Display | 15-1 |
| 15.3   | Block functions | 15-3 |
| 15.4   | The CE group editor | 15-4 |
| 15.5   | How to proceed when creating CEs | 15-5 |
| 15.6   | Creating graphics of CEs | 15-7 |
| 15.7   | The plausibility group editor | 15-9 |
| 15.8   | CE Cache configuration | 15-13 |
| 15.9   | Creating the program run in a CE | 15-13 |
| 15.10  | Reading in manufacturer libraries | 15-39 |

### 16 Printing

| 16.1   | Introduction | 16-1 |
| 16.2   | General print parameters | 16-2 |
| 16.3   | Print format editor | 16-3 |
| 16.4   | FBD/LD list | 16-4 |
| 16.5   | Extended IL list | 16-8 |
| 16.6   | Instruction list | 16-9 |
| 16.7   | Variable list | 16-10 |
| 16.8   | Cross reference list | 16-11 |
| 16.9   | Library | 16-12 |
| 16.10  | Logic plan diagram | 16-13 |
| 16.11  | Ladder diagram | 16-14 |
| 16.12  | Comment list | 16-15 |
| 16.13  | Online list | 16-16 |
| 16.14  | Text pages | 16-16 |
| 16.15  | Data range | 16-17 |
| 16.16  | Main lists | 16-17 |
| 16.17  | Modularization list | 16-18 |
| 16.18  | Total variable list | 16-19 |
| 16.19  | Total cross reference list | 16-20 |

### 17 Batch processing

| 17.1   | Principle | 17-1 |
| 17.2   | Special characters/functions | 17-1 |
| 17.3   | Background recording | 17-2 |
18 Terminal mode ........................................ 18- 1
19 Command overview .............................. 19- 1
  19.1 Help / displayable character
     Cursor control / Block commands .......... 19- 1
  19.2 Searching / searching and replacing ....
     inserting / deleting / comments .......... 19- 3
  19.3 Special commands / ............................
     ONLINE commands ............................ 19- 5
  19.4 Function keys ................................. 19- 7
20 Guide to time-optimized working ...... 20- 1
  20.1 Introduction .................................. 20- 1
  20.2 Translate changes (without reserve) ... 20- 2
  20.3 Translate changes with reserve ......... 20- 3
  20.4 Translate changes ........................... 20- 4
  20.5 Translate all ................................ 20- 5
  20.6 Modularized projects ...................... 20- 5

Appendix
1 Initialization of the laser
   printer HP Laserjet 4 ......................... A- 1
2 Initialization of Kyocera
   laser printers .................................. A- 4

Index

Abbreviations

AC31  Advant Controller 31
IL    Instruction list
AWP   User program (Anwenderprogramm AWP)
FBD   Function block diagram
LD    Ladder diagram
PM    Program module
PLC   Programmable Controller
CE    Connection element
VM    Variable module
SFC   Sequential function chart
<CR>  Carriage Return, Enter, Return

1 Introduction

1.1 Overview

1.1.1 Program creation

The result of program creation is a tested PLC program. Program creation is subdivided into:

- Program structuring
- Program input
- Program test
1.2 Program structuring
Solving a complex automation task is facilitated by breaking it down into individual subtasks that are easy to solve.

- **Non-modular programming**
The overall project can be broken down into individual subtasks:

```
  Overall project
     /   \
  /     \ 
subtask subtask subtask
```

Example:

```
  automation task
     \
  /   \
broken down into:
  /     \ 
  |       |
subtask subtask subtask
  |      |     |     |
initialization part system 1 part system 2 communication
```

- **Modular programming**
The more complex an automation task is, the more difficult it is to solve it. Again, however, it is easier to solve a task if individual elements can be combined in a subtask.

Example:

```
  automation task
     \
  /   \
broken down into:
  /     \ 
  |       |
subtask subtask subtask
  |      |     |     |
initialization: part system 1: part system 2: communication:
part system 2: - monitoring - control - feed - palleting - operating station - master computer
```
1.3 Program input
Input of a program for a programmable controller (PLC) is split into several steps:

- Programming optionally in:
  - Function block diagram (FBD)
  - Ladder diagram (LD)
  - Instruction list (IL)

- Translation to instruction list (IL):
  - FBD
  - LD

- Sending the program to the PLC

1.4 Program test
During the program test, a check is generally made as to whether or not the program fulfills the demands placed on it. The program test can be part of commissioning.
Testing tools are, for example:
- Status display
- Single cycle
- Breakpoint

1.5 Program structuring with 907 PC 33
The module and segment plan concept of the 907 PC 33 programming and test software serves to structure complex PLC programs.
907 PC 33 offers the possibility of splitting a program into program modules (PM) and variable modules (VM). In turn, the program modules allow splitting into segment plans.
Thanks to subdivision of the overall project into modules and segment plans, a high degree of flexibility is guaranteed for program creation, modification and testing. Large projects can be structured clearly and are therefore easier to handle. This program design minimizes times needed for program modifications.
o Non modular programming

Using 907 PC 33, the overall project can be broken down into individual segment plans (subtasks):

Example:

The individual segment plans are linked to each other, thereby forming the overall project.

Note:
For effective work with the 907 PC 33, we recommend that small segment plans be used if possible, because this allows changes to be translated and sent more quickly.

Depiction with 907 PC 33

Non-modular programming corresponds to editing one single program module (PM) that is simultaneously the overall project. A variable module cannot be created in this case. There is only one working level. After you have called 907 PC 33, you are in this working level, the module editing level.
Modular programming

Individual segment plans (elements) constitute a program module (subtask). The overall project consists of individual program modules.

Example:

The individual segment plans are linked to each other, thereby forming the overall project.

Note:
For effective work with the 907 PC 33, we recommend that small segment plans be used if possible, because this allows changes to be translated and sent more quickly.
Depiction with 907 PC 33

Working levels in 907 PC 33:
- Overall project level
- Module management level
- Module editing level
**Explanatory notes**

**Overall project editing level**

An overall project is a complete PLC program. After you have called a modularized project in 907 PC 33 you are in the overall project level. The functions relating to the overall project, for example project management or printing in relation to the overall project, can be called in this level. Here a PM or VM can be selected in a menu for editing. This switches you to the module editing level. In the “Edit” option, the modularization editor can be called and you are switched to the module management level.

**Module management (modularization editor)**

In this level, the program modules (PM) and variable modules (VM) are defined and assigned to each other (-> modularization editor). Moreover, the sequence of the program modules is defined.

**Module editing level**

The individual program or variable modules are edited here. You can switch between the program and variable modules as you wish.

**Program modules**

Program and variable modules are means of structuring an overall project. This structuring into clearly arranged, e.g. functionally or logically coherent, entities facilitates program testing and commissioning. Program modules contain the program entered in the chosen programming language. In turn, the program modules are split into segment plans. When printed out, a segment plan results in 4 DIN A4 pages (see figure below).

Programming languages are:
- Sequential function chart (SFC)
- Function block diagram/ladder diagram (FBD/LD)
- Extended instruction list (Ext. IL)
Variable modules

Variable modules (VM) are also a means of structuring an overall project. Variable modules contain (structured) variable areas that communicate with program modules.
Example of “communication between program modules and variable modules”:

Creation of variable modules is left to the user’s discretion.

1.6 Program input with the 907 PC 33

- Programming optionally in:
  - Function block diagram/ladder diagram: FBD/LD
  - Extended instruction list: Ext. IL

- Translation
  - FBD/LD
  - Ext. IL *)

  *) Translation is not necessary if no connection elements are used from the library.

- Sending the program to the programmable controller (PLC)
Absolute and symbolic programming

The 907 PC 33 programming and test system supports symbolic programming of PLC programs. The user has the possibility of first of all entering the instruction list or variable list symbolically by specifying the variable type (e.g. E, A, M) and by later adding the group and channel numbers or input/output addresses.

Symbolic input:

**IL:**

- ! E Symbol Long text
- &NE Symbol Long text
- = A Symbol Long text

**Variable list:**

- E Symbol Long text
- E Symbol Long text
- A Symbol Long text

Absolute input:

**IL:**

- ! E 0.00,00 Symbol Long text
- &NE E 0.00,01 Symbol Long text
- = A 0.01,00 Symbol Long text

**Variable list:**

- E 0.00,00 Symbol Long text
- E 0.00,01 Symbol Long text
- A 0.01,00 Symbol Long text

Function block diagram/ladder diagram (FBD/LD)

A standard editor allows programming with graphic symbols in both a function block diagram (FBD) and also a ladder diagram (LD). Mixed programming is supported, i.e. linking of contact networks with function block diagram elements or vice versa (FBD/LD). Therefore, programming in FBD/LD is suitable equally for simple logic combination controls in the ladder diagram and also for complex functions and arithmetics in the function block diagram. Comments related to the network can be entered to simplify program description.

Extended instruction list (Ext. IL)

In contrast to the generally known instruction list (IL), the Extended instruction list (Ext. IL) has improved characteristics:

- Clear representation of instructions with a symbol and long text in various layouts
- Cursor control with a MOUSE
- Simple selection of graphic element functions (CE = Connection element) from the FBD/LD library with the aid of a selection menu
- More extensive syntax checking of all variables

Standard features

- Automatic generation of sentence and word numbers
- Automatic addition of variable addresses *)
- Automatic column subdivision
- Input of comments

*) Examples:

* Required variable: E 0.00,00
  Key strokes: <E><0><0><,><0>

* Required variable: A 0.10,03
  Key strokes: <A><1><0><,><3>

* Required variable: A 1.01,01
  Key strokes: <A><1><,><1><,><1>

* Required variable: MW 0004,01
  Key strokes: <M><W><4><,><1>
1.7 Program test with 907 PC 33

An extensive spectrum of tools is available for program testing.

Status display in:
- Function block diagram/ladder diagram
- Extended instruction list
- Variable list

Online program modifications

Online lists in which the statuses of selected variables can be displayed.

Overview of online functions:
- Start program
- Abort program
- Stop program
- Continue program
- Program status
- Single cycle on/off
- Single step on/off
- Set breakpoint
- Display breakpoint
- Delete breakpoint (single)
- Delete breakpoint (all)
- Display variable status
- Trigger to variable
- Trigger to time
- Jog
- Overwrite
- Variable list status display
- FORCE I/O variables
- Delete FORCE values
- Display FORCE values

The nature and scope of the online functions depend on the performance classes of the individual control systems used. You can find in-depth descriptions in the system-specific parts of the 907 PC 33 description for the respective control system.

System-specific parts of 907 PC 33 are:
- 907 PC 331 for ABB Procontic CS31, Advant Controller 31
- 907 PC 332 for ABB Procontic T200
1.8 Multiuser program creation

The 907 PC 33 programming and test system enables the user to make use of various methods of working when creating programs.

**Top down program creation**

The individual subtasks (= program modules) belonging to an automation task (= overall project) can be solved by different persons.
Bottom up program creation

The user first of all assumes that subdivision of his automation task into segment plans suffices. He later discovers that a very large number of variables has to be used and subdivision into programming modules will improve the arrangement. Even at this time 907 PC 33 allows subdivision into program modules and variable modules.

### 1.9 Programming environment

![Diagram of programming environment]

<table>
<thead>
<tr>
<th>target system</th>
<th>PC</th>
<th>printer</th>
<th>PROM programmer</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABB Procontic T200 (ZE)</td>
<td>07 PH 32</td>
<td>07 DR 12</td>
<td></td>
</tr>
<tr>
<td>ABB Procontic CS31</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Advant Controller 31 Series 90</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Advant Controller 31 Series 40 &amp; 50</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

07SK50

07SK90

07SK81

Centronics cable
Cable connections:

07 SK 61  Connection of 07 ZE 60, 07 ZE 61, 07 ZE 62, 07 ZE 63 to COM1 of the PC
(available as 07 SK 62 for a 9 pole serial interface of the PC)

07 SK 90  Connection of 07 KR 91, 07 KT 92/93/94, or 07 KP 62 to COM 1 of the PC

Centronics cable  Standard Centronics cable for connection of the 07 DR 12 printer to parallel port of the PC

07 SK 25  Connection of the PROM programmer 07 PP 31/32/33 to COM1 of the PC
(available as 07 SK 26 for a 9 pole serial interface of the PC)

07 SK 50  Connection of 07 CR/CT41, 07 KR/KT51 to COM1 of the PCs

1.9.1 Programming unit (PC)
The 907 PC 33 programming and test system is executable on:

Commercially available IBM compatible personal computers type 07 PH 32 (Compaq) with the following technical features:

- 80386 processor
- at least 4 MByte RAM
- Hard disk drive
- 3 1/2" 1.44 MByte floppy drive
- Serial interface EIA-RS-232 for the control system
- Parallel or second serial interface for a printer
- MS-DOS operating system V5.0 or higher
- Optional MOUSE for fast cursor control
- Cache memory to increase operating speed

Main memory
The 907 PC 33 adapts as far as possible to the existing memory conditions. However, there should be no other programs in the memory, such as network drivers or MS-DOS operator interfaces. A cache memory should always be installed since it increases the processing speed. This cache memory is included in the scope of delivery on the 07 PH 32.

The maximum available memory space for 907 PC 33 is displayed beneath the logo when invoked. If there is inadequate memory available, this is indicated by a corresponding message. The variable memory with a maximum length of 64000 bytes is crucial to the length of a program module. Symbolic variables and long texts are also stored in the variable memory. The number of variables increases or decreases, dependent upon whether symbolic variables and/or long texts are used. The maximum required memory space for a variable with symbol and long text is 61 bytes.

Example:

Information below applies per program module:

- 64000 bytes / 61 bytes = 1049 variables with symbol and long text
- 64000 bytes / 31 bytes = 2064 variables with symbol only
- 64000 bytes / 21 bytes = 3047 variables without symbol or long text

Hard disk drive
The programming and test system requires many disk access operations when operating. In order to keep access times as short as possible, it is advisable to set up a cache memory. Such programs are included in the scope of delivery of the MS-DOS operating system or are available from PC dealers or software houses.
1.9.2 Data backup

Besides the extensive data backup measures which 907 PC 33 performs automatically, you should always back up the project data on diskettes. 907 PC 33 offers a corresponding function for this.

1.9.3 MS-DOS file CONFIG.SYS

In order to ensure that the program runs optimally, it is necessary to set up an appropriate environment on the PC. 907 PC 33 should be used only on a PC which has at least an 80386 processor. Since the configuration of file CONFIG.SYS depends upon the hardware of the PC used, we are only able to give an example configuration for optimum operation below for an 80386 processor and MS-DOS V5.0.

```plaintext
DEVICE=C:\DOS\HIMEM.SYS
DEVICE=C:\DOS\EMM386.EXE noems
DOS=HIGH,UMB
DEVICEHIGH=C:\DOS\ansi.sys
DEVICEHIGH=C:\DOS\SMARTDRV.SYS 2048 512 /increase if there is more memory
DEVICEHIGH=C:\DOS\SETVER.EXE
COUNTRY=049,437,C:\DOS\COUNTRY.SYS
DEVICEHIGH=C:\DOS\DISPLAY.SYS CON=(EGA,,1)
DEVICEHIGH=C:\mouse.sys (a mouse driver in the AUTOEXEC.BAT should be loaded with LOADHIGH)
FILES=25
BUFFERS=25
```

1.9.4 Mouse operation

Using a mouse to operate 907 PC 33 and to control the cursor facilitates working with the programming and test system substantially. To do this, the mouse must be connected to a corresponding interface of the PC and the mouse driver must be installed in the PC. Here also, attention should be paid to using or installing a mouse driver in such a way that it uses the least amount of space in the PC’s main memory. Refer to the respective manufacturers’ manuals for information.
2 Installation

2. 907 PC 33 files

Note
The installation procedure and how to start the programming and test software for the respective ABB Procontic PLCs are described in detail in the respective system specific product binder.

Brief-description of the 907 PC 33 files:

Files of a specific PLC

Following files are necessary to start 907 PC 33:

PLCname.exe  Program for the PLC
PLCname.scr  Initial overlay
PLCname.cfg  Standard configuration file/Name of the last project
PLCname.hkf  Hardware (color) configuration
PLCname.tex  Messages of the program 907 PC 33
PLCname.hlf  Help texts and allocation of function keys
PLCname.mnu  Setup of the menu windows
PLCname.mmn  Setup of the main menu
PLCname.pas  Password file
PLCname.trs  Configuration for the syntax-compiler
PLCname_f.exe  Color configuration
PLCname_s.exe  Backup data
PLCname.ter  Interface parameters for terminal operation
PLCname.dfe  Standard print format for lists

Further files “PLCname.**” are needed with some PLCs.
Following files can exist optionally:

PLCname.fb  Connection element library (T200 only)

Project files

Depending on which editors were used and entered data following files belong to a project:

projectname.cfg  Project configuration
projectname.sym  List of variables
projectname.dbr  Data areas (from the data-area editor)
projectname.dfe  Print format (Head- and footlines)
projectname.dpm  Print parameters for all lists
projectname.err  Error file of the compilers
projectname.edp  Parameters for editor setup (e. g. scroll speed)
projectname.bld  Hardcopies (<Ctrl>-K-D) overtaken from 907 PC 33
projectname.hmp  Menu-setups (Main menu)
projectname.ref  References of variables (with modularized projects)
projectname.pas  Password release
projectname.oe  Online list
projectname.be  Breakpoint list
projectname.vex  Variable export / import
projectname.pg1  Index file  (Program code of FBD/LD, Ext. IL)
projectname.pg2  Reference file  (Program code of FBD/LD, Ext. IL)
projectname.pg3  Data  (Program code of FBD/LD, Ext. IL)
libraryname.ve1  Index file  (CE library)
libraryname.ve2  Reference file  (CE library)
libraryname.ve3  Data  (CE library)
For the ABB Procontic T200 a file
    projectname.cod   PLC-code (Hex) for transmitting programs

is created.

During the program run further files named “temp????.????” and “temp ????.???” are created. These
work files are copied into the project name files after 907 PC 33 is stored/quit.

Following files can exist with modularized projects:
overallprojectname.mod   Structure of modularization (from the modularization editor)
overallprojectname.rep   Report file of an update (ASCII file)
overallprojectname.pas   Password release

For some PLCs further files named projectname.* or overallprojectname.* are created.

2.2 Reserve files
907 PC 33 often works on the harddisk using different files at the same time. To make sure that all
writing operations on the harddisk are finished in
the necessary files properly, even if there is no space left on the harddisk, an empty reserve file of
configurable size named

    _RESERVE.DAT

is created on the current drive (normally the drive containing 907 PC 33) in the 907 PC 33 directory
while starting 907 PC 33.

Should there be no diskspace left on this drive after a writing operation, a warning message is displayed
and the reserve file is reduced in steps of 250 KBytes so that all necessary writing operations are
finished.

After that exit 907 PC 33 and provide some space on the harddisk.

CAUTION:
The advantages of the reserve file can only be used with the current drive. If further drives exist and one
of them, which is not the current drive, has got no space left, it is not possible to release any reserve
space, as no reserve file exists on this drive.

2.3 Auxiliary files
Notice:
To secure a proper program run following minimal values have to be enterd in the file

    CONFIG.SYS
    BUFFERS = 25
    FILES = 25

These parameters are checked by the installation program and set to the correct values if required.

2.4 Subdirectory structure
Paths can also be entered with all file names.
This makes following directory structure possible:
1. Current directory (no path)
The 907 PC 33 user must be in this directory to start the program.
Contents:
    – PLCname.* (All 907 PC 33 files)
– All temporary files.
  (After finishing 907 PC 33 correctly all the temporary files are deleted).

To avoid detrimentally affecting the processing speed, no projects or project files should be created in this directory.

2. Project directory
This path name for the project is entered in the project data editor.
Contents:
– All Data of the project.

3. Library directory
This subdirectory contains the connection element library. The name of the library is entered with the path name in the menu for the program configuration. The default setting is \BIB\<libname>.

4. Further directories
All blocks can be saved into an arbitrary directory and loaded back.
The input of the filenames with paths is always done in the same window.

Example for a path specification: C:\ABB-SPST200\PROJECT\file name
or PROJECT\file name

Notice:
After a project has been modularized it is possible that this structure has changed. See chapter „modularization“ for further information.

2.5 Password protection
907 PC 33 makes it possible to enter passwords with different steps of privileges.

After a project is started the password must be entered by the user.

After that it is checked by all 907 PC 33 functions if the privilege step allows to do the appropriate operation. If not a message appears saying that the entered password doesn't allow to do the appropriate function.

Passwords are entered and deleted in the password editor. The owner of a privilege level can display, delete or re-enter his own password and the passwords of the levels below.

As the encryption algorithm is very complicated, it takes very long to decode a password and therefore this display is only conceived for cases in which users have forgotten their passwords. Passwords are only displayed as from the user’s own privilege level.

There are 4 hierarchical levels:
1. Project manager (G)
   Universal access!
Following installation, the first password is always the name of the PLC:
   Advant Controller 31: AC31 or ac31
   ABB Procontic CS31: CS31 or cs31
   ABB Procontic T200: T200 or t200
The password can then be modified to suit users’ own requirements.
2. Planner (P)
   In the modularization editor, all write operations on the hard disk are inhibited.
   When the editor is called, a message appears to the effect that changes made in the editor will always be rejected. This call must be acknowledged.
3. Commissioning (I)
In all editors, all write operations on the hard disk are inhibited.
The same message as for the 2nd level appears in all editors. (Everything remains permitted in online mode)

4. Final user (E)
All process influencing online functions are locked. These are following functions:
- start PLC
- break
- stop PLC
- continue PLC
- single cycle
- single step
- set breakpoint
- display breakpoint
- delete single breakpoints
- delete all breakpoints
- send simulation list to PLC
- jog

Following functions stay available:
- display status
- status of variables
- trigger variable
- trigger time
- change status of chosen variables

Notice:
The upper functions belong to a major group of all possible ONLINE functions that are not completely implemented in every PLC.
### 2.6 Password editor

Passwords are entered and deleted with the password editor. The owner of a privilege level can display, delete or re-enter passwords of his own and lower levels.

Passwords can contain up to ten arbitrary characters.

<table>
<thead>
<tr>
<th>Name</th>
<th>Pr.</th>
<th>Name</th>
<th>Pr.</th>
<th>Name</th>
<th>Pr.</th>
</tr>
</thead>
<tbody>
<tr>
<td>T200</td>
<td>G</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>t200</td>
<td>G</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Proj</td>
<td>P</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Commi</td>
<td>I</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>End</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Example of the overlay of the password editor, after `<F1> “Display “` had been pressed.

**Display passwords `<F1>`**

Passwords are only displayed as from the user’s own privilege level.

As the encryption algorithm is very complicated, it takes very long to decode a password, when an old computer is used.

**Enter new password `<F2>`**

After a password has been entered one of the following privilege levels must be chosen (see pages 2-6 and 2-7).


**Delete password `<F3>`**

If a password is to be deleted, the privilege level to be entered must be equivalent to the one belonging to the password.
2.7 Starting 907 PC 33

907 PC 33 is started - the logo appears.

There are three alternatives in the logo:

- **<F1>**
  Exit 907 PC 33.

- **<F2>**
  After the password has been entered the menu level of the last handled project is taken up.
  The projectname is read from the file “PLCname.cfg” and displayed behind <F2>.

- **<F3>**
  After the password has been entered the menu level of the last handled module is taken up. This
  can be a program module or a variable module of a modularized project.
  The last handled module is displayed behind “<F3>”.

- **<any key>**
  The project data editor appears. The project name can be confirmed with the <Enter key>, or a new
  project name can be entered and confirmed with the <Enter key> or an existing project can be
  selected via <Ctrl>-U.
  After the password is entered, data for the project, library and controller can be entered.
  After entering the data, quit the project editor with <Esc>.
  The menu level of the project appears.

Note:
If you wish to access a project which is on a diskette, ensure that the diskette is not write-protected,
otherwise access is not possible.

In the menu level all data of the last setting are preset.

907 PC 33 automatically detects modularized or not modularized projects and displays the affiliated menu for function selection.

If there is not enough main memory available the following window is displayed below the logo:

```
Not enough main memory!
162064 Byte are missing.
Program can not be started!

Please change PC configuration and restart!
<Key>
```

In this case more memory space has to be provided.
3 Data backup

3.1 Principle
All data belonging to a project are saved on the harddisk under the file name

```
projectname.*
```

907 PC 33 gives the possibility to discard any modifications when exiting the program. Every editor writes modifications that have been done with it into its file

```
TEMP$$$.???
```

So created TEMP-files are always written into the directory where 907 PC 33 was started.

This makes it possible to continue at the state of when 907 PC 33 had last been started in case of a sudden program interrupt (e.g. electricity breakdown).

If 907 PC 33 is exited with “End” (or before a new project name is entered in “project data” without discarding changes) all project files “projectname.*” with an existing TEMP-file are deleted and the existing TEMP-files are renamed into “projectname.*”.

If 907 PC 33 is exited with “Abort” (or before a new project name is entered in “project data” with discarding changes) all TEMP-files, and so all modifications are deleted.

If 907 PC 33 is interrupted in any other way (illegal program break, reset, program breakdown) all project files keep the state of when 907 PC 33 had last been started. Eventually some TEMP-files with the last modifications will exist.

907 PC 33 detects this illegal condition after it has been restarted and first of all the following notice appears:

```
The project PROJ1 contains errors !!!
Your last changes are stored in:
TEMP$$$.PRO   TEMP$$$.KFG   TEMP$$$.SYM   TEMP$$$.AWL

These files may contain errors.
Observe caution in the further course of processing.

You have 3 possibilities:
1. Continue working before the last change (delete TEMP$$$.*)
2. Accept the last change (RISK! PROJ1.* will be overwritten)
3. Transfer the last change to a new project
<Esc> Abort

Please make a choice
```

Equivalent to the possible alternatives the system must be told how to remove this illegal condition.

The savest is to discard the last modifications.

If only few changes were made the first alternative should be chosen. The second alternative should only be chosen if there is a backup of the whole project.

With the last alternative it is possible to work under a different project name. Make sure that all project files are without errors before copying them into the former project name.
3.2 Saving data on diskette

It is possible to save data onto diskette. A backup like this maintains modularized or not modularized projects and libraries.

After the function is selected from the main menu level the following menu is displayed for saving data onto diskette:

<table>
<thead>
<tr>
<th>Project name:</th>
<th>\ABB-SPS\T200\PROJECTS\TEST</th>
</tr>
</thead>
<tbody>
<tr>
<td>Destination:</td>
<td>A:TEST</td>
</tr>
<tr>
<td>Save project {Y/N} :</td>
<td>Y</td>
</tr>
<tr>
<td>Save library {Y/N} :</td>
<td>Y</td>
</tr>
</tbody>
</table>

The field “Project name” contains the current project, “destination drive” contains disk drive A: with the project as in “project name”. It is also possible to enter paths, although it is recommended not to enter relative paths.

Then it is stated if a project and library is to be saved or only one of the two.

The selected elements are compressed and then saved by pressing <F1>. Before saving it is checked if there is enough space on the medium in the destination drive. If this is not the case an error message is displayed.

The backup files that are created are:
- `<filename>.S_B` for libraries
- `<filename>.S_U` for not modularized projects
- `<filename>.S_M` for modularized projects

The backup is aborted by pressing <ESC>. 

The above described automatical backup does not replace a doubled backup of all the project files (projectname.*) and libraries, which always has to be done by the user.
Restoring

It is possible to restore projects and libraries that have been saved in such a way. After the function is selected from the main menu level the following menu is displayed for restoring data from diskette:

<table>
<thead>
<tr>
<th>Restoring data</th>
<th>TEST</th>
</tr>
</thead>
<tbody>
<tr>
<td>Backup file___</td>
<td>A:\TEST</td>
</tr>
<tr>
<td>Destination___</td>
<td>C:</td>
</tr>
</tbody>
</table>

Please enter the destination drive. <F1> Restore project <Esc> Abort restore

The field “Backup file” contains the backup file of the set drive. Drive C: is preset in the field “destination”. The backup file is loaded and decompressed by pressing <F1>. The project can be edited in the usual way. The destination drive is tested for enough space before loading.

The restore is aborted by pressing <ESC>.
3.3 Copy project

It is possible to copy projects. This covers both modularized and unmodularized projects and libraries. The function is called up via the menu points project man. -> Copy project. The following menu appears:

<table>
<thead>
<tr>
<th>Source project:</th>
<th>SPS_PRG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dest. project:</td>
<td>Y</td>
</tr>
<tr>
<td>Copy project:</td>
<td>Y</td>
</tr>
<tr>
<td>Copy library:</td>
<td>Y</td>
</tr>
<tr>
<td>Copy all files to project:</td>
<td>Y</td>
</tr>
</tbody>
</table>

Enter the name of source project.  
<F1> Start copy    <Esc> Abort

The path and name of the project being copied are entered in the “Source project” field. A Project overview can be called up with <Ctrl>-U.

The name of the new project is entered in the “Target project” field. Any desired directory can be extended here. If the desired directory does not yet exist, it is created during copying. Next, state whether the project and library or only one of the two should be copied.

The desired elements are copied after <F1> is pressed, depending on the setting. In the case of modularized projects, it is necessary to state a different target directory.

Copy all files to the project:

(Y): If the library is also to be copied, it is copied into the same project directory as the target project. In the case of modularized projects, this allows you to determine whether all program and variable modules are to be copied into the same directory as the target project. However, any existing path structure of the source project is lost here.

(N): If the library is to be copied, the directory BIB is created under the directory of the target project and the library is copied to it. All paths of the project being copied are copied relative to the path of the new project.

Example: Copy a modularized project stating ‘Copy all files: N’
Source project is C:\ABB-SPS\AC31\PROJEKT\PROG1. The target project should be: C:\ABB-SPS\AC31\PROJNEW\PROG1. The copied project is then located in: PROG1.MOD in C:\ABB-SPS\AC31\PROJNEW. The individual program modules are then located in C:\ABB-SPS\AC31\PROJNEW\ABB-SPS\AC31\PROJEKT. This means that the PMs are created in a path relative to the new directory PROJNEW.

Copying can be aborted with <ESC>. After copying, you can return to the menu level via <Esc>. 

---

3-4 907 PC 33/General Part/Issued: 11.98
The 907 PC 33 Menu interface serves for a fast and direct selection of any 907 PC 33 function with the keyboard or a connected mouse.

The menu interface consists of two areas:
- The main menu is located in the upper 3 lines.
- Depending on the selection of a main menu, a window appears in the lower area showing all the functions belonging to this menu (submenu).

907 PC 33 notes the most recent selected main menu/submenu combination. This means that the most recently selected menu is also displayed after the 907 PC 33 is restarted.

Selecting a menu via the keyboard

1st possibility:
- Select the appropriate line of the main menu with the <PgUp> or <PgDn> keys.
- Select the main menu with the <Arrow-left> or <Arrow-right> keys.
- Select the submenu with the <Arrow-up> or <Arrow-down> keys.

The appropriate function is started by pressing the <Enter> key.

2nd possibility:
- A specific function group is also selected after the first letter of the function group's description has been entered.
  (Should more function groups begin with the same letter, the next following group beginning with the same letter is selected.)
- After the function's number is entered the equivalent function is started.
Selecting functions with the mouse

– After moving the mouse in the equivalent direction the wanted submenu can be selected in the main menu.
– Then the right mouse button must be pressed. By moving the mouse in the window the wanted function can be selected.
  The right button switches from selection of the main menu in the top field to selection of the submenu in the window.
– The equivalent function is started after the left mouse button is pressed.

DOS-Shell (DOS commands)

After the selection of this submenu point it is possible to leave the menu interface and return to DOS. Then DOS commands or external programs that don’t use more than 200 kB memory can be called.
To keep the data entered in 907 PC 33 return to the menu interface by entering EXIT <Return>.

Quitting the menu

The menu interface is quit

– via Project man. -> Quit system or
– via <Esc>.

If changes were made, you are asked whether the changes should be incorporated in the project or not. Selecting the Abort option returns you to the menu.
5 Project data

5.1 Introduction

The name and path of the project, which is to be handled is entered in the project data editor. Further entries like project description, project name, and library name etc. are possible. With modularized projects the project can only be changed in the overall project. Any other entries are also permitted in subprojects.

The program part is called every time starting 907 PC 33, except if <F2> was selected meaning to go to the main menu. The editor can also be called from the main menu.

After changing the project or starting 907 PC 33 a password for the equivalent project is requested.

The text fields for the entries of name and description of the project or subproject will be printed on the cover sheets of printed lists and can also be put into the head- and footlines of the lists.

<table>
<thead>
<tr>
<th>907 PC 33</th>
<th>Project data</th>
<th>EXAMPLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLC Manufacturer: ABB</td>
<td>PLC name:</td>
<td></td>
</tr>
<tr>
<td>Project: \ABB-SPS\AC31\PROJEKT\EXAMPLE</td>
<td>Library: \ABB-SPS\AC31\BIB\VE_BIB</td>
<td></td>
</tr>
<tr>
<td>PLC type: KR/KT V1</td>
<td>Timeout: T3000</td>
<td></td>
</tr>
<tr>
<td>Responsible: Brown</td>
<td>Last modific.: 30.06.92</td>
<td></td>
</tr>
<tr>
<td>Project designation: comment text 1</td>
<td>comment text 2</td>
<td></td>
</tr>
</tbody>
</table>

The name of the last handled project is stored and appears as default.
The other data are stored and appear (if they already exist) after entering the name of the project and the password.

5.2 Description of the fields

To edit the single fields the cursor is moved to the wanted field with the cursor keys <Cursor Up>, <Cursor Down>, <Tab right>, <Tab left>. The texts are then edited with the field editor (see field editor).

Parameters that can't be modified

PLC manufacturer: ABB
PLC name: Display of the designation of the PLC.

Parameters that can be edited

In the following the parameters that can be edited are explained.

The text fields for the entries of name and description of the project or subproject will be printed on the cover sheets of printed lists and can also be put into the head- and footlines of the lists.
Project
The path and name of the project that is to be handled and under which the entered data are to be saved onto the harddisk with the appropriate extensions are entered in the project field.

After the program is started again the name of the file which was last handled is displayed in this field and the cursor is placed in this field. After calling the project data editor the cursor is placed in the field Library. A message is displayed asking if the last handled project is to be saved or not, if it is moved to the field project file with the key <Cursor Up>.

The field for the project file can max. contain 60 characters, from which 52 characters are for the disk drive and path (for example “C:\ABB-SPS\AC31\PROJEKT\”) and max. 8 characters are used for the name of the project.
An overview of all projects can be called by pressing <Ctrl-U>.

Library name
The path and name for the library are entered in this field. The extensions needed to create the complete name of the three files belonging to the library “.VE1”, “.VE2”, and “.VE3” are appended automatically.

The field for the name of the library can max. contain 60 characters, from which 52 characters are for the disk drive and path (for example “C:\ABB-SPS\AC31\BIB”) and max. 8 characters are used for the name of the library.
An overview of all libraries can be called by pressing <Ctrl-U>.

Type of PLC
If a PLC is delivered in different versions, for example with further instructions and/or with different numbers of IL lines, the type of PLC can be selected in this field. A window with the different types of PLC is displayed by pressing <Space bar> and the wanted PLC is selected by moving the cursor with the cursor keys <Cursor Up> / <Cursor Down>.
Accept with <Enter>, abort with <Esc> or <Space bar>.

Caution:
Modifications at this point affect the management of the main memory of the programming system and the internal table of permitted IL instructions. For this reason check if the project file entered in the project data suits the selected type of PLC.
Responsible

The name of the programmer can be entered in this field.

Last modification

The programmer can enter the date of the last modification in this field. After the project has been modified at a later time the current date is entered in this field automatically. For modularized projects the date is also modified in the module layer.

The date in the project data of the overall project layer is modified if changes were done in the overall project layer.

It is possible to retain the entered date, even after modifications, by entering a space in front of the date.

Timeout

It is possible that the computer is waiting for a character from the serial interface with any online function and that it can’t receive anything because of a fault. In this case the function is aborted after a certain time so that the program can be continued.

There are two possibilities of setting this time:

1. Enter “xxxxx” (e.g.). This makes sure that the loop is run xxxxx times before the function is aborted.
   In this case the actual time depends of the speed of the computer.

2. Enter “Txxxxx” (e.g. “T15000”). This ensures an actual time independant of the computer speed in milliseconds (e.g. 15000 ms).

Should 907 PC 33 display “Timeout” with any online function although the PLC is connected and switched on and the proper cable is used, enlarge the timeout value in steps up to max. 900000.

If the transmission works perfectly and the time until the online functions are aborted is too large, reduce the timeout value in steps down to 10000.

Project designation

Two lines of text can be entered for a further comment of the project in this field.

Project description

Further 8 lines of text can be entered to describe the project. This text like the rest of the project data will appear on the cover sheets of lists.

Further project specific texts can be edited with the text editor (see text editor).

Leaving the project data editor

The project data editor is left by pressing <ESC>. If any modifications were done to the project data, a message is displayed asking if these modifications are to be accepted or cancelled.

Note:

With the project documentation, various details from the project data are received as follows in the footers:
6 Configuration

6.1 Data area editor

The data necessary for the automatical setting of flags and jump labels are entered in the data area editor. Automatic flag issuing for a) Connecting lines between CEs and b) for intermediate flags within CEs.

The following data are entered in the data area editor:

Modul address:

The PLC specific address of the program module is entered here if the project is modularized. 907 PC 33 assigns the address by itself.

The address is needed to transfer data to the PLC and to receive data from the PLC (see also system-specific part of the affiliated PLC system).

Global flags for Extended IL and FBD/LD:

The area (begin/end) is entered for every type of flag. The automatic global flags are then taken from this area during the appropriate compilation. A global assigned flag is only assigned once.

Local flags for Extended IL and FBD/LD:

The area (begin/end) is entered for every type of flag from which the automatic local flags are to be taken from during the appropriate compilation. A locally issued flag is issued only once with a segment plan, and can therefore be issued again in the next segment plan.

Global jump labels for Extended IL and FBD/LD:

The area (begin/end) is entered for the global jump labels, from which the automatic global jump labels are taken from during the appropriate compilation.

Local jump labels for Extended IL and FBD/LD:

The area (begin/end) is entered for the local jump labels, from which the automatic local jump labels are taken from during the appropriate compilation.

Editing keys:

<Enter> : The input is taken over and the cursor moves to the next input field in the edit mode. Otherwise only cursor movement to the next input field.

<Cursor Up> : The cursor moves an input field upwards. The display is scrolled if the cursor is placed at the edge of the screen.

<Cursor Down> : The cursor moves an input field downwards. The display is scrolled if the cursor is at the edge of the screen.

<RTAB> and <LTAB> : The cursor moves to the left or right input field.

<Home> : The cursor moves to the first input field (PLC path).

<Esc> : If the edit mode is switched on it is left and the former field content is kept. Otherwise the editor is left. A message is displayed asking if the modifications are to be accepted or cancelled.
6.2 Terminal operation

All parameters of the serial interface for terminal operation (see Terminal operation) can be set in this configuration menu.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Channel</td>
<td>COM1, COM2</td>
</tr>
<tr>
<td>Baud rate</td>
<td>300, 600, 1200, 2400, 4800, 9600, 19200</td>
</tr>
<tr>
<td>Data bits</td>
<td>7, 8</td>
</tr>
<tr>
<td>Stop bits</td>
<td>1, 2</td>
</tr>
<tr>
<td>Parity</td>
<td>none, odd, even</td>
</tr>
<tr>
<td>Protocol</td>
<td>RTS/CTS, XON/XOFF</td>
</tr>
<tr>
<td>Echo</td>
<td>Yes, No</td>
</tr>
<tr>
<td>Line feed</td>
<td>CR/LF, CR</td>
</tr>
<tr>
<td>Contr. ch.</td>
<td>Yes, No</td>
</tr>
</tbody>
</table>

Caution:
Only the parameters for terminal operation are set. The other parameters of the serial interface for e.g. online data transfer with the PLC or communication with an EPROM programmer are internally set and can’t be modified! The exception to this is the ABB Procontic T200 PLC, for which the interface parameters can be modified.

The parameters of the serial interface are written to a file T200.TER or MPST.TER after leaving the configuration editor.

The interface is initialized according to the contents of this file when working with terminal operation.

The menu Terminal emulation in 907 PC 331/Advant Controller 31 is a terminal operation for the ABB Procontic CS31 or Advant Controller 31 resp. with the interface parameters (cannot be modified) for the controller.

Adjustable parameters
List of parameters that can be modified:

Channel:
The physical interface (a lot of computers are fitted with two serial interfaces) that is to be used with terminal operation is selected.

Baudrate:
The baudrate must be set equivalent to that of the communication device.

Data bits:
It can be chosen between 7 bits (pure ASCII transfer) and 8 data bits.

Stop bits:
Set to 1 or 2 stop bits according to the communication device.
Parity:
Set to odd/even or no parity according to the communication device.

Protocol:
A pure hardware handshake using the RTS/CTS lines follows if “RTS/CTS” is selected.
Software handshake using the XON/XOFF protocol follows if “XON/XOFF” is selected.

Echo:
The characters entered with the keyboard are displayed on the screen before they are sent to the serial interface if “yes” is selected. The characters entered with the keyboard are not displayed on the screen if “no” (no echo) has been set.

Line feed:
To avoid problems with the line feed when displaying the characters on the screen, a line feed is generated after every “CR” when displaying the character on the screen and sending the entered character over the serial interface, if “CR/LF” is set.
Linefeeds read by the serial interface are ignored if “CR” is set.

Contr. ch.:
Some PLCs will interpret control characters if this position is set to “yes”.

Selecting the parameters
The selected position is displayed inverse for each parameter.
The display and modification of a parameter is done for the selected channel (COM1 or COM2).
Modify a parameter for the selected channel by moving the cursor to the appropriate line with the <Cursor Up> or <Cursor Down> keys.
Select one of the alternatives by pressing <Cursor left> or <Cursor right>.

Leaving the configuration menu
The selected parameters are saved to the ASCII-file “PLCname.TER” by pressing <ESC>. The parameters of the serial interface for terminal operation are valid until the configuration menu is selected and the parameters are modified again.

6.3 Color configuration
It is possible for the user to configure the colors in 907 PC 33 with the 907 PC 33 color configuration program.
The setup is written into the file “PLCname.HKF” and read by 907 PC 33 after starting the program.
The color setup is ignored automatically by computers using a monochrom graphics adapter. The attributes for normal, bright and inverse characters are used instead.
The 907 PC 33 color configuration program is started from 907 PC 33 by selecting the appropriate menu point.
Move the cursor to the appropriate menu point by using the cursor keys or mouse. Select with <Enter> or the left mouse button.
After selection in the menu “Color adjustment” either a further menu or the color palette is displayed.
An example of the current setting is displayed on the right half of the screen. By moving the cursor in the color palette the chosen setting is changed.
The colors can be chosen in normal and in flashing display by pressing <space bar>.
The colors of the variables that are not zero can be set for the single editors in the menu “online”.
The colors of the lines can be set for the FBD/LD editor in addition.

– lines zero
– lines not zero
– undefined lines

The color of the ONLINE headline can also be set here for all editors.

Notice that the selected color for set variables differs from the background colors in the single editors so that set variables are displayed well.

It is possible to switch off the colors in the menu “Monitor” for computers that work with color graphic adapters and monochrome monitors or plasma monitors. 907 PC 33 then uses the attributes for normal, bright and inverse characters.
8 Field editor

8.1 Commands

The field editor is always activated if the cursor is placed at a position for text input (in a line) and the user has entered a character which is valid for the field editor.

Edit modes

There are two edit modes when a text input is requested:

– Insert
– Overwrite

When the insert mode is switched on the following valid characters are inserted in front of the last character at the current cursor position.

The following characters are overwritten when overwrite mode is switched on.

The edit modes are switched by pressing <Ins>. The current edit mode stays valid until <Ins> is pressed again.

The edit mode is recognized by the shape of the cursor:

- == No text input possible, because the cursor isn’t placed on an appropriate field (for example on a line in the FBD/LD editor).
- == Overwrite
- == Insert

The insert mode is active when starting 907 PC 33.

Following commands are possible in the field editor:

<table>
<thead>
<tr>
<th>Input</th>
<th>Reaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>displayable</td>
<td>The entered character is displayed and copied into the internal field buffer.</td>
</tr>
<tr>
<td>&lt;Crsr left&gt;</td>
<td>The cursor moves one position to the left.</td>
</tr>
<tr>
<td>&lt;Ctrl&gt;-E</td>
<td>The cursor moves one position to the left.</td>
</tr>
<tr>
<td>&lt;Backspace&gt;</td>
<td>The cursor moves one position to the left.</td>
</tr>
<tr>
<td></td>
<td>The character at the cursor position is deleted.</td>
</tr>
<tr>
<td></td>
<td>All characters right of the cursor position are moved one position to the left.</td>
</tr>
<tr>
<td>&lt;Crsr right&gt;</td>
<td>The cursor moves one position to the right.</td>
</tr>
<tr>
<td>&lt;Ctrl&gt;-D</td>
<td>The insert mode is switched on.</td>
</tr>
<tr>
<td>&lt;Ins&gt;</td>
<td>If the insert mode is switched on all further entered valid characters are inserted in front of the character at the cursor position. The insert mode is switched over by pressing &lt;Ins&gt; again.</td>
</tr>
<tr>
<td>&lt;Ctrl&gt;-V</td>
<td>The character at the cursor position is deleted.</td>
</tr>
<tr>
<td>&lt;Del&gt;</td>
<td>The character at the cursor position is deleted.</td>
</tr>
<tr>
<td>&lt;Ctrl&gt;-G</td>
<td>All characters from the cursor position to the end of the field are deleted.</td>
</tr>
<tr>
<td>&lt;Ctrl&gt;-T</td>
<td>Move the cursor eight characters to the right in the field</td>
</tr>
<tr>
<td>&lt;Ctrl&gt;-Q-Y</td>
<td>Move the cursor eight characters to the left in the field</td>
</tr>
<tr>
<td>&lt;Ctrl&gt;&lt;Crsr right&gt;</td>
<td>Move the cursor to the beginning of the field</td>
</tr>
<tr>
<td>&lt;Ctrl&gt;&lt;Crsr left&gt;</td>
<td>Move the cursor to the end of the field</td>
</tr>
<tr>
<td>&lt;Ctrl&gt;&lt;Home&gt;</td>
<td>Move the cursor eight characters to the right in the field</td>
</tr>
<tr>
<td>&lt;Ctrl&gt;&lt;End&gt;</td>
<td>Move the cursor eight characters to the left in the field</td>
</tr>
<tr>
<td>&lt;Esc&gt;</td>
<td>Abort the field editing.</td>
</tr>
<tr>
<td></td>
<td>The former content stays in the field.</td>
</tr>
<tr>
<td></td>
<td>The field editor is exited.</td>
</tr>
</tbody>
</table>
The entered text or number is taken over.

