

rho 3

PIC Programming Software manual



Version

102



BOSCH
Automationste

rho 3

PIC Programming Software manual

1070 072 159-102 (97.07) GB



Reg. Nr. 16149-03

© 1995-1997

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

Discretionary charge xx.– DM



Safety information and additional literature

Before you start working with PIC programming, we recommend that you thoroughly familiarise yourself with the contents of this manual. Keep this manual in a place where it is always accessible to all users.

Standard operation

This instruction manual presents a comprehensive set of instructions and information required for the standard operation of the described products. The products described hereunder are used in conjunction with PIC programming a rho 3.0 controller.

The products described hereunder

- were developed, manufactured, tested and documented in accordance with the relevant safety standards. In standard operation, and provided that the specifications and safety instructions relating to the project phase, installation and correct operation of the product are followed, there should arise no risk of danger to personnel or property.
- are certified to be in full compliance with the guidelines on electromagnetic compatibility (89/336/EEG, 93/68/EEG and 93/44/EEG). Specific compliance with harmonized standards EN 50081-2 and EN 50082-2 is hereby certified.
- are designed for operation in an industrial environment. Prior to the intended installation and/or operation within a private residence or business area, on retail premises or in a small-industry setting, the user will be required to obtain a single operating license issued by the appropriate national authority or approval body. In Germany, this is the Federal Institute for Posts and Telecommunications, and/or its local branch offices.

Qualified personnel

This instruction manual is designed for specially trained PLC personnel. The relevant requirements are based on the job specifications as outlined by the German Electrical and Electronics Manufacturers' Association (ZVEI). Please refer to the following German-language publication:

Weiterbildung in der Automatisierungstechnik

Hrsg.: ZVEI und VDMA

MaschinenbauVerlag

Postfach 71 08 64

60498 Frankfurt

This instruction manual addresses RC project designers.

Interventions in the hardware and software of our products which are not described in this instruction manual may only be performed by our skilled personnel.

Unqualified interventions in the hardware or software or non-compliance with the warnings listed in this instruction manual or indicated on the product may result in serious personal injury or damage to property.

Qualified personnel are persons who

- as **planning personnel**, are familiar with the safety guidelines used in electrical engineering and automation technology.
- as **operating personnel**, are familiar with the equipment used in the field of automation technology and are thus familiar with the operating instructions in this manual.
- as **commissioning personnel**, are authorized to commission, ground and classify electric circuits and devices/systems in accordance with the relevant safety standards.

Safety instructions on the control components

The following warnings and notices may be indicated on the control components themselves and have the following meaning:



Danger: High voltage!



Danger: Battery acid!



Electrostatically-sensitive components!



Disconnect at mains before opening!



Pin for connecting PE conductor only!



This connection for functional earthing or low-noise earth only!



For screened conductor only!

**Safety instructions in this manual**

These symbols are used throughout this manual subject to the following conditions.

**DANGER**

This symbol is used to warn of the presence of **dangerous electrical current**. Insufficient or lacking compliance with these instructions can result in **personal injury**.

Safety instructions accompanied by this symbol are serially numbered, for example 0.1. The appendix provides translations of the safety notes shown here in all the official EC languages.

**DANGER**

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

Safety instructions accompanied by this symbol are serially numbered, for example 0.1. The appendix provides translations of the safety notes shown here in all the official EC languages.

**CAUTION**

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

Safety instructions accompanied by this symbol are serially numbered, for example 0.1. The appendix provides translations of the safety notes shown here in all the official EC languages.



This symbol is used to inform the user of special features.

Amendments

Amendments to this instruction manual vis-à-vis edition 101 are indicated by means of a vertical, black line.

Symbols used



This sign shows that the manual is describing an activity which you have to perform, e.g.:

- ★ Insert disk 1 into the floppy disk drive.



We would greatly appreciate any contributions to improve this manual. If you have any suggestions, please fill out the page provided at the end of this manual.



Safety instructions

**DANGER****0.1****Danger to persons and equipment!****Test every new program before operating the system!****CAUTION****0.2****Danger to the module!****Do not insert or remove the module when the control is switched on! This can destroy the module. Switch off or remove the power supply module of the control, external power supply and signal voltage before inserting or removing the module!****CAUTION****0.3****Danger to the module!****All ESD protection measures must be observed when using the module! Avoid electrostatic discharges!**

Observe the following protective measures for electrostatically endangered modules (EEM)!