The field editor is exited.

List files.

Change numbers (see next page).

### 8.2 Number conversion

#### Call the function of changing numbers

There is a program that converts numbers into other numbering systems available in the field editor that is activated by pressing <Ctrl>-W.

After <Ctrl>-W has been pressed the following window is displayed in the upper or lower half of the screen depending on the cursor position:

<table>
<thead>
<tr>
<th>Number conversion</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>8/16/32 D/H/B</td>
<td>u/s</td>
</tr>
<tr>
<td>8/16/32 D/H/B</td>
<td></td>
</tr>
</tbody>
</table>

After the program has been activated the cursor is placed in the upper area of the “Numbering system conversion” and a value can be entered. The field can always be activated during conversion by pressing <home>. The converted value is displayed in the lower area. If an error occurs during conversion an error message is displayed in that field.

#### Selection of numbering system

The “Format” field is activated by pressing <Enter>. The word length and base of the value that is to be converted (top) as well as of the converted value (bottom) can be entered in this field.

The parameters are selected with the cursor keys and <Enter>.

The selected parameters are displayed inverse.

It also can be entered if the value of the number to be converted is signed (s), or unsigned (u). The parameter is automatically set to “s” with negative decimal values.

Following word lengths are possible:

- (8) Bit not used
- (16) Bit Word
- (32) Bit Double word

Following bases are possible:

- (H)exadecimal
- (B)inary
- (D)ecimal

#### Exiting the number conversion

The converted value is taken over with <space bar>. The conversion is aborted without taking over the value with <Esc>.
8.3 File overview

It is possible to display a window containing an overview of all files and directories of the harddisk or of a diskette by pressing <Ctrl>-U, when a filename is requested.

The file list contains the files and subdirectories. The display of the files differs according to how the function was called. If the function was called from the field "Project name", all projects in the current path are listed. If the file list was called from the field “Library”, all libraries in the current path are displayed.

It is possible to take over a filename and its path by moving the bar onto the filename and pressing <Enter>. The cursor is placed on the wanted file by using the cursor keys or entering the first letter of the file. Should further files begin with the same letter the bar skips to the next appropriate filename after the letter has been entered again.

It is also possible to change into a different directory by selecting the appropriate directory. The upper directory is selected by moving the bar onto "..". The name of the current directory is chosen without filename by moving the bar onto ".".

Should there be more entries than displayed in the window, it is possible to scroll the display with help of the cursor keys, <PgUp>, <PgDn>, <Home> and <End>.

Exit the file overview

The file list is exited by pressing <ESC>.

It is also possible to select files with the mouse. The left mouse button is identical to <Enter>, the right one is identical to <space bar>.
Selection of display parameters

A selection of display criterions is shown by pressing <space bar>.

The parameters are selected with the cursor keys.
The selected parameters are displayed inverse.
The new file overview is then displayed after pressing <Enter>.

Files:
- Proj: all projects (name: "*.HMP") in selected directory.
- Block: all program files (name: "*.PG?") in selected directory.
- Bibl: all libraries (name: "*.VE?") in selected directory.
- Files: all files of entered name in selected directory.

Sort order:
- Unsorted: Files/projects as entered in directory.
- Alphabetical: Files/projects sorted in alphabetical order of names.
- Date: Files/projects sorted in order of date of entry.

Directories:
- Display: Directories are displayed and can be changed.
- No display: Directories are not displayed.

Drives:
Existing drives are allocated, displayed and changed to.
9 Modularization

9.1 Introduction

Modularization permits a project to be subdivided into several modules.

The advantages of modularization include:

a) Clear processing of large projects

b) Modularization as early as the planning stage; a set of problems is broken down, right from the very start, into several initialization, main and part-programs for instance

c) Modularization for increasing flexibility; a project can be handled more easily owing to fast modification and adaption

This means that modularization can be used to subdivide projects into logical structures and to support different PLC configurations by a breakdown of the program and data ranges.

It is also possible to process and manage projects which could not be processed without modularization owing to their size which would require too much memory.

A distinction is made between program modules (PMs) and variable modules (VMs) in a modularized project.

Program modules contain program data and variables whilst variable modules contain only variables.

Several program modules can access the variables of the variable module. Connection of program modules and variable modules is defined with the aid of the modularization editor.

A distinction is made in the PMs between local variables and external variables. Local variables are managed only in the program modules, and external variables are also managed in the VMs. This permits a variable which is required in several PMs to be managed centrally in one VM. Changes to this variable should then only be implemented in the corresponding VM.

If a project which has not been modularized to date becomes excessively large, it can be split or segmented into two or more sub-projects. These sub-projects are combined to form one modularized overall project. (The corresponding procedure is described in chapter “Segmenting a program module”).
9.2 Basic elements of modularization

Main project layer

Main project function selection

............. Edit ..........................................

Modularization editor

Program modules | Variable modules

Module layer

Progr. module funct. sel.

......................

PM1

PM2

PMn

Var. module funct. sel.

......................

VM1

VM2

VMn
Structure of a project

The main program consists of several subprograms, the PMs, which themselves consist of several submodules that are known from the normal linear programming.

There are two different working layers:
- main project layer
- module layer

Starting 907 PC 33 (modularized project)

After 907 PC 33 is started the main project layer is displayed.
Main project layer
Functions that are based on the main project are available in the main project layer, for example project management, print (main project). The modularization editor can be called under the menu item “Edit”. The modularization editor is used for entering program modules and setting references to these (see modularization editor).

Module layer
The single PMs or VMs are edited in this layer.
The handling of a PM is similar to handling unmodularized programs.
It is possible to change between handling PMs and VMs e.g.
PM3 —> PM1, PM2 —> VM3, VM2 —> PM2.

Changing to the main project layer
<Esc>
Leaving 907 PC 33
<Esc> in the main project layer.

Program modules (PM) und Variable modules (VM)
A PM is structured like a complete non-modularized project. A VM only consists of a variable management.
Variables used in different PMs can and should be listed in the VM’s variable editors and managed centrally. The modularization editor is used for assigning PMs and VMs (see modularization editor). VMs can be assigned to different PMs and PMs can be assigned to different VMs.
A variable editor belongs to every PM and is used for listing all variables that are used by the PM program. Global defined variables that are listed in a VM are marked with an “E” (external).
Modifications of VMs can be transferred to the assigned PMs with help of the update function. This update function can be called from the modularization editor by pressing <F3> or per menu. See “Variable editor” and “Update function” for further manipulation of variables.

Unmodularized projects
The main project layer and the module layer are one layer with unmodularized projects.

Assignment of program modules and variable modules

Variables used in PM1 and PM2 are listed in VM1. Variables used in PM2 and PM3 are listed in VM2.
Two PMs are assigned to each VM1 and VM2. Two VMs are assigned to PM2, PM1 and PM3 can each have access to one VM.
Which programming styles are supported?

1. TOP-DOWN-programming:
The complete project is split into independent subtasks (PMs). The variables have to be organized using little memory.

The single tasks are now programmed by different programmers.
2. BOTTOM-UP-programming:
The programming had already started and later the project had to be split into modules to keep a clear order in the project or because too many variables had to be managed. Now the project is split into PMs and VMs.
3. Different PMs can be programmed with different editors (FBD/LD, Extended IL). This makes it possible to select the suiting editor for each part.

4. Program blocks that have been programmed for different projects can easily be modified and used again. In the lower example PM3 belonging to the main project 1 is used again as PM 2 in main project 2.
9.3 907 PC 33 menu interfaces

Menu layers

Principally three menu layers are used in the modularized version:

1. Main project function selection
   Functions based on the main project are available in this menu.

2. Program module function selection
   This is the menu layer for handling program modules. It is equivalent to the menu in the version without modularization.

3. Variable module function selection
   Functions that are needed to handle variable modules are available in this layer.

Switching between the single menu layers

After 907 PC 33 is started with a modularized project the overall project function selection is displayed. The modularization editor is called for processing the project first. The file names of the PMs and VMs are edited here and assigned to each other in a matrix. The single PMs and VMs entered in the modularization editor can be called from the overall project layer for editing.

9.4 Subdirectory structure

The structure of subdirectories change from the original structure (see chapter “Installation - structure of subdirectories“) as follows:

Project directory

This directory is the main project directory. The path for the main project is entered in the project data editor.

Contents:
- All data of the main project.

Module directories

These directories exist additionally n times to the non-modularized version for all program and variable modules.

The paths and filenames of the modules are entered in the modularization editor.
Contents:

- modulename.*
  (All files of the appropriate module; they are equivalent to the project files in the version without modularization)

9.5 Procedure for creating a modularized project

9.5.1 Creating a new modularized project

Select the function “Modularization” in the main menu.

Now the filename of the first program module (PM) must be entered. Present PLC programs (FBD/LD, Extended IL) that have been programmed as the name of the main project will be taken over under the entered PM name.

The modularization editor is called.

Program and variable modules can be defined in the modularization editor and can be called for editing.

A complete definition of a module makes it necessary to enter a designation and a filename in an appropriate window. The filenames must differ from each other.

Thereafter a module can be called for editing (press <F1> for program module, <F2> for variable module).

The access rights of each program module to the present variable modules is defined in the matrix of the modularization editor.

A program module can access all variables in the variable modules that have been assigned to that program module. If a variable is read from a variable module into a program module (referencing), that variable is marked as an external variable with an “E” in the variable editor. 907 PC 33 keeps the connection to the variable module. Other variables (local variables) are not specifically marked.

It is not necessary to set up variable modules when creating a modularized project. It is recommended for projects containing several program modules for reasons of a clear overview to create variable modules containing variables used in different program modules, though.

After the modularization editor is exited and after every new start of the project 907 PC 33 will notice that the project is modularized.

Now the function selection menu for the main project is displayed.

To handle a module select the function “Edit PM” or “Edit VM” from the main project layer.

9.5.2 Dividing an excessively large program module

Should a program module have become too complex or if it must be split for other reasons (e.g. for reasons of clear overview or structurization), this can be done with the editor the program module had been created with.

Proceeding:

1. Enter new program module in the modularization editor.
   It is recommended to give the same access rights to variable modules that have been given to the original program module. Otherwise the existing connections of variables in the new program module must be split from the variable modules.

2. Call the program editor in the program module to be split.

3. Call the management of segment plans in the FBD/LD editor or in the Extended IL editor and move the cursor onto the segment plan, that is to be split.
   Press <Shift><F1> and enter the filename of the new program module.
   The current program module’s files that aren’t program specific are copied into the new program module.
   All segment plans from the current cursor position in the management are copied into the new program module.
The copied program parts are deleted from the current program module after finishing the operation successfully.

The compiled IL file is deleted.

4. Recompile the program in the current program module.
5. Recompile the program in the new program module.
6. Delete all unused variables in both program modules.

### 9.5.3 Modularization of a project because of too many variables

If a project has got to be modularized because it isn’t possible to enter new variables in the symbol management select the function “Modularization” in the main menu and accept the present project as program module.

After that the program must be split (see “dividing program modules”). Make sure that few common variables are used in the new program modules.

After the program has been split the variables that aren’t used are deleted from the program modules so that more memory is available for new entries.

It is recommended to assign the variables to different variable modules because of the huge amount of variables in this case. Select logic assignment criterions (e.g. all inputs or all variables for machine A).

For that the variable modules have to be defined and the access rights of them have to be given to the program modules in the modularization editor. Now the variables that are used in the program modules can be inserted block by block in the variable modules or connections to existing variables in program modules can be set with the variable block transfer function (<Ctrl>-K-T).

### 9.5.4 Modularization of a project because of a too-large PLC program

Should a normal PLC program of a not-modularized project have become too complex or should the project have to be split because of structurization select the function “Modularization” in the main menu. Then follow the steps of “Creating a modularized project”. Now the program has become the first subproject in the modularized main project. Finally this subproject must be split. See “Dividing a too complex program module” to carry out this operation as often as the program has to be split.
9.6 Modularization editor

Program and variable modules are defined and variable modules are assigned to program modules with the modularization editor.

Several functions of the modularization editor can also be selected per menu.

Modularization editor with displayed main menu. The cursor is placed on program module 1 and variable module 7 in the matrix.

The filenames of the program and variable modules PMs and VMs on which the cursor is placed on are displayed in the bottom field.

The PM and VM the cursor is placed on is displayed in the top right. The current main project is displayed in the top window (Project _:).

It is possible to enter up to 255 PMs and 40 VMs. The display is scrolled accordingly.

The order of the program modules and the order of transmitting them to the PLC is the same as the above mentioned order in the modularization editor.

Operating the menus

The menu is displayed in the modularization editor by pressing <space bar> or the right mouse button.

The functions can be selected with the <Crsr-Up>, <Crsr-Dwn> keys or by moving the mouse up- or downwards.

The selected functions are called by pressing <Enter> or the left mouse button.

The menu is left without selecting a function by pressing <space> or the right mouse button.

Layout:

The MOD editor has got two layouts.

The layout menu is displayed on the right side of the editor by pressing <Ctrl><F1>.

1. PM - VM The program modules are displayed in the left window. Twelve program modules are listed, but on the right side above the matrix only the variable module of the cursor column is displayed.
2. VM - PM  The variable modules are listed in the left window. Twelve variable modules are listed, but on the right side above the matrix only the program module of the cursor column is displayed.

Layout 1 is chosen (PM - VM) for the following description.

**Inserting a PM designation:**
A line is inserted in the matrix by pressing <Ctrl>-N. The cursor is placed at the appropriate position in the left window so that the module description can be entered (max. 30 characters).

The input is confirmed with <Enter> and aborted with <Esc>. The same module designations can’t be given to other modules.

A window is then displayed requesting the file name of the new entered module. The first 8 characters of the module designation without blanks and special characters are the suggested selection.

If the cursor is placed underneath of the last PM (last available cursor position), the insert modus is switched on by pressing an alphanumeric key. The input is accepted as described above.

**Inserting a VM description:**
A column is inserted in the matrix by pressing <Ins>. The cursor is placed in the input field above the matrix. The module designation can be entered. The input is accepted with <Enter> or aborted with <Esc>.

Notice:
The last two functions work the other way round for layout 2.

**Editing filenames**
The PM’s and VM’s filenames are entered with a specific editor. The editor is called with <F7>. The menu is displayed in the right half of the editor and PM or VM can be selected.

An appropriate filename for every designation is entered with a specific editor. The designation can also be edited.

The same filename can’t be given to further modules.

<table>
<thead>
<tr>
<th></th>
<th>Project: C:\907PC33\MB\EXAMPLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Des.:</td>
<td>Air</td>
</tr>
<tr>
<td>Dat.:</td>
<td>C:\907PC33\SPS\EXE\AIR</td>
</tr>
<tr>
<td>Des.:</td>
<td>Vertical_cylinder</td>
</tr>
<tr>
<td>Dat.:</td>
<td>C:\907PC33\SPS\EXE\VERTCYL</td>
</tr>
<tr>
<td>Des.:</td>
<td>Visual_control</td>
</tr>
<tr>
<td>Dat.:</td>
<td>C:\907PC33\SPS\EXE\VC</td>
</tr>
<tr>
<td>Des.:</td>
<td>Lift</td>
</tr>
<tr>
<td>Dat.:</td>
<td>C:\907PC33\SPS\EXE\LIFT</td>
</tr>
<tr>
<td>Des.:</td>
<td>Power_limitation</td>
</tr>
<tr>
<td>Dat.:</td>
<td>C:\907PC33\SPS\EXE\PWRLIMIT</td>
</tr>
</tbody>
</table>

Modularization editor with displayed window for editing designations and filenames of program modules.
The completeness of the filenames is checked when the MOD editor is exited. The paths are checked when a PM or VM is called. The editor is left with <Esc>.

Searching for and replacing paths:

It is possible to call up search for and replacement of paths in the file name editor with <F1>. Parts of whole paths can be exchanged for different ones here. You can state whether they are to be replaced forwards or backwards from the cursor position.

<table>
<thead>
<tr>
<th>Old path:</th>
<th>C:\907PC33\SPS\EXAMPL1\</th>
</tr>
</thead>
<tbody>
<tr>
<td>New path:</td>
<td>C:\907PC33\SPS\EXAMPL2\EXAMPLE\</td>
</tr>
<tr>
<td>Direct.:</td>
<td>forwards backwards</td>
</tr>
</tbody>
</table>

If a directory does not yet exist, it is automatically set up after an enquiry.

Note:
It is not possible to copy projects with this.

If you change the path names of the file names of PMs, you must, if necessary, also change the path-naming VMs to which reference is made. The path of the main project (i.e. where the modularized file <name>.MOD is found) is not changed.

Creating references:
Asterisks (“*”) mark the access rights of PMs to VMs.
References are set or (if set) deleted after cursor movement by pressing <Enter>.
A reference is set with <F5>, the cursor moves to the next line.
A reference is deleted with <F6>, the cursor moves to the next line.

Deleting a line:
The current line according to the cursor position is deleted with <Ctrl>-<Y>.
The line can’t be deleted if it contains references.

Deleting columns:
The current column according to the cursor position is deleted with <Del>.
The column can’t be deleted if it contains references.

Handling a PM:
For editing a program module the function „Edit PM“ is called from the main project layer.

Handling a VM:
For editing a variable module the function “Edit VM” is called from the main project layer.

Editing a present PM:
The designation of the PM (or VM with layout 2) according to the cursor position can be edited by pressing an alphanumeric key or <Shift>-<F1>.

Editing an existing VM:
The designation of the VM (or PM with layout 1) according to the cursor position can be edited by pressing an alphanumeric key or <Shift>-<F2>.
List of commands:

Function keys:

- `<F3>` update of program modules (see below)
- `<F5>` mark and move cursor one position further down
- `<F6>` delete mark and move cursor one position further down
- `<F7>` call editing menu
- `<Shift><F1>` according to layout edit PM / VM in left window
- `<Shift><F2>`
- `<Ctrl><F1>` call layout menu

Keys for editing:

- `<Enter>` set or delete mark
- `<Crsr left>` move cursor one column to the left
- `<Crsr right>` move cursor one column to the right
- `<Crsr Up>` move cursor one line upwards
- `<Crsr Dn>` move cursor one line downwards
- `<PgUp>` move cursor one page up
- `<PgOn>` move cursor one page down
- `<Home>` move cursor to the first column of the first line
- `<End>` move cursor to the first column of the last line
- `<Ctrl>-N` insert a line in the matrix I
- `<Ctrl>-Y` delete a line in the matrix
- `<Ins>` insert a column in the matrix
- `<Del>` delete a column in the matrix

9.7 Change module

To edit a module it first must be called. A module is called from the main menu of the main project layer. Furthermore it is possible to change to a different PM from a program editor of the current edited PM. The program editor of the called PM is then displayed. This function is executed with `<Ctrl>-<F4>`.

A window containing all available program modules is displayed:

![Program modules window](image-url)

- AIR
- VERTCYL
- VC
- LIFT
- PWRLIMIT
- * TEMPCONT
- FUMES
- COOL
Program editor of a PM with displayed PM selection.

The window contains the filenames of the program modules and the program modules themselves. A “*” in front of the filename of the program module indicates that this program module hasn’t been compiled to IL yet.

In the upper example there is no compiled IL for the program modules “Vertical_cylinder” and “Temperature_control”.

9.8 Variable editor

Entering variables in a program module of a modularized project

A variable is searched for in the main memory (meaning the symbol management of the PM) after it is entered. The variable is accepted if the search was successful (as in unmodularized projects).

If it has not been found it is searched for in all the VMs that the program module can access. If it now has been found (should several variables have been found a variable selection follows, see below) it is taken, otherwise the entered variable is accepted as a local variable.

If VMs aren’t to be searched because of time reasons only the memory can be searched (see below: Configuring the search system of variables).

Example:

Input: “E  em_stop” (Variable is not listed in the symbol management of the program module).

All symbol files of the variable modules which the program module can access are searched for all variables with the operand “E” and the symbol “em_stop”.

If none of the variable modules contain the variable “E em_stop” it is entered in the symbol management of the program module locally.

Several variable modules/variable selection

If one or more variable modules contain the variable a menu window is displayed for selection of the present entries:

<table>
<thead>
<tr>
<th>M</th>
<th>E</th>
<th>File</th>
<th>Variable</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM</td>
<td>1</td>
<td>E 00,01</td>
<td></td>
<td>em_stop</td>
</tr>
<tr>
<td>VM</td>
<td>1</td>
<td>E 00,02</td>
<td></td>
<td>em_stop</td>
</tr>
<tr>
<td>VM</td>
<td>8</td>
<td>E 00,03</td>
<td></td>
<td>em_stop</td>
</tr>
</tbody>
</table>

Switch to a different layout with <Ctrl>-<F1> or <Tab right>:

<table>
<thead>
<tr>
<th>M</th>
<th>E</th>
<th>File</th>
<th>Module designation</th>
<th>Variable</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM</td>
<td>1</td>
<td>Control panel</td>
<td>E 00,01</td>
<td></td>
<td>em_stop</td>
</tr>
<tr>
<td>VM</td>
<td>1</td>
<td>Control panel</td>
<td>E 00,02</td>
<td></td>
<td>em_stop</td>
</tr>
<tr>
<td>VM</td>
<td>8</td>
<td>Conveyor belt</td>
<td>E 00,03</td>
<td></td>
<td>em_stop</td>
</tr>
</tbody>
</table>

In this menu the wanted variable is chosen by cursor movement onto the desired variable and pressing <Enter>.
The variable is marked as external “E” and entered in the symbol management. Internally the connection of the variable to the program module is saved in the variable module.

The variable selection is exited without accepting a variable with <ESC>.

Marked entries are noticed with an asterisk “*” in the “M” column and external ones are marked with “E” in the “E” column.

If no filename was entered the variable is used in the present PM.

The variable selection window makes it possible to swap variables between variable and program modules with modularized projects.

A window for variable selection is always displayed with the functions “Insert” and “Modify” once further variables have been found in the variable modules.

Following functions are available for displaying and modifying variables:

Cursor movement:
   <CurUp>, <CurDn>, <PgUp>, <PgDn>, <Home>, <End>

Switch layout (long text of variables or module designation of VMs):
   <Ctrl>-<F1>, <Tab right>

Mark (single) variables for entry:
   <F5> Mark
   <F6> Delete mark

Mark variables (as block) for entry:
   <Ctrl>-K-B Mark beginning of block
   <Ctrl>-K-K Mark end of block
   <Ctrl>-K-H Delete the block mark

Leave the window without accepting:
   <ESC>

Leave the window accepting the current variable according to the cursor position and marked variables:
   <Enter>

Notice:
The blockmarks have got no effect for functions, with which the selection window is displayed to select “one” variable as alternative variable.

Example:
Modify a variable, display of alternative variables by pressing <F7>.

Functions of the variable editor with modularized projects

**Modularization menu in the program module**

1. Small update:
   Function cannot be carried out here -> see function under Variable module, Page 9-26. Changes made to variables in the variable module are corrected in access-entitled PMs. Call-up only possible in the variable editor of a variable module.

2. Ext. var. -> loc. var. <Shift><F4>:
   This function converts all external variables within a marked block into local variables by deleting the connections (references) to variable modules. If no block marking is present, the connection (reference) to the variable module is deleted in the case of the current external variable at which the cursor is located. The conversion is indicated by deletion of the “E”.

---

9-16 907 PC 33/General Part/Issued: 11.98
3. **Alternative variables <F7>:**

This function allows further variables to be displayed in addition to the current variables at which the cursor is located and enables the current variable to be replaced by one the displayed variables. This involves all variable modules to which the PM has access entitlement being searched through and any found variables are displayed in the display window. A distinction can be made between three cases with regard to the selection:

1. When the current variable is completely defined both absolutely and symbolically (operand identifier, operand number and symbol are given), there is a search for all variables which correspond to the current one in absolute and symbolic designation.

2. When the current variable is given only in absolute terms (operand identifier and group number but no symbol), there is a search for variables with the same operand identifier and the same symbol. For example, “E 00,00 EMERGENCY STOP” and “E 00,02 EMERGENCY STOP” are found in addition to “E EMERGENCY STOP”, but not “M 00,00 EMERGENCY STOP”.

3. When the current variable has been programmed symbolically (operand identifier and symbol but no group number), there is a search for variables with the same operand identifier and the same symbol. For example, “E 00,00 EMERGENCY STOP” and “E 00,02 EMERGENCY STOP” are found in addition to “E EMERGENCY STOP”, but not “M 00,00 EMERGENCY STOP”.

The selection window is displayed for selecting inputs which exist several times when the variable is present several times in the variable modules in the access. In addition, all variables entered in the PM which correspond with the current variables are shown.

4. **Variable selection <F8>:**

   Display of all variables of the selected VM in the variable selection window with the possibility to transfer variables:

   First, a selection list appears with all the variable modules referenced on this PM. If we select from the list of variable modules one to which the current program module has access (position cursor and press <Enter key>), all variables of this variable module are displayed by pressing <Enter>.

   One or more variables can be selected and transferred from the list of displayed variables with the aid of the functions of the selection window.

5. **Variable block transfer <Ctrl>K-T:**

   From program module to variable module:

   When editing in the variable editor of a program module, this transfer function allows all local variables of a marked block to be entered in a variable module. The variable module can be selected from the list of the variable modules accessed.

   All transferred local variables in the program module become external variables as a result of the variable transfer.

   This function also allows a variable to be newly entered in a variable module without the need to exit the program module.

   A variable can be directly transferred into a referenced variable module by <F9> in the program editors with the individual variable editors.

   If a variable is already external, it cannot be transferred to another VM.

6. **Config. variable search <Ctrl><F2>:**

   The search behavior for variables can be adjusted here. If the search is not required in the variable modules (i.e. new variables should always be entered as local), this can be set in the variable editor and in the single-line variable editor of the PM. After <Ctrl><F2>, a window appears in which the current setting (inverse) is displayed:

<table>
<thead>
<tr>
<th>Search for variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>In memory and in VMs</td>
</tr>
<tr>
<td>Only in memory</td>
</tr>
<tr>
<td>&lt;ESC&gt; Abort</td>
</tr>
<tr>
<td>&lt;Enter&gt; Accept</td>
</tr>
</tbody>
</table>
This setting can be changed with the cursor or tab keys or with the space bar. The new setting is accepted in 907 PC 33 with <Enter>. With <ESC>, the window is exited without the change being accepted.

This setting remains valid until it is changed again or until 907 PC 33 is exited. The standard setting during start-up of 907 PC 33 is always “Search in memory and in VMs”.

Functions of the variable editor with modularized projects

Menu Modularization in variable module

1. Small update <F3>:

Modifications done to variables in a variable module are transferred to the according program modules by marking the modified variables as block. A “small update” is performed by pressing <F3>. These changes are done in the appropriate symbol files. This function has got no effect on the modified variables in the program modules if the variable editor is left without accepting the modifications.

Note:
If a variable is deleted in a VM, this change can only be processed by the update in the modularization editor (Section 9.11).

2. Ext. var. -> loc. var.:

Function cannot be executed here -> only from the variable editor in the program module.

3. Alternative variables:

Function cannot be executed here -> only from the variable editor in the program module.

4. Variable selection <F8>:

Display all variables of the selected PM in the variable selection window/no transfer possibility:

First, a selection list appears showing all program modules which can be referenced on this VM. Select a program module from the list (position cursor and press <Enter key>) to which the current variable module has access. This allows all variables of this program module to be displayed by pressing the <Enter key>. No variables can be transferred from the selection window here. All selection functions are therefore ignored for variable transfer in the variable selection window.

5. Block transfer <Ctrl>-K-T:

From the variable module to the program module:
During editing in the variable editor of a variable module, this transfer function allows all variables of a marked block to be inserted into a program module. The program module can be selected from the list of program modules which have access rights to the variable module. With the variable transfer, all variables transferred into the program which were previously local here or not present become external variables.

Functions of the single-line variable editor in FBD/LD and ext. IL

Variable selection for the variable block transfer VM -> PM <F8>

First, a selection list appears with all variable modules which can be referenced on this PM. If you select a VM from the list (position with cursor and press <Enter> to which the current program module has access, all variables of this variable module are displayed by pressing the <Enter key>.

The functions of the variable selection window are available (see Page 9-22f). This means that several variables or only those variables at which the cursor is located can be transferred from the variable module to the variables list of the program module.

Variable selection for variable block transfer PM -> VM <F9>

In the FBD/LD and ext. IL editor, a variable can be transferred with <F9> directly into a referencing variable module with the single-line variable editor.
9.9 Reserve configuration/module configuration

Reserve configuration (907 PC 332 only)

In case of modularized projects it is possible to set the size of a code reserve for the single modules and the size of a segment plan reserve for every segment plan belonging to the current project with the reserve configuration in the main project layer. After calling the function the following window is displayed:

<table>
<thead>
<tr>
<th>Reserve configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default module reserves: 0</td>
</tr>
<tr>
<td>Default segment plan reserves: 0</td>
</tr>
</tbody>
</table>

The values can be entered absolute or in percent.

The editor is exited by pressing <Esc>.

Module configuration

In the module layer the module configuration is available, in which the data range to be used, the flag setting and a module reserve only belonging to this module are entered.

Setting the flag range to be used, one can select whether the data range of the main project layer or the data range of the modules is to be used.

The flag setting states if local or global flags are to be used.

A module reserve only for the current module is set in the module configuration. It is independent of the module reserve set in the reserve configuration of the main project layer.

A segment plan specific reserve is set by using the @RES command. The @RES command is described in the chapter “translator” of the program editors.

After calling the module configuration the following window is displayed:

<table>
<thead>
<tr>
<th>907 PC 331: Module configuration</th>
<th>907 PC 332: Module configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data area: Overall proj. Module</td>
<td>Data area: Overall proj. Module</td>
</tr>
<tr>
<td>Default-FA: Global Local</td>
<td>Default-FA: Global Local</td>
</tr>
</tbody>
</table>

Select the displayed parameters with the cursor keys.

The value for the module code reserve can be entered in percent or absolute.

The editor is exited by pressing <Esc>.

To explain the procedure the following example shows a modularized project containing two program modules.

The reserve configuration of the main project layer is set as follows:

<table>
<thead>
<tr>
<th>Reserve configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default module reserves: 20</td>
</tr>
<tr>
<td>Default segment plan reserves: 5</td>
</tr>
</tbody>
</table>
The field "Module code_res" in the module configuration of PM1 is empty, PM2 has the following module configuration:

```
<table>
<thead>
<tr>
<th>Module configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data area:</td>
</tr>
<tr>
<td>Default-FA:</td>
</tr>
<tr>
<td>Module-Code-Res:</td>
</tr>
<tr>
<td>Word no.:</td>
</tr>
</tbody>
</table>
```

The setup of the program looks as follows:

**PM1**

- Segment pl. 1
  - SP res. for SP1
    - 5 NOP res.
  - Segment pl. 2
  - SP res. for SP2
    - 5 NOP res.
  - Segment pl. 3
  - SP res. for SP3
    - 5 NOP res.
  - Module reserve
    - 20 Module reserve

**PM2**

- Segment pl. 1
  - SP res. for SP1
    - 5 NOP res.
  - GRES 15 C
  - Segment pl. 2
  - SP res. for SP2
    - 15 NOP res.
  - GRES 0 C
  - Segment pl. 3
  - Module reserve
    - from module conf.
  - 40 Module reserve

The reserve for SP3 in PM2 was set to zero with the command "@RES 0 C".

### 9.10 Data ranges

**Flags assignment**

Intermediate flags are used for the direct connections of connection elements in a FBD/LD network. It is possible to set a fixed data range for every module or to use the data range of the main project layer. In the module configuration it is possible to toggle between "main project" and "module" in the field "Data range". Should "Module" be set, the data range of the module is used for setting the intermediate flags.

The data range for every module must then be entered in their data range editors. The data ranges of the single modules must differ from each other.

Should "Main project" be set, the flags are continuously taken from the data range of the main project. It then isn’t necessary to enter data ranges for the single modules.

The following example shows the effect of the data range selection in the module configuration. In both program modules the field "Data range" is set to "Module" in the module configuration.
### Data ranges of program module 1.

<table>
<thead>
<tr>
<th>Mod. addr:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>FBD/LD / ext.IL - GLOBAL FLAG</td>
<td></td>
</tr>
<tr>
<td>Bit flag</td>
<td>M 00,00</td>
</tr>
<tr>
<td>Word flag</td>
<td>MW 000,00</td>
</tr>
<tr>
<td>FBD/LD / ext.IL - LOCAL FLAG</td>
<td></td>
</tr>
<tr>
<td>Bit flag</td>
<td>M 08,00</td>
</tr>
<tr>
<td>Word flag</td>
<td>MW 008,00</td>
</tr>
<tr>
<td>FBD/LD / ext.IL - JUMP LABELS</td>
<td></td>
</tr>
<tr>
<td>Global jump labels</td>
<td>MRK 000</td>
</tr>
<tr>
<td>Local jump labels</td>
<td>MRK 011</td>
</tr>
</tbody>
</table>

### Data ranges of program module 2.

<table>
<thead>
<tr>
<th>Mod. addr:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>FBD/LD / ext.IL - GLOBAL FLAG</td>
<td></td>
</tr>
<tr>
<td>Bit flag</td>
<td>M 03,00</td>
</tr>
<tr>
<td>Word flag</td>
<td>MW 003,00</td>
</tr>
<tr>
<td>FBD/LD / ext.IL - LOCAL FLAG</td>
<td></td>
</tr>
<tr>
<td>Bit flag</td>
<td>M 09,00</td>
</tr>
<tr>
<td>Word flag</td>
<td>MW 009,00</td>
</tr>
<tr>
<td>FBD/LD / ext.IL - JUMP LABELS</td>
<td></td>
</tr>
<tr>
<td>Global jump labels</td>
<td>MRK 006</td>
</tr>
<tr>
<td>Local jump labels</td>
<td>MRK 014</td>
</tr>
</tbody>
</table>
The main data range editor has got the following content:

<table>
<thead>
<tr>
<th>Program module</th>
<th>From</th>
<th>To</th>
</tr>
</thead>
<tbody>
<tr>
<td>PM1</td>
<td>M 00,00</td>
<td></td>
</tr>
<tr>
<td>PM2</td>
<td>M 03,00</td>
<td>M 02,15</td>
</tr>
</tbody>
</table>

Main data range editor with the data ranges of both program modules.

In case of modularized projects, the flag ranges belonging to the main project are displayed, edited and checked in the main project layer with the data range editor.

**Display:**
The program modules and their flag ranges are displayed in the first line.
The flag ranges are switched by using the keys <F1> (page forwards) and <F2> (page backwards).

**Checking the flag ranges:**
It is checked if the entered flag ranges overlap by pressing <F3>. The message line displays which ranges are being checked.

If errors are found they are displayed as followed:
The message line displays the flag range of the program module which overlaps with the inverse displayed program module and flag range.

An appropriate message is displayed if no errors are found.

**Editing keys:**

- `<Enter>` : Accept input and move cursor to the next field (in edit mode), otherwise move cursor to the next field
- `<CurUp>` : Move cursor one field upwards. The display is scrolled if the cursor is placed at the top edge of the screen.
- `<CurDn>` : Move cursor one field downwards. The display is scrolled if the cursor is placed at the bottom edge of the screen.
- `<RTAB>` and `<LTAB>` : Moves cursor to the left or to the right field.
- `<Home>` : Moves cursor to the first field.
- `<End>` : Moves cursor to the last field.
- `<PgUp>` : Page backwards.
The program modules contain the following program examples:

**FBD/LD of PM1:**

![FBD/LD of PM1 diagram]

**FBD/LD of PM2:**

![FBD/LD of PM2 diagram]

After compiling the modules the instruction lists are as follows:

<table>
<thead>
<tr>
<th>PM1</th>
<th>PM2</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000</td>
<td>00000</td>
</tr>
<tr>
<td>00001 &amp;</td>
<td>00001 /</td>
</tr>
<tr>
<td>00002 =</td>
<td>00002 =</td>
</tr>
<tr>
<td>00003</td>
<td>00003</td>
</tr>
<tr>
<td>00004 !</td>
<td>00004 !</td>
</tr>
<tr>
<td>00005 /</td>
<td>00005 &amp;</td>
</tr>
<tr>
<td>00006 =</td>
<td>00006 =</td>
</tr>
</tbody>
</table>

PM1 has used the intermediate flag M 0,0 for the connection of the output of the AND-CE and the input of the OR-CE. The flag was taken from the data range of PM1.

In the instruction list of PM2 the intermediate flag M 3,0 was used for the direct connection of the CEs. It was taken from the data range of PM2.
If the data range in the module configuration of the modules was set to main project, the data range editors of the program modules display the ranges that were entered in the main project layer:

<table>
<thead>
<tr>
<th>PLC</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Mod. addr:</td>
<td></td>
</tr>
<tr>
<td>FBD/LD / ext.IL - GLOBAL FLAG</td>
<td></td>
</tr>
<tr>
<td>Bit flag</td>
<td>M 00,00</td>
</tr>
<tr>
<td>Word flag</td>
<td>MW 000,00</td>
</tr>
<tr>
<td>FBD/LD / ext.IL - LOCAL FLAG</td>
<td></td>
</tr>
<tr>
<td>Bit flag</td>
<td>M 08,00</td>
</tr>
<tr>
<td>Word flag</td>
<td>MW 008,00</td>
</tr>
<tr>
<td>FBD/LD / ext.IL - JUMP LABELS</td>
<td></td>
</tr>
<tr>
<td>Global jump labels</td>
<td>MRK 000</td>
</tr>
<tr>
<td>Local jump labels</td>
<td>MRK 011</td>
</tr>
</tbody>
</table>

Content of the data ranges of the program modules 1 and 2 and main project layer.

The program modules contain the following program examples:

FBD/LD of PM1:

```
E 0,0
E 0,1
&
/
A 1,0
```

FBD/LD of PM2:

```
E 1,0
E 2,1
/
&
A 10,0
```
After the modules have been compiled their instruction lists look as follows:

PM1:
00000 00000 ! E 0,0
00001 & E 0,1
00002 = M 0,0
00003
00004 ! M 0,0
00005 / E 0,2
00006 = A 1,0

PM2:
00000 00000 ! E 1,0
00001 / E 2,1
00002 = M 0,1
00003
00004 ! M 0,1
00005 & E 3,2
00006 = A 10,0

The intermediate flags have now continuously been taken from the data range of the main project layer.
It isn’t necessary to enter ranges for the single modules.

9.11 Updating program modules

In case of modularized projects, variables of a variable module can be used by several program modules at the same time (references).

The advantage of this organization is that so defined variables can be accessed centrally in the variable module by the different program modules. It is also possible to manage variables used by different program modules centrally.

Modifications done to variables in variable modules have to be transferred to the program modules that contain these modified variables.

To receive a flexibility as high as possible when working at program and variable modules this update is done with an update function for all modified variables at any time with an update function, and not for every single variable.

The main feature of the update function is:
a) update by acceptance of modified variable modules into the program module;
b) setting of new references of previously locally used variables in the program module.

The update function is started in the modularization editor by pressing <F3>. A window for the setting of the update parameters is displayed:
1. Variables

If “External” is selected, the references of the external variables in the program modules are checked. If differences are found it is tried to enter the change in the program module. If this isn’t possible the appropriate reference is deleted and the variable becomes local. Changing the access rights of a program module to variable modules or deleting a variable in the variable module can cause an invalid reference for an external variable in the program module. In this case this variable also becomes local in the program module.

If “Local” is selected, all local variables in the program modules are compared with the variables of the according variable modules. If two variables with identical absolute designations and symbols have been found a reference is set between program module and the according variable module. If several variables in the variable modules identical to the one in the program module have been found, a reference is set to the first variable found.

If “All” has been selected an update is done for external variables followed by an update for the local variables. Therefore alternative variables are searched for the variables that have become local in the first run. For this reason the results of the first run for every program module, then a line with “Signs of equality” followed by the results of the second run for every program module are listed in the report file (see below), when this update parameter has been selected.

2. Range

An update can be performed for the current program module, for a list of selected program modules or for all program modules of a modularized project.

In case of the update for a list of selected program modules a selection window is displayed where the required PMs have to be marked with key <F5> before the update is started (a set marking can be cancelled with <F6>). This window is exited (abort update)(384,741),(550,757) with <ESC> or continued (do update) with <ENTER>.

After the parameters have been set the update is performed by pressing <ENTER>. If <ESC> has been pressed the update is aborted and the modularization editor is displayed.

Three phases are run: collect variables, compare variables and handle the program modules. If “All” variables have been selected these phases are run through twice. The bottom line of the update window displays the current phase.

The update is processed in temporary files. The new symbol files of the program modules (including the necessary modifications for the update) have the name “modulename.STP” (symbol, temporary) and the reference file has the name “modulename.RTP”.

These files will be renamed (accept update) or deleted (abort update) at the end of the update.

9.12 Report file

During the run of the update all actions are written into a report file (mainprojectname.REP). After the update is finished 907 PC 33 automatically starts the 907 PC 33 text editor to display the report file.

The lines of the report file containing variables that couldn’t automatically be updated are displayed by pressing <Shift>-<F6>.

Actions are always written into the file “mainprojectname.REP” so that the actions of the last update are overwritten. If this isn’t wanted the former report file must be renamed before an update is performed.

A message in the report file consists of two or three lines. The first line always contains the variable of the program module, the second line contains the action done and eventually the third line contains the new updated variable. Such messages are seperated with empty lines. “Minus characters” are written in a line between two program modules followed by the name of the program module. The messages in the report files have got following meanings:

“Reference to VM: VM NAME deleted”

The reference was deleted. If the “minus” character is displayed instead of the name of the variable
module, the module couldn’t be found. This case is possible if an access right had existed for a variable module before it was deleted in the modularization editor.

“Reference to VM: couldn’t delete VM NAME”

A reference was supposed to have been deleted, but it couldn’t be deleted during the update. It must be deleted manually in the equivalent program module.

“Was updated to (from VM: VM NAME):”

The update was successful. The following line contains the updated variable.

“Could not be updated to (VM: VM NAME):”

The update was not successful. This is the case, if the referenced variable is already listed in the program module, for example. The equivalent reference has been deleted and the variable has become local.

“Reference to VM: VM-NAME has been set:”

The search for an alternative variable was successful. There is a clear reference possibility.

“Reference to VM: VM NAME couldn’t be set”

A reference was supposed to have been set. For some specific reasons it couldn’t be set during the update. An alternative variable must be searched for in the appropriate program module.

Editing the report file doesn’t effect the update function.

After the text editor is exited with <ESC> a message is displayed asking if the performed update is to be accepted or aborted.

Examples:
The variable module “KEYS” contains the variable “E 00,00”. Then it is entered as external in the program module for the first time. Finally the symbol of this variable is entered as “em_stop” in the variable module.

1. During the update for external variables the following message is written into the report file:
   E 00,00
   Was updated to (from VM: KEYS):
   E 00,00   em_stop

2. If the variable “E 00,00   em_stop” had been entered in the program module for the first time before the update, the change can’t be entered with the same parameter setting. (“E 00,00” can’t be changed to “E 00,00   em_stop” because this input already exists in the symbol management of the program module. If the “External”-notice hadn’t been given to “E 00,00   em_stop” a new reference can be set to VM “KEYS” for local variables.) “E 00,00” is listed as a local variable in the program module and the following message is written into the report file:
   E 00,00
   could not be updated to (from VM: KEYS):
   E 00,00   em_stop

3. If an access right to the variable module “KEYS” had been deleted in the modularization editor, the following message is displayed (The name of the variable module couldn’t be located because of the missing access right!):
   E 00,00
   Reference to VM: - has been deleted”

Notice:
An update can take a rather long time for reasons of complexibility and size of the program actions.
9.13 List of commands

Variable editor in the variable module:

- `<Ctrl>-K-T` Transfer block of variables (variables are inserted in the PM)
- `<F3>` “small update”, changes to variables in a VM are transferred to the PMs which can access this VM (Block command)
- `<F8>` List of variables of a PM in the variable selection window (Variables cannot be taken over into the VM)

Variable editor in the program module:

- `<Ctrl>-K-T` Transfer block of variables (former local variables are inserted into a VM and become external in the PM)
- `<F4>` Convert external variable(s) into local variable(s)
- `<F7>` Display and selection of alternative variables
- `<F8>` Display the variables of a VM in the variable selection window (variables can be taken over into the program module)
- `<Ctrl><F2>` Configure the search system of variables (search only in memory / in memory and variable modules)

Single-line variable editor:

- `<Ctrl><F2>` Configure the search system of variables (search only in memory / in memory and variable modules)
- `<F8>` Variable selection for variable block transfer VM -> PM
- `<F9>` Variable selection for variable block transfer PM -> VM

Program editors (FBD/LD, Extended IL):

- `<Shift><F1>` Split program module (segment plan or step chain management or direct in IL)

Modularization editor:

- `<F3>` Update program modules
10 FBD/LD editor

10.1 Explanation of terms

The 907 PC 33 FBD/LD editor makes it possible to program with graphic symbols. It is possible to program with the function block diagram (FBD) or the ladder diagram (LD) or mixed (FBD/LD).

FBD/LD editor display:

```
+---------------------------------+ +---------------------------------+ +---------------------------------+
| PROJECT: EXAMPLE                |  | SP_No: 1 ABS                    |
| +---------------------------------+ +---------------------------------+ +---------------------------------+ |
| TX: 0 TY: 0                      |
```

**Explanation of terms**

Connection element is abbreviated to “CE” in the following description.

“Mark” means: Move the cursor to the appropriate position and mark it by pressing <Enter> or the left mouse button.

**Headline**

The following fields are displayed in the headline (above of the frame):

- **PROJECT**: The currently handled project and segment plan description is displayed
- **SP-NO**: The number of the current segment plan is displayed
- **ABS/SYM**: The current variables display (absolute or symbolically) is displayed
- **TX**: Display of the column number
- **TY**: Display of the line number

10.2 Keys

**Using the mouse buttons for Yes/No requests:**

Questions that have to be answered with yes (Y) or no (N) can be answered with the keyboard and with the mouse. The left mouse button is equivalent to “yes” and the right one is equivalent to “no”.
Deleting marks with the mouse:
Keep the left mouse button pressed and press the right mouse button.

Changing the scroll speed with the mouse:
If the cursor is placed on the outer frame of the FBD/LD editor, the scroll speed can be changed with the mouse.

The scroll speed is reduced a step by pressing the left mouse button and incremented by pressing the right mouse button.

Function keys:

<table>
<thead>
<tr>
<th>Key</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1</td>
<td>delete a column</td>
</tr>
<tr>
<td>F2</td>
<td>insert a column</td>
</tr>
<tr>
<td>F3</td>
<td>compile the FBD/LD</td>
</tr>
<tr>
<td>F4</td>
<td>ONLINE on/off</td>
</tr>
<tr>
<td>F5</td>
<td>insert a comment line</td>
</tr>
<tr>
<td>F6</td>
<td>delete a comment line</td>
</tr>
<tr>
<td>F7</td>
<td>editor for comments</td>
</tr>
<tr>
<td>F8</td>
<td>library</td>
</tr>
<tr>
<td>F9</td>
<td>toggle between SYM/ABS</td>
</tr>
<tr>
<td>F10</td>
<td>help</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Key</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shift F1</td>
<td>split the current segment plan at the cursor position</td>
</tr>
<tr>
<td>Shift F2</td>
<td>link the current segment plan with the following one</td>
</tr>
<tr>
<td>Shift F3</td>
<td>search for and replace CE</td>
</tr>
<tr>
<td>Shift F4</td>
<td>save project</td>
</tr>
<tr>
<td>Shift F5</td>
<td>CE-selection menu</td>
</tr>
<tr>
<td>Shift F6</td>
<td>display compiler errors</td>
</tr>
<tr>
<td>Shift F7</td>
<td>search for undefined variable fields</td>
</tr>
<tr>
<td>Shift F8</td>
<td>search for CE</td>
</tr>
<tr>
<td>Shift F9</td>
<td>search for unoccupied CE connection</td>
</tr>
<tr>
<td>Shift F10</td>
<td>management menu</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Key</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ctrl F1</td>
<td>select size of layout (menu)</td>
</tr>
<tr>
<td>Ctrl F5</td>
<td>set scrollspeed</td>
</tr>
<tr>
<td>Ctrl F6</td>
<td>switch further display of contact variables on/off</td>
</tr>
<tr>
<td>Ctrl F7</td>
<td>display knots</td>
</tr>
<tr>
<td>Ctrl F8</td>
<td>display CE documentation</td>
</tr>
<tr>
<td>Ctrl F9</td>
<td>display orientation column</td>
</tr>
<tr>
<td>Ctrl F10</td>
<td>switch further cursor on/off</td>
</tr>
</tbody>
</table>

Function keys:

<table>
<thead>
<tr>
<th>Key</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alt F1</td>
<td>load CE with connections</td>
</tr>
<tr>
<td>Alt F2</td>
<td></td>
</tr>
<tr>
<td>Alt F3</td>
<td></td>
</tr>
<tr>
<td>Alt F4</td>
<td></td>
</tr>
<tr>
<td>Alt F5</td>
<td></td>
</tr>
<tr>
<td>Alt F6</td>
<td></td>
</tr>
<tr>
<td>Alt F8</td>
<td></td>
</tr>
</tbody>
</table>
Edit commands:

- `<Enter>` : mark or confirm input
- `<Ins>` : insert contact
- `<Del>` : delete contact / CE / connection or CE junction at cursor position
- `<Ctrl>-J` : delete mark
- `<Ctrl>-N` : insert line
- `<Ctrl>-Y` : delete line
- `<Ctrl>-Z` : call single line editor
- `<Ctrl>-Q-W` : create all horizontal connections of the CE at the appropriate side of the CE

Search commands:

- `<Ctrl>-Q-F-A` : search for absolute variables
- `<Ctrl>-Q-A-A` : search and replace absolute variables
- `<Ctrl>-L` : repeat search

Block commands:

- `<Ctrl>-K-B` : mark beginning of block
- `<Ctrl>-K-K` : mark end of block
- `<Ctrl>-K-H` : delete block mark
- `<Ctrl>-K-C` : copy block
- `<Ctrl>-K-W` : write block
- `<Ctrl>-K-R` : read block
- `<Ctrl>-K-Y` : delete block

Display compiled FBD/LD-IL:

- `<Ctrl>` <F3> : display segment plans of compiled IL

Cursor movement:

- `<Cursor left>` : Moves cursor one position to the left. The display is scrolled to the right if the cursor is placed at the left edge of the screen.
- `<Cursor right>` : Moves cursor one position to the right. The display is scrolled to the left if the cursor is placed at the right edge of the screen.
- `<Cursor Up>` : Moves cursor one position upwards. The display is scrolled downwards if the cursor is placed at the top edge of the screen.
- `<Cursor Down>` : Moves cursor one position downwards. The display is scrolled upwards if the cursor is placed at the bottom edge of the screen.
- `<Tab left>` : Moves cursor one tab position to the left.
- `<Tab right>` : Moves cursor one tab position to the right.
- `<PgUp>` : Moves cursor one page backwards.
- `<PgDn>` : Moves cursor one page forwards.
- `<Home>` : **Jump to the beginning**
  1. The cursor isn’t placed in the 1st line of the segment plan: The cursor jumps to the first line of the segment plan.
  2. The cursor is already placed in the 1st line of the segment plan: The cursor jumps to the 1st line of the former segment plan.
- `<End>` : **Jump to the end**
  1. The cursor isn’t placed in the last line of the segment plan: The cursor moves to the last line of the segment plan.
  2. The cursor is already placed in the last line of the segment plan: The cursor jumps to the last line of the following segment plan.
10.3 Menu control

Several functions of the FBD/LD editor can be called per menu.

FBD/LD editor with displayed main menu.

Using the menus

The menu is displayed in the FBD/LD editor by pressing the spacebar or right mouse button.
The single functions are selected by moving the cursor with the <CrsrUp> and <CurDn> keys or equivalent movement with the mouse.
The selected function is actually called by pressing <Enter> or the left mouse button.
The functions are also selected by striking the highlighted letter of the function (hotkey).
The menu is left without calling the selected function by pressing the right mouse button or the spacebar.
Notice:
The menu is always used the same way and the menu functions work the same way when editing.
Therefore the menu functions aren’t deliberately described.

10.4 Management

A program is put together from several segment plans with the FBD/LD editor.
The management of the single segment plans is done with the segment plan-management.
The management is displayed in the right side of the FBD/LD editor by pressing <Shift><F10>.

Entering a new segment plan:

1. Move the inverse bar where the new segment plan is to be inserted.
   To append a new segment plan move the inverse bar underneath of the last entry of the list.
2. Insert the new segment plan by pressing <Ctrl>-N.
3. Edit the new segment plan by pressing <Enter>. 
Example:

Segment plan with displayed management. Segment plan 6 (still without designation) has just been inserted.

A '*' at the variable means, that the variable cannot be displayed completely (see chapter 10.6, Layouts).

**Editing the designation of segment plans:**
1. Move the inverse bar to the wanted segment plan.
2. The designation is edited by entering an alphanumeric character or pressing <cursor right>.
3. Finish editing by pressing <Enter>.

**Changing the segment plan:**
1. Move inverse bar to the appropriate segment plan.
2. Call the segment plan by pressing <Enter>.

**Deleting a segment plan:**
1. Move inverse bar to the appropriate segment plan.
2. Delete the segment plan by pressing <Ctrl>-Y.

Notice:
As it isn’t possible to delete the current segment plan a different one has to be called before deletion.

**Block commands:**
Single or several segment plans can be marked as a block. These marked blocks can be moved, copied, deleted, written onto the harddisk and read from the harddisk.
Segment plans belonging to a marked block are marked with a "*" in the first column.
Affiliated commands are described in chapter “Command overview”.

**Mark single segment plans:**
1. Move inverse bar to the appropriate segment plan.
2. Mark the segment plan by pressing <F5>.
Delete marks of single segment plans:
1. Move inverse bar to the appropriate segment plan.
2. Delete the mark by pressing <F6>.

Mark beginning of a block:
1. Move inverse bar to the first segment plan of the block.
2. Mark the segment plan as beginning of a block by pressing <Ctrl>-K-B. A “*” is displayed in front of the segment plan stating a block mark.

Mark end of a block:
1. Move inverse bar to the last segment plan of the block.
2. Mark the segment plan as the end of the block by pressing <Ctrl>-K-K. All segment plans belonging to the block are marked with “*”.

Delete block mark:
1. Block marks are deleted by pressing <Ctrl>-K-H.

Delete block:
1. Mark the block as described above.
2. Delete the block by pressing <Ctrl>-K-Y.

Move block:
1. Mark the block as described above.
2. Move the inverse bar to the new initial position the block is to be moved to.
3. Move the block by pressing <Ctrl>-K-V.

Copy block:
1. Mark the block as described above.
2. Move the inverse bar to the new initial position the block is to be copied to.
3. Copy the block by pressing <Ctrl>-K-C.

Write block onto harddisk:
1. Mark the block as described above.
2. Call the block write function by pressing <Ctrl>-K-W.
3. Enter the filename of the block in the displayed window.
   The block is saved by confirming the filename with <Enter>.

Notices:
If during saving a file it is maked out, that on the hard disk in the same directory a block or project with the same filename already exists, a message is displayed asking if the file is to be overwritten.
Segment plans that have been saved as block are handled as single projects.

Reading a block from the harddisk:
1. Move the inverse bar to the position the new block is to be read to.
2. Call the block read function by pressing <Ctrl>-K-R.
3. Enter the filename of the block in the lower status line. After input the block is read by pressing <Enter>.
10.5 Management (Searching)

Searching for a segment plan:
It is possible to search for a number or name of a segment plan in the management. The search menu is displayed in the management by pressing <Shift><F3>. The following figure shows the search menu:

<table>
<thead>
<tr>
<th>Search for segment plan/step c</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name/No.:</td>
</tr>
<tr>
<td>Type : Name</td>
</tr>
<tr>
<td>Direct.: down</td>
</tr>
</tbody>
</table>

Explanation of the parameters:
- Name/No.: Input of the name or number that is to be searched for. If the segment plan's description isn't known completely, entering a part of the name is sufficient. A segment plan called "Turn on motor" is also found if "on" has been entered as name.
- Type: This field states if a number or name is searched for.
- Direct.: Selects the direction of the search (inverse bar).

After the search has been started by pressing <Enter> the inverse bar moves to the first found input. The search is continued by pressing <Ctrl>-L.

10.6 Layouts

It is possible to display the FBD/LD editor with three different layouts. After pressing <Ctrl><F1> an appropriate menu is displayed. A layout is selected by moving the inverse bar onto the wanted layout. The default is "5 characters/tab" when starting the FBD/LD editor. The following examples show the three layouts:

Layout "3 characters/tab" with displayed layout menu. A '*' at the variable means, that the variable cannot be displayed completely.
Orientation column:
An orientation column is displayed in all three layouts in the FBD/LD editor by pressing <Ctrl><F9>. Networks are displayed as a line in the orientation column. Comments are marked as an inverse double line.

Additional cursor display:
An additional cursor is displayed on the top and bottom frame by pressing <Ctrl><F10>. If the orientation column is displayed an additional cursor is also displayed there.

The additional cursor is toggled by pressing <Ctrl><F10>.

The following example shows a display with orientation column and additional cursor.

10.7 Connection elements (CEs)

Displaying an CE:
1. Move the cursor to the wanted position the CE is to be placed at (from tabulator position 3 on) with the cursor keys or mouse.
2. Enter the name of the CE via keyboard
3. Confirm with <Enter>.

Selecting a CE from the CE menu:
1. Move the cursor to the wanted position the CE is to be placed at with the cursor keys or mouse.
2. Call the CE menu by pressing <Shift><F5>.
3. Move the inverse bar onto the wanted CE.
4. Accept the CE by pressing <Enter> or the left mouse button or leave the CE-menu without accepting the CE with <ESC> or the right mouse button.
Displaying the CE documentation:
1. Move the cursor inside of the CE of which the documentation is to be displayed with the cursor keys or mouse.
2. The text editor containing the CE documentation of the current CE is displayed by pressing <Ctrl><F8>. or
1. Call the CE menu by pressing <Shift><F5>.
2. Move the inverse bar onto the wanted CE.
3. The text editor containing the CE documentation of the current CE is called by pressing <F4>.

Displaying CEs with connections:
1. Move the cursor to the wanted position the CE is to be placed at with the cursor keys or mouse.
2. Call the CE menu by pressing <Alt><F8>.
3. The wanted CE is placed in the area of the cursor and is displayed with connections by moving the inverse bar onto the wanted CE and pressing <Enter>. The connections on the input side of the CE are pulled to the left side of the screen. If this isn’t possible because of the position of the CE no connections on the input side of the CE are displayed.

Notice:
If a collision with different elements or mark would occur the CE is placed in the near area of the cursor with enough space, so that connections can be entered without having to move the CE.

FBD/LD editor with CEs and displayed CE menu.
There are some of the library functions available with displayed CE menu:

- <Shift><F1> : display all CEs
- <Shift><F2> : CE group selection
- <Shift><F3> : CE group assignment
- <F8> : Plausibility group editor
- <F9> : Toggle order criterion (order in order of input or alphabetical).

The library is called by pressing <F8>. The library functions according to the chapter “CE library” can be executed.
10.8 Moving CEs

Moving single CEs (horizontal and vertical)

1. Move the cursor inside of the frame of the CE with the cursor keys or mouse.
2. Mark the CE by pressing <Enter> or the left mouse button.
3. Move the cursor to the place that the CE’s marked position is to be moved to.
4. Confirm with <Enter> or the left mouse key.

The following example shows the elements of the last example after movement.

Notice:

Movement is only possible if none of the inputs or outputs are connected to the single CEs.
An error message is displayed if a collision with other elements or marks would occur.

10.9 Moving CE connections

Moving the connections of a CE is only possible in vertical direction.

1. Move the cursor to where the CE connection touches the frame of the CE with the cursor keys or mouse.
2. Mark position by pressing <Enter> or the left mouse button.
3. Move the cursor to the place within the column the marked connection is to be moved to.
4. Confirm with <Enter> or the left mouse button.

The following example shows the changes done after connections have been moved (always relating to the initial CE):
10.10 Inverting CE connections

1. Move the cursor next to the connection which is to be inverted with the cursor keys or mouse.
2. Mark the position by pressing <Enter> or the left mouse button.
3. Invert the connection by pressing <Enter> or the left mouse button.

Notice:
The CE library contains the definitions of which inputs and outputs can be inverted.
The following example shows how to invert the upper left connection of the "/" CE:

The cursor shows the position for inverting a connection.

**10.11 Duplicating CE connections**

An input or output can be duplicated if it is needed more than once.

The CE library contains the definitions of which connections can be duplicated.

1. Move the cursor to where the CE connection touches the frame of the CE with the cursor keys or mouse.
2. Mark the position by pressing <Enter> or the left mouse key.
3. Move the cursor one position to the left or to the right.
4. Confirm with <Enter> or the left mouse button.

In the following example the bottom inputs of the two upper CEs are duplicated:
Notices:
The duplication number is automatically appended to the max. two digit designation of the connection, which can be duplicated.

It is possible to connect up to 241 inputs and 241 outputs to a CE.

The duplication type when duplicating several connections at once is defined in the CE editor (see library CE editor) under “Duplication type”.

An error message is displayed if it is tried to duplicate a connection which can’t be duplicated.

10.12 Deleting CE connections

There is a possibility of deleting duplicated CE connections.

1. Move the cursor to the connection which is to be deleted with the cursor keys or mouse.
2. Delete the connection by pressing <Enter> or the left mouse button twice or delete connection by pressing <Del>.

Notice:

Only CE connections that aren’t connected to other lines can be deleted. An error message is displayed if it is tried to delete occupied CE connections.

The following example shows deleted duplicated connections:

![Diagram showing deleted connections](image)

The cursor shows the position for deleting a connection.

10.13 Deleting CEs

There is a possibility of deleting CEs in the segment plan if necessary.

1. Move the cursor inside of the CE which is to be deleted with the cursor keys or mouse.
2. Mark the CE by pressing <Enter> or the left mouse button.
3. Delete the CE by pressing <Enter> or the left mouse button.
Notice:
A safety request is displayed in the status line as all CE-connections and connections to other lines are also deleted.

The \textquotedbl/\textquotedbl CE is to be deleted in the following example:

Delete CE \textquotesingle?\textquotesingle —> Confirm with \textless Enter\textgreater, else other key
Segment plan before deleting the \textquotesingle/\textquotesingle CE. The cursor is placed inside of the \textquotesingle/\textquotesingle CE.

10.14 Deleting/swapping CEs

Segment plan after the \textquotesingle/\textquotesingle-CE has been deleted.
Swapping CEs

The CE of an FBD/LD network can be swapped with a CE from the CE library, if their connections are compatible.

When swapping CEs the features of the old CE are transferred to the new one. This goes for the following features:

1. Inversion permitted (Yes/No)
2. Display the formal-parameter-description (Yes/No)
3. Duplication permitted (Yes/No)

Procedure:
1. Move the cursor to the name of the CE which is to be swapped.
2. Enter the name of the new CE or select the new CE with the CE selection menu.
3. Swap the CEs by pressing <Enter>.

10.15 Setting initial contact

Set an initial contact network
1. Move the cursor to the line the initial contact network is to be inserted with the cursor keys or mouse.
2. Set initial contact network by pressing <Alt><F1>.

Set an initial contact:
1. Move the cursor to the position the initial contact is to be inserted with the cursor keys or mouse.
2. Set initial contact by pressing <Alt><F2>.

The following example shows an initial contact network and a simple initial contact underneath:

Display with the layout „3 characters/tab“.
**10.16 Inserting serial contact**

**Insert a serial contact:**

1. Move the cursor to the position the contact is to be inserted in the contact network with the cursor keys or mouse.

   then:

2. Mark the position by pressing <Enter> or the left mouse button.

3. Move the cursor one position to the left or to the right with the cursor keys or mouse.

4. Confirm by pressing <Enter> or the left mouse button.

   or:

2. Insert contact by pressing <Alt><F3>.

   or:

2. Press <Ins>.

A contact is to be inserted behind the present contact in the following example:

![Diagram of contact network with insertions](image)

Segment plan before inserting: mark and cursor are on the right side next to the contact.

**Notice:**

If there isn’t enough space it is provided automatically. If space can’t be provided an error message is displayed.
Segment plan after insertion.

10.17 Inserting parallel contact

1. Move the cursor to the left side of the contact to which the contact is to be placed parallel to with the cursor keys or mouse.

then:

2. Mark the position by pressing <Enter> or the left mouse button.
3. Move one position downwards with the cursor keys or mouse.
4. Confirm with <Enter> or with the left mouse button.

or:

2. Insert contact by pressing <Alt><F4>.

A contact is to be inserted parallel to the left one in the following example:

Segment plan before inserting; the mark is on the left side of the left contact, the cursor is placed underneath of the mark.
Notice:
If it isn’t possible to insert a contact underneath of a present contact it is eventually possible to insert it above of the present one.

Segment plan after insertion.

**Insert a contact parallel to several contacts:**

1. Move the cursor to the left side of the contact, to which the parallel path is to be opened, with the cursor keys or mouse.
2. Mark the position with <Enter> or the left mouse button.
3. Move the cursor to the right side of the contact, where the parallel path is to be closed, with the cursor keys or mouse. The cursor and mark must be in the same line.
4. Confirm with <Enter> or <Alt><F4> or with the left mouse button.

A contact is to be connected parallel to the path containing the contacts “E 00,00” and “E 00,02” in the following example:

Notice:
The variable descriptions (e.g “E 00.00”) depends from the used PLCs and can differ from this description. How to enter variables is described later (see display of variables).
Segment plan before inserting; the mark is placed left of the contact “E 00,00”, and the cursor is placed right of the contact “E 00,02”.

Segment plan after insertion.
10.18 Deleting contacts

1. Move the cursor on to the contact which is to be deleted with the cursor keys or mouse (not to the middle of contact).
2. Mark the position with <Enter> or the left mouse button.
3. Delete the contact with <Enter> or the left mouse button.

Notice:

The parallel connections are also deleted if the last contact of a parallel path is deleted. If necessary the deleted path is filled with the contacts of parallel paths underneath automatically.

The contact “E 02,00” is to be deleted in the following example:

Segment plan before deleting; the mark and cursor are placed at the same position on the right side of the contact.

Segment plan after the contact has been deleted.
10.19 Setting output contact

1. Move the cursor to the last tab position of the contact network with the cursor keys or mouse.
2. Insert the output contact by pressing \(<\text{Alt}><\text{F5}>\) or \(<\text{Alt}><\text{F6}>\).

Notices:

If the output contact has been inserted with \(<\text{Alt}><\text{F6}>\), it is automatically set to “S” (Set), with \(<\text{Alt}><\text{F5}>\) to “ ” (assignment).

The negation “/” and reset “R” cannot be set with the keyboard. They are inverted the same way CE connections are inverted.

Further output contacts can be set parallel to an output contact. (See “Inserting parallel contacts”).

The following example shows how to insert output contacts:

Segment plan before setting an output contact; the cursor is placed in the last tab position of the contact network.
Segment plan after the output contact has been set with <Alt><F5>.

Segment plan after the output contact has been set with <Alt><F6>.
10.20 Vertical contact junctions
There is a possibility of inserting or deleting vertical parallel connections into/from contact networks.

Inserting a vertical parallel contact connection:
1. Move the cursor to the wanted column (contacts must be left and right of this position) of the top connection line with the cursor keys or mouse.
2. Mark the position with <Enter> or the left mouse button.
3. Move the cursor to the column of the bottom connection line (contacts must be left and right of this position) with the cursor keys or mouse.
4. Confirm with <Enter> or the left mouse button.

Deleting a vertical parallel contact connection:
1. Move the cursor to the connection which is to be deleted.
2. Mark the position with <Enter> or the left mouse button.
3. Confirm with <Enter> or the left mouse button.

Notice:
Vertical contact connections can only be deleted if the deletion doesn’t interfere with the function of the contact network.

The following example shows how to insert a vertical contact connection:

Segment plan before inserting; the mark is placed on the top connection line, the cursor is placed on the bottom connection line.
Segment plan after insertion.

10.21 Inverting contacts

1. Move the cursor to the middle of the contact which is to be inverted with the cursor keys or mouse.
2. Invert the contact by pressing <Enter> or the left mouse button or (if already inverted) cancel inversion.

Notice:
Output contacts are toggled in the following order to ““ (assignment), “/” (negation), “S” (set), “R” (reset).
The contact “E 00,00” is inverted and the output contact “M 00,00” is changed to a reset contact in the following example.
10.22 Display of variables

There are different possibilities of displaying variables in the FBD/LD-editor:

1. Display contact variables and CE variables absolute.
2. Display contact variables and CE variables symbolical.
3. Display contact variables absolute and symbolical.

The display is toggled between absolute and symbolic display by pressing <F9>.
The additional display for contact variables is toggled on or off with <Ctrl><F6>.

The following examples show the different possible displays of variables:

1. **Absolute display of variables**:

   ![Example Diagram]

   Absolute display of contact variables.
Absolute display of CE variables.

2. Symbolical display of variables:

Symbolical display of contact variables.
3. Absolute and symbolical display of variables:

Symbolical display of contact variables.
10.23 Entering variables

Variables and symbols can be entered in the appropriate variable fields in the FBD/LD editor.

1. Move the cursor onto a variable. The variable or symbol input mode is switched on by pressing an alphanumerical key. Now the variable or symbol is entered or edited, if already available.

2. The variable is displayed formatted after pressing <Enter>. If the input was finished by pressing <TAB> the cursor jumps to the next empty variable field inside of a line from left to right and then from top to bottom inside of a segment plan.

Notices:

- Variable fields are displayed as "........" in CE networks. They are displayed above of the appropriate contact in contact networks and are not specifically marked.
- If the additional display is switched on in a contact network the top variable field is not an input field.
- Should the variable or symbol not be entered in the variable list or symbol list, the single line editor is activated (see single line editor).
- If changes are also done to variables already entered apart from entering the new one, the segment plan is displayed according to these changes after the single line editor is left.
- An error message is displayed if an invalid input was entered.

Activating the single line variable editor

If the cursor is placed on a variable field the single line editor is activated by pressing <Ctrl>-Z.

The following example shows absolute variables in a formatted display:

![Absolute variables in a formatted display.](image)

Absolute variables in a formatted display.
10.24 Direct CE output

In the CE library (see CE library) can be defined for which CEs outputs can be set directly.

1. Call the CE.
2. Move the cursor somewhere over the frame of the CE.
3. Input of a valid character via keyboard activates the variable or symbol input mode and then the
   variable or the symbol can be input or edited, if already available.
4. The variable is displayed formatted by pressing <Enter>.

Notice:
If the variable or the symbol isn’t listed in the variable or symbol list yet it is automatically entered into
this list.

10.25 Creating connections

There are different possibilities of connecting two elements in the FBD/LD editor:

1. Connection variable - variable
2. Connection variable - CE
3. Connection CE - variable
4. Connection CE - CE
5. Connection Connection line - CE
6. Connection Connection line - variable
7. Connection output contact network - input CE
8. Connection output CE - input kontakt network

Notice:
Vertical connections are straight from left to right and from top to bottom.
To connect two elements it is necessary to mark one position and place the cursor to the next position.
It doesn’t matter which of the elements have been marked and which one the cursor has been moved
to. The following examples only show one possibility.
The following examples show how to connect elements.
The CE "/" only serves as example. The CE "/" in the ABB Procontic manufacturer library has only one output. This output cannot be duplicated.

Segment plan with two CEs and two contact-networks without connections.

**Connection variable - variable (direct assignment):**

1. Move the cursor to the position of the first variable with the cursor keys or mouse.
2. Mark the position with <Enter> or the left mouse button.
3. Move the cursor to the position of the second variable with the cursor keys or mouse.
4. Connect the two positions by pressing <Enter> or the left mouse button. The following CEs must be available in the library for the FBD/LD compiler: ::, :=, :=W

The following example shows the connection of two variables.
The first variable is placed left in the first line, the second one is placed right in the second line:

Segment plan before direct assignment of variables. The mark is placed in the top left corner and the cursor is placed at the right in the second line.
Segment plan after connection was performed.

**Connection variable - CE:**

1. Move the cursor to the position of the variable with the cursor keys or mouse.
2. Mark the position with <Enter> or the left mouse button.
3. Move the cursor to the appropriate CE with the cursor keys or mouse.
4. Connect the two positions by pressing <Enter> or the left mouse button.

Notice:
If the mark is placed in the same line or underneath of the CE input, the elements are connected with a straight line.

The following examples show the connection of variables with the “/” CE:

Segment plan before connection.
Segment plan after connection was performed.

**Connection of CE - variable:**

1. Move the cursor to the appropriate output of the CE with the cursor keys or mouse.
2. Mark the position with <Enter> or the left mouse button.
3. Move the cursor to the position of the variable with the cursor keys or mouse.
4. Connect the two positions by pressing <Enter> or the left mouse button.

Notice:

If the cursor is in the same line of the CE output or underneath of it, the two elements are connected with a straight line.

The following example shows the connection of the "&" CE and a variable:

Segment plan before connection, the output of the CE "&" is marked and the cursor is placed at the bottom right.
Segment plan after the connection was performed.

**Connection of CE - CE:**

1. Move the cursor to the wanted output of the CE with the cursor keys or mouse.
2. Mark the position with <Enter> or the left mouse button.
3. Move the cursor to the wanted input of the CE with the cursor keys or mouse.
4. Perform connection by pressing <Enter> or the left mouse button.

The following example shows how to connect the output of the “/” CE with the top input of the “&” CE:

Segment plan before connection. The top output of the “/” CE is marked and the cursor is placed on the top input of the “&” CE.
Segment plan after the connection was performed.

**Connection of connection line - CE:**

1. Move the cursor onto the connection line with the cursor keys or mouse.
2. Mark the position with <Enter> or the left mouse button.
3. Move the cursor to the appropriate input of the CE with the cursor keys or mouse.
4. Connect the two positions by pressing <Enter> or the left mouse button.

The following example shows how to connect the top connection-line with the top input of the “/” CE:

Segment plan before connection. The top connection line is marked and the cursor is placed on the top input of the “/” CE.
Segment plan after the connection was performed.

**Connection of connection line - variable:**

1. Move the cursor onto the connection line with the cursor keys or mouse.
2. Mark the position with <Enter> or the left mouse button.
3. Move the cursor to the appropriate variable with the cursor keys or mouse.
4. Perform the connection by pressing <Enter> or the left mouse button.

The following example shows how to connect the connection line of the “/” and “&” CEs with a variable:

Segment plan before connection. The connection line of the “/” CE and “&” CEs is marked and the cursor is placed on the right above of the “&” CE.
Segment plan after the connection was performed.

**Connection of contact network - CE:**

1. Move the cursor to the last tab position of the contact network with the cursor keys or mouse.
2. Mark the position with <Enter> or the left mouse button.
3. Move the cursor to the input of the appropriate CE with the cursor keys or mouse.
4. Perform the connection by pressing <Enter> or the left mouse button.

The following example shows how to connect the last position of the left contact network with the bottom input of the "&" CE:

Segment plan before connection, the last tab position of the contact network is marked and the cursor is placed on the bottom input of the "&" CE.
Segment plan after the connection was performed.

**Connection of CE - contact network:**

1. Move the cursor to the output of the appropriate CE with the cursor keys or mouse.
2. Mark the position with <Enter> or the left mouse button.
3. Move the cursor to the beginning of the contact network with the cursor keys or mouse.
4. Connect the two positions by pressing <Enter> or the left mouse button.

The following example shows how to connect the bottom input of the "/" CE with the right contact network:

Segment plan before connection, the bottom output of the "/" CE is marked and the cursor is placed on the initial part of the right contact network.
Segment plan after the connection was performed.

There is a possibility of connecting all inputs or outputs of a CE with variables at once for CE - variable or variable - CE connections.

1. Move the cursor to one output or input of the appropriate CE with the cursor keys or mouse. Mark the position with <Enter> or the left mouse button.
2. Move the cursor to the wanted column with the cursor keys or mouse.
3. Connect all CE-connections to variables by pressing <Ctrl>-Q-W.

Notice:
If a position hasn’t been marked and at an input or output <Ctrl>-Q-W has been pressed the connections are written to the outer right or left variable fields.

The following example shows a CE with all inputs connected to variable fields:

CE after all inputs have been connected; the cursor is placed on the input from where the keys <Ctrl>-Q-W had been pressed.
10.26 Deleting connections

1. Move the cursor to the connection which is to be deleted with the cursor keys or mouse.
2. Delete the connection by pressing <Enter> or the left mouse button twice
   or
   delete the connection by pressing <Del>.

In the following example the connection of the top output of the “/” CE to the top input of the “&” CE is deleted:

Segment plan before deletion; the cursor is placed on the connection of the “/” and “&” CEs. The connection is deleted by pressing <Enter> or the left mouse button twice.

Segment plan after deleting the connection.
10.27 Display junctions

There is a possibility of displaying knots.

The additional display of knots is toggled by pressing <Ctrl><F7>.

The following example shows the additional display of knots:

Segment plan with displayed knots.

Notice:
The knots displayed with the “*” character (when printing) differ from the character on the monitor display (04H, rhomb), that can’t be printed.

10.28 Inserting/deleting lines

There is a possibility of inserting or deleting lines in / from a segment plan.

Inserting lines:
1. Move the cursor to the position the line is to be inserted at with the cursor keys or mouse.
2. Insert a line above of the cursor position by pressing <Ctrl>-N.

Deleting lines:
1. Move the cursor onto the line which is to be deleted with the cursor keys or mouse.
2. Delete the line by pressing <Ctrl>-Y.

Notice:
The lines that are to be deleted cannot contain any further elements apart from vertical connection lines. Otherwise the line isn’t deleted and an error message is displayed.
A line is to be inserted above of the “/” CE in the following example:

Segment plan before inserting. The cursor is placed on the top frame of the “/” CE.

Segment plan after insertion.
10.29 Inserting/deleting columns

There is a possibility of inserting or deleting columns within a network.
The area that is to be moved is limited to the upper and lower free line.

**Inserting columns:**
1. Move the cursor to the position the column is to be inserted in the network with the cursor keys or mouse.
2. Insert the column by pressing <F2>.

Notice:
Columns can only be inserted if there is enough space for movement or if columns can be deleted from the right area.

**Deleting columns:**
1. Move the cursor to the column which is to be deleted with the cursor keys or mouse.
2. Delete the appropriate column by pressing <F1>.

Notice:
Columns only containing connection lines can be deleted.

The following example shows how to insert a column front of the “&” CE:

![Diagram showing column insertion](image)

Segment plan before inserting; the cursor is placed left of the “&” CE.
Segment plan after insertion.

**10.30 Comments/Control commands**

Insert comment line:
An empty line or a line only containing vertical connection lines is marked as a comment line in a segment plan by pressing <F5>. If the cursor is placed on a comment line or directly underneath of one a further comment line is inserted.

The shape of the cursor changes from block cursor to text cursor in comment lines. The comment editor is activated by entering an alphanumerical character or <F7>. The input is accepted by pressing <Esc>.

Notice:
Comments beginning with a semikolon “;” are listed in the IL after compilation.

Deleting comment lines:
If the cursor is placed on the comment line it is deleted by pressing <F6>.

The following example shows a comment line:

```
PROJECT: EXAMPLE
SP_No:  2  ABS
TX: 7   TY: 15

............

; This is a comment line that is inserted in the instruction list.

............
```
Control commands:
Comment lines beginning with "@" characters are interpreted as control commands by the FBD/LD compiler.

10.31 Searching

Search menu:
Depending on the selected search function the appropriate search menu is displayed in the FBD/LD editor at the top right. In this menu the search-object, range to be searched and search direction is entered.

The following example shows the FBD/LD editor with displayed search menu "Search absolute variable":

All search functions:
Change between the single fields with the cursor keys. The search function is started after the parameters were entered by pressing <Enter>. The search menu is exited without searching by pressing <Esc>. If the stated object is found the cursor is placed on this object. The search is continued by pressing <Ctrl>-L.

Parameters:
Area : The area the object is to be searched in is stated with this parameter. It is possible to search in the complete project, in a module (only in the case of modularized projects), in the current segment plan or in a block marked in the segment plan management.

Dir. : It is to be stated, whether the selected area from the cursor position on should be searched through up or down.

A message is displayed in the status line if the object wasn’t found.

The following search possibilities are available:

Search for variable (<Ctrl>-Q-F-A):
The search menu “Search for variables” appears when the key combination <Ctrl>-Q-F-A is entered. The desired variable must now be entered. A distinction is made between three cases:
- Only the absolute designation ("Abs. Var.") is entered. In this case, all variables are searched for with this designation, irrespective of any symbol being present.

- Only the symbolic designation ("Sym. Var.") is entered. All variables which have this symbol are then found, irrespective of the absolute designation.

- Both fields are filled out. In this case, the search is also unambiguous in the cases where an absolute variable exists with different symbols (or vice versa).

The search is started after the other parameters are entered and the <Enter key> is pressed.

**Search for a CE (<Shift>+<F8>):**
After the CE name entered the search is started by pressing <Enter>. The cursor is placed on the first CE found.

**Search for unoccupied CE connections (<Shift>+<F9>):**
After entering the parameters and the starting the search the cursor is placed on the first unoccupied connection.

**Search for not absolutely defined variable fields (<Shift>+<F7>):**
After entering the parameters and starting the search the cursor is placed on the first connection that isn’t assigned to a variable. Connections to symbolical variables with entered operand identifiers but missing operand numbers are also found.

**Continue search (<Ctrl>-L):**
The appropriate search is continued by pressing <Ctrl>-L. The cursor is placed on the next object found.

---

### 10.32 Search and replace

It is possible to search for variables and CEs and to replace them with other ones. If a search condition is found, the cursor jumps to the corresponding program point and you are asked whether this point should be replaced. The cursor then jumps to the next program point corresponding to the search condition. If all points are to be replaced without interrogation, this can be done with the “global” setting. After all points have been replaced, the number of changed program points is displayed in the message line.

#### Search and replace variables (<Ctrl>-Q-A-A):

The appropriate search menu is displayed in the editor after the above combination keys is pressed or after this function has been selected in the menu.

The absolute and/or symbolic designation of the (old) variable to be replaced can be entered in the top two fields. A distinction is made between the following cases when entering the new variable designation:

- Only the absolute designation (“New abs. Var.”) is entered. In this case, all variables found receive the new absolute designation. Any existing symbols remain unchanged.

- Only the symbolic designation (“New sym. Var.”) is entered. In this case, all variables found receive the new symbol. An already existing absolute variable designation is preserved.

- If both designations are entered, the absolute designation and the symbol are replaced for the variables found.

The function is started after the range and direction are selected and after the <Enter key> is pressed.

#### Replacing ranges of variables (<Ctrl>-Q-A-W):

It is possible to replace entire variable ranges with <Ctrl>-Q-A-W. To do so, state the “First variable”, the “Last variable” and the “Destination variable” in the search menu. The appropriate variable ranges are replaced after starting with the <Enter key>.
Note:
If variables are replaced, no plausibility test is run for CE parameters.

Example:
The variables “E 00,00” to “E 03,15” are to be replaced by the variables “E 03,00” to “E 06,15”.
Enter “E 00,00” for the first variable. The end variable becomes “E 03,15” and “E 03,00” is the destination variable.

Searching for and replacing CEs (<Shift><F3>):
The appropriate search menu is displayed in the FBD/LD editor after the keys <Shift><F3> are pressed. The search is started with the <Enter key> after the old and new CE names are entered.

Note:
So that CEs can be replaced, the number of connections, plausibility groups and defined doublings of both CEs must be the same.

10.33 Dividing/linking segment plans
There is the possibility of splitting the current segment plan or linking the current segment plan with the following one.

Split segment plan:
1. Move the cursor to the position the segment plan is to be split.
2. Split the segment plan by pressing the function keys <Shift><F1>.
The rest of the current segment plan underneath of the cursor position is entered in the segment plan management with the same designation.
It contains the next following number of the current segment plan. The segment plan numbers of following segment plans are incremented.

Notice:
The segment plan that was created with this function is handled as any other segment plan in the segment plan management (e.g: move).

Link segment plans:
The current segment plan is linked with the next following one by pressing <Shift><F2>. The next following segment plan is appended to the end of the current segment plan and deleted from the segment plan management. The segment plan numbers of the following segment plans are decremented.

Notice:
A message is displayed if the current segment plan can’t be linked with the following one.

10.34 Handling blocks
In segment plans of the FBD/LD editor it is possible to read, write, copy, move and delete blocks.
The file format of the blocks is identical to the blocks of the segment plan management. The block files can be created and read from both layers. That makes it possible to read a block, that was saved as a block in a segment plan, into the segment plan management as a new segment plan.
The block has to be marked for write and delete operations.
The block marks are displayed as lines in the FBD/LD editor. The block marks can be set anywhere apart from inside of a network.
The order of marking is arbitrary.
Mark block:

Mark beginning of block  <Ctrl>-K-B
Mark end of block        <Ctrl>-K-K
Delete block marks       <Ctrl>-K-H

Save block:
A marked block in a segment plan is saved after a filename was entered.

Save block            <Ctrl>-K-W
The block marks are deleted in the FBD/LD after the block has been saved.

Read block:
The block is inserted into the segment plan at the current cursor position after a filename has been entered. The cursor must be placed in an empty line.

Read block            <Ctrl>-K-R
Copy block             <Ctrl>-K-C

Copying blocks:
The previously marked block is inserted into the segment plan at the current cursor position. The cursor must be in a blank line here.

Moving block:
The marked block is inserted into the segment plan at the current cursor position and is deleted in the old position.
The cursor must be in a blank line here.

Move block             <Ctrl>-K-V

If the block contains several segment plans a message is displayed asking if the read segment plans are to be linked.

If they aren’t to be linked or if there isn’t enough space in the current segment plan, the segment plans that haven’t been linked are appended to the current one in the segment plan management.

If there are networks under the cursor position they are placed in a further segment plan behind the read ones.

This is also done if there isn’t enough space available in the current segment plan when inserting a block containing a segment plan.

Deleting blocks:
The marked block in the current segment plan is deleted. The content underneath of the block is moved to the position of the deleted block.

Delete block           <Ctrl>-K-Y

10.35 Plausibility check

The plausibility check tests if assignments of variables and connections to CEs follow the rules of the program logic and PLC at an early point of programming before the program is compiled to IL.

Groups of operands are created in the library with the plausibility group editor. A group containing a list of operands is then assigned to every CE connection with the CE editor.

Criterions for the plausibility check are the data size of variables (bit, word, ...), and kind of operands (input, output, constant, flag,...). The plausibility check is based on the stated variable connection possibilities that were entered with the CE graphic. This makes the programming system as flexible as possible according to PLC features and requirement of the program logic.
Three cases are tested with the plausibility check:

1. Connection of variable and CE.
2. The direct connection of two CEs or the connection of operand - CE in the FBD or the connection operand - LD network is a vertical line.
3. Connection of variable - variable.

1:
In this case it is checked if the variable is listed in the entered group of this CE connection.

2:
The compiler needs a flag for this connection which is automatically taken from the flag pool. The data width of the flag must be unique. To assure this a common group is created containing the flag types of the plausibility group of the output of the left CE and the flags of the plausibility group of the input of the right CE. Then it is checked if a unique flag is listed in this common group.

A message is displayed and the input is rejected if there are several flag types which make unique flag assignment impossible or if the common group is empty.

3:
The direction of the assignment is tested in this case: values can’t be assigned to constants, for example, but a constant can be assigned to a variable.

If errors according to the plausibility check should occur while editing it must be checked, if the written program requires such a connection and if the PLC supports it. If this is the case groups must be modified or defined anew in the plausibility group editor or CE editor.

10.36 Error file
If errors should occur in the instruction list while compiling the FBD/LD, the errors are written into a file called “projectname.ERR”.

The contents of this file is displayed in the bottom line of the FBD/LD editor by pressing <Shift><F6> (several times if necessary).

If possible, the cursor automatically moves to the position in the FBD/LD where the error occurred. This position can be wrong if changes were done to the FBD/LD after compilation.

Exit
The FBD/LD - editor is exited by pressing <ESC>.

A request is displayed asking if changes are to be stored (Yes) under the project name or not (No). The FBD/LD editor is started again by pressing <ESC> once again.

10.37 Compiler
The FBD/LD-compiler reads the entered network structure of CEs or contacts from the FBD/LD file. The segment plans of the FBD/LD are compiled to instruction list using the instruction lists of the CEs in the library.

The compilation is done in three phases:

Phase 1
Phase 1 performs a line analysis and detects missing variables. Also a “KOPNETZ” line is implemented for every LD network and the LD networks are saved in a temporary file.

Phase 2
Phase 2 calculates a table that contains an order in which the CEs must be compiled. Contact networks are handled as CEs with one input and one output.
It is compiled from the top left to the bottom right.
As soon as every input line of a single CE is in a defined state, the CE is entered in the list.

Example:

```
E 0000
A 0000

E 0001
E 0003

E 0004
E 0005

E 0006
E 0007

d1
A 0001

/CE 1

/CE 2

& CE 3

E 0003
A 0000

KOP 1

E 0003
A 0004

d4
A 0005

d5

E 0006
E 001

I 0008

CE 4

A 0002
```

Compilation order:
d1 - d2 - CE 1 - d4 - d5 - CE 3 - CE 2 - LD 1 - d3 - CE 4

Pseudoflags are automatically set for direct connections of two CEs and eventually in contact networks.
They are replaced with appropriate flags from the flagpool stated in the data range editor during code generation.

Any errors that occurred during compilation are written into a file:

```
projectname.err
```

and are displayed in the FBD/LD editor.

**Phase 3**
The prepared FBD/LD is converted to instruction list in phase 3. The compiler accesses the CE/IL library.
The appropriate CE IL of every CE or direct connection is read.
By means of the names of the formal parameters an assignment to the current parameters is performed.
The current parameter at the terminal of a CE is entered in the (possibly duplicated) IL at the corresponding position.
Should one of the variable types be invalid in the IL, a message is displayed.
If the FBD/LD was compiled successfully further references are written into the FBD/LD-files for an ONLINE diagnosis.
If “GLOBAL” assignments for flags was set, the status of direct connections of CEs are also displayed in the online display of the FBD/LD.
Caution:
The library must contain the following CEs for direct assignments, as long as the PLC supports the specific data types.

```
:= for bit values
:=W for word values
:=D for doubleword values
```
Compiling segment plans:
Only the segment plans that were modified or edited are compiled when compiling segment plans.
If a network was inserted into a present project, reserves are needed. There is a possibility of setting a code and a segment plan reserve in the reserve configuration.
The program setup then looks as follows:

<table>
<thead>
<tr>
<th>SP reserve for SP1</th>
<th>NOP reserve</th>
</tr>
</thead>
<tbody>
<tr>
<td>Segment plan 1</td>
<td></td>
</tr>
<tr>
<td>SP reserve for SP2</td>
<td>NOP reserve</td>
</tr>
<tr>
<td>Segment plan 2</td>
<td></td>
</tr>
<tr>
<td>SP reserve for SP3</td>
<td>NOP reserve</td>
</tr>
<tr>
<td>Segment plan 3</td>
<td></td>
</tr>
<tr>
<td>Module reserve</td>
<td>Module reserve</td>
</tr>
</tbody>
</table>

These reserves are preset in the reserve configuration. The following window is displayed after the reserve configuration was selected per menu:

<table>
<thead>
<tr>
<th>Reserved configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default-F:</td>
</tr>
<tr>
<td>Module_code_res:</td>
</tr>
<tr>
<td>SP reserve:</td>
</tr>
<tr>
<td>Default SP res_:</td>
</tr>
</tbody>
</table>

Default-F: Preset of which flags are to be used (global/local)
Module_code_res: Preset of the size of the code-reserve
SP reserve: Preset of the size of the segment plan reserve
Values can be entered absolute or in percent.
There is also a possibility of entering parameters for single segment plans independent of the preset values.
Control commands beginning with “@” are available for setting parameters in single segment plans. These commands must be entered at the top of a segment plan. Only comments are allowed above of these commands.
**Modes of setting flags**

For direct connections of CEs in a FBD/LD network flags are set. There are two forms of setting flags for this use. When global flags are set, a unique flag is set for every line in the project. If local flags are set, a flag is set for every line in a segment plan, but the same data range is used for setting flags in the next segment plan. This means that one flag is used for several lines of different segment plans.