- The employees responsible for the storage, transport and handling must be trained in ESD protection.
- EEMs must be stored and transported in the protective packaging specified.
- EEMs may basically only be handled at special ESD work places set up specifically for this purpose.
- Employees, work surfaces and all devices and tools, which could come into contact with EEMs must be same potential (e.g. earthed).
- Wear an approved earthing strap around your wrist. The grounding bracelet must be connected via a cable with integrated 1 M Ω resistance with the work surface.
- EEMs may on no account come into contact with chargeable objects, these include most plastics.
- When inserting EEMs into devices and removing them, the power source of the device must be switched off.

Contents

	Page	
1	Introduction	1–1
2	PIC Versions	
2.1	PIC250 module in rho 3.1, w/ 64 inputs, 40 outputs, 0.2 A	2–1
2.2	PIC Input / Output Connections	2–2
2.3	PIC Signals and direct signals	2–3
2.4	Running SoftPIC on the rho 3.0	2–4
3	PIC Instructions	
3.1	Bit Operations	3–1
3.2	Time / Counter Operations	3–1
3.3	Operand Labelling	3–1
3.4	BAPS-PIC250 Instruction Set (Standard)	3–2
3.5	BAPS-PIC250 Instruction Set Extension in Version 200 .	3–3
3.5.1	New Program Structure (Version 200)	3–3
3.5.2	Previous Program Structure (Version 100)	3–3
3.5.3	Combination Instruction	3–4
3.5.4	No-operation Instruction	3–4
3.5.5	INCLUDE Instruction	3–4
3.6	BAPS-PIC250 Instruction Set Extensions in Version 200 in conjunction with ROPS Version W3C	3–5
3.6.1	Channel Numbers for Temporary Markers	3–5
3.6.2	Declaration of Constants	3–5
3.6.3	Standard Constants	3–7
3.6.4	New Standard Types	3–7
3.6.5	ARRAY Declaration	3–8
3.6.6	Introduction of Byte and/or Word Instructions	3–9
3.6.7	Multiple Assignment	3–9
3.6.8	Compare Operators	3–10

	Page
3.6.9	Line Numbers for Lines of Commentary 3–10
3.6.10	Backward Jump Warning 3–11
3.6.11	Language-dependent Code 3–11
4	PIC Program Generation
4.1	Using BAPS to generate a PIC Program 4–1
4.2	Programming Tools 4–2
4.3	Editor 4–3
4.4	ONLINE Connectivity Application 4–4
4.5	PIC250 Development System 4–5
5	PIC, CL300 & CL400 Address Ranges 5–1
5.1	Marker Assignment 5–2
6	Examples
6.1	Programming Example – AND 6–1
6.2	Programming Example – OR 6–2
6.3	Programming Example – NOT 6–3
6.4	Programming Example – AND before OR 6–4
6.5	Programming Example – OR before AND 6–5
6.6	Programming Example – Set / Reset 6–6
6.7	Programming Example – Conditional Jump 6–7
6.8	Programming Example – Positive Transition 6–8
6.9	Programming Example – Negative Transition 6–9
6.10	Programming Example – TIME (Raising Delay) 6–10
6.11	Programming Example – TIME (Falling Delay) 6–11
6.12	Programming Example – Counter 6–12
6.13	Programming Example – Trigger Pulse 6–14
6.14	Programming Example – Cycling ON/OFF w/ Pushbutton .. 6–15
6.15	Programming Example – Flasher 6–16
6.16	Programming Example – Byte / Word Instructions .. 6–18
6.17	Programming Example – Fields 6–19
6.18	Programming Example – Compare Operations 6–20