The flag setting (global/local) is stated with the reserve configuration or entered as control command in a comment field of the editors.

Syntax:  
\[ \text{@GLO} \quad \text{(global flags)} \]  
\[ \text{@LOC} \quad \text{(local flags)} \]

If the segment plan does not contain either of the two commands, the type of flag assignment is determined by the default setting. These two commands are also valid for the further segment plans. If the flag assignment is to be set differently in the next segment plan, enter the corresponding @GLO- or @LOC command there.

**Code reserve (907 PC 332 only)**

To save having to transmit the complete program to the PLC it is possible to enter a code reserve for every segment plan.

The reserve is entered in a comment field of the appropriate editor.

Syntax:  
\[ \text{@RES [No.] [%] [options]} \]

with:  
No.: Reserve  
%: Percent or absolute  
options: C = Code reserve (always absolute,"%" is ignored)

A @RES command is valid for the current segment plan.

If a segment plan doesn't contain this command, the default reserve of the reserve configuration is used.

**Module reserve**

The module reserve is set in the reserve configuration.

**Compilation modes**

There are three modes when compiling the segment plans:

- Modifications
- Modifications and reserves
- All

**a) Compiling modifications**

The modified segment plans are compiled without taking notice of new entered reserves. If the former reserves are too small an error message “Reserve too small” is displayed. In this case the compilation mode “Modifications and reserves” must be chosen.

As the individual PLC systems react differently when activating changes, in all cases you should observe the explanations of the “Program changes” function beforehand in the PLC communication menu in the ONLINE part of the system specific description.

**b) Modifications and reserves**

In this mode all reserves are set again:

The PLC code is eventually moved (because of modified code reserve).

Jump labels have to be reassigned when compiling to PLC code and the complete code has to be transmitted to the PLC.

If the ranges of the flagpools are modified, the automatically assigned flags have to be moved and the whole project to be transferred.
c) Compiling all
The project is handled as if it never had been compiled. All segment plans are compiled.

**10.38 Displaying the compiled IL**

After the FBD/LD has been compiled successfully it is possible to display the single segment plans containing the compiled IL. Change to other segment plans with the segment plan management.

As the size of IL to be displayed is limited, it is possible that the message “Can’t display compiled segment plan IL, please split segment plan.”. In this case the segment plan must be split in the editor a level further up with the according functions before the compiled IL can be displayed.

The IL is displayed with the extended IL editor and all cursor and search functions are available. Only the edit functions are locked so that the IL can’t be modified.

**10.39 Saving project**

It is possible to save the FBD/LD project without having to leave the editor.

After pressing <Shift>-F4 a savety request is displayed.

“Save project <Y/N>”

The function is aborted by pressing <ESC>

Caution:

It is not possible to cancel changes done to the project when leaving 907 PC 33 after the project was saved with this function.
11 Extended IL editor

11.1 Introduction
The extended IL editor makes it possible to enter Advant Controller PLC instructions.

Variables (PLC inputs, outputs, flags etc.) can be entered with symbols and long texts. This supports symbolical programming.

The line and sentence numbers are created automatically.

Comments can be inserted at any place.

The extended IL editor makes it possible to call CEs which are listed in the CE library. The program code belonging to the appropriate CE is then used as a macro.

Absolute variables have to be assigned to the formal parameters of the CE.

Instruction list of the extended IL editor with the "&" CE as macro:

<table>
<thead>
<tr>
<th>Sn</th>
<th>Ln</th>
<th>Command / Variable</th>
<th>Symbol</th>
<th>Long text</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000</td>
<td>00000</td>
<td>!</td>
<td>E 0.00,02</td>
<td>Fill</td>
</tr>
<tr>
<td>00001</td>
<td>=S</td>
<td>A 0.03,11</td>
<td>Fillvalf</td>
<td>Close fill valve</td>
</tr>
<tr>
<td>00002</td>
<td>00003</td>
<td>; Reached upper fill level</td>
<td>Fillpump</td>
<td>OFF Fill valve OFF</td>
</tr>
<tr>
<td>00004</td>
<td>;</td>
<td>00005</td>
<td>$ &amp;</td>
<td></td>
</tr>
<tr>
<td>00006</td>
<td>M 005,03</td>
<td>Fill OK</td>
<td>Fill level OK</td>
<td></td>
</tr>
<tr>
<td>00007</td>
<td>E 0.01,05</td>
<td>Flow</td>
<td>Flow available</td>
<td></td>
</tr>
<tr>
<td>00008</td>
<td>M 001,02</td>
<td>flag</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00009</td>
<td>00010</td>
<td>!</td>
<td>M 001,01</td>
<td></td>
</tr>
<tr>
<td>00011</td>
<td>=S</td>
<td>A 0.03,01</td>
<td>Fillpump</td>
<td>Fill pump OFF</td>
</tr>
<tr>
<td>00012</td>
<td>=S</td>
<td>A 0.03,11</td>
<td>Fillvalf</td>
<td>Fill valve off</td>
</tr>
<tr>
<td>00013</td>
<td>=S</td>
<td>A 0.03,12</td>
<td>Flow</td>
<td>Flow measurement OFF</td>
</tr>
<tr>
<td>00014</td>
<td>00015</td>
<td>; Fillpump</td>
<td>OFF, CLOSE fill valve, flow measurement OFF</td>
<td></td>
</tr>
<tr>
<td>00016</td>
<td>;</td>
<td>00017</td>
<td>Conveyor belts ON, Robotarm ON</td>
<td></td>
</tr>
<tr>
<td>00018</td>
<td>!</td>
<td>A 0.03,01</td>
<td>Fillpump</td>
<td>Fill pump OFF</td>
</tr>
</tbody>
</table>

Headline

PROJECT: The currently handled project and segment plan is displayed.

SP NO: The number of the current segment plan is displayed.

The insert mode or overwrite mode is displayed according to the used mode on the right side in the headline.

11.2 Keys

Using the mousebuttons for Yes/No requests:

Questions that have to be answered with yes (Y) or no (N) can be answered with the keyboard and with the mouse. The left mousebutton is equivalent to "yes" and the right one is equivalent to "no".

Changing the scroll speed with the mouse:

If the cursor is placed on the outer frame of the extended IL editor, the scroll speed can be changed with the mouse keys.
The scroll speed is reduced a step by pressing the left mouse button and incremented by pressing the right mouse button.

The scroll speed can also be modified with a menu that is displayed by pressing <Ctrl><F5>.

Function keys:

- `<F1>` : invert a CE parameter
- `<F3>` : compile the extended IL
- `<F4>` : ONLINE on/off
- `<F8>` : library
- `<F10>` : help
- `<Shift>` <F1> : split the current segment plan at the cursor position
- `<Shift>` <F2> : link the current segment plan with the following one
- `<Shift>` <F3> : search for and replace CE
- `<Shift>` <F4> : save project
- `<Shift>` <F5> : CE selection menu
- `<Shift>` <F6> : display compiler errors
- `<Shift>` <F7> : search for undefined variable field
- `<Shift>` <F8> : search for CE
- `<Shift>` <F9> : search for unoccupied CE connection
- `<Shift>` <F10> : segment plan management
- `<Ctrl>` <F1>` : select layout (menu)
- `<Ctrl>` <F3> : display compiled IL
- `<Ctrl>` <F5> : set scroll speed per menu
- `<Ctrl>` <F8> : display CE documentation

Edit commands:

- `<Enter>` : confirm the input
- `<Ins>` : insert mode (only for the current field)
- `<Del>` : delete character
- `<Ctrl>-N` : insert line or duplicate block (depending from the current line the cursor is placed in)
- `<Ctrl>-Y` : delete line/doubling block/connection element (depending from the current line the cursor is placed in)
- `<Ctrl>-Z` : call single line editor for variables

Cursor movement:

- `<Cursor left>` : Moves cursor one position to the left. The display is scrolled if the cursor is placed at the edge of the screen.
- `<Cursor right>` : Moves cursor one position to the right. The display is scrolled if the cursor is placed at the edge of the screen.
- `<Cursor up>` : Moves cursor one position upwards. The display is scrolled if the cursor is placed at the top edge of the screen.
- `<Cursor down>` : Moves cursor one position downwards. The display is scrolled if the cursor is placed at the bottom edge of the screen.
- `<Tab left>` : Moves cursor one tab position to the left (onto the previous input field).
- `<Tab right>` : Moves cursor onetab position to the right (onto the next input field).
- `<PgUp>` : Moves cursor one page backwards.
- `<PgDn>` : Moves cursor one page forwards.
1. The cursor isn’t placed in the 1st line of the segment plan:
   The cursor jumps to the first line of the segment plan.
2. The cursor is already placed in the 1st line of the segment plan:
   The cursor jumps to the 1st line of the former segment plan.

1. The cursor isn’t placed in the last line of the segment plan:
   The cursor moves to the last line of the segment plan.
2. The cursor is already placed in the last line of the segment plan:
   The cursor jumps to the last line of the following segment plan.

Search commands

<Ctrl>-Q-F-A : search for variables
<Ctrl>-Q-F-S : search for sentence number
<Ctrl>-Q-F-W : search for word number
<Ctrl>-Q-F-Z : search for line number
<Ctrl>-Q-F-B : search for command

<Ctrl>-Q-A-A : search and replace absolute variable
<Ctrl>-Q-A-W : search and replace range

<Shift><F7> : search for unoccupied variable field
<Shift><F8> : search for CE

<Ctrl>-L : repeat search

Block commands (inside of a segment plan)

<Ctrl>-K-B : mark beginning of block
<Ctrl>-K-K : mark end of block
<Ctrl>-K-H : delete block mark
<Ctrl>-K-Y : delete block
<Ctrl>-K-V : move block
<Ctrl>-K-C : copy block
<Ctrl>-K-W : write block
<Ctrl>-K-R : read block

Exit:
<Esc> : leave the editor
11.3 Menu

All functions of the extended IL editor can be called per menu.

Extended IL editor with displayed main menu:

Using the menus

The menu is displayed in the Extended IL editor by pressing the spacebar or right mouse button.
The single functions are selected by moving the cursor with the <CrsrUp> and <CurDn> keys or equivalent movement with the mouse.
The selected function is actually executed by pressing <Enter> or the left mouse button.
The functions are also selected by striking the highlighted letter of the function (hotkey).
The menu is left without executing the selected function by pressing the right mouse button or the space bar.
Notice:
The menu is always used the same way and the menu functions work the same way when editing. Therefore the menu functions aren’t deliberately described.

11.4 Management

A program is put together from several segment plans with the Extended IL editor. The management of the single segment plans is done with the segment plan management.
The management is displayed in the right side of the Extended IL editor by pressing <Shift><F10>.

Entering a new segment plan:

1. Move the inverse bar where the new segment plan is to be inserted. To append a new segment plan move the inverse bar underneath of the last entry of the list.
2. Insert the new segment plan by pressing <Ctrl>-N.
3. Edit the new segment plan by pressing <Enter>.
Example

Segment plan with displayed management. Segment plan 5 (still without designation) has just been inserted.

**Editing the designation of segment plans:**

1. Move the inverse bar to the wanted segment plan.
2. The designation is edited by entering an alphanumeric character or pressing <cursor-right>.
3. Finish editing by pressing <Enter>.

**Changing the segment plan:**

1. Move inverse bar to the appropriate segment plan.
2. Call the segment plan by pressing <Enter>.

**Deleting a segment plan:**

1. Move inverse bar to the appropriate segment plan.
2. Delete the segment plan by pressing <Ctrl>-Y.

**Notice:**

As it isn’t possible to delete the displayed segment plan a different one has to be called before deletion.

**Block commands:**

Single or several segment plans can be marked as a block. These marked blocks can be moved, copied, deleted, written onto the harddisk and read from the harddisk.

Segment plans belonging to a marked block are marked with a "*" in the first column.

Mark single segment plans:

1. Move inverse bar to the appropriate segment plan.
2. Mark the segment plan by pressing <F5>.
Notice:
It is possible to mark several single segment plans, that don’t follow each other in the management as one block with this function.

Delete the marks of single segment plans:
1. Move inverse bar to the appropriate segment plan.
2. Delete the mark by pressing <F6>.

Mark beginning of a block:
1. Move inverse bar to the first segment plan of the block.
2. Mark the segment plan as beginning of a block by pressing <Ctrl>-K-B. A "*" is displayed in front of the segment plan stating a block mark.

Mark the end of a block:
1. Move inverse bar to the last segment plan of the block.
2. Mark the segment plan as the end of the block by pressing <Ctrl>-K-K. All segment plans belonging to the block are marked with "*".

Delete the blockmark:
1. Blockmarks are deleted by pressing <Ctrl>-K-H.

Delete the block:
1. Mark the block as described above.
2. Delete the block by pressing <Ctrl>-K-Y.

Move block:
1. Mark the block as described above.
2. Move the inverse bar to the new initial position the block is to be moved to.
3. Move the block by pressing <Ctrl>-K-V.

Copy block:
1. Mark the block as described above.
2. Move the inverse bar to the new initial position the block is to be copied to.
3. Copy the block by pressing <Ctrl>-K-C.

Write block onto harddisk:
1. Mark the block as described above.
2. Call the block write function by pressing <Ctrl>-K-W.
3. Enter the filename of the block in the displayed window. The block is saved by confirming the filename with <Enter>.

Notices:
Should the filename already exist on the harddisk in the same directory (being a former block or a project name) a message is displayed asking if the file is to be overwritten.

Segment plans that have been saved as block can be handled as single projects.

Reading a block from the harddisk:
1. Move the inverse bar to the position the new block is to be read to.
2. Call the block read function by pressing <Ctrl>-K-R.
3. Enter the filename of the block in the displayed window. The block named as the filename just entered is read by pressing <Enter>.
11.5 Management (Searching)

Searching for a segment plan:

It is possible to search for a number or name of a segment plan in the management. The search menu is displayed in the management by pressing <Shift><F3>.

The following figure shows the search menu:

<table>
<thead>
<tr>
<th>Name/No.</th>
<th>Type</th>
<th>Directi.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Name</td>
<td>down</td>
</tr>
<tr>
<td></td>
<td>Number</td>
<td>up</td>
</tr>
</tbody>
</table>

Explanation of the parameters:

Name/No. : Input of the name or number that is to be searched for. If the segment plan's designation isn't known completely a part of the name can be entered. A segment plan called "Turn on motor" is also found if "on" has been entered as name.

Type : This field states if a number or name is searched for.

Directi. : Selects the direction of the search (as of the inverse bar).

After the search has been started by striking the <Enter> key the inverse bar moves to the first found input. The search is continued by pressing <Ctrl>-L.

11.6 Layouts

It is possible to display the Extended IL editor with three different layouts.

After pressing <Ctrl><F1> a menu with the different layouts is displayed. Layout 1 displays the line numbers in the first column. Additionally layout 2 displays the set numbers and in addition layout 3 displays the word numbers.

The default is layout 2 when starting the Extended IL editor.

The following examples show the three layouts:

```
<table>
<thead>
<tr>
<th>LN</th>
<th>Command / Variable</th>
<th>Symbol</th>
<th>Long text</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000</td>
<td>!</td>
<td>E 0.00,02</td>
<td>Fill level indicator</td>
</tr>
<tr>
<td>00001</td>
<td>=S</td>
<td>A 0.03,11</td>
<td>Fillvalv Close fillvalve</td>
</tr>
<tr>
<td>00002</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00003</td>
<td>;En: Reached upper fill level: Fillpump OFF, CLOSE fillvalve</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00004</td>
<td>;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00005</td>
<td>$ &amp;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00006</td>
<td>=E</td>
<td>M 005,03</td>
<td>Fill OK Fill-level OK</td>
</tr>
<tr>
<td>00007</td>
<td>=E</td>
<td>E 01,05</td>
<td>Flow Flow available</td>
</tr>
<tr>
<td>00008</td>
<td>=M</td>
<td>M 001,02</td>
<td>flag</td>
</tr>
<tr>
<td>00009</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0010</td>
<td>!</td>
<td>M 001,01</td>
<td>flag</td>
</tr>
<tr>
<td>0011</td>
<td>=S</td>
<td>A 03,01</td>
<td>Fillpump Fillpump OFF</td>
</tr>
<tr>
<td>0012</td>
<td>=S</td>
<td>A 03,11</td>
<td>Fillvalv close fill valve</td>
</tr>
<tr>
<td>0013</td>
<td>=S</td>
<td>A 03,12</td>
<td>Flow Flow measurement OFF</td>
</tr>
<tr>
<td>0014</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0015</td>
<td>!</td>
<td>;Fillpump OFF, Fillvalve OFF, Flow measurement OFF</td>
<td></td>
</tr>
<tr>
<td>0016</td>
<td>;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0017</td>
<td>:Conveyor belts ON, Robotarm ON</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0018</td>
<td>!</td>
<td>A 03,01</td>
<td>Fillpump Fillpump OFF</td>
</tr>
</tbody>
</table>
```

Layout 1 (with line numbers)
### Layout 2 (with sentence and line numbers)

<table>
<thead>
<tr>
<th>SN</th>
<th>LN</th>
<th>Command / Variable</th>
<th>Symbol</th>
<th>Long text</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000</td>
<td>00000</td>
<td>!</td>
<td>E</td>
<td>0.00,02 Fill-level indicator</td>
</tr>
<tr>
<td></td>
<td>00001</td>
<td>=S</td>
<td>A</td>
<td>0.03,11 Fillvalv Fillvalve closed</td>
</tr>
<tr>
<td></td>
<td>00002</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00003</td>
<td>00003</td>
<td>;Reached upper fill level: Fillpump OFF Fillvalve closed</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>00004</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>00005</td>
<td>$ &amp;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00006</td>
<td>00006</td>
<td>M</td>
<td>005,03</td>
<td>Fill OK Fill-level OK</td>
</tr>
<tr>
<td>00007</td>
<td>00007</td>
<td>E</td>
<td>0.01,05</td>
<td>Flow Flow available</td>
</tr>
<tr>
<td>00008</td>
<td>00008</td>
<td>M</td>
<td>001,02</td>
<td>flag</td>
</tr>
<tr>
<td>00009</td>
<td>00009</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00010</td>
<td>00010</td>
<td>!</td>
<td>M</td>
<td>001,01 flag</td>
</tr>
<tr>
<td>00011</td>
<td>00011</td>
<td>=S</td>
<td>A</td>
<td>0.03,01 Fillpump Fillpump OFF</td>
</tr>
<tr>
<td>00012</td>
<td>00012</td>
<td>=S</td>
<td>A</td>
<td>0.03,11 Fillvalv Fillvalve closed</td>
</tr>
<tr>
<td>00013</td>
<td>00013</td>
<td>=S</td>
<td>A</td>
<td>0.03,12 Flow Flow measurement OFF</td>
</tr>
<tr>
<td>00014</td>
<td>00014</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00015</td>
<td>00015</td>
<td>;Fillpump OFF, Fillvalve OFF, Flow measurement OFF</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>00016</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00017</td>
<td>00017</td>
<td>;Conveyor belts ON, Robotarm ON</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00018</td>
<td>00018</td>
<td>!</td>
<td>A</td>
<td>0.03,01 Fillpump Fillpump OFF</td>
</tr>
</tbody>
</table>

### Layout 3 (with sentence, line and word numbers)

<table>
<thead>
<tr>
<th>SN</th>
<th>LN</th>
<th>WN</th>
<th>Command / Variable</th>
<th>Symbol</th>
<th>Long text</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000</td>
<td>00000</td>
<td></td>
<td>!</td>
<td>E</td>
<td>0.00,02 Fill-level indicator</td>
</tr>
<tr>
<td></td>
<td>00001</td>
<td>00001</td>
<td>=S</td>
<td>A</td>
<td>0.03,11 Fillvalv Fillvalve closed</td>
</tr>
<tr>
<td></td>
<td>00002</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00003</td>
<td>00003</td>
<td></td>
<td>;Reached upper fill level: Fillpump OFF Fillvalve closed</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>00004</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>00005</td>
<td>00005</td>
<td>$ &amp;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00006</td>
<td>00006</td>
<td>00006</td>
<td>M</td>
<td>005,03</td>
<td>Fill OK Fill-level OK</td>
</tr>
<tr>
<td>00007</td>
<td>00007</td>
<td>00007</td>
<td>E</td>
<td>0.01,05</td>
<td>Flow Flow available</td>
</tr>
<tr>
<td>00008</td>
<td>00008</td>
<td>00008</td>
<td>M</td>
<td>001,02</td>
<td>flag</td>
</tr>
<tr>
<td></td>
<td>00009</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00010</td>
<td>00010</td>
<td>00010</td>
<td>!</td>
<td>M</td>
<td>001,01 flag</td>
</tr>
<tr>
<td>00011</td>
<td>00011</td>
<td>00011</td>
<td>=S</td>
<td>A</td>
<td>0.03,01 Fillpump Fillpump OFF</td>
</tr>
<tr>
<td>00012</td>
<td>00012</td>
<td>00012</td>
<td>=S</td>
<td>A</td>
<td>0.03,11 Fillvalv Fillvalve closed</td>
</tr>
<tr>
<td>00013</td>
<td>00013</td>
<td>00013</td>
<td>=S</td>
<td>A</td>
<td>0.03,12 Flow Flow measurement OFF</td>
</tr>
<tr>
<td></td>
<td>00014</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00015</td>
<td>00015</td>
<td>00015</td>
<td>;Fillpump OFF, Fillvalve OFF, Flow measurement OFF</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>00016</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00017</td>
<td>00017</td>
<td>00017</td>
<td>;Conveyor belts ON, Robotarm ON</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00018</td>
<td>00018</td>
<td>00018</td>
<td>!</td>
<td>A</td>
<td>0.03,01 Fillpump Fillpump OFF</td>
</tr>
</tbody>
</table>
11.7 Fields

The following fields are displayed in the Extended IL editor depending on the layout:

**LN:**
The line numbers are displayed in this field. The line number begins with 0 and is increased with every line in every segment plan by 1.

**SN:**
The instruction list is sentence oriented. The sentence numbers are created automatically and displayed in this field every time a new sentence begins. An empty line is inserted between the single sentences to keep a clear overview.

Sentence numbers are only displayed using the layouts 2 and 3.
Sentence numbers start at 0 in every segment plan and increase by 1 with every further step.

**WN:**
A word number is displayed in this field for every entered instruction.
The word number states the position the instruction is placed in the user program memory in the PLC.
Word numbers are only displayed using the layout 3 and in ONLINE mode.

Notice:
Word numbers are created during compilation and are only displayed correctly in the compiled Extended IL. If the Extended IL hasn’t been compiled yet it is possible that invalid word numbers are displayed.

**Command / variable:**
In this field the command and designation of the absolute variable is entered and displayed formatted. The formatting is done after finishing input.

Identification of CEs in the command / variable field of the Extended IL editor:

CE definition: The line containing the definition of the CE in the Extended IL editor is marked with the ‘$’ character. It is followed by a space and the CE’s name.

CE parameters: The lines containing the CE parameters are shown as following:

```
  __|      input
   O |    inverted input
    |      output
   O |    inverted output
```

These characters are followed by the formal parameters of the CE according to the definitions done in the CE editor.

A direct connection of an CE is occupied with by the formal parameter ‘@DIR’.

**Symbol:**
The symbol is entered and displayed in this field. The symbol can contain up to 10 characters.

**Long text:**
A detailed description of a variable or a symbol is entered in this field. The long text can contain up to 30 characters. The long text is edited by calling the single-line editor by pressing <Ctrl>-Z.

**Status line:**
The bottom line is there for displaying messages which support the programmer with his work. Input errors are also displayed.
ONLINE:
In the ONLINE mode this field shows the status of bit values and the value of non-bit values.

Display of an Extended IL with running PLC.

The status of the PLC is displayed in the left half of the screen (second line). The notice on the mode of ONLINE representation is stated on the right side (see chapter "ONLINE" - the key functions in ONLINE mode are also explained in this chapter).

11.8 Commands
Before the first input in the Extended IL editor or after confirming an input with <Enter> the cursor is placed on the first position of the "Command / Variable" field.

Editing a field
The field editor is activated by entering a character, which is valid for the field editor. The shape of the cursor states if input is possible at the current position (Cursor = '_') or input not possible (Cursor = '❚').

The field editor makes it possible to edit the appropriate input field. It is automatically tested if the entered operand is permitted with the entered instruction.

Commands of the field editor are described in the chapter "field editor". The input is automatically formatted after leaving the "Command / variable" field.

Notice that the Extended IL editor doesn’t allow direct inputs modifying a variable/ symbol/ text assignment. To perform changes like this the single line editor for variables must be called by pressing <Ctrl>-Z after placing the cursor on the appropriate line.

Inserting lines
It is possible to insert a line between two lines by pressing <Ctrl>-N.

Now the insert mode is switched on. A line is inserted in the line above of the current cursor position and the cursor is then placed into the new line. The insert mode stays active when input is confirmed with <Enter>.

The edit mode is displayed on the right side of the headline.

Deleting lines
The line according to the cursor position is deleted by pressing <Ctrl>-Y.
11.9 Comments

Input of comments:
There is a possibility of inserting any number of comment lines at any position in the Extended IL editor. Comment lines have to begin with a semicolon (";").

A comment placed between two sentences is automatically assigned to the second sentence, so that the empty line separating the two sentences is inserted between the first sentence and the comment.

To assign a comment to the upper set "SE" (end of sentence) has to be entered behind the comment. Now the empty line is inserted above of the following sentence.

11.10 Connection elements

Entering an CE in the Extended IL editor:
1. Move the cursor to the wanted line with the cursor keys or mouse.
2. Enter the ‘$’ character as CE definition followed by a space and name of the CE.
3. Confirm with <Enter>.

Selecting an CE from the CE menu:
1. Move the cursor to the wanted line with the cursor keys or mouse.
2. Call the CE menu by pressing <Shift><F5>.
3. Move the inverse bar onto the wanted CE.
4. Accept the CE by pressing <Enter> or the left mouse button or leave the CE menu without accepting the CE with <ESC> or right mouse button.

<table>
<thead>
<tr>
<th>SN</th>
<th>LN</th>
<th>Command / Variable</th>
<th>Symbol</th>
<th>Long</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000</td>
<td></td>
<td>!</td>
<td>Fill</td>
<td>Fill-level indica</td>
</tr>
<tr>
<td>00001</td>
<td>=S</td>
<td>A 0.03,11</td>
<td>Fillval</td>
<td>Fill valve closed</td>
</tr>
<tr>
<td>00002</td>
<td></td>
<td>;Reached upper fill level:</td>
<td>Fillpump</td>
<td>OFF Fill valve close</td>
</tr>
<tr>
<td>00004</td>
<td></td>
<td>;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00005</td>
<td>&amp;</td>
<td>M 005,03</td>
<td>Fill OK</td>
<td></td>
</tr>
<tr>
<td>00007</td>
<td></td>
<td>E 0.01,05</td>
<td>Flow</td>
<td>Flow available</td>
</tr>
<tr>
<td>00008</td>
<td></td>
<td>M 001,02</td>
<td>flag</td>
<td></td>
</tr>
<tr>
<td>00009</td>
<td></td>
<td>;</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Extended IL editor with inserted CE '/' and displayed CE library.
Some library functions are also active apart from the CE selection at this point:

- `<Shift><F1>`: display all CEs
- `<Shift><F2>`: CE group selection
- `<Shift><F3>`: CE group assignment
- `<F8>`: Plausibility group editor
- `<F9>`: Order criterion (order in order of input or alphabetical)

The library is called by pressing `<F8>`. The library functions according to the chapter "CE library" can be executed.

**Entering the parameters:**

The parameters of a CE are entered by entering absolute variables in the "Command / variable" field or symbolical variables in the symbol field.

**11.11 Plausibility check**

To avoid invalid input a plausibility test is performed the same way as in the FBD/LD editor after a variable was assigned to a CE parameter.

It is checked if the entered operand is listed in the plausibility group that was selected in the CE editor for the appropriate CE parameter.

**Displaying the CE documentation**

1. Move the cursor inside of the CE of which the documentation is to be displayed with the cursor keys or mouse.
2. The text editor containing the CE documentation of the current CE is displayed by pressing `<Ctrl><F8>`.

or

1. Call the CE menu by pressing `<Shift><F5>.
2. Move the inverse bar onto the wanted CE.
3. The text editor containing the CE documentation of the current CE is called by pressing `<F4>`.

Notice:

To be able to enter an invalid operand, a different group must be entered for this parameter in the CE editor or the parameter must be entered in the present group. The CE editor is called from the Extended IL with displayed CE menu (Plausibility group editor, call by pressing `<F8>`). It is not possible to change the CEs or the plausibility groups in the case of manufacturer CEs.

The designation of the CE must then be reentered in the line defining the CE so that the change of plausibility group becomes active (swapping CE).
11.12 Duplicating CE connections

An input or output can be duplicated if it is needed more than once.

The CE library contains the definitions of which connections can be duplicated.

1. Move the cursor to the parameter line of the connection that is to be duplicated with the cursor keys or mouse.
2. Duplicate the connection by pressing <Ctrl>-N.

The following example shows fictitious CE with its initial assignment and the CE with several times duplicated inputs and outputs:

<table>
<thead>
<tr>
<th>SN</th>
<th>LN</th>
<th>Command / Variable</th>
<th>Symbol</th>
<th>Long text</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$ T3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00000</td>
<td>00000</td>
<td>E100 E</td>
<td>0.01,05</td>
<td>Flow</td>
</tr>
<tr>
<td>00002</td>
<td></td>
<td>E200 E</td>
<td>0.00,02</td>
<td>Fill</td>
</tr>
<tr>
<td>00003</td>
<td></td>
<td>A100 M</td>
<td>001,01</td>
<td>flag</td>
</tr>
<tr>
<td>00004</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00001</td>
<td>00005</td>
<td>$ T3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00006</td>
<td></td>
<td>E100 E</td>
<td>0.01,05</td>
<td>Flow</td>
</tr>
<tr>
<td>00007</td>
<td></td>
<td>E200 E</td>
<td>0.00,02</td>
<td>Fill</td>
</tr>
<tr>
<td>00008</td>
<td></td>
<td>E101</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00009</td>
<td></td>
<td>E201</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00010</td>
<td></td>
<td>E102</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00011</td>
<td></td>
<td>E202</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00012</td>
<td></td>
<td>E103</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00013</td>
<td></td>
<td>E203</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00014</td>
<td></td>
<td>A100 M</td>
<td>001,01</td>
<td>flag</td>
</tr>
<tr>
<td>00015</td>
<td></td>
<td>A101</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00016</td>
<td></td>
<td>A102</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00017</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Notices:

The duplication number is automatically appended to the max. two digit destination of the connection, which can be duplicated.

Up to 241 inputs and 241 outputs can be connected to a CE.

The duplication type when duplicating several connections at once is defined in the CE editor (see library CE editor) under "Duplication type".

An error message is displayed if it is tried to duplicate a connection which can't be duplicated.

11.13 Inverting CE connections

Inputs and outputs of a CE can be inverted.

The CE library contains the definitions of which connections can be inverted.

1. Move the cursor to the parameter line of the connection that is to be inverted with the cursor keys or mouse.
2. Invert the connection with <F1>. It is reinverted (former state) by pressing <F1> once again.

The inversion is shown with the 'o' character.

The following example shows an "&" CE with inverted input and output:
11.14 Deleting CE connections

There is a possibility of deleting duplicated CE connections.

1. Move the cursor to the parameter line of the connection which is to be deleted with the cursor keys or mouse.
2. Delete the connection by pressing <Ctrl>-Y.

Notice:

CE connections with variables are also deleted without warning.

11.15 Deleting/Swapping CEs

There is a possibility of deleting CEs in the segment plan if necessary.

1. Move the cursor to the line containing the CE, which is to be deleted, with the cursor keys or mouse.
   The line is designated with the "$" character.
2. Delete the CE by pressing <Ctrl>-Y.

Notice:

A safety request is displayed on the status line as all parameters are also deleted.

**Swapping CEs**

The CE of an Extended IL can be swapped with a CE from the CE library, if their connections are compatible.

When swapping CEs the features of the old CE are transferred to the new one. This goes for the following features:

1. Inversion permitted (Yes/No)
2. Display the formal parameter name (Yes/No)
3. Duplication permitted (Yes/No)
4. Plausibility test of connected variables

Procedure:

1. Move the cursor to the designation field of the CE which is to be swapped.

---

**Table:**

<table>
<thead>
<tr>
<th>SN</th>
<th>LN</th>
<th>Command / Variable</th>
<th>Symbol</th>
<th>Long text</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000</td>
<td>00000</td>
<td>!</td>
<td>E</td>
<td>0.00,02</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>=S</td>
<td>A</td>
</tr>
<tr>
<td>00002</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00003</td>
<td>;Reached upper fill level: Fillpump OFF Fillvalve closed</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00004</td>
<td>;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00005</td>
<td>$ &amp;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00006</td>
<td>M</td>
<td>005,03</td>
<td>Fill OK</td>
<td>Filllevel OK</td>
</tr>
<tr>
<td>00007</td>
<td>E</td>
<td>0.01,05</td>
<td>Flow available</td>
<td></td>
</tr>
<tr>
<td>00008</td>
<td>o</td>
<td>M</td>
<td>001,02</td>
<td>flag</td>
</tr>
<tr>
<td>00009</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00010</td>
<td>!</td>
<td>M</td>
<td>001,01</td>
<td>flag</td>
</tr>
<tr>
<td>00011</td>
<td>=S</td>
<td>A</td>
<td>0.03,01</td>
<td>Fillpump Fillvalve OFF</td>
</tr>
<tr>
<td>00012</td>
<td>=S</td>
<td>A</td>
<td>0.03,11</td>
<td>Fillvalve closed</td>
</tr>
<tr>
<td>00013</td>
<td>=S</td>
<td>A</td>
<td>0.03,12</td>
<td>Flow measurement OFF</td>
</tr>
<tr>
<td>00014</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00015</td>
<td>;Fillpump OFF, Fillvalve OFF, Flow measurement OFF</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00016</td>
<td>;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00017</td>
<td>;Conveyor belts ON, Robotarm ON</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00018</td>
<td>!</td>
<td>A</td>
<td>0.03,01</td>
<td>Fillpump Fillvalve OFF</td>
</tr>
</tbody>
</table>

---

907 PC 33/General Part/Issued: 11.98
2. Enter the description of the new CE or select the new CE with the CE selection menu.
3. Swap the CEs by pressing <Enter>.

Notice:
If the graphics of a CE had been modified in the library it must also be swapped in the Extended IL editor so that the changes become active in the Extended IL.

11.16 Searching

Search menu:
Depending on the selected search function the appropriate search menu is displayed in the Extended IL editor at the top right. In this menu the search object, range to be searched and search direction is entered.

The following example shows the search menu "Search for variable":

<table>
<thead>
<tr>
<th>Search for variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abs. Var.</td>
</tr>
<tr>
<td>Sym. Var.</td>
</tr>
<tr>
<td>Range: Project</td>
</tr>
<tr>
<td>Module mark.SP</td>
</tr>
<tr>
<td>cur.SP</td>
</tr>
<tr>
<td>Direct.: forwards</td>
</tr>
<tr>
<td>backwards</td>
</tr>
</tbody>
</table>

All search functions:
Change between the single fields with the cursor keys. The search function is started after the parameters were entered by pressing <Enter>. The search menu is exited without searching by pressing <Esc>. If the stated object is found the cursor is placed on this object. The search is continued by pressing <Ctrl>-L.

Parameters:
Range : The range in which the corresponding object is to be searched for.
        It is possible to search in the entire project, in the module (only in the case of modularized projects), in a block marked in the segment plan manager or in the current segment plan.
Direct. : Selects the direction of the search (as of the inverse bar).
A message is displayed in the status line if the object wasn’t found.

The following search possibilities are available:

Search for variables (<Ctrl>-Q-F-A):
The "Search for variables" search menu appears after the key combination <Ctrl>-Q-F-A is entered. The variables being searched for must now be entered. A distinction is made between 3 cases here:
– Only the absolute designation ("Abs. Var.") is entered. In this case, all variables with this designation are searched for, irrespective of any symbol present.
– Only the symbolic designation ("Sym. Var.") is entered. Then, all variables which have the symbol are found, irrespective of the absolute designation.
– Both fields are filled out. In this case, the search is also unambiguous even in cases where an absolute variable exists with different symbols (or vice versa).
The search is started after entering the usual parameters and pressing the <Enter key>. 

907 PC 33/General Part/Issued: 11.98
Search for line number (<Ctrl>-Q-F-Z):
After <Ctrl>-Q-F-Z is entered, the program demands that the line number to be jumped to be entered. The cursor then jumps to the desired location.

Search for a sentence number (<Ctrl>-Q-F-S):
A sentence is searched for by pressing <Ctrl>-Q-F-S.

Search for a word number (<Ctrl>-Q-F-W):
It is possible to search for an entered word number by pressing <Ctrl>-Q-F-W.

Notice:
The word numbers are created during compilation and are displayed in the compiled Extended IL correctly. If the Extended IL hasn’t been compiled yet it is possible that invalid word numbers are displayed.

Search for commands with and without operand (<Ctrl>-Q-F-B):
A command is searched for by pressing <Ctrl>-Q-F-B followed by the wanted program line, for example !PE, NOP or !E 00,00.

Search for not absolutely defined variable fields (<Shift><F7>):
After entering the parameters and starting the search the cursor is placed on the first connection that isn’t assigned to a variable. Connections to symbolical variables with entered operand labels but missing operand numbers are also found.

Search for a CE (<Shift><F8>):
After the CE name was entered the search is started by pressing <Enter>. The cursor is placed on the first CE found.

Search for unoccupied CE connections (<Shift><F9>):
After entering the parameters and starting the search the cursor is placed on the first unoccupied connection in the Extended IL editor.

Continue search (<Ctrl>-L):
The appropriate search is continued by pressing <Ctrl>-L. The cursor is placed on the next object found. If no further object is found a message is displayed in the status line.

11.17 Search and replace
It is possible to search for variables and CEs and replace them with other ones. If a search condition is found, the cursor jumps to the corresponding program point and you are asked whether this point should be replaced. Next, the cursor jumps to the next program point corresponding to the search condition. If all points are to be replaced without interrogation, this can be done with the “global” setting. After all points are replaced, the number of changed program points is displayed in the message line.

Search for and replace variables (<Ctrl>-Q-A-A):
The appropriate search window is displayed in the IL after the above combination of keys is pressed or after this function has been selected in the menu.
The absolute and/or symbolic designation of the (old) variable to be replaced can be entered in the top two fields. A distinction is made between the following cases when entering the new variable destination:
– Only the absolute designation (“New abs. Var.”) is entered. In this case, all variables found receive the new absolute designation. Any existing symbols remain unchanged.
– Only the symbolic designation (“New sym. Var.”) is entered. In this case, all variables found receive the new symbol.
– If both designations are entered, the absolute designation and the symbol are replaced for the variables found.

The function is started after the range and direction are selected.

Replacing ranges of variables (<Ctrl>-Q-A-W):

There is a possibility of replacing ranges of variables. The "Start variable", "End variable" and the "target variable" must be entered. Then the stated range is replaced by pressing <Enter>.

Notice:

No type or group check is carried out for the CE parameters during the replacement.

Example:

The variable range "E 00,00" to "E 03,15" must be replaced with "E 03,00" to "E 06,15". The start variable is "E 00,00", the end variable is "E 03,15" and the target variable is "E 03,00".

Searching for and replacing CEs (<Shift><F3>):

The appropriate search menu is displayed in the ext. IL after the keys <Shift><F3> are pressed. The search is started with the <Enter key> after the old and new CE names are entered.

Note:

So that CEs can be replaced, the number of connections, the plausibility groups and the defined doublings of both CEs must be the same.

11.18 Splitting/linking segment plans

There is the possibility of splitting the current segment plan or linking the current segment plan with the following one.

Split segment plan into two segment plans:

1. Move the cursor using the cursor keys or the mouse to the position the segment plan is to be split.
2. Split the segment plan by pressing the function keys <Shift><F1>.

The rest of the current segment plan underneath of the cursor position is entered in the segment plan management with the same designation.

It contains the next following number of the current segment plan. The segment plan numbers of following segment plans are incremented.

Notice:

The segment plan that was created with this function is handled as any other segment plan in the segment plan management (e.g: move).

A segment plan can’t be divided inside of a CE.

Link two segment plans to one segment plan:

The current segment plan is linked with the next following one by pressing <Shift><F2>. The next following segment plan is appended to the end of the current segment plan and deleted from the segment plan management. The segment plan numbers of the following segment plans are decremented by one.

Notice:

A message is displayed if the current segment plan can’t be linked with the following one.
11.19 Block processing

In segment plans of the Extended IL editor it is possible to read, write, copy, move and delete blocks.
The block marks are displayed as inverse lines in the Extended IL editor. If the cursor is placed on a CE while marking, the whole CE is marked.
The order of marking is arbitrary, an already marked block is enlarged if necessary.

Marking a block:

Mark beginning of block  \(<Ctrl>-K-B\)
Mark end of block  \(<Ctrl>-K-K\)
Delete block marks  \(<Ctrl>-K-H\)

Save block:
A marked block in a segment plan is saved under a filename to be entered.

Save block  \(<Ctrl>-K-W\)

Read block:
The block is inserted into the segment plan at the current cursor position after a filename has been entered. The cursor must not be placed on a CE.

Read block  \(<Ctrl>-K-R\)

If the block contains several segment plans a message is displayed asking if the read segment plans are to be linked.

If they aren’t to be linked or if there isn’t enough space in the current segment plan, the segment plans that haven’t been linked are appended to the current one in the segment plan management.

If there are instructions under the cursor position, where the block should be inserted into the segment plan, they are placed in a further segment plan behind the read ones.

This is also done if there isn’t enough space available in the current segment plan when inserting a block containing a segment plan.

Notices:
Before reading or writing a block the filename of the file must be entered, in which it is to be written or read from. Should a block already be present with the same filename when saving a block it must be stated if the present block is to be overwritten or if the current block is to be appended to the present one on the harddisk.

The file format of the blocks is identical to the blocks of the segment plan management. The block files can be created and read from both layers. That makes it possible to read a block, that was saved as a block in a segment plan, into the segment plan management.

It is not possible to move marked blocks to other segment plans or to copy them. In this case the block functions of the segment plan management must be used.

Move block:
The marked block is copied above the cursor position and deleted at the initial position. The cursor must be placed outside of the block but not on a parameter line of a CE.

Move block  \(<Ctrl>-K-V\)

Copy block:
To copy a marked block to the position above of the cursor, the cursor must be placed outside of the block but not on a parameter line of a CE.

Copy block  \(<Ctrl>-K-C\)
Delete block:
The marked block of the current segment plan is deleted. The content underneath of the block is moved to where the content of the block used to be.

\[\text{Delete block} \quad \text{<Ctrl>-K-Y}\]

11.20 Error file
If errors should occur in the instruction list while compiling the Extended IL, a message "Errors during Extended IL translation!" is displayed.

The errors are written into a file called "projectname.ERR".

The contents of this file is displayed in the bottom line of the Extended IL editor by pressing <Shift><F6> (several times if necessary).

If possible, the cursor automatically moves to the position in the Extended IL where the error occurred. This position can be wrong if changes were done in the Extended IL after compilation.

Exit
The Extended IL editor is exited by pressing <ESC>.

A request is displayed asking if changes are to be saved or not.

The Extended IL editor is restarted by pressing <ESC> once again.

11.21 Compiler
The edited Extended IL must be compiled if it contains CEs.

For that the translator accesses the CE IL library.

The instruction lists of the CEs listed in the Extended IL are read.

The CEs in the Extended IL editor are replaced with their appropriate instruction lists that were created with the CE editor.

If necessary parts of the instruction lists of the CE are duplicated according to the CE connections in the Extended IL editor.

The formal parameters (PP) used in the CE editor are replaced with the appropriate variables entered at the CE connections in the Extended IL editor during compilation. The current variables are assigned to the formal parameters using their designations.

Should one of the variable types be invalid in the Extended IL, a message is displayed.

Valid word numbers are created during compilation.

If errors occur during compilation they are written into the file "projectname.ERR"

Displaying the translated IL (<Ctrl><F3>)
After the Extended IL has been compiled successfully it is possible to display the compiled IL. Change to other segment plans with the segment plan management.

As the size of IL to be displayed is limited, it is possible that the message "Can't display compiled segment plan IL, please split segment plan.". In this case the segment plan must be split in the editor a level further up with the according functions before the compiled IL can be displayed.

The IL is displayed with the Extended IL editor and all cursor and search functions are available. Only the edit functions are locked so that the IL can't be modified.
Compiling segment planwise:
Only the segment plans that were modified or edited are compiled when compiling segment plans.
If a network was inserted into a present project, reserves are needed.
There is a possibility of setting a code and a segment plan reserve in the reserve configuration.
The program setup then looks as follows:

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Segment plan 1</td>
<td>NOP reserve</td>
</tr>
<tr>
<td>Segment plan 2</td>
<td>NOP reserve</td>
</tr>
<tr>
<td>Segment plan 3</td>
<td>NOP reserve</td>
</tr>
<tr>
<td>Module reserve</td>
<td>Module reserve</td>
</tr>
</tbody>
</table>

These reserves are preset in the reserve configuration. The following window is displayed after the reserve configuration was selected per menu:

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Default-F</td>
<td>global</td>
<td>local</td>
</tr>
<tr>
<td>Module_code_res</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Default_SP_res</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

Default-F: Preset of which flags are to be used (global/local)
Module_code_res: Preset of the size of the code reserve
SP_reserve: Preset of the size of the segment plan reserve
Values can be entered absolute or in percent.
Code reserve
To save having to transmit the complete program to the PLC it is possible to enter a code reserve for every segment plan.
The reserve is entered in a comment field of the appropriate editor.
Syntax: \( \texttt{@RES [no.]} [\%] \texttt{[options]} \)
with:
no.: Reserve
\%: Percent or absolute
options: C = Code reserve (always absolute, "\%" is ignored)
A @RES command is valid for the current segment plan.
If a segment plan doesn’t contain a @RES command, the default reserve of the reserve configuration is used.

Module reserve
The module reserve is set in the reserve configuration.

Compilation modes
There are three modes when compiling the segment plans:
– modifications
– modifications and reserves (only valid for ABB Procontic T200)
– all
a) Compiling modifications
The modified segment plans are compiled without taking notice of new entered reserves. If the former reserves are too small an error message "Reserve too small" is displayed. In this case the compilation mode "changes and reserves" must be chosen.

b) Modifications and reserves (only valid for ABB Procontic T200)
In this mode all reserves are set again:
- The PLC code is eventually moved (because of modified code reserve)
Jump labels have to be reassigned when compiling to PLC code and the complete code must be transmitted to the PLC.

c) Compiling all
The project is handled as if it never had been compiled. All segment plans are compiled.

11.22 Saving data
It is possible to save the data, which were input in the Extended IL under the project name without having to leave the editor.
After pressing \(<\text{Shift}>\)-F4 a savety request is displayed.
"Save project <Y/N>"
The function is aborted by pressing \(<\text{ESC}>\)
Caution:
It is not possible to cancel changes done to the project when leaving 907 PC 33 after the project was saved with this function.
12 Variable editor

12.1 Principle of operation

The variable editor always contains a sorted complete list of all entered absolute variables.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Symbol</th>
<th>Long text</th>
</tr>
</thead>
<tbody>
<tr>
<td>A 03,11</td>
<td>Fillvalv</td>
<td>Filling valve closed</td>
</tr>
<tr>
<td>A 03,12</td>
<td>Flow</td>
<td>Flow measurement OFF</td>
</tr>
<tr>
<td>A 04,00</td>
<td>Message</td>
<td>Warning: Level above high limit</td>
</tr>
<tr>
<td>A 04,01</td>
<td>Alarm</td>
<td>No flow at drain</td>
</tr>
<tr>
<td>A 04,02</td>
<td>Drainvalv</td>
<td>Open drain valve</td>
</tr>
<tr>
<td>A 04,04</td>
<td>Alarm</td>
<td>Drain valve alarm</td>
</tr>
<tr>
<td>A 04,05</td>
<td>Suction ON</td>
<td>Suction pump ON</td>
</tr>
<tr>
<td>A 04,07</td>
<td>Flow</td>
<td>Flow measurement ON</td>
</tr>
<tr>
<td>A 04,10</td>
<td>Alarm</td>
<td>Flow measurement alarm</td>
</tr>
<tr>
<td>A 04,11</td>
<td>Drain CLOSED</td>
<td>Closed drain valve</td>
</tr>
<tr>
<td>A 04,12</td>
<td>Suction OFF</td>
<td>Suction pump OFF</td>
</tr>
<tr>
<td>A 04,13</td>
<td>Flow</td>
<td>Flow measurement OFF</td>
</tr>
<tr>
<td>A 05,00</td>
<td>Alarm</td>
<td>Belt running 1 alarm</td>
</tr>
<tr>
<td>A 05,02</td>
<td>Alarm</td>
<td>Light barrier 1 alarm</td>
</tr>
<tr>
<td>A 05,08</td>
<td>Swivel arm</td>
<td>Start automatic swivel arm</td>
</tr>
<tr>
<td>A 05,12</td>
<td>Hydraulics</td>
<td>Hydraulic arm pickup position</td>
</tr>
<tr>
<td>A 05,13</td>
<td>Solenoid</td>
<td>Activate solenoid</td>
</tr>
</tbody>
</table>

A new variable will be included in the variable list if it has been entered in an editor for program creation (SFC, FBD/LD, Ext. IL) or in the variable editor.

Therefore, various methods of working are possible depending on requirements:

1. The PLC program including absolute variables can be entered first of all in one of the program creation editors. All absolute variables entered will appear automatically in the variable editor and, in this editor, can be described in further detail later by means of their symbols and long texts.

2. All absolute variables, including their symbols and long texts, needed later in the PLC program can be entered first of all in the variable editor. When creating the program in the used editor, the symbol and long text when editing a variable or the absolute variable and long text when entering the symbol will then be displayed automatically.

3. An absolute variable will be described completely in a program creation editor by its symbol and long text. In this case, the variable editor will then only serve to provide an overview of all absolute variables existing in the program.

4. A diversity of further possibilities is provided by combining the first three methods of working.
Which of these possibilities is chosen essentially depends on your usual method of working and also whether you wish to enter a new PLC program or document an existing one.
12.2 Input fields/Commands

Fields in the variable editor

E An “E” (external) is displayed in this field when a variable has been imported from a variable module (see modularization).

Variable A new variable can be entered in this field or an existing one edited in order to make a new variable entry. After you have moved out of the input line, all new variables will be sorted automatically into the correct positions.

Symbol A new symbol can be entered in this field or an existing one edited.

Long text A new long text can be entered in this field or an existing one edited.

Commands in the variable editor

Cursor control commands

- `<Cursor up>`: Scrolling one line down
- `<Cursor down>`: Scrolling one line up
- `<Enter>`: Acceptance of a modified line
- `<RTAB>`: Next field position on the right
- `<LTAB>`: Next field position on the left
- `<PgUp>`: Page up
- `<PgDn>`: Page down
- `<Home>`: Cursor to start of file
- `<End>`: Cursor to end of file
- `<Ctrl>-N`: Insert line
- `<Ctrl>-Y`: Delete line
- `<Ctrl>-B`: Delete up to end of field
- `<F1>`: Definition of not permissible input fields
- `<F2>`: Definition of the cursor starting position
- `<F9>`: Changeover of absolute/symbolic sorting

Search commands

- `<Ctrl>-Q-F-A`: Search for variable
- `<Ctrl>-Q-F-S`: Search for variables not needed
- `<Ctrl>-L`: Repeat search

Block commands

- `<F5>`: Set block marking
- `<F6>`: Reset block marking
- `<Ctrl>-K-B`: Mark start of block
- `<Ctrl>-K-K`: Mark end of block
- `<Ctrl>-K-H`: Delete marking
- `<Ctrl>-K-W`: Save block on hard disk
- `<Ctrl>-K-R`: Load block from hard disk
- `<Ctrl>-K-Y`: Delete variables in block
- `<Ctrl>-Q-A-W`: Replace variable range

When loading a block from the hard disk, it is also possible to load variable blocks produced when storing blocks in the FBD/LD or in the extended IL.

12.3 Special commands

Special commands in the variable editor

Importing and exporting variables:

- Import variables `<Ctrl>-K-I`
- Export variables `<Ctrl>-K-E`
- Import CAD/CAE variables  <Ctrl>-K-G (see also Page 12-10f)
- Export CAD/CAE variables  <Ctrl>-K-F (see also Page 12-10f)

Deleting all unused variables in a block
Mark the block. All variables that are needed neither in the FBD/LD or in the extended IL can be deleted out of a marked area by pressing <Ctrl>-K-Y.

Important:
Naturally, it is only possible to delete an absolute variable or symbol if the variable or symbol is no longer needed in the PLC program.
In all other cases, an error message will be issued.

Search for variable entries no longer contained in a PLC program
Every absolute variable or symbol once entered in the editors will remain, even if it no longer occurs in the PLC program, until the variable and symbol has been deleted in the variable editor by means of a deletion command.
The <Ctrl>-Q-F-S command can be used to find the variables and symbols no longer needed. Once these keys have been pressed, a search will be started and the cursor will move to the next absolute variable that is no longer needed.
If necessary, the complete entry can then be deleted by means of the “Delete line” command.

Replace variable ranges
Ranges of variables can be replaced by means of the keystroke combination <Ctrl>-Q-A-W.
In the header line you will be prompted to enter the start and end variables of the range you wish to replace, plus the start variable of the target range. Usually, the source variables will then be moved into the target range once you have acknowledged the message stating that the entry does not yet exist.
Within this context, the following cases may occur:
1. The target variable does not exist:
   A new entry is made (in the target range) after the appropriate message has appeared.
2. One of the target variables already exists:
   All source variables will be replaced up to the first existing target variable. A corresponding message will appear.

Defining the not valid input fields
Depending on your method of working, new entries or changes will not be made in all fields in the variable editor. To avoid having to tab past the field you do not need, an input field can be disabled or enabled again by pressing the function key <F1> followed by the designation of the field.
Above the bottom double line on the screen, an “E” for allowed tab positions or an “S” for the cursor’s starting position will indicate which fields you can move to by pressing the tab keys.

Defining the cursor’s starting position
The cursor will be positioned in the variable field every time you call the variable editor or move out of a line.
In the variable editor, the cursor’s starting position can be changed by pressing the function key <F2> followed by the designation of the field. For instance, this is expedient whenever only ever long texts ought to be entered for existing variables and symbols. In this case, you save yourself the bother of having to press the tab key several times in each line.
The cursor’s starting position is indicated by an “S” above the bottom double line on the screen and, of course, can be changed at any time.
Sorting variables
You can press the function key <F9> to toggle between the two possible criteria for sorting variables:

a) Alphabetical sorting of symbols  
b) Sorting according to absolute variables

12.4 Single line variable editor
When new variables or symbols are entered outside of the variable editor, or by pressing the <Ctrl>-Z keys, the program changes automatically to the single line variable editor.

Inputs made before changing to this editor appear in it as defaults.

Several possibilities of making further inputs are available at this point:
1. Pressing the <Enter> key:
   The variable or symbol will be entered in the variable or symbol list and the program will return to the editor concerned.
   
   Note:
   For symbolic programming, the operand identifier must be entered in the single line variable editor in addition to the symbol. The absolute values, for instance group and channel numbers, may be omitted. They must naturally be added before loading the program in the PLC.

2. In the single line variable editor, the inputs are completed by additionally entering the symbol (variable) and the long text.
   Once all inputs have been made, press <Enter> to return to the corresponding editor.

3. Pressing the <Esc> key:
   Abortion, i.e. the variable or symbol is not transferred to the variable or symbol management and you are returned to the corresponding editor.

In the single line variable editor, it is possible to scroll up and down through the variable or symbol list by means of the <Cursor up> or <Cursor down> keys or using the <PgUp> and <PgDn> keys.

Important:
Any number of variable and symbol combinations are possible, i.e. several symbols can be allocated to one variable or several variables to one symbol.

12.5 Variable export
Variable export enables you to store all variables, symbols and long texts belonging to a project in an external ASCII file in order to be able to process this data further using any chosen word processing program.

This function is called in the variable editor by pressing <Ctrl>-K-E.

The name of this file must be specified at the start. The default is:
   “project name.VEX”.

If a file already has the name you have entered, a message will be issued and you will have a possibility of aborting.

Each text for the
   Variable  
   Symbol  
   Long text
is entered one below the other in the ASCII file.

If a component does not exist, there will be a blank line in its place.
The created ASCII file can be edited and, if necessary, read in later again with the “variable import” function. If you intend to read it in later, first insert a control line into the first line of the file (e.g. @V, see Pages 12-7 to 12-9).

12.6 Variable import

The variable import function allows you to import or modify variables, symbols and long texts stored in an external ASCII file.

It is called up within the variable editor by <Ctrl>-K-I.

The first line of this external ASCII file must a control line (e.g. @V) so that the read-in mode is defined.

Important note:

On execution of the variable import function, no check is made as to whether a change to an entry is allowed in the way it has been done.

Problems may therefore occur when modifying variables already used in other editors and when replacing them by variables of a different type.

In such cases, you are therefore advised to save all previous changes in the project before starting the variable import function (quit the 907 PC 33 by selecting “Quit system”). If necessary, all changes caused in other editors by the import function can then be reversed (quit the 907 PC 33 by selecting “Abort”).

The name of the file containing the variables you wish to import must be specified at the start (default: project name.VEX).

The ASCII file must contain each text for the

<table>
<thead>
<tr>
<th>Variable</th>
<th>Symbol (max. 10 characters)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Long text</td>
<td>Long text (max. 30 characters)</td>
</tr>
</tbody>
</table>

located one below the other.

Below, the sum of the 3 lines is referred to as an “Entry”. If one of the components does not exist, there must a blank line in its place.

Control lines may be placed before or between the entries. A line is interpreted as a control line if there is a control character (“@”) on the extreme left position. All further entries will then be moved by this one line.

The first line must be a control line to ensure that read mode is defined.

A new control line will overwrite the previous one.

The following control characters are currently implemented:

- V To overwrite a variable.
- S To overwrite a symbol.
- L To overwrite a long text.

The following combinations are possible:

- @L (Edit long text)
  A search is made for the entry where the variable and symbol agree.
  The previous long text is overwritten by the new one. All entries that are not found are transferred to the variable list as new entries.

- @V Edit variable
  A search is made for the entry in which the symbol agrees.
  In this entry, the old variable is overwritten by the new one. All entries that are not found are transferred to the variable list as new entries.

- @VL Edit variable and long text
  A search is made for the entry in which the symbol agrees.
In this entry, the previous variable is overwritten by the new one and the previous long text is also overwritten by the new one.

All entries that are not found are transferred to the variable list as new entries.

**@S Edit symbol**
A search is made for the entry in which the variable agrees.

In this entry, the previous symbol is overwritten by the new one. All entries that are not found are transferred to the variable list as new entries.

**@SL Edit symbol and long text**
A search is made for the entry in which the variable agrees.

In this entry, the previous symbol is overwritten by the new one and the previous long text by the new long text.

All entries that are not found are transferred to the variable list as new entries.

**@VS Edit variable and symbol**
Here, two entries always belong together. The first of the two entries contains the previous variable and symbol and the second one contains the new ones.

A search is made for the entry in which the variable and symbols agree with the previous entry.

In this entry, the previous variable and symbol are overwritten by the new ones, i.e. a previous entry is replaced by a new one except for the long text.

The previous entry must be complete, i.e. it must consist of the variable and symbol if these components exist.

**@VSL Edit variable, symbol and long text**
Here also, two entries always belong together. The first one contains the previous variable and symbol and the second entry contains the new variable, the new symbol and the new long text.

A search is made for the entry in which the variable and symbol agree with the previous entry.

In this entry, the previous variable, symbol and long text are overwritten by the new variable, symbol and long text, i.e. a previous entry is replaced completely by a new one.

The previous entry must be complete, i.e. it must consist of the variable and symbol if these components exist.

If the following errors occur, an error message will be issued and the variable import function will be aborted:

- The specified ASCII file does not exist
- The first line of the file is not a control line
- Control lines are located in the wrong positions (not between complete entries)
- A control character line does not contain interpretable control characters.

It is not possible to import or edit an entry in the following cases. The entries which have not been imported will be written into an ASCII error file with the name “filename.VER”:

- A variable cannot be translated (invalid syntax of a variable)
- An entry found is not unique. This may occur in the “@V” or “@VL” combinations if the symbol for which you are searching or, in the case of the “@S” or “@SL” combinations, the variable for which you are searching exists several times.
- The previous entry is not found when using the combinations “@VS” or “@VSL” - A symbol’s maximum length has been exceeded
- A long text’s maximum length has been exceeded
- A text constant’s maximum length has been exceeded

This error file continues to exist until the variable import function is called again.
The following line appears at the end of the variable import function:
New: xx Modified: xx Errors: xx Adopt changes (Y/N)

The number of the new and modified entries and the number of entries containing errors are specified (these are located in the error file).

At the end you can specify whether you wish to adopt the changes made into the current symbol file or reject them.

12.7 Importing data from CAD/CAE systems

Preparing the variable lists for CAD/CAE planning systems for circuit diagrams using 907 PC 33

907 PC 33 offers the possibility of exporting variables in order to provide the variable lists (input/output levels) for a CAD/CAE system in order to create circuit diagrams.

Using 907 PC 33, you can also again read a file prepared accordingly by the CAD/CAE system and use it as a variable list (by importing variables).

A diskette can be used as the transfer medium. The respective system will read the files from this diskette.

Method:

1. Create the transfer file using 907 PC 33:
   - Select the variable editor
   - Enter <Ctrl>-K-F or select function via menu
   - Specify the path and file name in which you wish to save the file (do not specify any file name and extension that already exist or are used elsewhere by 907 PC 33)

2. Using 907 PC 33, read the file created by the CAD/CAE system:
   - Select the variable editor
   - Enter <Ctrl>-K-G or select function via menu
   - Specify the path and file name in which the file has been saved by the CAD/CAE system.

Variable codes

- LE Binary input
- LA Binary output
- AE Analog input
- AA Analog output
- WE Word input
- WA Word output

Format

The files transferred from 907 PC 33 or from the CAD/CAE system are ASCII files with the following format:

```
<Variable><CR><Symbol><CR><Long text><0>
```

Example for ABB Procontic T200:

- Variable: LE0.01,01
- Symbol: up to 10 characters
- Long text: up to 30 characters
- 0: End identifier
The following CAD/CAE systems have an ABB Procontic/Advant Controller 31 PLC module for importing or exporting data:

<table>
<thead>
<tr>
<th>Vendor</th>
<th>System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aucotec</td>
<td>ELCAD</td>
</tr>
<tr>
<td>Wiechers &amp; Partner</td>
<td>EPLAN</td>
</tr>
<tr>
<td>Aucos</td>
<td>ELEKTROCAD (in preparation at Aucos)</td>
</tr>
<tr>
<td>MG-Data</td>
<td>AUTOCAD, MG-CAD (in preparation at MG-Data)</td>
</tr>
<tr>
<td>SI-tronik</td>
<td>ELCADSY (in preparation at SI-tronik)</td>
</tr>
</tbody>
</table>

Transfer of data using an ABB Procontic/AC31 PLC module saves the need to enter variable lists twice. The CAD/CAE system combines and formats the data in this module and stores it in a file so that it can be read again by means of 907 PC 33 and used as a variable list.

If the PLC program has been created first, 907 PC 33 is capable of storing the variable list in the same format. The CAD/CAE system imports this data and writes it directly into the plan concerned.

PLC input unit in the CAD/CAE plan (example):

```
<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>E0.00,00</td>
<td>E0.00,01</td>
<td>E0.00,02</td>
<td>E0.00,03</td>
<td>E0.00,04</td>
<td>E0.00,05</td>
<td>E0.00,06</td>
<td>E0.00,07</td>
<td></td>
</tr>
</tbody>
</table>
```

Filling valve closed
Flow off
Level >
Drain valve 3
Belt running
Light barrier
Solenoid
Swivel arm

In the CAD diagram, the symbols are placed next to the respective graphic element (e.g. switch).
13  Text constant editor

If text constants (#"...) are allowed for a PLC, a few additions apply to input of variables. This applies to all editors.

Only ever the first 10 characters of a text constant are output in the editors, and also in the lists. All texts are given in the text constant list.

If, in one of the editors, you move the cursor to a variable field containing a text and press <Cursor right>, a single line window allowing you to make an input will appear in the bottom line on the screen. A text constant containing up to 250 characters can be entered here.

In the FBD/LD editor, you move the cursor to the variable field and enter "#".

When making a new input in the variable field, a branch to the bottom line can be requested by entering #".

<table>
<thead>
<tr>
<th>Sent. Word</th>
<th>Com. SF</th>
<th>OpID</th>
<th>GRN.CN</th>
<th>Symbol</th>
<th>Long text</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000</td>
<td>00000</td>
<td>!</td>
<td>A</td>
<td>04,00</td>
<td></td>
</tr>
<tr>
<td>00002</td>
<td>/</td>
<td>A</td>
<td>04,01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00004</td>
<td>=</td>
<td>A</td>
<td>00,00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00006</td>
<td>!</td>
<td>E</td>
<td>00,00</td>
<td>INPUT</td>
<td>Auxiliary switch</td>
</tr>
<tr>
<td>00008</td>
<td>=</td>
<td>A</td>
<td>00,00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00010</td>
<td>!</td>
<td>E</td>
<td>00,02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00011</td>
<td>&amp;N</td>
<td>B</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00012</td>
<td>=BS</td>
<td></td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00013</td>
<td>FB</td>
<td>0</td>
<td>TEX</td>
<td>0</td>
<td>Print module</td>
</tr>
</tbody>
</table>

#"Here is the window for the text

Any required control characters are entered as \xxx, e.g. <CR> with \013. Text constants can be treated fully symbolically. Here, #" serves as the operand identifier.
14 Text editor

The text editor enables you to edit any chosen ASCII files with a width of up to 255 characters per line. As only a part of the text is stored in the main memory and is accordingly loaded automatically from the disk when needed, the total size of the text file is limited only by your computer’s operating speed and disk capacity.

You can scroll both horizontally and vertically in the text window.

Besides the mode (overwrite/insert) and the file name, the line number, column number and current number of characters up to the cursor’s position are also displayed in the top line.

The following commands can be executed with in the text editor:

Cursor control:

- `<Cursor left>` The cursor moves one character to the left / scrolling on the left margin
- `<Cursor right>` The cursor moves one character to the right / scrolling on the right margin
- `<Cursor up>` The cursor moves one line up / scrolling at the top margin
- `<Cursor down>` The cursor moves one line down / scrolling at the bottom margin
- `<LTAB>` Next tab on the left / scrolling on the left margin
- `<RTAB>` Next tab on the right / scrolling on the right margin
- `<Ctrl><Cursor right>` The cursor moves to the start of the next word on the right
- `<PgUp>` The cursor moves up by one page
- `<PgDn>` The cursor moves down by one page
- `<Home>` Cursor to the start of the line, then start of the page and then start of the file
- `<End>` Cursor to the end of the line, then end of the page and then end of the file

Block processing:

- `<Ctrl>-K-B` Mark start of block (the block is displayed inverse)
- `<Ctrl>-K-K` Mark end of the block. The block’s contents are transferred to an internal buffer. The block marking is deleted.
- `<Ctrl>-K-H` Delete block marking.
- `<Ctrl>-K-C` Copy block from internal buffer to text.
- `<Ctrl>-K-Y` The block’s contents are transferred to the internal buffer and the block is deleted from the text.
- `<Ctrl>-K-W` Write block to file.
- `<Ctrl>-K-R` Insert block from file at the cursor’s current position.

As long as a block is marked, all editing functions are disabled and the current start of the block is displayed in the top line.

Editing functions:

- `<Enter>` In insert mode, everything from the cursor’s current position onward is moved to the next line and the cursor is placed at the start of the next line. In overwrite mode, the cursor is only placed at the start of the next line.
- `<Del>` Delete character at the cursor’s position
- `<Ctrl>-N` Insert line
- `<Ctrl>-Y` Delete line
- `<Ctrl>-Q-Y` Delete up to the end of the line from the cursor’s current position
- `<Ins>` Switch between insert and overwrite mode.

Search functions:

- `<F5>` Search for text (enter the search text in the window)
- `<F6>` Search for line number
- `<Ctrl>-L` Repeat search (when searching for text)
Terminating or aborting editing:

<Esc> Terminate or abort editing
Terminate commenced command or the text editor, with the possibility of rejecting changes, saving changes or returning to the editor (<Esc>).
15 Library

15.1 Introduction

Connection elements (CE) are the devices of the function block diagram that is programmed with the FBD/LD editor. They consist of a graphic and a program code. They can also be used in the extended IL. There they are used as macros listing program parts that are often repeated.

The creation of the graphic, the definition of the program code and recording of the CE is done with the library.

There are two groups of CEs: finished CEs that are delivered together with the 907 PC 33 programming system are manufacturer-CEs and the user-CEs are defined by the PLC programmer himself.

Existing CEs can be implemented into the code of a new CE, so nested CEs are possible.

Basically a user-CE can have the same description as a manufacturer-CE. In this case the user-CE is used during compilation, so that the delivered CE is hidden.

15.2 Display

To create a CE the menu function “Library” must be selected in the 907 PC 33 main menu and CE library must be selected in the submenu.

The following figure is displayed:

<table>
<thead>
<tr>
<th>B</th>
<th>CE name</th>
<th>CE comment</th>
<th>Grf</th>
<th>Src</th>
<th>IL</th>
<th>Doc</th>
<th>M/U</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;</td>
<td>AND</td>
<td>* eIL * * * H 26.03.92</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>*</td>
<td>Multiplication</td>
<td>* eIL * * * H 26.03.92</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>*:</td>
<td>Multiplication with division</td>
<td>* eIL * * * H 26.03.92</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>*D</td>
<td>Multiplication, double word</td>
<td>* eIL * * * H 26.03.92</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>+</td>
<td>Addition</td>
<td>* eIL * * * H 26.03.92</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>+D</td>
<td>Addition, double word</td>
<td>* eIL * * * H 26.03.92</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>Subtraction</td>
<td>* eIL * * * H 26.03.92</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-D</td>
<td>Subtraction, double word</td>
<td>* eIL * * * H 26.03.92</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/</td>
<td>OR</td>
<td>* eIL * * * H 26.03.92</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>:=$=</td>
<td>Division</td>
<td>* eIL * * * H 26.03.92</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>:d</td>
<td>Input in FBD not possible</td>
<td>eIL * H 26.03.92</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>:=DW</td>
<td>Input in FBD not possible</td>
<td>eIL * H 04.11.93</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>:=WO</td>
<td>Input in FBD not possible</td>
<td>eIL * H 26.03.92</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>:D</td>
<td>Division, double word</td>
<td>* eIL * * * H 26.03.92</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;</td>
<td>Less than</td>
<td>* eIL * * * H 26.03.92</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;=</td>
<td>Less than or equal to</td>
<td>* eIL * * * H 26.03.92</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;&gt;</td>
<td>Unequal</td>
<td>* eIL * * * H 26.03.92</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;D</td>
<td>Less than, double word</td>
<td>* eIL * * * H 26.03.92</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Headlines

There are two headlines in the library:

CE lib.: SER 90  CEs: 149  Sort.: Alpha  Editor: Ext. IL

Group: ALL

Group: The CE group containing the CEs is displayed.

CEs: The number of listed CEs is displayed in this field.

Sort.: The order that the CEs are listed in is displayed. It is possible to order alphabetical or the order the CEs were entered.
Editor: The set editor for the creation of the CEs of present basic CEs is displayed in this field. The extended IL or the FBD/LD editor can be used.

**Fields:**

The cursor is moved to any field with the cursor keys.

The single fields have got following meanings:

- **B:** An asterisk (“*”) displays that the CE in this line is marked as a block.
- **M:** An asterisk (“*”) displays that the IL data belonging to this CE is held in the memory while compiling the program entered in the FBD/LD or extended IL editor (see chapter 15.8. CE cache configuration).

**CE name:**

This field contains the CE’s designation.

**CE comment:**

A short description (max. 30 characters) of the CE is entered in this field.

**Grf:**

The CE graphic editor that the graphic of a CE is created with is started by pressing <Enter>. The CE is displayed in the FBD/LD editor the way it had been created with this editor. After the graphic has been created this field is marked with an “*”.

**Src:**

The preset editor (FBD/LD or extended IL, toggle with <F2>) is started so that the program code is edited. This field is marked with “FL” or “eIL” depending on the used editor after editing is finished.

**IL:**

The compiled instruction list is displayed.

**Doc:**

The text editor is started so that a detailed documentation of the CE is edited.

**M/U:**

It is displayed if the CE is listed in the manufacturer (H) or user library (A) (field can’t be edited).

**Date:**

The date of the last modification of the CE is displayed (Field can’t be edited).

A different layout displaying the time of the last modification on the right side next to the date is shown by pressing <Ctrl>-<F1> (Field can’t be edited). The “CE comment” field is shortened. The former layout is displayed by pressing <Ctrl>-<F1> again.

**Keys**

- **<Ins>,<Ctrl>-N:** A new CE is inserted
- **<Del>,<Ctrl>-Y:** The current CE is deleted.
- **<F1>:** Starts the preset editor for editing a CE.
- **<F2>:** Selects the editor (FBD/LD or extended IL).
- **<F3>:** Starts the CE editor.
- **<F4>:** Starts the text editor.
- **<F5>:** The field “B” or “M” is marked with a “*”, that means the CE in this line is marked as a block.
- **<F6>:** The mark in field “B” or “M” is deleted.
- **<F7>:** The fields “CE name” and “CE comment” (depending on cursor position) are edited.
- **<F8>:** The plausibility group editor is started.

With this editor the operands (Bit input, word flag,...) that are to be connected to a connection of a CE are defined.
<F9>: The display order of the CEs is selected. It is possible to display in alphabetical order or in the order the CEs were entered.

<Shift>-<F1>: All CEs are displayed.

<Shift>-<F2>: The present CE groups are listed. The contents of the groups are displayed by moving the cursor onto the appropriate group and pressing <Enter>.

<Shift>-<F3>: Marked CEs are assigned to a different group.

<Shift>-<F4>: CEs can be assigned to specific groups. An editor is started so that a new group is edited by pressing <Shift>-<F4>.

<Ctrl>-<F1>: The layout is toggled. As described above also the time of the last modification of the CE is displayed next to the date.

<Ctrl>-<F5>: Set scroll speed or mouse intensity.

<Alt>-<F1>: CE cache configuration

<Alt>-<F2>: Duplicate CE

If all CEs are displayed, the menu for selecting the CE group into which the new duplicated CE is to be included appears.

After selecting a CE group and pressing the <Enter key>, a blank line is inserted before the duplicated CE. A new CE name must be entered.

Exception: The same name can be given when copying manufacturer CEs. Duplication then occurs.

If you already find yourself in a group during duplication (screen display is not ALL), it is not necessary to select a CE group into which the CE is to be included. The new CE is automatically allocated to the group currently displayed.

<Alt>-<F3>: In the case of the CE marked in column B, the time stamp is updated with the current time/date. Only user CEs can be updated.

15.3 Block functions

Several functions for transferring blocks to different libraries are available. It is possible to start a library, mark a few CEs in the field “B” and then assign these to a different library.

Mark block:

Single devices are marked by pressing <F5>, several devices are marked as a block by pressing <Ctrl>-K-B (beginning) / <Ctrl>-K-K (end). The block marks are deleted individually by pressing <F6> or with the “delete block marks” function <Ctrl>-K-H.

Write block (<Ctrl>-K-W):

The CEs marked as a block are written onto the harddisk/diskette as a new library after entering a filename. An already present library is overwritten after a safety request was displayed.

Read block (<Ctrl>-K-R):

After entering a filename the CE library is appended to the current displayed one. If the current displayed library contains CEs with the same designation as the ones in the read file, they are replaced by the ones in the read file.

Delete block (<Ctrl>-K-Y):

The marked CEs are deleted after a safety request was displayed. Manufacturer-CEs can’t be deleted by the user.
15.4 The CE group editor

The CEs can be assigned to specific groups in the library. For example it is sensible to assign logic CEs (AND, OR, NOT, ...) to a group ‘LOGIC’ and to assign arithmetic CEs (ADD, SUB, ...) to a group ‘ARITHMETIC’. These groups are only for order reasons. It is not possible to have two CEs with the same designation in different groups as CEs are identified by a unique designation. Only manufacturer-CEs can be overwritten by user-CEs with the same designation (Both CEs are present in the library, but the FBD/LD and extended IL editors always use the user CE).

The structure of groups is identical to the structure of directories (MSDOS) so that subgroups can be defined in groups again.

Display groups:
The CE groups are listed on the right side of the screen as described in chapter “Keys” by pressing <Shift><F2>.

Example of a structure of a group:

<table>
<thead>
<tr>
<th>B</th>
<th>M</th>
<th>CE name</th>
<th>CE comment</th>
<th>Grf</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>&amp;</td>
<td>AND</td>
<td></td>
</tr>
<tr>
<td>*</td>
<td></td>
<td>/</td>
<td>OR</td>
<td></td>
</tr>
<tr>
<td>*</td>
<td>S</td>
<td>SET</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R</td>
<td></td>
<td></td>
<td>RESET</td>
<td></td>
</tr>
</tbody>
</table>

The content of the group the cursor is placed on is displayed by pressing <Enter>.

Defining a new group:
The present groups are displayed by pressing <Shift>-<F4>. The cursor is placed at the position the new group is to be inserted behind of.

A line, in that the new group definition is entered, is inserted by pressing <Ctrl>-N or <Ins> (or select “Insert a line” per menu). The group is defined after pressing <Esc> and confirmation.

Insert a line with <Ctrl>-N.
Delete a line with <Ctrl>-Y.
Rename a group by pressing <F7>.

Assign CEs to a different group:
First of all the appropriate CEs must be marked. The present groups are displayed by pressing <Shift>-<F3>. The marked CEs are assigned to a group by selecting one and pressing <Enter>. The moved CEs and the CEs already listed in the group are now displayed.
15.5 How to proceed when creating CEs

Library -> Call up CE library

Press <Ctrl>-N

Select CE group to which the CE is to be allocated

Press <Enter key>

Enter name of CE

Press <Enter key>

CE is inserted in alphabetical order

Set cursor to the CE comment of the new CE

A
Note on creating the CEs:
Program compilation of the CEs in FBD or ext. IL is limited to one segment plan. If the program is longer, the new CE must be divided up into several CEs.
15.6 Creating graphics of CEs

The graphic of a CE must be defined so that the CE can be used in the FBD/LD and extended IL. The graphic is defined with the graphic editor.

Move the cursor onto the “Grf” field and call the editor by pressing <Enter>.

The NAND (NOT-AND) gate is to be defined as an example. It contains an & and a =N block. The & and =N blocks are presumed listed in the library.

First of all the CE designation and comment are entered.

Entering the CE designation:

Move the cursor onto the “CE name” field and press <Enter> or <F7>. Enter the designation of the CE (in this case “NAND”).

Entering the comment of the CE:

Move the cursor to the “CE comment” field and press <Enter> or <F7>. Enter the short designation of the CE (for example “logical NOT-AND”).

Documentation:

There is a possibility of writing a detailed documentation of the CE. This can contain for example further description of the CE, name of the programmer, last modification etc.

Move the cursor onto the “Doc” field and press <Enter>.

The 907 PC 33 text editor is started and information can be entered.

Creation of the graphics

Move the cursor to the “Grf” field and press <Enter>. The following figure is displayed:

<table>
<thead>
<tr>
<th>Plaus. group</th>
<th>Dty</th>
<th>Inv</th>
<th>Dpl</th>
<th>Mod</th>
<th>No</th>
<th>No Mod</th>
<th>Dpl</th>
<th>Inv</th>
<th>Dty</th>
<th>Plaus. group</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The cursor is placed on the left side (input side) under the name of the CE. The formal parameter designation of the first input is requested. Enter “E1” for example. The line “ALL Y N Y 0 00” is displayed left next to the input after pressing <Enter> (see below).

The CE is enhanced further downwards by pressing <CurDn>. So space for the second input is made and “E2” is entered as second formal parameter designation. Now move the cursor to the output side of the CE by pressing <Crsr right> and enter “A1” as output formal parameter designation.
The plausibility group is selected. It defines which operands are to be assigned to the CE connections. A group that was defined in the plausibility group editor is selected by moving the cursor onto “Plaus. group” “ALL” in this case and pressing <Enter>. A window with present groups is displayed and a group is selected by moving the cursor onto the wanted group with the cursor keys and pressing <Enter>. The “BIT” group was selected in the example for all connections. The definitions of the already available plausibility groups can be printed with the menu functions “Print 2” and “CE library” or can be displayed in the plausibility group editor. For an exact description of the plausibility group editor see chapter 15.6.

“Dty” - default “Y”
Connection required: at this point it is possible to state if connection is required or not, meaning if the CE-connection has to be occupied or not necessarily (“Y”) or (“N”).

“Inv” - default “N”
Inversion: the possibility of inverting this connection is permitted or locked by pressing <Y> or <N>.

“Dpl” - default “Y”
Display: the possibility of displaying the formal parameter designations in the FBD/LD or extended IL editors is permitted or locked by pressing “Y” or “N”. Normally the formal parameter designations aren’t displayed with standard CEs like AND, but it is useful to display them in more complicated CEs to state the meaning of the CE connections.

Duplication of CE connections:
Of course there are applications needing an AND gate with more than two inputs. There is a possibility of enlarging the number of inputs of a CE by duplicating an input. For that the duplication type and duplication block number must be entered at one CE connection.

“Dpl” - default “0”
The duplication type is defined for the current CE connection.
There are two duplication types, 0 and 1:
If several CE connections with the same duplication block number are to be duplicated, every single duplicated CE connection is placed underneath of the original CE connections with the same connection using duplication type 0. The CE connections are duplicated as a block using duplication type 1.

Example:

<table>
<thead>
<tr>
<th>before duplication</th>
<th>after duplication</th>
<th>before duplication</th>
<th>after duplication</th>
</tr>
</thead>
<tbody>
<tr>
<td>X0</td>
<td>X0</td>
<td>X0</td>
<td>X0</td>
</tr>
<tr>
<td>Y0</td>
<td>X1</td>
<td>Y0</td>
<td>Y0</td>
</tr>
<tr>
<td>Z0 =&gt;</td>
<td>Y0</td>
<td>Z0 =&gt;</td>
<td>Z0</td>
</tr>
<tr>
<td></td>
<td>Y1</td>
<td></td>
<td>X1</td>
</tr>
<tr>
<td></td>
<td>Z0</td>
<td></td>
<td>Y1</td>
</tr>
<tr>
<td></td>
<td>Z1</td>
<td></td>
<td>Z1</td>
</tr>
</tbody>
</table>

“No” - default “00”

The duplication block number is entered at this point. It defines which CE connections are to be duplicated as a block as described above. The duplication block number is from 0 to 15. Block 0 isn't duplicated.

The graphic editor is exited after a safety request asking if changes are to be accepted by pressing <ESC>. The main display of the library function is displayed. The “Grf” field is marked with a “*”.

15.7 The plausibility group editor

A plausibility test is done when editing a project with the FBD/LD or extended IL editor. It is checked if the entered operand (input, flag,...) is listed in the plausibility group of the CE. In this way erroneous inputs can be avoided in the FBD/LD editor and Extended editor in most cases.

So the plausibility group of a CE connection contains a list of valid operands. There is a possibility of defining plausibility groups. A group can also consist of subgroups. In this way bracketing is realized.

The definitions of already present plausibility groups can be printed out with the menu function “Print 2” and “Print CE library”. 907 PC 33 can manage a maximum of 128 plausibility groups.

Linkages

The operands are explicitly assigned to a group with link symbols.

There are four linking systems: common content “&”, complete content “/” and their negations “&N”, “/N”.

Example:

A group is supposed to contain bit inputs and bit flags, that means only bit inputs and bit flags are to be connected to the CE connection.

The linkage is the following:

/ E <- Operand bit input
/ M <- Operand bit flag

Now the group is only to contain bit flags. It is presumed that the groups “BIT” and “FLAGS” are available:

/ BIT <- Group BIT
& FLAGS <- Group FLAGS

Now the group is supposed to contain the subgroup “OUTPUT”. This subgroup contains a group “FLAGS”, that aren’t to be connected:

/ OUTPUT <- Group OUTPUT
&N FLAGS <- Group FLAGS

In this way specific groups are created containing explicitly required operands.
The following figure appears after starting the editor by pressing <F8>:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>The group marked with an asterisk &quot;*&quot; in this line is marked as a block.</td>
</tr>
<tr>
<td></td>
<td>The user can’t access groups marked with an &quot;X&quot;.</td>
</tr>
<tr>
<td>Plaus. group</td>
<td>The present plausibility groups are listed in this field.</td>
</tr>
<tr>
<td>Operands</td>
<td>All operands are listed.</td>
</tr>
<tr>
<td>Link.</td>
<td>The link-symbols are listed in this field.</td>
</tr>
<tr>
<td>Group definition</td>
<td>The operands and subgroups of a group are listed in this field. The link</td>
</tr>
<tr>
<td></td>
<td>symbols are displayed in the left half of the field and the actual groups</td>
</tr>
<tr>
<td></td>
<td>and operands are listed in the right half.</td>
</tr>
</tbody>
</table>

**Keys**

**In group selection mode:**

- `<Ins>`: A line is inserted at the current cursor position.
- `<Ctrl>-N`: A line is inserted at the current cursor position.
- `<Ctrl>-<F1>`: All available operands or all operands according to the linkage are listed in the field „Operands“.
- `<F5>`: The group in the current line is marked as a block.
- `<Ctrl>-K-B`: The marked groups are deleted.

**In edit group mode:**

- `<Ctrl>-N`: A line is inserted at the current cursor position.
- `<Ctrl>-Y`: An operand or a group at the cursor position is deleted.
The cursor is placed on the plausibility group field. It is possible to edit a group by moving the cursor onto the wanted group and pressing <Enter>.

The following figure appears:

<table>
<thead>
<tr>
<th>B</th>
<th>M</th>
<th>CE name</th>
<th>CE comment</th>
<th>Grf</th>
<th>Src</th>
<th>IL</th>
<th>Doc</th>
<th>M/U</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>B</td>
<td>Plaus. group</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X</td>
<td></td>
<td>ALL</td>
<td>E</td>
<td>/</td>
<td></td>
<td>E</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X</td>
<td></td>
<td>FLAGS</td>
<td>EW</td>
<td>&amp;</td>
<td></td>
<td>A</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X</td>
<td></td>
<td>KOP_IN</td>
<td>PP</td>
<td>/N</td>
<td></td>
<td>M</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X</td>
<td></td>
<td>KOP_OUT</td>
<td>A</td>
<td>&amp;N</td>
<td></td>
<td>S</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X</td>
<td></td>
<td>DIRECT-CONS</td>
<td>AW</td>
<td></td>
<td>K</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X</td>
<td></td>
<td>BIT-OUT</td>
<td>M</td>
<td></td>
<td></td>
<td>PP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X</td>
<td></td>
<td>BIT-IN</td>
<td>MW</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X</td>
<td></td>
<td>BIT-OUT-FB</td>
<td>K</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X</td>
<td></td>
<td>WORD-OUT</td>
<td>KW</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X</td>
<td></td>
<td>WORD-IN</td>
<td>KD</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X</td>
<td></td>
<td>WORDCONST #</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X</td>
<td></td>
<td>DWORD-OUT</td>
<td>#H</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The operands belonging to the group „BIT-IN“ are listed in the „Group definition“ field. The operands according to the linkage are listed in the „Operands“ field by pressing <Ctrl><F1>. The top right field will display „display group“.

<table>
<thead>
<tr>
<th>B</th>
<th>M</th>
<th>CE name</th>
<th>CE comment</th>
<th>Grf</th>
<th>Src</th>
<th>IL</th>
<th>Doc</th>
<th>M/U</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>B</td>
<td>Plaus. group</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X</td>
<td></td>
<td>ALL</td>
<td>E</td>
<td>/</td>
<td></td>
<td>E</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X</td>
<td></td>
<td>FLAGS</td>
<td>PP</td>
<td>/N</td>
<td></td>
<td>M</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X</td>
<td></td>
<td>KOP_IN</td>
<td>A</td>
<td>&amp;N</td>
<td></td>
<td>S</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X</td>
<td></td>
<td>DIRECT-CONS</td>
<td>AW</td>
<td></td>
<td>K</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X</td>
<td></td>
<td>BIT-OUT</td>
<td>M</td>
<td></td>
<td></td>
<td>PP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X</td>
<td></td>
<td>BIT-IN</td>
<td>MW</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X</td>
<td></td>
<td>BIT-OUT-FB</td>
<td>K</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X</td>
<td></td>
<td>WORD-OUT</td>
<td>KW</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X</td>
<td></td>
<td>WORD-IN</td>
<td>KD</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X</td>
<td></td>
<td>WORDCONST #</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X</td>
<td></td>
<td>DWORD-OUT</td>
<td>#H</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

All available operands are displayed in the „operands“ field by pressing <Ctrl><F1> once again. The „select group“ mode is switched to by pressing <Esc>.
Editing a group:

A group is inserted by pressing <Ctrl>-<N>. The group name must be entered and confirmed with <Enter>. To specify or modify the content of the group move the cursor onto the group name and press <Enter>. Now groups and operands with their link symbols are selected by cursor movement and confirmation with <Enter>. These are then listed in the „group definition“ field.

It is also possible to directly enter operands in this field by moving the cursor onto the „group definition“ field. It is possible to check the result of the linkage of the selected groups at any time by pressing <Ctrl>-<F1>. Now the operands according to the definition of the group are listed in the „operands“ field.

Recursions (a group contains itself as subgroup) are invalid.

Deleting groups:

Groups are deleted by marking them (see block functions) as a block (press <F5> or <Ctrl>-K-B/<Ctrl>-K-H) and then pressing <Ctrl>-K-Y. The groups marked with an „X“ can’t be deleted. If a group is marked that is listed in a different group as a subgroup it also isn’t deleted.

A maximum of 128 groups can be entered that themselves contain a maximum of 20 devices. If a specific group has to contain more than 20 devices subgroups must be defined and listed in this group.

The new entered or modified groups are accepted by pressing <ESC> and accepting the changes.

Obligatoric plausibility groups

For the internal run of the Compiler and control functions 907 PC 33 requires the groups

- MERKER (FLAGS)
  (all operand types used as automatic temporary flags)
- KOP IN  (LD_IN)
  (all operand types that can be connected to a ladder diagram input)
- KOP OUT (LD_OUT)
  (all operand types that can be connected to a ladder diagram output)

and a list of all operands to which this operand type can be assigned to:

<table>
<thead>
<tr>
<th>can be assigned to</th>
</tr>
</thead>
<tbody>
<tr>
<td>0P 1</td>
</tr>
<tr>
<td>0P 2</td>
</tr>
<tr>
<td>0P 4</td>
</tr>
<tr>
<td>0P 5</td>
</tr>
<tr>
<td>etc.</td>
</tr>
</tbody>
</table>

These data and the group ‘ALLE’ (ALL) are automatically created when creating a new library.

Should one of these groups be edited for the first time the field group definition is empty. If data is entered in the field ‘group definition’ all presets are deleted (after accepting the changes) and the group content is set to the new edited group definition.

The same goes for editing (only possible in manufacturer library) the table of operand assignments, with the difference that the group definition is kept after saving the edited groups and recalling, while this field is empty for the operand types.
15.8 CE Cache configuration

There is a possibility of defining a memory range in which the CEs marked in the „M“ column are held during compilation of the program entered in the FBD/LD or extended IL editor.

A window is displayed in the library so that the CE cache is switched on and the size is entered by pressing <Alt><F1>:

<table>
<thead>
<tr>
<th>B</th>
<th>M</th>
<th>CE name</th>
<th>CE comment</th>
<th>Grf</th>
<th>Src</th>
<th>IL</th>
<th>Doc</th>
<th>M/U</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;</td>
<td></td>
<td>AND</td>
<td></td>
<td>eIL</td>
<td>*</td>
<td>*</td>
<td>H</td>
<td>26.03.92</td>
<td></td>
</tr>
<tr>
<td>*</td>
<td></td>
<td>Multiplication</td>
<td></td>
<td>eIL</td>
<td>*</td>
<td>*</td>
<td>H</td>
<td>26.03.92</td>
<td></td>
</tr>
<tr>
<td>:</td>
<td></td>
<td>Multiplication with division</td>
<td></td>
<td>eIL</td>
<td>*</td>
<td>*</td>
<td>H</td>
<td>26.03.92</td>
<td></td>
</tr>
<tr>
<td>*D</td>
<td></td>
<td>Multiplication, double word</td>
<td></td>
<td>eIL</td>
<td>*</td>
<td>*</td>
<td>H</td>
<td>26.03.92</td>
<td></td>
</tr>
<tr>
<td>+</td>
<td></td>
<td>Addition</td>
<td></td>
<td>eIL</td>
<td>*</td>
<td>*</td>
<td>H</td>
<td>26.03.92</td>
<td></td>
</tr>
<tr>
<td>+D</td>
<td></td>
<td>Addition</td>
<td></td>
<td>eIL</td>
<td>*</td>
<td>*</td>
<td>H</td>
<td>26.03.92</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td></td>
<td>Subtraction</td>
<td></td>
<td>eIL</td>
<td>*</td>
<td>*</td>
<td>H</td>
<td>26.03.92</td>
<td></td>
</tr>
<tr>
<td>-D</td>
<td></td>
<td>Subtraction</td>
<td></td>
<td>eIL</td>
<td>*</td>
<td>*</td>
<td>H</td>
<td>26.03.92</td>
<td></td>
</tr>
<tr>
<td>/</td>
<td></td>
<td>Division</td>
<td>CE IL in memory</td>
<td>eIL</td>
<td>*</td>
<td>*</td>
<td>H</td>
<td>26.03.92</td>
<td></td>
</tr>
<tr>
<td>:</td>
<td></td>
<td>Division</td>
<td>Max. memory size</td>
<td>2048</td>
<td>eIL</td>
<td>*</td>
<td>*</td>
<td>H</td>
<td>26.03.92</td>
</tr>
<tr>
<td>:=DW</td>
<td></td>
<td>Input in FBD not possible</td>
<td></td>
<td>eIL</td>
<td>*</td>
<td>H</td>
<td>04.11.93</td>
<td></td>
<td></td>
</tr>
<tr>
<td>:=WO</td>
<td></td>
<td>Input in FBD not possible</td>
<td></td>
<td>eIL</td>
<td>*</td>
<td>H</td>
<td>04.11.93</td>
<td></td>
<td></td>
</tr>
<tr>
<td>:D</td>
<td></td>
<td>Division, double word</td>
<td></td>
<td>eIL</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>26.03.92</td>
<td></td>
</tr>
<tr>
<td>&lt;=</td>
<td></td>
<td>Less than</td>
<td></td>
<td>eIL</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>26.03.92</td>
<td></td>
</tr>
<tr>
<td>&lt;&gt;</td>
<td></td>
<td>Unequal</td>
<td></td>
<td>eIL</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>26.03.92</td>
<td></td>
</tr>
<tr>
<td>&lt;D</td>
<td></td>
<td>Less than, double word</td>
<td></td>
<td>eIL</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>26.03.92</td>
<td></td>
</tr>
</tbody>
</table>

A CE cache defined in that way will shorten the compilation time of the extended IL or FBD/LD program - especially an Personal Computers without cache or with a small cache installed, e.g. 256 kB cache. If it is possible, for example with the help of the SMARTDRIVE program (contained in DOS 5.0) to set up a cache of 1 to 2 MB, the CE cache configuration is not necessary.

The size of the memory that is to be used for this function can be entered in the bottom line. It is limited to 8 kB (8192 Bytes). The CE cache is created in the conventional DOS memory of 640 kB. If more CEs are marked than fit in the CE cache, as many CEs as possible are included, with maximum utilization of the cache.

15.9 Creating the program run in a CE

The actual program of the CE is written with the FBD/LD or extended IL editor. The editor is selected by pressing <F2>.

Programming by means of the extended IL editor is done by entering instructions, operands and CEs. Additionally, CEs can be called.

The following pages explain the creation of the program with the extended IL. Existing control commands are also explained and a CE is listed for every command as an example. After that an example using the FBD/LD-editor is shown.

Creating the program run with the IL

The compilation mode can be Set by enter-ing specific 907 PC 33 control commands in the CE instruction list. Further setting is done with the Parameter line of the CE editor (see creating the graphic).
The single control commands are shown and explained in examples. A larger example is shown at the end of the chapter.

The following control commands can be used in a CE:
- Replace Operand with current CE Parameter (PP 0..15)
- Repeat IL instructions for duplicated Parameters ([1..15, ] 1..15)
- Request global flag (@GLO)
- Request local flag (@LOC)
- Set beginning of duplication (@START)
- Set end of duplication (@END)
- Recursive access to parameters in duplication (@OFF)

1. CE without parameters

The IL entered in the CE instruction list editor is taken over into the IL without changes.

Element CE1:

<table>
<thead>
<tr>
<th>Plaus. group</th>
<th>Dty</th>
<th>Inv</th>
<th>Dpl</th>
<th>Mod</th>
<th>No</th>
<th>No Mod</th>
<th>Dpl</th>
<th>Inv</th>
<th>Dty</th>
<th>Plaus. group</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

==Creating the CE with the extended IL editor:==

```
SN  LN  Instr./ Variable  Symbol  Long text
0000 000000; The CE IL is taken over 1:1
000001 !  M  0000,00
000000 =  M  0000,01
```

==Editing the CE with the extended IL editor:==

```
SN  LN  WN  Instr./ Variable  Symbol  Long text
00000 000000; CE without parameters
000001 000000 $ CE1
```

==compiled IL:==

```
SN  LN  Instr./ Variable  Symbol  Long text
00000 ;CE without parameters
        ;The CE IL is taken over 1:1
000000 !  M  0000,00
000001 =  M  0000,01
```

15-14
2. Parameters not capable of duplication

The position of where the variable is to be entered for a CE connection must be entered in the CE IL.
This is done with „PP 0“. (The PP with values larger than 0 are used for parameters, which are capable of duplication.

For CE parameters not capable of duplication „0“ must always (!) be entered. This „PP 0“ replaces the absolute designation when entering commands.

For a correct assignment of the parameters of the CE to be compiled the name of eth CE connection must also be entered in the Symbol of „PP 0“.

Now a 1:1-assignment of CE graphic and CE IL is possible.

Element: CE2

<table>
<thead>
<tr>
<th>Plaus. group Dty Inv Dpl Mod No</th>
<th>No Mod Dpl Inv Dty Plaus. group</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIT_IN N Y Y 0 00</td>
<td>E1</td>
</tr>
<tr>
<td>BIT_IN N Y Y 0 00</td>
<td>E2</td>
</tr>
</tbody>
</table>

Creating the CE with the extended IL editor:

<table>
<thead>
<tr>
<th>SN</th>
<th>LN</th>
<th>Instr. / Variable</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000</td>
<td>00001</td>
<td>PP 00</td>
<td>E1</td>
</tr>
<tr>
<td>00000</td>
<td>00002</td>
<td>PP 00</td>
<td>E2</td>
</tr>
<tr>
<td>00000</td>
<td>00003</td>
<td>PP 00</td>
<td>A</td>
</tr>
</tbody>
</table>

Editing the CE with the extended IL editor:

<table>
<thead>
<tr>
<th>SN</th>
<th>LN</th>
<th>WN</th>
<th>Instr. / Variable</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000</td>
<td>00001</td>
<td>00000 $ CE2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00000</td>
<td>00002</td>
<td>00000—E1 M 0000,00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00000</td>
<td>00003</td>
<td>00001—E2 M 0000,01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00000</td>
<td>00004</td>
<td>00002—A M 0000,02</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

compiled IL:

<table>
<thead>
<tr>
<th>SN</th>
<th>LN</th>
<th>Instr. / Variable</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000</td>
<td>00001</td>
<td>M 0000,00</td>
<td></td>
</tr>
<tr>
<td>00000</td>
<td>00002</td>
<td>M 0000,01</td>
<td></td>
</tr>
<tr>
<td>00000</td>
<td>00003</td>
<td>M 0000,02</td>
<td></td>
</tr>
</tbody>
</table>
3. Parameters capable of duplication

In the CE-IL it is to be stated which duplication block belongs to a parameter. This is done by entering „PP n“, with n as duplication block number. The symbol contains the name of the CE connection.

Also the IL code that is to be duplicated must be entered. That is done with the special commands „[n“ and „]n“. n is also the duplication block number. The range inside of the brackets is repeated as often as the duplication block was duplicated. A different parameter is placed at „PP n“ during each repetition.

Element: CE3

<table>
<thead>
<tr>
<th>Plaus. group</th>
<th>Dty</th>
<th>Inv</th>
<th>Dpl</th>
<th>Mod No</th>
<th>No Mod</th>
<th>Dpl</th>
<th>Inv</th>
<th>Dty</th>
<th>Plaus. group</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| BIT_IN | N | Y | Y | 0 | 00 | E1 |
| BIT_IN | N | Y | Y | 0 | 00 | A  |

Creating the CE with the extended IL editor:

<table>
<thead>
<tr>
<th>SN</th>
<th>LN</th>
<th>Instr. / Variable</th>
<th>Symbol</th>
<th>Long text</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000</td>
<td></td>
<td>LD</td>
<td>PP 00</td>
<td>E1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>;enter different parameter here</td>
</tr>
<tr>
<td>[</td>
<td></td>
<td></td>
<td>01</td>
<td>]</td>
</tr>
<tr>
<td>00001</td>
<td></td>
<td>&amp;</td>
<td>PP 01</td>
<td>E2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00002</td>
<td></td>
<td>=</td>
<td>PP 00</td>
<td>A</td>
</tr>
</tbody>
</table>

Editing the CE with the extended IL editor:

<table>
<thead>
<tr>
<th>SN</th>
<th>LN</th>
<th>Instr. / Variable</th>
<th>Symbol</th>
<th>Long text</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000</td>
<td></td>
<td></td>
<td></td>
<td>;Parameters capable of duplication</td>
</tr>
<tr>
<td>00000</td>
<td></td>
<td></td>
<td>$</td>
<td>CE3</td>
</tr>
<tr>
<td>00002</td>
<td></td>
<td></td>
<td>E1</td>
<td>M 0000,00</td>
</tr>
<tr>
<td>00003</td>
<td></td>
<td></td>
<td>E200</td>
<td>M 0000,01</td>
</tr>
<tr>
<td>00004</td>
<td></td>
<td></td>
<td>E201</td>
<td>M 0000,02</td>
</tr>
<tr>
<td>00005</td>
<td></td>
<td></td>
<td>E202</td>
<td>M 0000,03</td>
</tr>
<tr>
<td>00006</td>
<td></td>
<td></td>
<td>A</td>
<td>M 0000,04</td>
</tr>
</tbody>
</table>

compiled IL:

<table>
<thead>
<tr>
<th>SN</th>
<th>LN</th>
<th>Instr. / Variable</th>
<th>Symbol</th>
<th>Long text</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000</td>
<td></td>
<td></td>
<td></td>
<td>;Parameters capable of duplication</td>
</tr>
<tr>
<td>00000</td>
<td></td>
<td></td>
<td>!</td>
<td>M 0000,00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>;enter different parameter here</td>
<td></td>
</tr>
<tr>
<td>00001</td>
<td></td>
<td></td>
<td>&amp;</td>
<td>M 0000,02</td>
</tr>
<tr>
<td>00002</td>
<td></td>
<td></td>
<td>&amp;</td>
<td>M 0000,03</td>
</tr>
<tr>
<td>00004</td>
<td></td>
<td></td>
<td>=</td>
<td>M 0000,04</td>
</tr>
</tbody>
</table>
4. Nested duplication

The IL ranges that are to be duplicated can also be nested. The internal duplications are then repeated several times.

Element: CE4

```
BIT_IN N Y Y 0 04  ---- CE4
BIT_IN N Y Y 0 08  ---- E1
```

Creating the CE with the extended IL editor:

```
00000   04     ;Duplication block numbers don't have to begin with 1
00000   08
00001 &  PP 08  E2
          ;The duplication block can also contain a variable from the outer block
00002 /  PP 04  E1
00002 ]  04,
          10
00003 =  PP 10  A
        10
```

Calling the CE in the extended IL editor:

```
00000   ;nested duplication
00001  00000 $ CE4
00002  00000   E100  M 0000,00
00003  00001   E101  M 0000,01
00004  00002   E200  M 0000,02
00005  00003   E201  M 0000,03
00006  00004   E202  M 0000,04
00007  00005   A00  M 0000,05
00008
```
<table>
<thead>
<tr>
<th>SN</th>
<th>LN</th>
<th>Instr. / Variable</th>
<th>Symbol</th>
<th>Long text</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000</td>
<td>00000</td>
<td>; Nested duplication</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00001</td>
<td>&amp;</td>
<td>M 0000,02</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>; The duplication block can also contain a variable from the outer block</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00002</td>
<td>/</td>
<td>M 0000,00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00003</td>
<td>&amp;</td>
<td>M 0000,03</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>; The duplication block can also contain a variable from the outer block</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00004</td>
<td>/</td>
<td>M 0000,00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00005</td>
<td>&amp;</td>
<td>M 0000,04</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>; The duplication block can also contain a variable from the outer block</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00006</td>
<td>/</td>
<td>M 0000,00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00007</td>
<td>!</td>
<td>M 0000,01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00008</td>
<td>&amp;</td>
<td>M 0000,02</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>; The duplication block can also contain a variable from the outer block</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00009</td>
<td>/</td>
<td>M 0000,01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00010</td>
<td>&amp;</td>
<td>M 0000,03</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>; The duplication block can also contain a variable from the outer block</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00011</td>
<td>/</td>
<td>M 0000,01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00012</td>
<td>&amp;</td>
<td>M 0000,04</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>; The duplication block can also contain a variable from the outer block</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00013</td>
<td>/</td>
<td>M 0000,01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00014</td>
<td>=</td>
<td>M 0000,05</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
5. Requesting global flags

Sometimes it is necessary to hold a value of a variable of the last cycle (e.g. edge detection, historical values).

The management of these internal flags is done by 907 PC 33 (see data ranges). These flags are requested with the "@GLO" command.

The type of variable behind "@GLO" defines the type of requested flag.

The next flag of the global flag range is used.

A flag can be requested several times inside of a duplication block.

A global label is also requested with the "@GLO" command if the PLC supports these. The mechanism is the same as with flags.

Element: CE5

```
<table>
<thead>
<tr>
<th>Plaus. group</th>
<th>Dty</th>
<th>Inv</th>
<th>Dpl</th>
<th>Mod</th>
<th>No</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

```
BIT_IN  N Y Y 1 01 CE5  E 01 0 Y Y N BIT_OUT
```

Creating the CE with the extended IL editor:

<table>
<thead>
<tr>
<th>SN</th>
<th>LN</th>
<th>Instr./Variable</th>
<th>Symbol</th>
<th>Long text</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000</td>
<td></td>
<td>[</td>
<td>01</td>
<td></td>
</tr>
<tr>
<td>00000</td>
<td>@GLO</td>
<td>M</td>
<td>old_val</td>
<td></td>
</tr>
<tr>
<td>00000</td>
<td>!N</td>
<td>M</td>
<td>old_val</td>
<td></td>
</tr>
<tr>
<td>00001</td>
<td>&amp;</td>
<td>PP 01</td>
<td>E</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>;Impulse at positive signal</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00002</td>
<td></td>
<td>=PP 01</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>00001</td>
<td></td>
<td>PP 01</td>
<td>E</td>
<td></td>
</tr>
<tr>
<td>00004</td>
<td></td>
<td>M</td>
<td>old_val</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>]</td>
<td>01</td>
<td></td>
</tr>
</tbody>
</table>

Calling the CE in the extended IL editor:

<table>
<thead>
<tr>
<th>SN</th>
<th>LN</th>
<th>WN</th>
<th>Instr./Variable</th>
<th>Symbol</th>
<th>Long text</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000</td>
<td></td>
<td></td>
<td>;requesting a global flag</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00001</td>
<td>0000</td>
<td>0  CE5</td>
<td>00000,00</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td>00002</td>
<td>0000</td>
<td>0  E00</td>
<td>00000,01</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td>00003</td>
<td>0001</td>
<td>0  E01</td>
<td>00000,03</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td>00004</td>
<td>0002</td>
<td>0  A00</td>
<td>00000,04</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td>00005</td>
<td>0003</td>
<td>0  A01</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00006</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SN</td>
<td>LN</td>
<td>Instr. / Variable</td>
<td>Symbol</td>
<td>Long text</td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>-----</td>
<td>------------------</td>
<td>--------</td>
<td>--------------------------------</td>
<td></td>
</tr>
<tr>
<td>00000</td>
<td></td>
<td></td>
<td></td>
<td>;requesting a global flag</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>!N</td>
<td>M 0002,00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00001</td>
<td>&amp;</td>
<td>M 0000,00</td>
<td></td>
<td>;Impulse at positive signal</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>M 0000,03</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00002</td>
<td></td>
<td>!</td>
<td>M 0000,00</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>=</td>
<td>M 0002,00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00003</td>
<td></td>
<td>!N</td>
<td>M 0002,01</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>&amp;</td>
<td>M 0000,01</td>
<td></td>
<td>;Impulse at positive signal</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>M 0000,04</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00007</td>
<td></td>
<td>!</td>
<td>M 0000,01</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>=</td>
<td>M 0002,01</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
6. Requesting local flags

Sometimes variables for keeping results for further use are needed. Local flags are requested analog to
global ones with the @LOC command.

They are only valid inside of this CE and then they are assigned anew.
The same goes for local labels as long as these are supported by the PLC.

Element: CE6

| CE6     | | | | |
|---------|-------|------|-------|
| BIT_IN  | N     | Y    | Y     | 0     | 00   | -    |
| BIT_IN  | N     | Y    | Y     | 0     | 00   | E2   |
| BIT_IN  | N     | Y    | Y     | 0     | 00   | E3   |

Creating the CE with the extended IL editor:

<table>
<thead>
<tr>
<th>SN</th>
<th>LN</th>
<th>Instr./ Variable</th>
<th>Symbol</th>
<th>Long text</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000</td>
<td>;</td>
<td>Inputs XOR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00000</td>
<td>@LOC</td>
<td>M</td>
<td>XOR</td>
<td></td>
</tr>
<tr>
<td>00001</td>
<td>PP</td>
<td>00</td>
<td>E1</td>
<td></td>
</tr>
<tr>
<td>00002</td>
<td>&amp;</td>
<td>PP 00</td>
<td>E2</td>
<td></td>
</tr>
<tr>
<td>00003</td>
<td>=</td>
<td>M</td>
<td>XOR</td>
<td></td>
</tr>
<tr>
<td>00004</td>
<td>!</td>
<td>PP 00</td>
<td>E1</td>
<td></td>
</tr>
<tr>
<td>00005</td>
<td>/</td>
<td>M</td>
<td>XOR</td>
<td></td>
</tr>
<tr>
<td>00006</td>
<td>=</td>
<td>M</td>
<td>XOR</td>
<td></td>
</tr>
</tbody>
</table>

Calling the CE in the extended IL editor:

<table>
<thead>
<tr>
<th>SN</th>
<th>LN</th>
<th>WN</th>
<th>Instr./ Variable</th>
<th>Symbol</th>
<th>Long text</th>
</tr>
</thead>
<tbody>
<tr>
<td>00001</td>
<td>;</td>
<td></td>
<td>requesting local flags</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00002</td>
<td>00000</td>
<td>-</td>
<td>E1</td>
<td>M</td>
<td>0000.00</td>
</tr>
<tr>
<td>00003</td>
<td>00001</td>
<td>-</td>
<td>E2</td>
<td>M</td>
<td>0000.01</td>
</tr>
<tr>
<td>00004</td>
<td>00002</td>
<td>-</td>
<td>E3</td>
<td>M</td>
<td>0000.02</td>
</tr>
<tr>
<td>00005</td>
<td>00003</td>
<td>-</td>
<td>E3</td>
<td>M</td>
<td>0000.03</td>
</tr>
<tr>
<td>00006</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SN</td>
<td>LN</td>
<td>Instr. / Variable</td>
<td>Symbol</td>
<td>Long text</td>
<td></td>
</tr>
<tr>
<td>----</td>
<td>----</td>
<td>-------------------</td>
<td>--------</td>
<td>-----------</td>
<td></td>
</tr>
<tr>
<td>00000</td>
<td>0000</td>
<td>requesting local flags</td>
<td>M 0000,00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00001</td>
<td>0001</td>
<td>inputs  XOR</td>
<td>M 0000,01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00002</td>
<td>0002</td>
<td>=</td>
<td>M 0003,00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00003</td>
<td>0003</td>
<td>!</td>
<td>M 0000,01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00004</td>
<td>0004</td>
<td>&amp;</td>
<td>M 0000,00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00005</td>
<td>0005</td>
<td>/</td>
<td>M 0003,00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00006</td>
<td>0006</td>
<td>=</td>
<td>M 0003,00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00007</td>
<td>0007</td>
<td>!</td>
<td>M 0003,00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00008</td>
<td>0008</td>
<td>&amp;</td>
<td>M 0000,02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00009</td>
<td>0009</td>
<td>=</td>
<td>M 0000,03</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00010</td>
<td>0010</td>
<td>!</td>
<td>M 0000,02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00011</td>
<td>0011</td>
<td>&amp;</td>
<td>M 0003,00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00012</td>
<td>0012</td>
<td>/</td>
<td>M 0000,03</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00013</td>
<td>0013</td>
<td>=</td>
<td>M 0000,03</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
7. New absolute begin of duplication

It is possible to ignore specific CE parameters when duplicating.
If the necessary CE parameters have been chosen right it is possible to create different code depending on the duplication number. This is shown in the following example.

This is useful for a shift register for instance.

The normal duplication begin is 1, that means the first replacement is done for the first CE parameter.

The replacement can be set at the m. CE parameter with the „@START m“ command. m is between 1 and the amount of duplications.

The „@START“ command only goes for the next „[“.
If m has been chosen too large no IL is created for this duplication.

Element: CE7

<table>
<thead>
<tr>
<th>Plaus. group</th>
<th>Dty</th>
<th>Inv</th>
<th>Dpl</th>
<th>Mod No</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>No</th>
<th>Mod</th>
<th>Dpl</th>
<th>Inv</th>
<th>Dty</th>
<th>Plaus. group</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Y</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

BIT_IN N Y Y O 01 E A 00 O Y Y N BIT_OUT

Creating the CE with the extended IL editor:

<table>
<thead>
<tr>
<th>SN</th>
<th>LN</th>
<th>Instr. / Variable</th>
<th>Symbol</th>
<th>Long text</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000</td>
<td>00000</td>
<td>@START 002</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00001</td>
<td></td>
<td>[]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00002</td>
<td></td>
<td>!</td>
<td>PP 00</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>/</td>
<td>PP 01</td>
<td>E</td>
</tr>
<tr>
<td></td>
<td></td>
<td>=</td>
<td>PP 00</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>01</td>
<td></td>
</tr>
</tbody>
</table>

Callin the CE in the extended IL editor:

<table>
<thead>
<tr>
<th>SN</th>
<th>LN</th>
<th>Instr. / Variable</th>
<th>Symbol</th>
<th>Long text</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>000000</th>
<th>00001</th>
<th>00000</th>
<th>CE7</th>
</tr>
</thead>
<tbody>
<tr>
<td>00002</td>
<td>00000</td>
<td>00001</td>
<td>00002</td>
</tr>
<tr>
<td>00003</td>
<td>00002</td>
<td>00003</td>
<td>00004</td>
</tr>
<tr>
<td>00005</td>
<td>00003</td>
<td>00005</td>
<td>00006</td>
</tr>
<tr>
<td>00007</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

compiled IL:

<table>
<thead>
<tr>
<th>SN</th>
<th>LN</th>
<th>Instr. / Variable</th>
<th>Symbol</th>
<th>Long text</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000</td>
<td>00000</td>
<td>; New absolute duplication begin</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00001</td>
<td></td>
<td>!</td>
<td>M 0000,04</td>
<td></td>
</tr>
<tr>
<td>00002</td>
<td></td>
<td>/</td>
<td>M 0000,01</td>
<td></td>
</tr>
<tr>
<td>00003</td>
<td></td>
<td>=</td>
<td>M 0000,04</td>
<td></td>
</tr>
<tr>
<td>00004</td>
<td></td>
<td>!</td>
<td>M 0000,04</td>
<td></td>
</tr>
<tr>
<td>00005</td>
<td></td>
<td>/</td>
<td>M 0000,03</td>
<td></td>
</tr>
<tr>
<td>00006</td>
<td></td>
<td>=</td>
<td>M 0000,04</td>
<td></td>
</tr>
</tbody>
</table>
8. New absolute end of duplication

Parameters can also be ignored at the end of a duplication analog to ‚@START‘ with the ‚@END‘ command.

„@END“ also only goes for the next „[“. If „@END“ is smaller than „@START“ no IL is created for this duplication.

Element: CE8

Creating the CE with the extended IL editor:

<table>
<thead>
<tr>
<th>SN</th>
<th>LN</th>
<th>Instruction / Variable</th>
<th>Symbol</th>
<th>Long text</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000</td>
<td>00000</td>
<td>!</td>
<td>#B 0</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>;Only the first two parameters</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00001</td>
<td>@END</td>
<td>002</td>
<td>E</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>[</td>
<td>01</td>
<td></td>
</tr>
<tr>
<td>00001</td>
<td>&amp;</td>
<td>PP 01</td>
<td>E</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>]</td>
<td>01</td>
<td></td>
</tr>
<tr>
<td>00002</td>
<td>=</td>
<td>PP 00</td>
<td>A</td>
<td></td>
</tr>
</tbody>
</table>

Calling the CE in the extended IL editor:

<table>
<thead>
<tr>
<th>SN</th>
<th>LN</th>
<th>Instruction / Variable</th>
<th>Symbol</th>
<th>Long text</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000</td>
<td>00000</td>
<td>00000</td>
<td>CE8</td>
<td></td>
</tr>
<tr>
<td>00002</td>
<td>00000</td>
<td>E00 M 0000,00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00003</td>
<td>00001</td>
<td>E01 M 0000,01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00004</td>
<td>00002</td>
<td>E02 M 0000,02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00005</td>
<td>00003</td>
<td>E03 M 0000,03</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00006</td>
<td>00004</td>
<td>A M 0000,04</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00007</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

compiled IL:

<table>
<thead>
<tr>
<th>SN</th>
<th>LN</th>
<th>Instruction / Variable</th>
<th>Symbol</th>
<th>Long text</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000</td>
<td>00000</td>
<td>!</td>
<td>#B 0</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>;Only the first two parameters</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00001</td>
<td>&amp;</td>
<td>M 0000,00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00002</td>
<td>&amp;</td>
<td>M 0000,01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00003</td>
<td>=</td>
<td>M 0000,04</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
9. New relative begin of duplication

If a value smaller than 1 is entered with the „@START“ command this value is taken relative to the end of the duplication block. The third CE parameter before last is started with if „@START -2“ was entered for instance. The begin is the sum of the duplication number and the value behind „@START“.

If the sum is smaller than 1 no IL code is created.

Element: CE9

<table>
<thead>
<tr>
<th>Plaus. group</th>
<th>Dty Inv Dpl Mod No</th>
<th>No Mod Dpl Inv Dty Plaus. group</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BIT_IN N Y Y 0 01</td>
<td>CE9 E A 0 0 Y Y N BIT_OUT</td>
</tr>
</tbody>
</table>

Creating the CE with the extended IL editor:

<table>
<thead>
<tr>
<th>SN</th>
<th>Ln</th>
<th>Instr. / Variable</th>
<th>Symbol</th>
<th>Long text</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td></td>
<td>@START -001</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0001</td>
<td></td>
<td>PP 00</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>0002</td>
<td></td>
<td>PP 01</td>
<td>E</td>
<td></td>
</tr>
</tbody>
</table>

Calling the CE in the extended IL editor:

<table>
<thead>
<tr>
<th>SN</th>
<th>Ln</th>
<th>Wn</th>
<th>Instr./ Variable</th>
<th>Symbol</th>
<th>Long text</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td></td>
<td></td>
<td>$ CE9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0002</td>
<td></td>
<td></td>
<td>E00 M 0000,00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0003</td>
<td></td>
<td></td>
<td>E01 M 0000,01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0004</td>
<td></td>
<td></td>
<td>E02 M 0000,02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0005</td>
<td></td>
<td></td>
<td>E03 M 0000,03</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0006</td>
<td></td>
<td></td>
<td>-A M 0000,04</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

compiled IL:

<table>
<thead>
<tr>
<th>SN</th>
<th>Ln</th>
<th>Instr. / Variable</th>
<th>Symbol</th>
<th>Long text</th>
</tr>
</thead>
<tbody>
<tr>
<td>00016</td>
<td></td>
<td>;New relative duplication begin</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0002</td>
<td></td>
<td>;Begin with the one before last</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0000</td>
<td></td>
<td>M 0000,04</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0001</td>
<td></td>
<td>M 0000,02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0002</td>
<td></td>
<td>M 0000,04</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0003</td>
<td></td>
<td>M 0000,04</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0004</td>
<td></td>
<td>M 0000,03</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0005</td>
<td></td>
<td>M 0000,04</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
10. New relative end of duplication

The end of duplication can be defined with the „@END“ command analog to the „@START“ command relatively to the amount of duplications.

Element: CE10

<table>
<thead>
<tr>
<th>Plaus. group</th>
<th>Dty</th>
<th>Inv</th>
<th>Dpl</th>
<th>Mod No</th>
<th>No Mod</th>
<th>Dpl</th>
<th>Inv</th>
<th>Dty</th>
<th>Plaus. group</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

```
BIT_IN  N  Y  Y  0  01
       E  A

00000 00000 0  #B  0
       00001   @END  -001
       01
       00001  &  PP 01  E
       01
       00002  =  PP 00  A
```

Creating the CE with the extended IL editor:

<table>
<thead>
<tr>
<th>SN</th>
<th>LN</th>
<th>WN</th>
<th>Instr./ Variable</th>
<th>Symbol</th>
<th>Long text</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000</td>
<td></td>
<td></td>
<td>;New relative duplication end</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00000</td>
<td>00001</td>
<td>00000</td>
<td>$ CE10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00002</td>
<td>00000</td>
<td></td>
<td>E00  M  0000,00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00003</td>
<td>00001</td>
<td></td>
<td>E01  M  0000,01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00004</td>
<td>00002</td>
<td></td>
<td>E02  M  0000,02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00005</td>
<td>00003</td>
<td></td>
<td>E03  M  0000,03</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00006</td>
<td>00004</td>
<td></td>
<td>A  M  0000,04</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Calling the CE in the extended IL editor:

<table>
<thead>
<tr>
<th>SN</th>
<th>LN</th>
<th>WN</th>
<th>Instr./ Variable</th>
<th>Symbol</th>
<th>Long text</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000</td>
<td></td>
<td></td>
<td>;New relative duplication end</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00000</td>
<td></td>
<td></td>
<td>#B  0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00001</td>
<td>&amp;</td>
<td>M</td>
<td>0000,00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00002</td>
<td>&amp;</td>
<td>M</td>
<td>0000,01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00003</td>
<td>&amp;</td>
<td>M</td>
<td>0000,02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00004</td>
<td>=</td>
<td>M</td>
<td>0000,04</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
11. Offset inside of a duplication

Sometimes two different CE parameters of a duplication block are linked. For the example of a shift register the parameter before must be accessed.

This is done with the „@OFF m“ command. m can be positive or negative. An „@OFF“ command only goes for the following instruction, after that the offset is set to zero.

Element: CE11

Creating the CE with the extended IL editor:

<table>
<thead>
<tr>
<th>SN</th>
<th>LN</th>
<th>Instr. / Variable</th>
<th>Symbol</th>
<th>Long text</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00000</td>
<td></td>
<td>@START 002</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00000</td>
<td></td>
<td>[</td>
<td>01</td>
<td></td>
</tr>
<tr>
<td>00000</td>
<td></td>
<td>PP 01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00001</td>
<td></td>
<td>@OFF -001</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00001</td>
<td></td>
<td>PP 01</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Calling the CE in the extended IL editor:

<table>
<thead>
<tr>
<th>SN</th>
<th>LN</th>
<th>WN</th>
<th>Instr./ Variable</th>
<th>Symbol</th>
<th>Long text</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000</td>
<td></td>
<td></td>
<td>;Offset inside of a duplication</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00000</td>
<td></td>
<td>00000</td>
<td>$ CE11</td>
<td>E</td>
<td></td>
</tr>
<tr>
<td>00003</td>
<td></td>
<td>00001</td>
<td>E01 M 0000,01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00004</td>
<td></td>
<td>00002</td>
<td>E02 M 0000,02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00005</td>
<td></td>
<td>00003</td>
<td>E03 M 0000,03</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>00006</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

compiled IL:

<table>
<thead>
<tr>
<th>SN</th>
<th>LN</th>
<th>Instr. / Variable</th>
<th>Symbol</th>
<th>Long text</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000</td>
<td></td>
<td>;Offset inside of a duplication</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>;Shift register</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00000</td>
<td></td>
<td>! M 0000,01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00001</td>
<td></td>
<td>= M 0000,00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00002</td>
<td></td>
<td>! M 0000,02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00003</td>
<td></td>
<td>= M 0000,01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00004</td>
<td></td>
<td>! M 0000,03</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00005</td>
<td></td>
<td>= M 0000,02</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
12. Maximum with absolute and (!) relative begin of duplication

If an absolute value was entered for the beginning and a relative value for the end with `.START` command the maximum is taken.

Element: CE12

<table>
<thead>
<tr>
<th>BIT_IN</th>
<th>N</th>
<th>Y</th>
<th>Y</th>
<th>0</th>
<th>01</th>
<th>CE12</th>
<th>E</th>
<th>A</th>
<th>00</th>
<th>0</th>
<th>Y</th>
<th>N</th>
<th>BIT_OUT</th>
</tr>
</thead>
</table>

Creating the CE with the extended IL editor:

<table>
<thead>
<tr>
<th>SN</th>
<th>LN</th>
<th>Instr. / Variable</th>
<th>Symbol</th>
<th>Long text</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td></td>
<td>;Begin with one before last</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00000</td>
<td></td>
<td>@START -001</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00000</td>
<td></td>
<td>;but at least with second</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00000</td>
<td></td>
<td>002</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00000</td>
<td></td>
<td>[</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00001</td>
<td></td>
<td>PP 00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00002</td>
<td></td>
<td>PP 01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00002</td>
<td></td>
<td>]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Calling the CE in the extended IL editor:

<table>
<thead>
<tr>
<th>SN</th>
<th>LN</th>
<th>WN</th>
<th>Instr. / Variable</th>
<th>Symbol</th>
<th>Long text</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000</td>
<td></td>
<td>;Absolute and relative duplication begin</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00001</td>
<td></td>
<td>00001 00000 $ CE12</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00002</td>
<td></td>
<td>0000 00000 --- E00 M 0000,00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00003</td>
<td></td>
<td>00001 --- E01 M 0000,01</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00004</td>
<td></td>
<td>00002 --- E02 M 0000,02</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00005</td>
<td></td>
<td>00003 --- E03 M 0000,03</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00006</td>
<td></td>
<td>00004 --- E04 M 0000,04</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00007</td>
<td></td>
<td>00005 --- E05 M 0000,05</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00008</td>
<td></td>
<td>00006 --- A M 0000,06</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00009</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

compiled IL:

<table>
<thead>
<tr>
<th>SN</th>
<th>LN</th>
<th>Instr. / Variable</th>
<th>Symbol</th>
<th>Long text</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000</td>
<td></td>
<td>;Absolute and relative duplication begin</td>
<td></td>
<td></td>
</tr>
<tr>
<td>000001</td>
<td></td>
<td>00000 00000 --- E00 M 0000,06</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00001</td>
<td></td>
<td>/ M 0000,04</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00002</td>
<td></td>
<td>= M 0000,06</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SN</th>
<th>LN</th>
<th>Instr. / Variable</th>
<th>Symbol</th>
<th>Long text</th>
</tr>
</thead>
<tbody>
<tr>
<td>00001</td>
<td></td>
<td>00003 ! M 0000,06</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00004</td>
<td></td>
<td>/ M 0000,05</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00005</td>
<td></td>
<td>= M 0000,06</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
13. Minimum with absolute and (!) relative duplication end

If an absolute value was entered for the beginning and a relative value for the end with the "@END" command the minimum is taken.

Element: CE13

Creating the CE with the extended IL editor:

<table>
<thead>
<tr>
<th>SN</th>
<th>LN</th>
<th>WN</th>
<th>Instr./Variable</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000</td>
<td>00000</td>
<td>00000</td>
<td>#B 0</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>;Until the one before last</td>
<td></td>
</tr>
<tr>
<td>00001</td>
<td>@END</td>
<td>-001</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>;but max to the third</td>
<td></td>
</tr>
<tr>
<td>00001</td>
<td>@END</td>
<td>003</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[</td>
<td></td>
</tr>
<tr>
<td>00001</td>
<td>&amp;</td>
<td>PP 01</td>
<td>E</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>]</td>
<td></td>
</tr>
<tr>
<td>00002</td>
<td>=</td>
<td>PP 00</td>
<td>A</td>
<td></td>
</tr>
</tbody>
</table>

Calling the CE in the extended IL editor:

<table>
<thead>
<tr>
<th>SN</th>
<th>LN</th>
<th>WN</th>
<th>Instr./Variable</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000</td>
<td>00001</td>
<td>00000</td>
<td>$ CE13</td>
<td></td>
</tr>
<tr>
<td>00002</td>
<td>00000</td>
<td></td>
<td>E00 M 0000,00</td>
<td></td>
</tr>
<tr>
<td>00003</td>
<td>00001</td>
<td></td>
<td>E01 M 0000,01</td>
<td></td>
</tr>
<tr>
<td>00004</td>
<td>00002</td>
<td></td>
<td>E02 M 0000,02</td>
<td></td>
</tr>
<tr>
<td>00005</td>
<td>00003</td>
<td></td>
<td>E03 M 0000,03</td>
<td></td>
</tr>
<tr>
<td>00006</td>
<td>00004</td>
<td></td>
<td>E04 M 0000,04</td>
<td></td>
</tr>
<tr>
<td>00007</td>
<td>00005</td>
<td></td>
<td>E05 M 0000,05</td>
<td></td>
</tr>
<tr>
<td>00008</td>
<td>00006</td>
<td>-A</td>
<td>M 0000,06</td>
<td></td>
</tr>
<tr>
<td>00009</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

compiled IL:

<table>
<thead>
<tr>
<th>SN</th>
<th>LN</th>
<th>WN</th>
<th>Instr./Variable</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000</td>
<td>00000</td>
<td></td>
<td>#B 0</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>;Until the one before last</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>;but max to the third</td>
<td></td>
</tr>
<tr>
<td>00001</td>
<td>&amp;</td>
<td>M 0000,00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00002</td>
<td>&amp;</td>
<td>M 0000,01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00003</td>
<td>&amp;</td>
<td>M 0000,02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00004</td>
<td>=</td>
<td>M 0000,06</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
14. Complete example

The signals of the inputs are to be linked exclusive-or although the PLC only knows the commands AND and OR in the example.

Element: CE14

Creating the CE with the extended IL editor:

<table>
<thead>
<tr>
<th>SN</th>
<th>LN</th>
<th>Instr. / Variable</th>
<th>Symbol</th>
<th>Long text</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td></td>
<td>@LOC M</td>
<td>XOR</td>
<td></td>
</tr>
<tr>
<td>0000</td>
<td></td>
<td>;Signal flag</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0000</td>
<td></td>
<td>@LOC M</td>
<td>signal</td>
<td></td>
</tr>
<tr>
<td>0000</td>
<td></td>
<td>;Special case first input</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0000</td>
<td></td>
<td>@START 001</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0000</td>
<td></td>
<td>@END 001</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0000</td>
<td></td>
<td>[</td>
<td>01</td>
<td></td>
</tr>
<tr>
<td>0000</td>
<td></td>
<td>@GLO M</td>
<td>form_con</td>
<td></td>
</tr>
<tr>
<td>0000</td>
<td></td>
<td>!N M</td>
<td>form_con</td>
<td></td>
</tr>
<tr>
<td>0000</td>
<td></td>
<td>&amp; PP 01</td>
<td>E</td>
<td></td>
</tr>
<tr>
<td>0000</td>
<td></td>
<td>= M</td>
<td>old_val</td>
<td></td>
</tr>
<tr>
<td>0000</td>
<td></td>
<td>]</td>
<td>01</td>
<td></td>
</tr>
<tr>
<td>0000</td>
<td></td>
<td>;keep first signal</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0000</td>
<td></td>
<td>00005 @GLO M</td>
<td>form_con</td>
<td></td>
</tr>
<tr>
<td>0000</td>
<td></td>
<td>00005 !N M</td>
<td>form_con</td>
<td></td>
</tr>
<tr>
<td>0000</td>
<td></td>
<td>00006 &amp; PP 01</td>
<td>E</td>
<td></td>
</tr>
<tr>
<td>0000</td>
<td></td>
<td>00007 = M</td>
<td>Signal</td>
<td></td>
</tr>
<tr>
<td>0000</td>
<td></td>
<td>00008 !N M</td>
<td>Signal</td>
<td></td>
</tr>
<tr>
<td>0000</td>
<td></td>
<td>00009 &amp; M</td>
<td>XOR</td>
<td></td>
</tr>
<tr>
<td>0000</td>
<td></td>
<td>00010 / M</td>
<td>Signal</td>
<td></td>
</tr>
<tr>
<td>0000</td>
<td></td>
<td>00011 &amp; M</td>
<td>XOR</td>
<td></td>
</tr>
<tr>
<td>0000</td>
<td></td>
<td>00012 = M</td>
<td>XOR</td>
<td></td>
</tr>
<tr>
<td>0000</td>
<td></td>
<td>00004 0013 ! PP 01</td>
<td>E</td>
<td>form_con</td>
</tr>
<tr>
<td>0000</td>
<td></td>
<td>00014 = M</td>
<td>form_con</td>
<td></td>
</tr>
<tr>
<td>0000</td>
<td></td>
<td>00005 ]</td>
<td>01</td>
<td></td>
</tr>
</tbody>
</table>
Calling the CE in the extended IL editor:

<table>
<thead>
<tr>
<th>SN</th>
<th>LN</th>
<th>WN</th>
<th>Instr./Variable</th>
<th>Symbol</th>
<th>Long text</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000</td>
<td>00000</td>
<td>00000</td>
<td>$ CE14</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00002</td>
<td>00000</td>
<td></td>
<td>E00 M 0000,00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00003</td>
<td>00001</td>
<td></td>
<td>E01 M 0000,01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00004</td>
<td>00002</td>
<td></td>
<td>E02 M 0000,02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00005</td>
<td>00003</td>
<td></td>
<td>E03 M 0000,03</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00006</td>
<td>00004</td>
<td></td>
<td>-A M 0000,04</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

00007
<table>
<thead>
<tr>
<th>SN</th>
<th>LN</th>
<th>Instr. / Variable</th>
<th>Symbol</th>
<th>Long text</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td></td>
<td>;Main example</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>;Result flag</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>;Signal flag</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>;Special case first input</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0000</td>
<td></td>
<td>!N</td>
<td>M</td>
<td>0002,02</td>
</tr>
<tr>
<td>0000</td>
<td></td>
<td>&amp;</td>
<td>M</td>
<td>0000,00</td>
</tr>
<tr>
<td>0000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0000</td>
<td></td>
<td>;keep first signal</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0002</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>;all inputs from the second</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>!N</td>
<td>M</td>
<td>0002,03</td>
</tr>
<tr>
<td>0003</td>
<td></td>
<td>&amp;</td>
<td>M</td>
<td>0000,01</td>
</tr>
<tr>
<td>0003</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0004</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0005</td>
<td></td>
<td>!N</td>
<td>M</td>
<td>0003,01</td>
</tr>
<tr>
<td>0009</td>
<td></td>
<td>&amp;</td>
<td>M</td>
<td>0003,00</td>
</tr>
<tr>
<td>0010</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0011</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0012</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0006</td>
<td></td>
<td>!N</td>
<td>M</td>
<td>0003,01</td>
</tr>
<tr>
<td>0018</td>
<td></td>
<td>&amp;</td>
<td>M</td>
<td>0003,00</td>
</tr>
<tr>
<td>0020</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0021</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0022</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0007</td>
<td></td>
<td>!N</td>
<td>M</td>
<td>0000,02</td>
</tr>
<tr>
<td>0023</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0024</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0008</td>
<td></td>
<td>!N</td>
<td>M</td>
<td>0002,05</td>
</tr>
<tr>
<td>0025</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0026</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0027</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0009</td>
<td></td>
<td>!N</td>
<td>M</td>
<td>0003,01</td>
</tr>
<tr>
<td>0028</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0029</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0030</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0031</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0032</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0010</td>
<td></td>
<td>!N</td>
<td>M</td>
<td>0000,03</td>
</tr>
<tr>
<td>0033</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0011</td>
<td></td>
<td>!N</td>
<td>M</td>
<td>0003,00</td>
</tr>
<tr>
<td>0035</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0012</td>
<td></td>
<td>!N</td>
<td>M</td>
<td>0003,04</td>
</tr>
<tr>
<td>0036</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Creating the program run in the CE with the FBD/LD-editor

The FBD/LD editor is selected with <F2> and started with <F1>.

Move the cursor further to the bottom and further to the right of the screen and enter the name of the &
connection element. The =N CE is also needed. Both CEs are assigned with connections.

There the formal parameter descriptions are entered as symbols. The absolute formal parameter
"PP 0" is entered in the single line editor. The so created program run must be compiled.

The compiled IL can be displayed with <Ctrl><F3>. This display is only valid for representation of CEs
in the extended IL and for display of flags for connection lines between the CEs.

As a standard, local flags are assigned to the lines between the CEs. If global flags should be used,
"@GLO" must be entered in the first line as comment.

See chapter FBD/LD for handling the FBD/LD editor.

A NAND created with the FBD/LD editor could look like this:
If the library layer is entered, the field „Src“ is marked with „FBD“. If the compilation was successful, the
field IL is marked with „*“.

Notes:
1. Direct connections are not allowed when creating a logic element in the FBD!

2. The command @GLO allows you to define whether global flags should be issued for the lines between
the CE/FBD.

But what must be given when CEs with direct flags, such as RS, SR, I+ or I-, are used in the CE-FBD
and a direct flag should be issued automatically? The following uses an example to describe a procedure
for solving this problem.
Step 1:
Define the CE in the FBD and specify PPO for all variables which are to be given later.

Note: The symbol names written in *italics* here appear on the monitor by switching over from absolute to symbolic display with <F9>. 
Step 2:
Define a CE in the extended IL editor by calling up the CE which you have created in step 1. You can now specify a global or local flag for the variable HM1. The variable then remains hidden within this CE and is issued automatically.
Creating the program run with the extended IL

The extended IL editor is selected with <F2> and started with <F1>.

The CEs & and = have to be entered and the formal parameters and formal parameter designations have to be entered. See chapter extended IL editor for handling the extended IL editor.

Such a NAND created with the extended IL editor could look like this:

```
CE: NAND\ SP-NO: 0001 Overwrite

<table>
<thead>
<tr>
<th>SN</th>
<th>ZN</th>
<th>Instr. / Variable</th>
<th>Symbol</th>
<th>Long text</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000 00000</td>
<td>$ &amp;</td>
<td>E1 PP 00</td>
<td>E1</td>
<td></td>
</tr>
<tr>
<td>00002</td>
<td>E2 PP 00</td>
<td>E2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00003</td>
<td>A1 PP 00</td>
<td>A1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00004</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00005</td>
<td>$ =</td>
<td>E1 PP 00</td>
<td>A1</td>
<td></td>
</tr>
<tr>
<td>00007</td>
<td>A0A1 PP 00</td>
<td>A1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00008</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

The field „Src“ is filled with „eIL“ and the IL field is marked with “*” after returning to the library layer.

Duplicating CEs

There is a possibility of duplicating CEs.

Only the CE graphic and the program code of an already existing CE are copied with this function. Position the Cursor on the CE to be copied and press <Alt>-F2.

If all CEs are displayed, the menu for selecting the CE group appears into which the new duplicated CE is to be included. After selecting a CE group and pressing the <Enter key> insert a blank line before the CE being duplicated. A new CE name must now be entered. Exception: The same name can be stated when copying manufacturer CEs.

The duplication procedure is then carried out.

If you are already in a group when duplicating (screen display is not ALL), it is not necessary to select a CE group into which the CE is to be included. The new CE is automatically allocated to the currently displayed group.
In the example above, position the cursor an the CE „<„ and press <Alt>-<F2>. After entering the new CE name the CE graphic, CE IL and existing marks in the „M“ or „B“ columns of the „INC“ CE are saved to the new CE name and the new CE is appended to the occurring order.

<table>
<thead>
<tr>
<th>B M</th>
<th>CE name</th>
<th>CE comment</th>
<th>Grf</th>
<th>Src</th>
<th>IL</th>
<th>Doc</th>
<th>M/U</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AND</td>
<td></td>
<td>*</td>
<td>eIL</td>
<td>*</td>
<td>*</td>
<td>H</td>
<td>26.03.92</td>
</tr>
<tr>
<td></td>
<td>*</td>
<td>Multiplication</td>
<td>*</td>
<td>eIL</td>
<td>*</td>
<td>*</td>
<td>H</td>
<td>26.03.92</td>
</tr>
<tr>
<td></td>
<td>*:</td>
<td>Multiplication with division</td>
<td>*</td>
<td>eIL</td>
<td>*</td>
<td>*</td>
<td>H</td>
<td>26.03.92</td>
</tr>
<tr>
<td></td>
<td>*D</td>
<td>Multiplication, double word</td>
<td>*</td>
<td>eIL</td>
<td>*</td>
<td>*</td>
<td>H</td>
<td>26.03.92</td>
</tr>
<tr>
<td></td>
<td>+</td>
<td>Addition</td>
<td>*</td>
<td>eIL</td>
<td>*</td>
<td>*</td>
<td>H</td>
<td>26.03.92</td>
</tr>
<tr>
<td></td>
<td>+D</td>
<td>Addition, double word</td>
<td>*</td>
<td>eIL</td>
<td>*</td>
<td>*</td>
<td>H</td>
<td>26.03.92</td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>Subtraction</td>
<td>*</td>
<td>eIL</td>
<td>*</td>
<td>*</td>
<td>H</td>
<td>26.03.92</td>
</tr>
<tr>
<td></td>
<td>-D</td>
<td>Subtraction, double word</td>
<td>*</td>
<td>eIL</td>
<td>*</td>
<td>*</td>
<td>H</td>
<td>26.03.92</td>
</tr>
<tr>
<td></td>
<td>/</td>
<td>OR</td>
<td>*</td>
<td>eIL</td>
<td>*</td>
<td>*</td>
<td>H</td>
<td>26.03.92</td>
</tr>
<tr>
<td></td>
<td>:</td>
<td>Division</td>
<td>*</td>
<td>eIL</td>
<td>*</td>
<td>*</td>
<td>H</td>
<td>26.03.92</td>
</tr>
<tr>
<td></td>
<td>:=</td>
<td>Input in FBD not possible</td>
<td>eIL</td>
<td>*</td>
<td>*</td>
<td></td>
<td>H</td>
<td>26.03.92</td>
</tr>
<tr>
<td></td>
<td>:==</td>
<td>Input in FBD not possible</td>
<td>eIL</td>
<td>*</td>
<td>*</td>
<td></td>
<td>H</td>
<td>04.11.93</td>
</tr>
<tr>
<td></td>
<td>:=:=WO</td>
<td>Input in FBD not possible</td>
<td>eIL</td>
<td>*</td>
<td>*</td>
<td></td>
<td>H</td>
<td>26.03.92</td>
</tr>
<tr>
<td></td>
<td>:D</td>
<td>Division, double word</td>
<td>*</td>
<td>eIL</td>
<td>*</td>
<td>*</td>
<td>H</td>
<td>26.03.92</td>
</tr>
<tr>
<td></td>
<td>&lt;.&lt;</td>
<td>Less than</td>
<td>*</td>
<td>eIL</td>
<td>*</td>
<td>*</td>
<td>H</td>
<td>26.03.92</td>
</tr>
<tr>
<td></td>
<td>&lt;.&lt;=</td>
<td>Less than or equal to</td>
<td>*</td>
<td>eIL</td>
<td>*</td>
<td>*</td>
<td>H</td>
<td>26.03.92</td>
</tr>
<tr>
<td></td>
<td>&lt;&gt;&lt;=..</td>
<td>Unequal</td>
<td>*</td>
<td>eIL</td>
<td>*</td>
<td>*</td>
<td>H</td>
<td>26.03.92</td>
</tr>
</tbody>
</table>
15.10 Reading in manufacturer libraries

There is a possibility of reading new manufacturer libraries by selecting the appropriate function in the 907 PC 33 main menu (Read manuf. library).

Note:
Manufacturer libraries cannot be read in from write-protected diskettes.

A window is displayed requesting the filename of the library that is to be read.

Enter name of manufacturer's library, please.

The entered manufacturer library is read. Everything is accepted into the new manufacturer library. Elements in the old library are overwritten by the same elements in the new library. Additional elements in the new library are appended and elements of the old library that aren't listed in the new one remain. This goes for CEs, plausibility groups and CE groups.

After reading in, the library data is reorganized. For example, non-used data is deleted.

Note:
If the message 'The block version is too old. Please call up block as library' appears, carry out the following:

Call up the menu Project man. -> Project data. Enter the name of the block (e.g. NEW) under library. Next, call up the CE library. Quit this again with <Esc>. In the project data, enter the 'old' library again into which the block is to be read. Then again read the manufacturers library of the block read with <Ctrl>-K-R in the case of user CEs.
16  Printing

16.1 Introduction

A printout of the lists given in the print menus can be produced for the purpose of complete documentation or verification of a program.

The corresponding cover sheet containing the project data can be printed out before every list. If required, all data entered in the print screens is stored in a file entitled „project name.DPM“.

All lists are page-formatted and have header and footer lines as defined in the print format editor (see print format editor).

For all lists it is possible to divert the printer output to a file bearing a freely chosen name. If several lists are output to the same file, these lists will be appended to each other.

Printing can be aborted by pressing the space bar.

The following lists are only necessary if projects have been modularized and can therefore only be selected in the „Total project function selection“ menu:

- Modularization list
- Total variable list
- Total cross reference list

If lists are printed containing the data of other lists e.g. the CE library, the printing parameters of the individual lists are used for the settings of these lists.

Note:

A DOS system message is displayed in the bottom line of the screen if the connection to the printer has been disconnected while printing or if the printer has not been connected when printing started.

Device driver error      Abort (A)     Repeat (R)     Ignore (I)

The data of the project retain unchanged if the print procedure cannot be repeated by pressing „R“ and has to be aborted by pressing „A“. When recalling 907 PC 33 the data saving displays the message

Project xxxx is erroneous !!!

In this case choose Option 1.
16.2 General print parameters

General format specifications applicable to all lists to be printed are defined in the „General print parameters“ screen.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Page length</td>
<td>72</td>
</tr>
<tr>
<td>Left margin</td>
<td>10</td>
</tr>
<tr>
<td>Papercut</td>
<td>N</td>
</tr>
</tbody>
</table>

**Explanation of parameters**

- **Page length**: Number of lines on one printed page (the default is 72).
- **Left margin**: Column width of the left margin.
- **Saving paper**: If Y (yes) is entered, page feed is suppressed in the case of some lists.

**Conversion file**

If conversion of characters is required, the file name in which the conversion is defined must be entered here.

For each character to be converted, the conversion file (ASCII file) must contain the following line:

```
\z\zw1;  (When printing, the character z is converted to the character zw1)
```

or

```
\z\zw1\zw2;  (When printing, the character z is converted to the characters zw1 and zw2)
```

Any chosen numbers between 0 and 255 are allowed for z, zw1 and zw2.

The semicolon may be followed by any chosen remark. It is possible to enter a maximum of 2 characters to which a character is to be converted when printing.

If the printer 07 DR 11 is connected, the file name „PLC_name.IT0“, e.g. T200.IT0, is entered.

**Initialization file**

The initialization file may contain an initialization text to be sent to the printer directly before printing. Control characters are entered by means of \xxx. In this way it is possible, for example, to initialize or adapt laser printers before printing.
Reinitialization file

When this file exists, its contents are sent to the printer at the end of the printing process.

Important note:

A text editor which does not write ETX at the end of the file must be used to create the initialization and reinitialization files. This is the case, for example, with the DOS editor EDIT and with the 907 PC 33's own text editor.

Initialization, reinitialization and conversion files are available for the following printer for 907 PC 33: 07 DR 11 with the associated files T200.ITO and CS31.ITO.

KYOCERA with the files INIT-A3.KYO (initialization file for A3), INIT-A4.KYO (initialization file for A4), KONVERT.KYO (conversion file), REINIT.KYO (reinitialization file).

HP LaserJet 4 with the files INIT.HP4 and REINIT.HP4

Printouts with KYOCERA:

The following printing parameters can be set:
For A3: Page length: 88; left edge: 0.
For A4: Page length: 88; left edge: 0.

Printing out with HP LaserJet 4:
For A4: Page length: 86; left edge: 10.

Further notes: See Appendix

---

16.3 Print format editor

The print format editor enables free design of header and footer lines for listings.

<table>
<thead>
<tr>
<th>907 PC 33</th>
<th>Print format editor</th>
<th>EXAMPLE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>File name: EXAMPLE</td>
<td>No. of header lines 3</td>
</tr>
<tr>
<td></td>
<td>.................................</td>
<td>@V</td>
</tr>
</tbody>
</table>

This text appears in listings' header lines

|           | ................................. | Page @@@+S | @V |

This text appears in listings' footer lines

Data entered is stored in a file with a freely chosen name and the automatic name extension „DFE“ and can therefore be edited further at any chosen time.

When printing listings, however, only the files with the names...
project_name.DFE
and
PLC_name.DFE
are used with the above priority for designing header and footer lines according to their presence. No header and footer lines will be printed if neither of the two files exists.

By way of default, 907 PC 33 contains the file with the designation „PLC name.DFE“. When printing, the data in this file will be output as the header and footer.

If a different format is ever needed for a special project, after editing it must be stored with the designation „project_name.DFE“. If you want to change the standard file so that all projects have the header and footer lines you require, load the file MPST.DFE (for 907 PC 331) or T200.DFE (for 907 PC 332) in the print format editor and modify them according to your requirements.

Input fields
The print format editor comprises 5 input fields between which you can toggle as required by pressing the cursor keys:

Filename
When you call the print format editor, a window first of all appears for you to enter the print format file’s designation.

The file name is entered without the extension „.DFE“, which is appended automatically. It can also include path information if you do not wish to store the file in the current directory.

When you call the editor, this is the name of the file to be edited or created. The project name appears by default and, if applicable, you only have to confirm it by pressing the <Return> key.

Before you terminate the editor, the filename window reappears. By pressing the <Return> key, you can store the data under the entered name or you can reject all changes by pressing the <Esc> key. It is therefore possible to store the data under a different name by changing the filename.

Number of header lines
Here you enter the number of lines to be kept free on a printed page for the header lines. A maximum of 4 header lines is possible. If you have entered more lines, an error message will be issued when you terminate the editor.

Number of footer lines
Here you enter the number of lines to be kept free at the end of each printed page for the entered footer line. A maximum of 10 footer lines is possible. If you enter more lines, an error message will be issued when you terminate the editor.

Field for entering the header lines
A maximum of 4 text lines to appear on printing pages as header lines can be edited in this field. As the number of characters/line on listings is more than 80 characters, you can scroll to the left and right in the window.

You can incorporate control characters (see control characters) anywhere in the text to enable automatic output of specific data.

All editing functions are listed overleaf.

Field for entering footer lines
Analogously to input of header lines, you can freely style a footer (up to 10 lines) here for listings.
Editing functions

The following functions are provided within the framed area:

- **<Cursor left>** The cursor moves 1 position to the left. Scrolling to the right from the left margin.
- **<Cursor right>** The cursor moves 1 position to the right. Scrolling to the left from the right margin.
- **<Cursor up>** The cursor moves 1 position up. At the top margin, it moves up to the next input field.
- **<Cursor down>** The cursor moves 1 position down. At the bottom margin, the cursor moves down to the next input field.
- **<Backspace>** The cursor moves 1 position to the left. The character at the cursor’s new position is deleted.
- **<Tab. left>** The cursor moves to the left by 10 positions.
- **<Tab. right>** The cursor moves to the right by 10 positions.
- **<Home>** The cursor moves to the top left corner.
- **<End>** The cursor moves to the bottom line.
- **<PgUp>** The cursor moves to the 1st line.
- **<PgDn>** The cursor moves to the last line.
- **<Enter>** The cursor moves to the start of the next line.
- **<Ins>** Insert mode is activated; this mode is deactivated again whenever any of the functions described here is called.
- **<Del>** The character at the cursor’s position is deleted. Everything in the line on the right of the cursor’s position is moved to the left by 1 character.
- **<Ctrl>-N** A line is inserted.
- **<Ctrl>-Y** A line is deleted.
- **<Ctrl>-Q-Y** All characters as from the cursor’s current position are deleted up to the end of the line.
- **<Esc>** The print format editor is terminated.

Control characters

Besides the texts that appear unchanged on the listings, the following control characters can be entered anywhere within the framed input field to achieve automatic output of specific data.

- **@N** Name of the project file (depending on the listing)
- **@D** Date of the project file (depending on the listing)
- **@U** Time of the project file (depending on the listing)
- **@T** Type of PLC from the project data
- **@H** Manufacturer of the PLC from the project data
- **@P** Line 1 of the project designation from the project data
- **@p** Line 2 of the project designation from the project data
- **@B** Responsible person from the project data
- **@A** Last change from the project data
- **@1** 1st text line of project data
- **@2** 2nd text line of project data
In respect of sheet width, A4 format is always assumed when creating header and footer lines. The control character "@V" therefore allows you to adapt to a different format without having to create additional header and footer lines for this format.

From the position of the ",@" onwards, the rest of the line is moved up to the right margin. The resulting gap is filled with the character directly following the "V".

**Formatting outputs of this data**

The number of "@" characters and the subsequent command character define the total width of the field. A "+" placed before the command character allows you to write right-justified text into the field.

Examples:

```
@@.WARNING@D
writes the date of the project file left-justified into a field of up to 11 characters.

@@@+S
writes the Page number right-justified into a field of up to 5 characters.
```

### 16.4 FBD/LD list

The FBD/LD listing is the printout of the program entered in the FBD/LD editor.

<table>
<thead>
<tr>
<th>Print parameters</th>
<th>EXAMPLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start SP No.___</td>
<td>1</td>
</tr>
<tr>
<td>End SP No.___</td>
<td>1000</td>
</tr>
<tr>
<td>Start SP name___</td>
<td></td>
</tr>
<tr>
<td>End SP name____</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Variables Abs./Symb. (A/S)__</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cross references (Y/N/G/V)___</td>
<td>N</td>
</tr>
<tr>
<td>Represenation (1..4)/(0..1):</td>
<td>1/0</td>
</tr>
<tr>
<td>Limit to DIN-A4 (Y/N)___</td>
<td>Y</td>
</tr>
<tr>
<td>Layout (1..3):</td>
<td>2</td>
</tr>
<tr>
<td>Print with cover sheet (Y/N)</td>
<td>Y</td>
</tr>
<tr>
<td>First page___</td>
<td>1</td>
</tr>
<tr>
<td>Print destin.__</td>
<td>PRN</td>
</tr>
</tbody>
</table>

The cursor moves to the next field when you press the <Enter> key.

You can switch between the individual fields by pressing the cursor keys.

When you press the function key <F1>, the parameters are stored and printing is started.
The parameters can be stored by pressing the function key <F2>.

Inputs in the print menu can be aborted by pressing <Esc>.

**Explanation of parameters**

- **Start SP No. ________**: Number of the segment plan where printing is to begin.
- **End SP No. __________**: Number of the segment plan where printing is to end.
- **Start SP name _______**: Name of the segment plan where printing is to begin.
- **End SP name _______**: Name of the segment plan up to which you wish to print.

**Variables Abs./Symb. (A/S) ____**: Display mode of variables in the FBD/LD list (absolute or symbolic).

**Cross references (Y/N/G/V) ___**: Here you specify whether or not you wish to include cross references to the instruction list in the FBD/LD list. These cross references can only be correct if the FBD/LD program has no longer been modified after translation. „G“ can only be entered for modularized projects. The cross references of the total reference list are output together with module names. In the case of the setting „V“, only the variables without cross references are output under the network.

**Representation (1..4)/(0..1) ____**: You can choose between 4 representation modes:
1: Standard DIN A4 output.
2: Standard output with additional details of each network’s first word number.
3: DIN A3 with details of each network’s first word number. All variables in the extreme left and right columns are printed with symbol or absolute designation and long text.
4: DIN A3 with additional details of each network’s first word number. Each left and right variable of a line is printed with symbol or absolute designation and long text.

You can additionally specify after the slash whether nodes are to be highlighted (1) or not (0).

**Limit to DIN-A4 (Y/N)_________**: Limiting for DIN-A4 printer. Everything that is wider than one line will be truncated.

**Layout (1..3) _______________**: Layout size for printing (see FBD/LD editor). Depending on the set layout, it can happen that variables cannot be printed completely. If this is the case, the variables are marked with a ‘*’.

**Printout with cover sheet (Y/N) __**: Here you specify whether the cover sheet is to be printed or not.

**First page _________________**: Page number of the first page printed. All further pages will each be incremented by one.

**Print destination_____________**: Unit on which you wish to make the output, e.g. printer (PRN), via V24 (LTP1) or to a file.
Example:
Variable output in display mode 3: Mode 3, for display mode 4: Mode 4

Form Feed
Before printing 907 PC 33 checks if the following network can still be printed out on the page which is currently output. If the space on the current page is not sufficient 907 PC 33 automatically issues a form feed command to the printer. This ensures that the complete network fits on one page if the size of the network doesn’t exceed the page size.

A control character for a form feed can be entered in the FBD/LD editor using the command „@FF“ in the first column of the desired separation line as a comment if a form feed is required when printing the FBD/LD (e.g. for structuring or the necessary separation of networks which exceed the page size). The further contents of the form feed lines is ignored when printing. When entering „@FF“ between two networks take care of the fact that all empty lines between „@FF“ and the start of the network are output an the next page. „@FF“ must be placed directly before the network if no empty lines shall be printed out.

16.5 Extended IL list
The extended IL list is a printout of the program entered in the extended IL editor.

<table>
<thead>
<tr>
<th>907 PC 33</th>
<th>Ext. IL printing</th>
<th>EXAMPLE</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Print parameters</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Start SP No.: 1</td>
<td>End SP No.: 1000</td>
<td></td>
</tr>
<tr>
<td>Start SP name:</td>
<td>End SP name:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Limit to DIN-A4 (Y/N): Y</td>
<td>Layout (1..3): 2</td>
<td></td>
</tr>
<tr>
<td>Printout with cover sheet (Y/N): Y</td>
<td>First page: 1</td>
<td></td>
</tr>
<tr>
<td>Print dest: PRN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Start printing at segment plan (1..)</td>
<td>&lt;F1&gt;Store and print parameters</td>
<td>&lt;ESC&gt;Abort</td>
</tr>
</tbody>
</table>
Explanation of the parameters

Start SP No. ________ : Number of the segment plan where printing is to begin.
End SP No. ________ : Number of the segment plan where printing is to end.
Start SP name _______ : Name of the segment plan where printing is to begin.
End SP name _______ : Name of the segment plan up to where you wish to print.
Limit to DIN-A4 (Y/N)_________ : Limiting for DIN-A4 printers. Everything that is wider than one line will be truncated.
Layout (1..3) ________________ : Layout size for printing:
narrow (1), medium (2), wide (3).
Printout with cover sheet (Y/N) ___ : Here you specify whether or not the cover sheet is also to be printed out.
First page _________________ : Page number of the first page printed. All further pages will each be incremented by one.
Print destination ____________ : Unit on which you wish to make the output, e.g. printer (PRN), via V24 (LPT1) or to a file.

16.6 Instruction list

The instruction list is the printout of the instruction list generated by translating the programs created in the Extended IL or FBD/LD editor.

<table>
<thead>
<tr>
<th>Start SP No. ___</th>
<th>End SP No. ___</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9999</td>
</tr>
</tbody>
</table>

Print parameters

<table>
<thead>
<tr>
<th>Print with cover sheet (Y/N)</th>
<th>First page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>1</td>
</tr>
</tbody>
</table>

Print destin. : PRN

Start printing at segment plan (1..)
<F1>Store and print parameters   <F2>Store parameters   <ESC>Abort

Explanation of the Parameters

Start SP No. ________________ : Number of the segment plan where printing is to start.
End SP No. ________________ : Number of the segment plan where printing is to end.
Start SP name _____________ : Name of the segment plan where printing is to start.
End SP name _____________ : Name of the segment plan where printing is to end.
Print with cover sheet (Y/N) ___ : Here you specify whether the cover sheet is also to be printed out or not.
First page _________________ : Page number where printing is to begin.
Print destin. ________________ : Unit on which you wish to make the output, e.g. printer (PRN), via V24 (LPT1) or to a file.
### 16.7 Variable list

The variables entered in the editors appear in this list along with the symbol and long text.

<table>
<thead>
<tr>
<th>Print parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Start variable</strong>:</td>
</tr>
<tr>
<td><strong>Start symbol</strong>:</td>
</tr>
<tr>
<td><strong>Sorting abs./symb. (A/S)</strong>:</td>
</tr>
<tr>
<td><strong>Print destin.</strong> : PRN</td>
</tr>
</tbody>
</table>

#### Explanation of the Parameters

- **Start variable** : Variable where printing is to begin.
- **Start symbol** : Symbol where printing is to begin.
- **End variable** : Variable where printing is to end.
- **End symbol** : Symbol where printing is to end.
- **Sorting abs./symb. (A/S)** : Here you specify whether the variables are to be output, sorted according to the absolute designations or according to symbols.
- **Printout with cover sheet (Y/N)** : Here you specify whether or not the cover sheet is also to be printed out.
- **First page** : Page number where printing is to begin.
- **Print destin.** : Unit on which you wish to make the output, e.g. printer (PRN), via V24 (LTP1) or to a file.
### 16.8 Cross reference list

The variables entered in the editors appear in this list with a reference to all word numbers where they occur.

<table>
<thead>
<tr>
<th>Print parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Start variable</strong>:</td>
</tr>
<tr>
<td><strong>Start symbol</strong>:</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sorting abs./symb. (A/S)</th>
<th>Print type (1..2):</th>
<th>Printout with cover sheet (Y/N):</th>
<th>First page:</th>
<th>Print destin.:</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>Y</td>
<td>1</td>
<td>PRN</td>
</tr>
</tbody>
</table>

- **Start printing at variable**:  
  - `<F1>` Store and print parameters
  - `<F2>` Store parameters
  - `<ESC>` Abort

---

### Explanation of the Parameters

- **Start variable**: 
  - Variable where printing is to begin.
- **Start symbol**: 
  - Symbol where printing is to begin.
- **End variable**: 
  - Variable where printing is to end.
- **End symbol**: 
  - Symbol where printing is to end.
- **Sorting abs./symb. (A/S)**: 
  - Here you specify whether the variables are to be output, sorted according to the absolute designations or according to symbols.
- **Print type (1..2)**: 
  - Unpacked (1) or packed (2) printing is possible.
  - Unpacked (1): For every command, the segment plan name and segment plan number in which this command/this variable is to be found are printed out also.
  - Packed (2): The commands for every variable are summarized; segment plan name and segment plan number are not printed out also.
- **Printout with cover sheet (Y/N)**: 
  - Here you specify whether or not the cover sheet is also to be printed out.
- **First page**: 
  - Page number where printing is to begin.
- **Print destin.**: 
  - Unit on which you wish to make the output, e.g. printer (PRN), via V24 (LPT1) or to a file.
### 16.9 Library

The library list is a list of the current library.

<table>
<thead>
<tr>
<th>Print parameters</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CE selection (A/M) ____ : Print all (A) CEs or only the CEs marked (M) in column B in the library</td>
<td>CE-Name ____ :</td>
</tr>
<tr>
<td>Sort. in CE groups (Y/N) ____ : Output sorted to CE groups</td>
<td>Sort. (A/E) ____ : A</td>
</tr>
<tr>
<td>CE groups (Y/N) ____ : Print CE group branch</td>
<td>Overview (Y/N) ____ : Y</td>
</tr>
<tr>
<td>CE groups (Y/N) ____ : Print CE group branch</td>
<td>IL (Y/N) ____ : Y</td>
</tr>
<tr>
<td>Sort. in CE groups (Y/N) ____ : Sorted in alphabetical order or in entered order</td>
<td>Print with cover sheet (Y/N) ____ : First page ____ : 1</td>
</tr>
<tr>
<td>Overview (Y/N) ____ : Print common CE table</td>
<td>Print destin. ____ : PRN</td>
</tr>
<tr>
<td>Plausibility groups (Y/N) ____ : Print plausibility group overview and setup</td>
<td>Documentation (Y/N) ____ : Print CE with documentation</td>
</tr>
<tr>
<td>Documentation (Y/N) ____ : Print CE with documentation</td>
<td>Graphic (Y/N) ____ : Print CE with graphic</td>
</tr>
<tr>
<td>Graphic (Y/N) ____ : Print CE with graphic</td>
<td>FBD/LD/Comf. AWL source (Y/N) ____ : Print CE with FBD/LD - / Ext. IL source</td>
</tr>
<tr>
<td>FBD/LD/Comf. AWL source (Y/N) ____ : Print CE with FBD/LD - / Ext. IL source</td>
<td>The settings of the print parameters for the FBD/LD printout or the ext. IL printout are used as print parameters. If, for example, absolute variable specification is selected for the 'normal' FBD/LD printout, the variables in the FBD/LD are also given absolutely here (e.g. PP 00).</td>
</tr>
<tr>
<td>IL (Y/N) ____ : In the case of CEs compiled in the FBD/LD, the translated IL of the CE-FBD/LD is also printed out</td>
<td>Print with cover sheet (Y/N) ____ : Here you specify whether or not the cover sheet is also to be printed out.</td>
</tr>
<tr>
<td>Print with cover sheet (Y/N) ____ : Here you specify whether or not the cover sheet is also to be printed out.</td>
<td>First page ____ : The number of the first page is entered in this field.</td>
</tr>
<tr>
<td>First page ____ : The number of the first page is entered in this field.</td>
<td>Print destin. ____ : Device to which the print is to be sent to.</td>
</tr>
</tbody>
</table>

- **Parameters**

- **CE selection (A/M) ____**: Print all (A) CEs or only the CEs marked (M) in column B in the library.
- **Sort. in CE groups (Y/N) ____**: Output sorted to CE groups.
- **Sort. (A/E) ____**: Sorted in alphabetical order or in entered order.
- **CE groups (Y/N) ____**: Print CE group branch.
- **Overview (Y/N) ____**: Print common CE table.
- **Plausibility groups (Y/N) ____**: Print plausibility group overview and setup.
- **Documentation (Y/N) ____**: Print CE with documentation.
- **Graphic (Y/N) ____**: Print CE with graphic.
- **FBD/LD/Comf. AWL source (Y/N) ____**: Print CE with FBD/LD - / Ext. IL source.
- **IL (Y/N) ____**: In the case of CEs compiled in the FBD/LD, the translated IL of the CE-FBD/LD is also printed out.
- **Print with cover sheet (Y/N) ____**: Here you specify whether or not the cover sheet is also to be printed out.
- **First page ____**: The number of the first page is entered in this field.
- **Print destin. ____**: Device to which the print is to be sent to.
16.10 Logic plan diagram

The logic plan is generated on the basis of the existing instruction list.

<table>
<thead>
<tr>
<th>Print parameters</th>
</tr>
</thead>
</table>
| **Start SP No.** : 1  
| **End SP No.** : 9999  |
| **Start SP name** :  
| **End SP name** :  |
| **Variables Abs./Symb. (A/S)** : A  
| **Layout (1..2)** : 1  |
| **Print with cover sheet (Y/N)** : Y  
| **First page** : 1  |
| **Print destin.** : PRN  

Start printing at segment plan (1.. )

<F1>Store and print parameters  
<F2>Store parameters  
<ESC>Abort

**Explanation of the parameters**

- **Start SP No.** : Number of the segment plan where printing is to begin.
- **End SP No.** : Number of the segment plan where printing is to end.
- **Start SP name** : Name of the segment plan where printing is to begin.
- **End SP name** : Name of the segment plan where printing is to end.
- **Variables Abs./Symb. (A/S)** : Here you specify whether the variables should be displayed absolutely or symbolically.
- **Layout (1..2)** : 1: The printout will be produced in DIN A4 size.  
                        2: The printout will be produced in DIN A3 size, with output of the variables (absolute designation, symbolic designation, long text).
- **Printout with cover sheet (Y/N)** : Here you specify whether the cover sheet is also be printed or not.
- **First page** : Page number where printing is to begin.
- **Print destin.** : Unit on which you wish to make the output, e.g. printer (PRN), via V24 (LPT1) or to a file.
16.11 Ladder diagram

The ladder diagram is generated on the basis of the existing instruction list.

### Explanation of the parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start SP No. ________</td>
<td>Segment plan number where printing is to begin.</td>
</tr>
<tr>
<td>End SP No. _________</td>
<td>Segment plan number where printing is to end.</td>
</tr>
<tr>
<td>Start SP name _______</td>
<td>Name of the segment plan where printing is to begin.</td>
</tr>
<tr>
<td>End SP name _______</td>
<td>Name of the segment plan where printing is to end.</td>
</tr>
<tr>
<td>Print with cover sheet (Y/N)____</td>
<td>Here you specify whether the cover sheet is also to be printed out or not.</td>
</tr>
<tr>
<td>First page _________________</td>
<td>Page number where printing is to begin.</td>
</tr>
<tr>
<td>Print destin. ________________</td>
<td>Unit on which you wish to make the output, e.g. printer (PRN), via V24 (LPT1) or to a file.</td>
</tr>
</tbody>
</table>
16.12 Comment list

The comment list provides an overview of all comments existing in the instruction list.

<table>
<thead>
<tr>
<th>Print parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start SP No. ___ : 1</td>
</tr>
<tr>
<td>Start SP name ___</td>
</tr>
</tbody>
</table>

Print with cover sheet (Y/N): Y  First page ___ : 1

Print destin. : PRN

Start printing at segment plan (1..)
<F1> Store and print parameters  <F2> Store parameters  <ESC> Abort

Explanation of the parameters

Start SP No. _______ : Segment plan number where printing is to begin.
End SP No. _______ : Segment plan number where printing is to end.
Start SP name _______ : Name of the segment plan where printing is to begin.
End SP name _______ : Name of the segment plan where printing is to end.
Print with cover sheet (Y/N) _______ : Here you specify whether or not the cover sheet is also to be printed.
First page number ____________ : Page number where printing is to begin.
Print destin. ________________ : Unit on which you wish to make the output, e.g. printer (PRN), via V24 (LPT1) or to a file.
### 16.13 Online list

The variables entered in the online list are printed along with the corresponding comments.

<table>
<thead>
<tr>
<th>Print parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Print with cover sheet ((Y/N)) :</td>
</tr>
</tbody>
</table>

- Also print cover sheet \((Y/N)\)
- \(<\text{F1}>\text{Store and print parameters}\>
- \(<\text{F2}>\text{Store parameters}\>
- \(<\text{ESC}>\text{Abort}\>

### Explanation of the parameters

- **Print with cover sheet \((Y/N)\)**: Here you specify whether or not the cover sheet is also to be printed.
- **First page** : Page number where printing is to begin.
- **Print destin.** : Unit on which you wish to make the output, e.g. printer (PRN), via V24 (LPT1) or to a file.

### 16.14 Text pages

Any chosen text files (ASCII files) can be printed here.

<table>
<thead>
<tr>
<th>Print parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filename : EXAMPLE.TXT</td>
</tr>
</tbody>
</table>

| Print with cover sheet \((Y/N)\) : | First page : 1 |

- Please enter name of file to be printed
- \(<\text{F1}>\text{Store and print parameters}\>
- \(<\text{F2}>\text{Store parameters}\>
- \(<\text{ESC}>\text{Abort}\>

---

16-16

907 PC 33/General Part/Issued: 11.98
Explanation of the Parameters

Print with cover sheet (Y/N) ____ : Here you specify whether or not the cover sheet is also to be printed.

First page _________________ : Page number where printing is to begin.

Print destin. ________________ : Unit on which you wish to make the output, e.g. printer (PRN), via V24 (LPT1) or to a file.

16.15 Data range

The data entered in the data area editor can be printed here.

---

<table>
<thead>
<tr>
<th>Print parameters</th>
<th>EXAMPLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Print with cover sheet (Y/N): Y</td>
<td>First page: 1</td>
</tr>
<tr>
<td>Print destin.: PRN</td>
<td></td>
</tr>
<tr>
<td>Also print cover sheet (Y/N)</td>
<td>Store and print parameters</td>
</tr>
</tbody>
</table>

---

Explanation of the parameters

Print with cover sheet (Y/N) ____ : Here you specify whether or not the cover sheet is also to be printed.

First page _________________ : Page number where printing is to begin.

Print destin. ________________ : Unit on which you wish to make the output, e.g. printer (PRN), via V24 (LPT1) or to a file.

16.16 Main lists

The following three lists (modularization lists, overall variables list and overall reference list) can only be printed with modularized projects.

It is also possible to print the FBD/LD, Extended IL, IL, Variables, Reference list and the Logic and ladder diagram belonging to the overall project from the main project layer.

Please refer to the appropriate specifications of the present chapter for the description of the affiliated masks and keyboard entries.
**16.17 Modularization list**

The data entered in the modularization editor can be printed here.

<table>
<thead>
<tr>
<th>Print parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start pro module: 1</td>
</tr>
<tr>
<td>Start var module: 1</td>
</tr>
<tr>
<td>Matrix (Y/N): Y</td>
</tr>
<tr>
<td>Print with cover sheet (Y/N): Y</td>
</tr>
<tr>
<td>Print destin.: PRN</td>
</tr>
</tbody>
</table>

Start printing at program module

Explanation of the parameters

- **Start pro module**: Number of the program module where printing is to begin.
- **End pro module**: Number of the program module where printing is to end.
- **Start var module**: Number of the variable module where printing is to begin.
- **End var module**: Number of the variable module where printing is to end.
- **Matrix (Y/N)**: Including printing of the matrix (Y/N).
- **With path (Y/N)**: Output module names with path (Y/N).
- **Print with cover sheet (Y/N)**: Here you specify whether or not the cover sheet is also to be printed.
- **First page**: Page number where printing is to begin.
- **Print destin.**: Unit on which you wish to make the output, e.g. printer (PRN), via V24 (LPT1) or to a file.
16.18 Total variable list
This list contains all variables of the overall project (from program modules and variable modules).

<table>
<thead>
<tr>
<th>Print parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start variable:</td>
</tr>
<tr>
<td>Start symbol:</td>
</tr>
<tr>
<td>Sorting Abs./Symb (A/S): A</td>
</tr>
<tr>
<td>Print with cover sheet (Y/N): N</td>
</tr>
<tr>
<td>Print destin.: PRN</td>
</tr>
</tbody>
</table>

Start printing at variable <F1>Store and print parameters <F2>Store parameters <ES>Abort

Explanation of the parameters

Start variable: Variable where printing is to start.
Start symbol: Symbol where printing is to start.
End variable: Variable where printing is to end.
End symbol: Symbol where printing is to end.
Sorting Abs./Symb: Here you specify whether the printout is to be sorted according to variables or symbols.
Layout (1..2): Here you specify whether the printout is to be produced with (2) or without module name (1).
Print with cover sheet (Y/N): You specify whether or not the cover sheet is also to be printed out.
First page: Page number where printing is to begin.
Print destin.: Unit on which you wish to make the output, e.g. printer (PRN), via V24 (LPT1) or to a file.
16.19 Total cross reference list

This list contains all references of the overall project (references for all program modules).

<table>
<thead>
<tr>
<th>Print parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start variable:</td>
</tr>
<tr>
<td>Start symbol:</td>
</tr>
<tr>
<td>Sorting Abs./Symb. (A/S):</td>
</tr>
<tr>
<td>Limit on DIN A4 (Y/N):</td>
</tr>
<tr>
<td>Print with cover sheet (Y/N):</td>
</tr>
<tr>
<td>Print destin.:</td>
</tr>
<tr>
<td>Print type (1...2):</td>
</tr>
<tr>
<td>First page:</td>
</tr>
</tbody>
</table>

**Explanation of the parameters**

- **Start variable**: Variable where printing is to begin.
- **Start symbol**: Symbol where printing is to begin.
- **End variable**: Variable where printing is to end.
- **End symbol**: Symbol where printing is to end.
- **Sorting Abs./Symb. (A/S)**: Here you specify whether the printout is to be sorted according to variables or symbols.
- **Print type (1...2)**: Unpacked (1) or packed (2) printing is possible.
  - Unpacked (1): For every command, the segment plan name and segment plan number in which this command / this variable is to be found are printed out also.
  - Packed (2): The commands for every variable are summarized; segment plan name and segment plan number are not printed out also.
- **Limit on DIN A4 (Y/N)**: If you print with layout 2, i.e. stating the program module names, only the segment plan number is printed, limited to DIN A4 (Y) and print type 1.
  - To obtain the segment plan name and number an the printout with layout 2 and print type 1, it is necessary to use a DIN A3 printer.
- **Layout (1..2)**: Here you specify whether the printout is to be produced without (1) or with (2) module name.
- **Print with cover sheet (Y/N)**: Here you specify whether or not the cover sheet is also to be printed.
- **Print destin.**: Page number where printing is to begin.
17 Batch processing

17.1 Principle

By means of the integrated batch processing facility, all inputs normally made on the keyboard can be read from a file. Thus, for instance, it is possible to print several lists successively without having to make inputs on the keyboard for each new list. To achieve this, all necessary keyboard inputs must be written to a so-called batch file beforehand.

When starting the program, the batch file is started from the initial screen by entering:

```
PLC name batch file name
```
or by entering <ALT>255 (The numbers must be entered on the separate digit keypad).

The batch file can be created by means of the incorporated text editor, or also using any other text editor.

It successively contains the characters that would otherwise have to be entered on the keyboard.

Control characters are entered as three characters preceded by a „\“ in „\xxx“ format.

„xxx“ stands for the ASCII code of the required character (see table of special characters overleaf).

If a line contains „\255...“, the commands that follow only apply internally to the batch process (see table of extended functions overleaf).

Example of a batch file with the name „EDIT.BA“ which calls a project with the name „PROJECT“ and then calls the instruction editor:

```
\255 Skip initial screen
\027
\255 Enter project name in the project data editor
\1008
\017\025
PROJECT\013
\255 Enter password
asdl013
\255 Terminate configuration
\027
\255 Call edit menu in the main menu
E
\255 Call instruction List
4
```

The batch process can be stopped by pressing <Ctrl>-S, it can be continued by pressing <Ctrl>-Q and can be aborted by pressing <Ctrl>-C.

In the batch file, the \ character assumes a special function:

```
\xxx The ASCII code xxx will be read on input
     (see table of special characters).
\ \ The character \ will be read
\255 Everything up to the end of the line will be treated as a comment (see table of extended functions).
```

17.2 Special characters/functions

Table of special characters and their meanings

<table>
<thead>
<tr>
<th>\xxx</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>\001</td>
<td>&lt;F1&gt; etc.</td>
</tr>
<tr>
<td>\002</td>
<td>&lt;F10&gt;</td>
</tr>
<tr>
<td>\003</td>
<td>&lt;Shift&gt;&lt;F1&gt; etc.</td>
</tr>
<tr>
<td>\004</td>
<td>&lt;Shift&gt;&lt;F10&gt;</td>
</tr>
<tr>
<td>\005</td>
<td>&lt;Ctrl&gt;&lt;F1&gt; etc.</td>
</tr>
</tbody>
</table>

907 PC 33/General Part/Issued: 11.98
Table of extended functions

Extended functions can now be addressed by means of „\255“:

\255P1 ... \255P8 A short (1) or long (8) pause will be inserted depending on the argument.
\255L1 ... \255L8 As in the case of \255P, a pause will be inserted, but the ONLINE status will also be displayed.
\255P9 The program will wait until the user has pressed a key.
The user can abort the batch by pressing <ESC>.
\255+ The lowermost line (25th line) will be saved.
\255- The lowermost line (25th line) will be fetched back from the data backup.
\255>Text The corresponding text will be output in line 25.
\255<File2 The batch will not be run from file2.

Each of these options can be followed by a comment. Introduce simple comments with „\255<Space bar>“.

17.3 Background recording

It is possible to record all processing operations for a project or a specific subarea by means of a batch file.
The file can be called anywhere by pressing the keystroke combination <Alt>254. The name of the file in which you wish to record the batch operations must then be entered.
If the file name is preceded by the character „@“ (e. g. @example.ba), this file will be closed after every write operation. Accordingly, if a system crash should occur, the recorded data will remain stored up to the last character.
Recording is terminated at the end of the program or by entering <Alt>254. If you end recording by pressing <Alt>254, the „STOP ?“ prompt will appear. Recording can be terminated immediately by pressing the <Enter> key. Alternatively, a text can be entered that appears in the batch file as a comment. If the text is preceded by a „@“, the text will be adopted completely so that control characters can also be entered. Recording is continued after input of the text.
A batch file generated in this way can be modified using the incorporated text editor or also using any other text editor.
18 Terminal mode

The „Terminal mode“ part of the program enables communication through the serial interface with other computers or with ABB Procontic/Advant Controller equipment without having to terminate the programming and testing system 907 PC 33.

After you have selected this program part, the file „PLC name.TER“ will first of all be loaded and the serial interface will be initialized corresponding to the contents of this file.

An error message will appear if this file should not yet exist (see configuration - terminal mode) or if initialization of the serial interface should not be possible for any other particular reasons.

After completion of initialization, the screen will be deleted and following activities will be executed successively within a loop:

1. When a character is received through the interface, it will be displayed.
2. If a character is entered on the keyboard, it will be sent through the interface. If echo mode is set for the parameters, the character will additionally be displayed.

The <F1> key can be pressed to interrupt this loop and thus terminate terminal mode.
### 19 Command overview

#### 19.1 Help/displayable character/cursor control/block commands

<table>
<thead>
<tr>
<th>Input</th>
<th>Function (brief description)</th>
<th>Futaba PGM 1907 PC 33/General Part/Issued: 11.98</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;F10&gt;</td>
<td>Help</td>
<td>E</td>
</tr>
<tr>
<td>Displayable</td>
<td>Character is displayed and transferred to the internal field buffer. The comment editor is called.</td>
<td>x</td>
</tr>
<tr>
<td>character</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>&lt;Cursor left&gt;</td>
<td>Cursor one position to the left; left margin: scrolling.</td>
<td>X</td>
</tr>
<tr>
<td>&lt;Backspace&gt;</td>
<td>Cursor one position to the left; Character beforehand is deleted;</td>
<td>X</td>
</tr>
<tr>
<td>&lt;Cursor right&gt;</td>
<td>Cursor one position to the right; right margin: scrolling</td>
<td>X</td>
</tr>
<tr>
<td>&lt;Cursor up&gt;</td>
<td>Cursor one line up; top margin: scrolling</td>
<td>X</td>
</tr>
<tr>
<td>&lt;Cursor down&gt;</td>
<td>Cursor down by one line; bottom margin: scrolling</td>
<td>X</td>
</tr>
<tr>
<td>&lt;Enter&gt;</td>
<td>Acknowledgment of a line. Scrolling up by one line; Line feed.</td>
<td>X</td>
</tr>
<tr>
<td>&lt;Return&gt;</td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

**Abbreviations**

Modularization editor
Function block diagram/ladder diagram editor
Extended instruction list editor
Variable editor
Text editor
Connection element library
Connection element editor
Field editor
## Cursor control

<table>
<thead>
<tr>
<th>Input</th>
<th>Function (brief description)</th>
<th>MOB</th>
<th>DX T O / E L I D D</th>
<th>FEX A R I E T E L I B</th>
<th>C E E L D E</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;PgDn&gt;</td>
<td>Page down.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;Home&gt;</td>
<td>The cursor moves to the start of the file.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;Pos1&gt;</td>
<td>The cursor moves to the first line in the segment plan or, if the cursor is already in the first line, then to the first line of the previous segment plan. Cursor to the start of the line and then to the start of the page and to the start of the file.</td>
<td></td>
<td></td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>&lt;Ctrl&gt;-End</td>
<td>The cursor moves to the end of the line/end of the field.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;Ctrl&gt;-Home</td>
<td>The cursor moves to the start of the line/ end of the field.</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;End&gt;</td>
<td>The cursor moves to the end of the file.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;Ctrl&gt;-Q-R</td>
<td>The cursor moves to the last line in the segment plan or, if the cursor is already in the last line, then to the last line in the next segment plan. Cursor to the end of the line, then end of the page, then end of the file.</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## Block commands

<table>
<thead>
<tr>
<th>Input</th>
<th>Function (brief description)</th>
<th>MOB</th>
<th>DX T O / E L I D D</th>
<th>FEX A R I E T E L I B</th>
<th>C E E L D E</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;Ctrl&gt;-K-B</td>
<td>Mark start of block.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;Ctrl&gt;-K-D</td>
<td>Saves the screen contents to ASCII file (projectname.BLD)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;Ctrl&gt;-K-K</td>
<td>Mark end of block.</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;Ctrl&gt;-K-H</td>
<td>Delete block marking.</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;Ctrl&gt;-K-V</td>
<td>Move block.</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;Ctrl&gt;-K-C</td>
<td>Copy block.</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;Ctrl&gt;-K-Y</td>
<td>Delete block. Delete all unused variables in the block.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;Ctrl&gt;-K-W</td>
<td>Save block to hard disk.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;Ctrl&gt;-K-R</td>
<td>Load block from hard disk.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### 19.2 Searching / searching and replacing / inserting / deleting / comments

#### Searching

| Input          | Function (brief description)                                                                 | M | O | B | D | X | E | D | V | E | R | E | X | T | E | D | F | L | E | D |
| <Ctrl>-Q-F-S  | Search for sentence number. Search for unused variables.                                    | x |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| <Ctrl>-Q-F-W  | Search for word number.                                                                     |   | x |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| <Ctrl>-Q-F-A  | Search for variables.                                                                      | x | x |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| <Ctrl>-Q-F-O  | Search for symbols.                                                                        | x | x |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| <Ctrl>-Q-F-B  | Search for command with and without operand.                                               | x |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| <Ctrl>-Q-F-Z  | Search for line number.                                                                    |   | x |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| <Ctrl>-L      | Repeat search.                                                                              | x | x | x |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| <Shift>+<F3>  | Search through segment plan for name/number (segment plan management).                    |   | x |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| <Shift>+<F8>  | Search for CE.                                                                              | x | x |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| <Shift>+<F9>  | Search for non-allocated connection.                                                        | x | x |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| <Shift>+<F7>  | Search for non-absolutely defined variable field.                                          |   | x |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |

#### Searching and replacing

| Input          | Function (brief description)                                                                 | M | O | B | D | X | E | D | V | E | R | E | X | T | E | D | F | L | E | D |
| <Ctrl>-Q-A-A  | Search for and replace variables within a block.                                            | x | x |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| <Ctrl>-Q-A-W  | Search for and replace variable ranges.                                                      | x | x | x |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| <Ctrl>-Q-A-O  | Search for and replace symbols                                                               | x | x |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| <Ctrl>-<F3>   | Search for and replace CE.                                                                   | x | x |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |

#### Inserting

| Input          | Function (brief description)                                                                 | M | O | B | D | X | E | D | V | E | R | E | X | T | E | D | F | L | E | D |
| <Ins>         | Activate insert mode. Insert column. Setting a serial contact                               | x | x | x | x | x | x | x |   |   |   |   |   |   |   |   |   |   |   |   |
| <Ctrl>-V      | Activate insert mode in the field editor; otherwise same as Ins.                            | x | x | x | x | x | x | x | x |   |   |   |   |   |   |   |   |   |   |   |
| <Ctrl>-N      | Insert line, permanent insert mode.                                                          | x | x | x | x | x | x | x | x |   |   |   |   |   |   |   |   |   |   |   |
### Deleting

<table>
<thead>
<tr>
<th>Input</th>
<th>Function (brief description)</th>
<th>M</th>
<th>O</th>
<th>B</th>
<th>D</th>
<th>E</th>
<th>D</th>
<th>T</th>
<th>R</th>
<th>E</th>
<th>L</th>
<th>I</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;Delete&gt;</code></td>
<td>The character at the cursor's position is deleted.</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;Ctrl&gt;</code>-G</td>
<td></td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;F6&gt;</code></td>
<td>Delete comment line.</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;Ctrl&gt;</code>-T</td>
<td>As from the cursor's position, delete all characters up to field end.</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;Ctrl&gt;</code>-Q-Y</td>
<td>As from the cursor's position, delete all characters up to the end of line.</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;Ctrl&gt;</code>-Y</td>
<td>Delete line.</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;Ctrl&gt;</code>-B</td>
<td>Delete up to end of line.</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;Ctrl&gt;</code>-J</td>
<td>Delete marking.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Comments

- Comment texts must be preceded by a semicolon (;).  
- Comments are placed at the end of the sentence.

| `<F5>` | Insert comment line. | x |
| `<F6>` | Delete comment line. | x |
# 19.3 Special commands / ONLINE commands

## Special commands

<table>
<thead>
<tr>
<th>Input</th>
<th>Function (brief description)</th>
<th>M</th>
<th>F</th>
<th>E</th>
<th>D</th>
<th>I</th>
<th>L</th>
<th>T</th>
<th>E</th>
<th>R</th>
<th>E</th>
<th>D</th>
<th>F</th>
<th>L</th>
<th>E</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;Esc&gt;</td>
<td>Abort field editing. The field's previous contents are left unchanged. Terminate the editor.</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>&lt;Spacebar&gt;</td>
<td>Display menu window.</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>&lt;Ctrl&gt;-Z</td>
<td>Jump to the single line variable editor.</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>&lt;F1&gt;</td>
<td>Definition of allowed input fields. Delete column.</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>&lt;F2&gt;</td>
<td>Definition of the cursor's start position. Insert column.</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>&lt;F4&gt;</td>
<td>ONLINE on/off</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>&lt;F9&gt;</td>
<td>Variable/symbol input switchover.</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>&lt;Shift&gt;&lt;F1&gt;</td>
<td>Divide segment plan.</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>&lt;Shift&gt;&lt;F2&gt;</td>
<td>Merge segment plans.</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>&lt;Shift&gt;&lt;F5&gt;</td>
<td>CE selection.</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>&lt;Shift&gt;&lt;F6&gt;</td>
<td>Error file evaluation.</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>&lt;Ctrl&gt;&lt;F3&gt;</td>
<td>Display the compiled IL.</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>&lt;Ctrl&gt;-Q-W</td>
<td>Create all horizontal connections on the corresponding side of a CE.</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>&lt;Ctrl&gt;-W</td>
<td>Conversion of number bases.</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Key sequence</td>
<td>Function (Brief description)</td>
<td>Function available as menu call</td>
<td>F E V B D R T / E L I D D L</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------</td>
<td>--------------------------------------------------------------------------------------------</td>
<td>---------------------------------</td>
<td>-----------------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;Alt&gt;-1</td>
<td>Translate and transmit program changes **)</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;Alt&gt;-2</td>
<td>Send program changes</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;Alt&gt;-3</td>
<td>Send program *)</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;Alt&gt;-4</td>
<td>Compare program *)</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;Alt&gt;-5</td>
<td>Send constants *) (only valid for 907 PC 331)</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;Alt&gt;-6</td>
<td>Compare constants *) (only valid for 907 PC 331)</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;Alt&gt;-7</td>
<td>Reactivate old program</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;Alt&gt;-8</td>
<td>Replace variable (only valid for 907 PC 332)</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;Alt&gt;-S</td>
<td>Start program *)</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;Alt&gt;-A</td>
<td>Abort program *)</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;Alt&gt;-H</td>
<td>Stop program</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;Alt&gt;-C</td>
<td>Continue program</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;Alt&gt;-D</td>
<td>Online list set to ONLINE</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>V</td>
<td>Variable status</td>
<td>x</td>
<td>xxx</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>Status trigger to variable</td>
<td>x</td>
<td>xxx</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;Alt&gt;-M</td>
<td>Cancel status trigger variable</td>
<td>x</td>
<td>xxx</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>V</td>
<td>Status trigger to time</td>
<td>x</td>
<td>xxx</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>O (Null)</td>
<td>Freeze status on screen</td>
<td>x</td>
<td>xxx</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>Status of selected variables On/Off</td>
<td>x</td>
<td>xxx</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>Read and display PLC status</td>
<td>x</td>
<td>xxx</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Collect for status of selected variables</td>
<td>x</td>
<td>xxx</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Adopt status to ONLINE list</td>
<td>x</td>
<td>xxx</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z</td>
<td>Single cycle On/Off</td>
<td>x</td>
<td>xxx</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>Single step On/Off</td>
<td>x</td>
<td>xxx</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>Jog</td>
<td>x</td>
<td>xxx</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>ONLINE list editor</td>
<td>x</td>
<td>xxx</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Collect for ONLINE list</td>
<td>x</td>
<td>xxx</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L</td>
<td>Enter breakpoint list</td>
<td>x</td>
<td>xxx</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Collect for breakpoints</td>
<td>x</td>
<td>xxx</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>Set breakpoints single</td>
<td>x</td>
<td>xxx</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>W</td>
<td>Set breakpoints all</td>
<td>x</td>
<td>xxx</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;Alt&gt;-W</td>
<td>Delete breakpoints all</td>
<td>x</td>
<td>xxx</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q</td>
<td>Display breakpoint</td>
<td>x</td>
<td>xxx</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P</td>
<td>Breakpoint pursuit On/Off</td>
<td>x</td>
<td>xxx</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Collect for overwrite</td>
<td>x</td>
<td>xxx</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>U</td>
<td>Overwrite single</td>
<td>x</td>
<td>xxx</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R</td>
<td>Overwrite all</td>
<td>x</td>
<td>xxx</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Collect for forcing</td>
<td>x</td>
<td>xxx</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>Force singly</td>
<td>x</td>
<td>xxx</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;Alt&gt;-F</td>
<td>Delete force singly (only valid for 907 PC 331)</td>
<td>x</td>
<td>xxx</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>Force all</td>
<td>x</td>
<td>xxx</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;Alt&gt;-E</td>
<td>Delete force all</td>
<td>x</td>
<td>xxx</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;Alt&gt;-D</td>
<td>Display force values</td>
<td>x</td>
<td>xxx</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X</td>
<td>Switch over Bit/Word</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>Switch over value representation dec./hex./oct.; menu will be displayed.</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*) Functions are also available on main menu level; no effect as hotkey.
**) Function can only be carried out if ONLINE is switched off.
### 19.4 Function keys

<table>
<thead>
<tr>
<th>Mod editor</th>
<th>FBD/LD</th>
<th>EXTENDED IL</th>
<th>Variable editor</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>F1</strong></td>
<td>Edit PM/VM descr. Layout menu</td>
<td>Delete column</td>
<td>CE param. inv. Layout menu</td>
</tr>
<tr>
<td>&lt;Shift&gt;-F1</td>
<td>&lt;Ctrl&gt;-F1</td>
<td>Divide SP</td>
<td>Divide SP</td>
</tr>
<tr>
<td>&lt;Alt&gt;-F1</td>
<td>Layout menu</td>
<td>[—] (—) —</td>
<td>Layout menu</td>
</tr>
<tr>
<td><strong>F2</strong></td>
<td>Edit VM/PM descr.</td>
<td>Insert column</td>
<td>Define inp. field</td>
</tr>
<tr>
<td>&lt;Shift&gt;-F2</td>
<td>&lt;Ctrl&gt;-F2</td>
<td>Merge SP</td>
<td>Conf.var.search</td>
</tr>
<tr>
<td>&lt;Alt&gt;-F2</td>
<td>[—] [—]</td>
<td>Merge SP</td>
<td></td>
</tr>
<tr>
<td><strong>F3</strong></td>
<td>Translate</td>
<td>Translate</td>
<td>&quot;small upd.&quot; VM</td>
</tr>
<tr>
<td>&lt;Shift&gt;-F3</td>
<td>Search/Replace CE</td>
<td>Search/Replace CE</td>
<td></td>
</tr>
<tr>
<td>&lt;Ctrl&gt;-F3</td>
<td>displ. trans1.IL</td>
<td>displ. trans1.IL</td>
<td></td>
</tr>
<tr>
<td>&lt;Alt&gt;-F3</td>
<td>[—] [—]</td>
<td>[—] [—]</td>
<td></td>
</tr>
<tr>
<td><strong>F4</strong></td>
<td>ONLINE ON/OFF</td>
<td>ONLINE ON/OFF</td>
<td>ONLINE on/off</td>
</tr>
<tr>
<td>&lt;Shift&gt;-F4</td>
<td>Save project</td>
<td>Save project</td>
<td>ext var&gt;loc PM</td>
</tr>
<tr>
<td>&lt;Ctrl&gt;-F4</td>
<td>Module change</td>
<td>Module change</td>
<td></td>
</tr>
<tr>
<td>&lt;Alt&gt;-F4</td>
<td>ONLINE ON/OFF</td>
<td>ONLINE ON/OFF</td>
<td></td>
</tr>
<tr>
<td><strong>F5</strong></td>
<td>Set reference</td>
<td>Ins. comment line</td>
<td>Set block mark.</td>
</tr>
<tr>
<td>&lt;Shift&gt;-F5</td>
<td>CE select. menu</td>
<td>CE select. menu</td>
<td></td>
</tr>
<tr>
<td>&lt;Ctrl&gt;-F5</td>
<td>Scroll speed</td>
<td>Scroll speed</td>
<td></td>
</tr>
<tr>
<td>&lt;Alt&gt;-F5</td>
<td>[—] [—]</td>
<td>[—] [—]</td>
<td></td>
</tr>
<tr>
<td><strong>F6</strong></td>
<td>Delete reference</td>
<td>Delete comm. line</td>
<td>Rev. block mark.</td>
</tr>
<tr>
<td>&lt;Shift&gt;-F6</td>
<td>Error display</td>
<td>Error display</td>
<td></td>
</tr>
<tr>
<td>&lt;Ctrl&gt;-F6</td>
<td>Cont. var.add.dis</td>
<td>Cont. var.add.dis</td>
<td></td>
</tr>
<tr>
<td>&lt;Alt&gt;-F6</td>
<td>[—] [—]</td>
<td>[—] [—]</td>
<td></td>
</tr>
<tr>
<td><strong>F7</strong></td>
<td>PM/VM file names</td>
<td>Comment editor</td>
<td>Alternative var.</td>
</tr>
<tr>
<td>&lt;Shift&gt;-F7</td>
<td>se. abs.def.Var.</td>
<td>se. abs.def.Var.</td>
<td></td>
</tr>
<tr>
<td>&lt;Ctrl&gt;-F7</td>
<td>Display node</td>
<td>Display node</td>
<td></td>
</tr>
<tr>
<td>&lt;Alt&gt;-F7</td>
<td>Alternative var.</td>
<td>Alternative var.</td>
<td></td>
</tr>
<tr>
<td><strong>F8</strong></td>
<td>Library</td>
<td>Library</td>
<td>Adopt var.</td>
</tr>
<tr>
<td>&lt;Shift&gt;-F8</td>
<td>Search for CE</td>
<td>Search for CE</td>
<td></td>
</tr>
<tr>
<td>&lt;Ctrl&gt;-F8</td>
<td>CE documentation</td>
<td>CE documentation</td>
<td></td>
</tr>
<tr>
<td>&lt;Alt&gt;-F8</td>
<td>CE with terminal</td>
<td>CE with terminal</td>
<td></td>
</tr>
<tr>
<td><strong>F9</strong></td>
<td>Toggle sym/abs.</td>
<td>se. alloc.CE ter</td>
<td>Sort. var/abs</td>
</tr>
<tr>
<td>&lt;Shift&gt;-F9</td>
<td>se. alloc.CE ter or. col. on/off</td>
<td>se. alloc.CE ter or. col. on/off</td>
<td></td>
</tr>
<tr>
<td>&lt;Ctrl&gt;-F9</td>
<td>or. col. on/off</td>
<td>or. col. on/off</td>
<td></td>
</tr>
<tr>
<td>&lt;Alt&gt;-F9</td>
<td>Help</td>
<td>Help</td>
<td></td>
</tr>
<tr>
<td><strong>F10</strong></td>
<td>Help</td>
<td>Help</td>
<td></td>
</tr>
<tr>
<td>&lt;Shift&gt;-F10</td>
<td>SP management</td>
<td>SP management</td>
<td></td>
</tr>
<tr>
<td>&lt;Ctrl&gt;-F10</td>
<td>Add.crusr. on/off</td>
<td>Add.crusr. on/off</td>
<td></td>
</tr>
<tr>
<td>&lt;Alt&gt;-F10</td>
<td>Help</td>
<td>Help</td>
<td></td>
</tr>
<tr>
<td>Mod editor</td>
<td>FBD/LD</td>
<td>Extended IL</td>
<td>Variable editor</td>
</tr>
<tr>
<td>------------</td>
<td>--------</td>
<td>-------------</td>
<td>----------------</td>
</tr>
<tr>
<td>F1</td>
<td>Delete column</td>
<td>Inv. CE param.</td>
<td>Define inp.field</td>
</tr>
<tr>
<td></td>
<td>Insert column</td>
<td>Translate</td>
<td>Define star.fiel</td>
</tr>
<tr>
<td>F2</td>
<td>Translate ONLINE ON/OFF</td>
<td>ONLINE ON/OFF</td>
<td>&quot;small Update&quot;</td>
</tr>
<tr>
<td>F3</td>
<td>ONLINE ON/OFF</td>
<td>Set block mark.</td>
<td>ONLINE ON/OFF</td>
</tr>
<tr>
<td>F4</td>
<td>Library</td>
<td>Res. block mark.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Help</td>
<td>Alternative var.</td>
<td></td>
</tr>
<tr>
<td>F5</td>
<td>Help</td>
<td>Adopt var.</td>
<td></td>
</tr>
<tr>
<td>F6</td>
<td>Help</td>
<td>Sort. var/ abs</td>
<td></td>
</tr>
<tr>
<td>F7</td>
<td>Help</td>
<td>Help</td>
<td></td>
</tr>
<tr>
<td>F8</td>
<td>Help</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F9</td>
<td>Help</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F10</td>
<td>Help</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<Shift>-F1  Edit PM/VM des.  Divide SP  Divide SP  Var.: ext->loc
<Shift>-F2  Edit VM/PM des.  Merge SP  Merge SP |
<Shift>-F3  |                  |                  |
<Shift>-F4  |                  |                  |
<Shift>-F5  |                  |                  |
<Shift>-F6  |                  |                  |
<Shift>-F7  |                  |                  |
<Shift>-F8  |                  |                  |
<Shift>-F9  |                  |                  |
<Shift>-F10 |                  |                  |

<table>
<thead>
<tr>
<th>Ctrl</th>
<th>-F1</th>
<th>Layout menu</th>
<th>Layout menu</th>
<th>Conf. var.search</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>-F2</td>
<td>displ. trans.IL</td>
<td>displ. trans.IL</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-F3</td>
<td>change module</td>
<td>change module</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-F4</td>
<td>scroll speed</td>
<td>scroll speed</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-F5</td>
<td>Add.disp.con.var</td>
<td>Add.disp.con.var</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-F6</td>
<td>Node display</td>
<td>Node display</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-F7</td>
<td>CE documentation</td>
<td>CE documentation</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-F8</td>
<td>Or. col. on/off</td>
<td>Or. col. on/off</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-F9</td>
<td>Add.crsr. on/off</td>
<td>Add.crsr. on/off</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-F10</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Alt -F1  CE with connection
20 Guide to time-optimized working

20.1 Introduction

The programming system offers various options for translating the programs written in programming languages FBD/LD or extended IL to the IL commands.

Programs which use no CEs in the extended IL do not need to be translated.

The various translation strategies differ as regards conversion of the programs to the IL commands and transfer of the program to the PLC.

The procedures differ dependent upon the capabilities of the PLC and can thus be described only in principle below.

In order to optimize the times for a change in the program, it is virtually always the case that only the program parts which have changed are translated and transferred to the PLC.

The following translation modes are available:

- Changes
- Changes and reserves
- All

If changes are made in the program, it is not necessary to translate the entire program. Rather, it is necessary to translate only the segment plans which have changed. This means that, when transferring the program, it is not necessary to transfer the entire program to the PLC either. This achieves a substantial timesaving as compared with translation of the complete program.
20.2 Translate changes (without reserve)

If changes are to be made in an existing project, it is possible to work with translator mode „CHANGES“ PLC-specifically without setting up reserves beforehand when translating the program to the IL commands.

In this case, only the segment plans in which something has changed are translated. This method of translation is the default method for translation of the programs to the IL commands.

If a change necessitates not only the changed segment plans being translated, the programming system signals this and performs the required translation automatically. The mode of translation is changed automatically if:

- a new project has been entered
- an old project has been transferred
- a change has been made in the data area editor
- a change has been made in the method of issuing flags (local, global)

Important:
If a change of a CE has been made in the library, this cannot be detected automatically. You must use „TRANSLATE ALL“.

The diagram below shows the procedure as the program becomes longer. If the PLC is not capable of shifting program ranges, the entire program must be transferred to the PLC.
20.3 Translate changes with reserve

This translator mode has two functions:

2.1 Insertion of reserves in an existing project.
   Reserves must be inserted in the program in order to achieve time-optimized working. This can be implemented with this translator mode after the corresponding commands have been inserted in the program.

2.2 Extending existing reserves.
   If existing reserves are to be extended or reduced, it is possible to process these changes with the „CHANGES WITH RESERVE“ translator mode.

In all the above-specified cases, everything has to be transferred again to the PLC with the ABB Procontic T200.

Creating reserves is described for the program editors in chapter „TRANSLATOR“.

The following diagram shows the procedure for inserting reserves.

```
FBD/LD or Extended IL  |  Segment plan-translated IL commands  |  PLC program in the PLC

segment plan 1  |  segment plan 1  |  ---------------
segment plan 2  |  ⇒  NOP reserve  |  ---------------
segment plan 3  |  ⇒  segment plan 2  |  ⇒  NOP reserve
⇒  NOP reserve  |  ⇒  segment plan 3  |  ⇒  NOP reserve
⇒  segment plan 4  |  ⇒  -----------------  |  ⇒  ---------------
                    |  ⇒  -----------------  |  ⇒  NOP reserve
                    |  ⇒  -----------------  |  ⇒  ---------------
                    |  ⇒  -----------------  |  ⇒  ---------------
```

907 PC 33/General Part/Issued: 11.98
20.4 Translate changes

If reserves have been inserted in the program, all changes can be processed with this translator mode. Since, now, only the changed segment plans are translated and, when transferring the program, only the changed segments are transferred, this method of working is the fastest method for all possible changes.

The reserves are automatically created anew if they are used up. The diagram below shows the procedure for inserting the changes.
20.5 Translate all

In the case of „TRANSLATE ALL“ translator mode, all segment plans are translated completely. In this case, the entire program must then be transferred to the PLC. This method of translation must be selected in the following cases:

1. If a change of a CE has been made in the library.
2. If all automatic flags are to be issued anew.

20.6 Modularized projects

Translation strategies for modularized projects

In the case of modularized projects, the translator modes described above act at program module level in precisely the same way as with a non-modularized project. Differences only relate to transfer of the program to the PLC.

Translate changes (without reserve)

If no reserves have been created, the function of „TRANSLATE CHANGES“ translator mode is identical as regards generation of the IL commands, both in the case of nonmodularized projects and in the case of modularized projects.

The procedure for transfer of a program to the PLC differs, dependent upon PLC.

If the PLC is not capable of shifting program ranges, you must transfer the entire project with menu option „TRANSFER PROGRAM“.
If the PLC is capable of shifting program ranges, you must transfer with "SEND CHANGES".

In this case, only the program part of the program module is transferred to the PLC.

The diagram below illustrates the procedure for transfer of changes of a program module to a PLC which is capable of shifting program ranges (ABB Procontic CS31 and Advant Controller 31).

Program module 1, the area shown with dashed lines, is increased, in size as the result of a change at the end of the first segment plan. The code of the program module is transferred to the PLC from these change through to the end of the module. The other modules 2 to 4, shown by the grid area, are shifted in the PLC.
**Translate changes with reserve**

The function of „CHANGES WITH RESERVE“ translator mode is the same in relation to generation of the IL commands, both in the case of non-modularized projects and in the case of modularized projects.

The procedure for transfer of a program to the PLC differs, dependent upon the capability of the PLC.

If the PLC is not able to shift program ranges, you must transfer the entire project to the PLC with menu option „SEND PROGRAM“.

If the PLC is capable of shifting PLC program ranges, you must transfer with „SEND CHANGES“.

In this case, only the code of the program module is transferred to the PLC.

The procedure for creating reserves is described in chapter Modularization.

The procedure is as shown in the previous diagram, but the requested reserves are also created for the segment plans in the program module.

**Translate changes**

The function of „TRANSLATE CHANGES“ translator mode is the same as regards generation of the IL commands, both in the case of non-modularized projects and in the case of modularized projects.

In this mode, you must transfer with „SEND CHANGES“. In this case, only the changes in the program module are transferred to the PLC.
The diagram below shows the procedure for inserting the changes.

**Translate all / reset flag assignment**

The function of „TRANSLATE ALL“ translator mode is essentially the same as regards generation of the IL commands, both in the case of non-modularized projects and in the case of modularized projects. The program must then be transferred to the PLC with „SEND PROGRAM“.

In the case of modularized projects which use the data range of the overall project level, you must perform a „RESET FLAG ASSIGNMENT“ at overall project level before „TRANSLATE ALL“.

This function releases the automatically assigned intermediate flags so that they can be assigned again.

This method of translation must be selected in the following cases:
1. If a change of a CE has been made in the library.
2. If all automatic flags are to be reassigned.

The diagram below shows the procedure for transfer of a modularized project which is to be transferred for the first time to the PLC. In this case, all program modules must be translated completely first and then transferred once to the PLC.
Appendix

1 Initialization of the laser printer HP LaserJet 4

The initialization file INIT.HP4 is used to initialize the HP LaserJet 4. Four options of the standard factory setting are changed with this file.

Esc & 1 1 0 Alignment = Landscape format
Esc & 1 4.06 C Vertical motion index (determines the page length)
Esc ( 10 U Character set = PC-8
Esc ( $ 20.00 H Character density = 20.00 characters per inch

The following files can be found in the path ABB-SPS\AC31:

- README.HP4 Description of the files INIT.HP4 and REINIT.HP4
- INIT.HP4 Initialization of the printer
- REINIT.HP4 Reinitialization of the printer

Enter the files as follows in the screen mask General Print Parameters (menu Print param. -> Print Parameters):

<table>
<thead>
<tr>
<th>907 PC 33</th>
<th>General print parameteres</th>
<th>EXAMPLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Page length : 86</td>
<td>Left margin : 10</td>
<td></td>
</tr>
</tbody>
</table>

Papersave : N

Convers. file : \ABB-SPS\CS31\INIT.HP4
Re-init. file : \ABB-SPS\CS31\REINIT.HP4

Please enter required page length (default: 72)
<F1> Store parameters  <ESC> Abort
We also recommend the following settings in the screen mask __FBD/LD Print/Print parameters__ for a FBD/LD printout (menu __Print 1 -> FBD/LD__):

<table>
<thead>
<tr>
<th>Print parameters</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Start SP No.____: 1</td>
<td>End SP No.____: 1000</td>
</tr>
<tr>
<td>Start SP name__:</td>
<td>End SP name__:</td>
</tr>
<tr>
<td>Variables Abs./Symb. (A/S)___: *</td>
<td></td>
</tr>
<tr>
<td>Cross references (Y/N/G/V)_______: *</td>
<td></td>
</tr>
<tr>
<td>Representation (1..4)/(0..1): 4 / 1</td>
<td></td>
</tr>
<tr>
<td>Limit to DIN-A4 (Y/N)__: Y</td>
<td>Layout (1..3)__: 2</td>
</tr>
<tr>
<td>Print with cover sheet (Y/N)__: *</td>
<td>First page__: 1</td>
</tr>
<tr>
<td>Print destin.__: PRN</td>
<td></td>
</tr>
</tbody>
</table>

Start printing at segment plan (1...)  
<F1>Store and print parameters  
<F2>Store parameters  
<ESC>Abort

* Setting arbitrary

On the following page you will find an example of a FBD/LD printout.
Teilplan: 1  Wort: 0
S 020,00 INITABL

Teilplan: 1  Wort: 4
S 020,01 INITABL
KD 01,02 SCHL.ZEIT
HILFSM.

Teilplan: 1  Wort: 9
M 040,07 HILFSM.

Teilplan: 1  Wort: 13
S 020,02 INITABL

Teilplan: 1  Wort: 17
M 014,02 HILFSM.
EDS 85,0
EDS 85,1
EDS 85,1
E 05,00
E 05,01
E 05,01
HANP
AUTO
AUTO

Einschaltung Hand/Automatik

ABB
ASEA BROWN BOVERI

Projektbez.: Testausdruck für HP LaserJet 4
Bearbeiter: He ABB/SST
Letzte Änderung: 02.02.94

TEST
Seite A-3
2 Initialization of Kyocera laser printers

The following files are available (under ABB-SPS\AC31) for initializing Kyocera laser printers:

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INIT-A4.KYO</td>
<td>Initialization of the printer for A4 format</td>
</tr>
<tr>
<td>INIT-A3.KYO</td>
<td>Initialization of the printer for A3 format</td>
</tr>
<tr>
<td>KONVERT.KYO</td>
<td>Conversion file</td>
</tr>
<tr>
<td>REINIT.KYO</td>
<td>Reinitialization of the printer</td>
</tr>
</tbody>
</table>

Enter the files in the screen mask **General Print Parameters** as follows (menu Print param. -> Print parameter):

<table>
<thead>
<tr>
<th>Page length</th>
<th>Left margin</th>
</tr>
</thead>
<tbody>
<tr>
<td>88</td>
<td>0</td>
</tr>
</tbody>
</table>

Papersave: N

**Convers. file**: \ABB-SPS\CS31\KONVERT.KYO

**Initial. file**: \ABB-SPS\CS31\INIT-A4.KYO

**Re-init. file**: \ABB-SPS\CS31\REINIT.KYO

Please enter required page length (default: 72)

<F1> Store parameters  <ESC> Abort

Recommended settings for a FBD printout -> see page A-2.

On the following Page you will find an example of a FBD/LD list printout.
Teilplan 1: INIT UND ZENTRALE

Teilplan: 1  Wort: 0
S 020,00  INITABL---INITABL

Teilplan: 1  Wort: 4
160000
S 020,01  INITABL---ESV
KÖ 01,02  SCHL.ZEIT---THO

Teilplan: 1  Wort: 9
M 040,07  HILFSM.---INITABL

Teilplan: 1  Wort: 13
S 020,02  INITABL---INIT-ENDE

Teilplan: 1  Wort: 17
M 014,02  Einschaltung Hand/Automatik
805 E5,0  E 05,00  HAND---SR
805 E5,1  E 05,01  AUTO---S
805 E5,1  E 05,01  AUTO---Q
R  Q  AUTO
Index

A

A3 format, ......................................................................................................................... 16-7
absolute variable display, .............................................................................................. 10-26
alternative variables, ..................................................................................................... 9-17

B

background recording, ................................................................................................... 17-2
batch processing, .......................................................................................................... 17-1
baudrate, ......................................................................................................................... 6-2
block command, ........................................................................................................... 10-5, 11-5, 19-1, 19-2
block functions, ............................................................................................................ 15-3
block marking, ............................................................................................................. 10-6, 10-48-11-5, 11-18
block processing, ......................................................................................................... 10-48, 11-18
blocks, .......................................................................................................................... 10-6
bottom-up program creation, ....................................................................................... 1-14, 9-6

C

call of the file overview, ............................................................................................... 8-3
CE cache configuration, ............................................................................................... 15-13
CE library, ..................................................................................................................... 15-1
CE without parameter, ................................................................................................. 15-14
change module, ............................................................................................................ 9-14
change, last, .................................................................................................................... 5-2
channel, ......................................................................................................................... 6-2
checking the flag ranges, ............................................................................................. 9-22
code reserve, .................................................................................................................. 11-21
color configuration, ....................................................................................................... 6-3
comment line, ............................................................................................................... 10-44, 11-11
comment list, ................................................................................................................ 16-15
comments, ...................................................................................................................... 11-11, 19-3, 19-4
connection elements, ................................................................................................... 15-1
conversion of numbers, ............................................................................................... 8-2
copying, ......................................................................................................................... 10-6, 11-6, 11-18
create project, ............................................................................................................... 9-9
creation of references, ................................................................................................ 9-13
creation of the graphics, ............................................................................................. 15-7
cross reference list, ....................................................................................................... 16-11
cross references, .......................................................................................................... 16-7

data area editor, ............................................................................................................ 6-1, 9-21
data area, ....................................................................................................................... 9-21
data backup, ................................................................................................................... 1-16, 3-1, 11-21
data ranges, ...................................................................................................................... 9-20, 16-17
delete block marking, .................................................................................................. 10-5, 11-6
delete block, .................................................................................................................. 10-5, 10-48, 11-6, 11-18, 15-3
delete CE connection, ................................................................................................. 10-14
delete CE, ................................................................. 10-15, 11-14
delete columns, ..................................................... 9-13, 10-43
delete connection, ................................................. 10-40
delete line, ......................................................... 9-13, 10-41, 11-10
delete unused variables in the block, .................. 12-3
deleting, ................................................................... 19-4
deletion of a line/column, .......................................... 9-13
deletion, ..................................................................... 1-4, 1-6
DIN A4, DIN A3, ....................................................... 16-7
direct CE output, ...................................................... 10-30
directory structure, .................................................. 2-2
display CE documentation, ...................................... 11-12
display of a CE, ......................................................... 10-9
display password, .................................................... 2-5
DOS commands, ...................................................... 4-2
DOS shell, ............................................................... 4-2
duplicate CEs, .......................................................... 15-37
duplication of CE connections, ......................... 10-13, 11-13
duplication, ......................................................... 10-13, 11-13
echo, ........................................................................... 6-3
edit existing VM/PM, .............................................. 9-13
editing keys (data area editor), .................. 6-1, 9-22
editing keys (modularization editor), ................ 9-14
editor (FBD/LD), ..................................................... 15-1
end of block, ....................................................... 10-6, 10-48, 11-6, 11-18
end SP name, .......................................................... 16-7
end SP No., ............................................................... 16-7
extended IL editor, .................................................. 11-1
extended IL, ............................................................. 1-10, 1-11

duplication, ......................................................... 10-13, 11-13

echo, ........................................................................... 6-3
edit existing VM/PM, .............................................. 9-13
editing keys (data area editor), .................. 6-1, 9-22
editing keys (modularization editor), ................ 9-14
editor (FBD/LD), ..................................................... 15-1
end of block, ....................................................... 10-6, 10-48, 11-6, 11-18
end SP name, .......................................................... 16-7
end SP No., ............................................................... 16-7
extended IL editor, .................................................. 11-1
extended IL, ............................................................. 1-10, 1-11

F
FBD/LD list, .......................................................... 16-6
FBD/LD, ................................................................. 1-18, 1-11, 10-1
field editor, ............................................................. 8-1, 11-10
fields, ........................................................................ 15-2
file overview, .......................................................... 8-3
flags, ......................................................................... 6-1
footer lines, ............................................................. 16-3, 16-4
function block diagram, ........................................ 10-1
function keys, ......................................................... 9-14, 19-7

global flags, ........................................................... 15-19
global/local, ......................................................... 11-20
| H   | header lines, .................................................................................................................. 16-3, 16-4 |
|     | help, .......................................................................................................................... 19-1 |
|     | hierarchical levels, ....................................................................................................... 2-3 |
|     | ........................................................................................................................................|
| I   | input field, ................................................................................................................... 5-2, 12-2 |
|     | input modes, ................................................................................................................... 8-1 |
|     | insert mode, ................................................................................................................... 11-10 |
|     | inserting a PM designation, ............................................................................................ 9-12 |
|     | inserting a VM designation, ............................................................................................. 9-12 |
|     | inserting, ........................................................................................................................ 10-18, 10-41, 10-43, 11-10, 19-3 |
|     | installation, ................................................................................................................... 2-1 |
|     | inverting, ....................................................................................................................... 10-12, 10-25, 11-13 |
| J   | jump labels, .................................................................................................................... 6-1 |
| K   | knots, ............................................................................................................................ 10-41 |
| L   | ladder diagram, ............................................................................................................... 10-1, 16-14 |
|     | last modification, .......................................................................................................... 5-3 |
|     | layout, ........................................................................................................................... 9-11, 10-7, 11-7 |
|     | library directory, .......................................................................................................... 2-3 |
|     | library name, ................................................................................................................... 5-2 |
|     | library, .......................................................................................................................... 16-12 |
|     | line number, .................................................................................................................... 11-9 |
|     | local flags, ..................................................................................................................... 15-21 |
|     | logic plan diagram, ......................................................................................................... 16-13 |
|     | long text, ....................................................................................................................... 11-9, 12-2, 12-5, 12-6 |
| M   | main lists, ...................................................................................................................... 16-17 |
|     | main memory, .................................................................................................................. 1-15 |
|     | management, ................................................................................................................... 10-4, 11-4 |
|     | manufacturer CEs, ............................................................................................................ 15-1 |
|     | margin, ............................................................................................................................ 16-2 |
|     | mark block, ................................................................................................................... 10-48, 11-18, 15-3 |
|     | mark, .............................................................................................................................. 10-1 |
|     | menu control, ................................................................................................................. 10-4, 11-4 |
|     | menu interface, ............................................................................................................... 4-1 |
|     | menu layers, .................................................................................................................... 9-8 |
|     | modular programming, .................................................................................................... 1-2, 1-5 |
|     | modularization editor, .................................................................................................... 9-11 |
|     | modularization list, ......................................................................................................... 16-17 |
modularization, ................................................................. 1-3
modularized projects, .................................................. 20-5
module change, ............................................................ 9-14
module configuration, ................................................. 9-19
module management level, ............................................ 1-6
module processing, ...................................................... 1-6, 9-4
module reserve, ............................................................ 11-21
mouse, ...................................................................... 1-16
move block, ................................................................. 10-6, 11-6, 11-18
MS-DOS file CONFIG.SYS, ......................................... 1-16
multiuser program creation, ........................................... 1-13

N
nested duplication, ........................................................ 15-17
non-modular programming, ......................................... 1-2, 1-4
number conversion, ..................................................... 8-2

O
ONLINE commands, overview, ................................. 19-6
ONLINE list, ................................................................. 16-16
ONLINE, ..................................................................... 11-10
operation of menus, .................................................... 9-11
orientation column, .................................................... 10-9
overall project, ............................................................ 1-6, 1-7, 9-1

P
parameters which can be modified, .................. 5-1, 6-2
parameters which cannot be modified, ............ 5-1
parameters, capable of duplication, .................. 15-16
parameters, not capable of duplication, .......... 15-15
parity, ....................................................................... 6-3
password editor, .......................................................... 2-5
password in case of project changeover, ............ 5-1
password input, ........................................................... 2-5
password, deletion of, ................................................ 2-5
path, ......................................................................... 2-3
plaus. group, default „All“ .......................................... 15-8
plausibility check, ....................................................... 10-48, 11-12
plausibility group editor, .......................................... 15-9
PLC type, ................................................................... 9-1, 9-4
print format editor, ..................................................... 16-3
print parameters, ....................................................... 16-2
printing, ..................................................................... 16-1
process VM, PM, ........................................................ 9-13
program creation, ....................................................... 1-1
program input, ............................................................ 1-3, 1-10
program module function selection, .................. 9-1
program module, ....................................................... 1-3, 1-6, 1-7, 1-8, 1-9, 9-1, 9-4, 9-15
program run (creation with extended IL), .......... 15-37

Index-4
907 PC 33/General Part/Issued: 11.98
program run (creation with FBD/LD editor), ................................................................. 15-33
program structuring, ........................................................................................................ 1-2
program test, ..................................................................................................................... 1-3, 1-12
programming unit, ............................................................................................................ 1-15
project data, ......................................................................................................................... 5-1, 16-1
project description, ............................................................................................................. 5-3
project designation, ............................................................................................................. 5-3
project directory, .................................................................................................................. 2-3, 9-8
project, ................................................................................................................................ 5-2
project, structure of, ............................................................................................................. 9-3

Q
quit (exit), .............................................................................................................................. 10-49, 11-19

R
read block, ............................................................................................................................ 11-18, 15-3
read block from harddisk, .................................................................................................... 11-6
read manufacturer library, .................................................................................................... 15-39
references, creation of, ....................................................................................................... 9-13
replace variables within a range, .......................................................................................... 10-46
replace, .................................................................................................................................. 10-46, 11-16, 19-3
reserve configuration, ......................................................................................................... 9-19
reserves, ................................................................................................................................. 10-52, 11-21
reset flag assignment, .......................................................................................................... 20-8
responsible, ............................................................................................................................ 5-3
restoring, ................................................................................................................................ 3-3

S
scroll speed, ........................................................................................................................... 11-1
search and replace variables, .............................................................................................. 10-46, 11-16
search for variables, ............................................................................................................. 9-17
search menu, ......................................................................................................................... 10-7, 11-7, 11-15
search, .................................................................................................................................... 19-3
segment plan, ......................................................................................................................... 1-4, 1-5, 1-7, 9-3, 10-5 10-21, 11-4, 11-20
single line variable editor, ................................................................................................... 9-18, 12-4
sorting, ................................................................................................................................... 12-4
split program, ....................................................................................................................... 9-9
splitting, .................................................................................................................................. 10-47, 11-17
start of block, ......................................................................................................................... 10-6, 10-48, 11-6, 11-18
start SP name, ....................................................................................................................... 16-6
start SP No., ........................................................................................................................... 16-6
start, ....................................................................................................................................... 2-6
status line, ................................................................................................................................ 11-9
store block on harddisk, ........................................................................................................ 11-6
store block, ................................................................................................................................ 11-18
symbol, .................................................................................................................................. 11-9, 12-6
<table>
<thead>
<tr>
<th>Term</th>
<th>Page Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td></td>
</tr>
<tr>
<td>terminal operation</td>
<td>6-2, 18-1</td>
</tr>
<tr>
<td>text constant editor</td>
<td>13-1</td>
</tr>
<tr>
<td>text editor</td>
<td>14-1</td>
</tr>
<tr>
<td>text pages</td>
<td>16-16</td>
</tr>
<tr>
<td>time-optimized working</td>
<td>20-1</td>
</tr>
<tr>
<td>timeout</td>
<td>5-3</td>
</tr>
<tr>
<td>total cross reference list</td>
<td>16-20</td>
</tr>
<tr>
<td>total variable list</td>
<td>16-19</td>
</tr>
<tr>
<td>translate all</td>
<td>20-5, 20-8</td>
</tr>
<tr>
<td>translate changes (with reserve)</td>
<td>20-2, 20-5</td>
</tr>
<tr>
<td>translate changes (without reserve)</td>
<td>20-2, 20-7</td>
</tr>
<tr>
<td>translate changes</td>
<td>20-4, 20-5</td>
</tr>
<tr>
<td>translator (compiler)</td>
<td>10-49, 11-19</td>
</tr>
<tr>
<td>U</td>
<td></td>
</tr>
<tr>
<td>update</td>
<td>9-25</td>
</tr>
<tr>
<td>user library (CEs)</td>
<td>15-1</td>
</tr>
<tr>
<td>V</td>
<td></td>
</tr>
<tr>
<td>variable block transfer</td>
<td>9-17, 9-18</td>
</tr>
<tr>
<td>variable list</td>
<td>16-16</td>
</tr>
<tr>
<td>variable module</td>
<td>1-9, 1-10, 9-1, 9-4, 9-17, 9-18</td>
</tr>
<tr>
<td>variable selection</td>
<td>9-18</td>
</tr>
<tr>
<td>variable</td>
<td>9-26, 12-2, 12-5</td>
</tr>
<tr>
<td>VM</td>
<td>9-1, 9-4</td>
</tr>
<tr>
<td>W</td>
<td></td>
</tr>
<tr>
<td>word number</td>
<td>11-9, 11-16</td>
</tr>
<tr>
<td>write Block</td>
<td>15-3</td>
</tr>
</tbody>
</table>
ABB Procontic Programming System

907 PC 33 General Part

AJP5 2040 00 AJP5 2040 00 AJP5 2040 00 AJP5 2040 00 AJP5 2040 00

ABB Procontic Programming System

907 PC 33 General Part

AJP5 2040 00 AJP5 2040 00 AJP5 2040 00 AJP5 2040 00 AJP5 2040 00

ABB Procontic Programming System

907 PC 33 General Part