	Page	
7	PIC & PROFI Software	
7.1	Loading and Saving with PROFI Software	7-1
7.2	Installing the Batch Files	7-2
7.3	Operation from within the Main Menu	7-3
7.4	Standard PIC Programs for PROFI Software	7-4
8	Bit Coupler	
8.1	Bit Coupling – PIC Implementation in External PLC .	8-1
8.2	Settings on the rho 3	8-2
8.3	PLC Settings	8-3
8.4	Bit Coupler in I/O Field	8-3
8.5	Bit Coupler in EI/EO Fields	8-3
8.6	Special Considerations – Bit Coupler in the CL400 ..	8-4
9	rho 3.0 <--> PLC Interface	
9.1	Introduction	9-1
9.2	KOMFIFO Module for rho 3.0 / PLC Communications	9-2
9.2.1	KOMFIFO Module Parameters:	9-2
9.2.2	DBLOAD Module Parameters:	9-2
9.2.3	Exchanging Interface Signals with the PLC	9-3
9.2.4	Example of a rho 3.0 <--> SPS Data Exchange	9-4
9.3	Start-up Characteristics with Several rho 3.0 in the PLC .	9-5
9.4	rho 3.0 in Decentralized Application	9-6
10	Menu Tree / Interface Programs	
10.1	Excerpt, PHG Menu Tree	10-1
10.2	Table of available Interface Programs	10-2

	Page
A	Appendix
A.1	Abbreviations A-1
A.2	SPS-Terminologie Deutsch/Englisch A-2
A.3	BAPS-Terminologie A-10
A.4	Safety instructions A-13
A.4.1	Dansk A-13
A.4.2	Deutsch A-14
A.4.3	Ελληνικά A-15
A.4.4	English A-17
A.4.5	Español A-18
A.4.6	Français A-20
A.4.7	Italiano A-21
A.4.8	Nederlands A-23
A.4.9	Português A-25
A.4.10	Suomi A-27
A.4.11	Svenska A-28

Illustrations

Fig.		Page
1-1	rho3 – Interface	1-1
2	Input / Output variants (without external PLC)	2-2
3	PIC Input / Output signals	2-3
4	SoftPIC for rho3.0	2-4
5	rho3.0 and SPS Interface	9-1
6	Data Transfer via KOMFIFO Module	9-3

1 Introduction

PIC Programmable Interface Controller

The **PIC250** programmable logic controller is used for *controlling machine-typical sequences* and links.

Independent of the type of links, the machine signals must simply be applied directly to the input and output terminals of the PIC250.

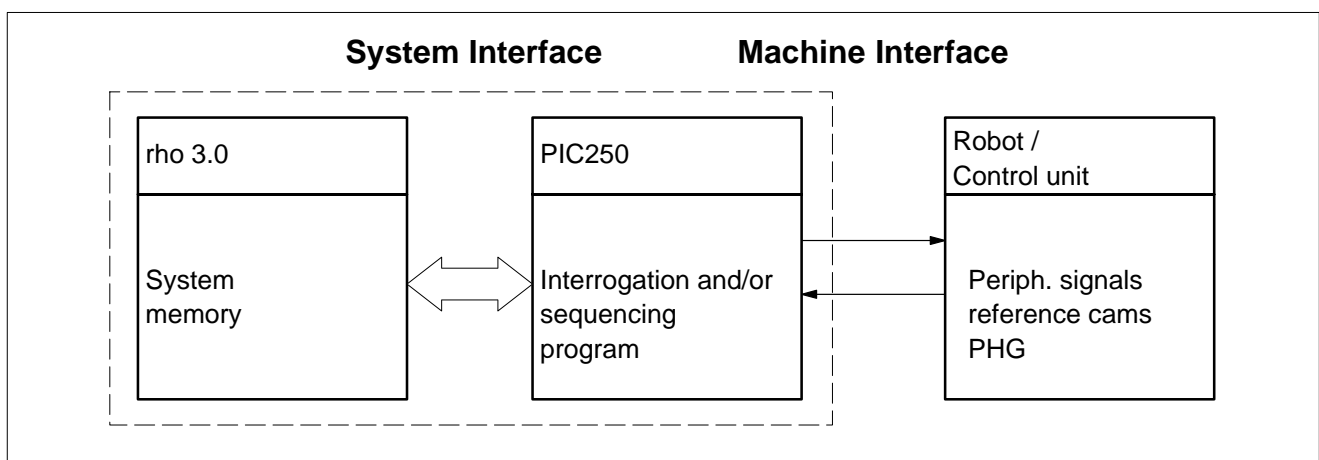


Fig. 1-1 rho3 - Interface

Functioning as the mediator between controller and machine, the PIC250 features an internal interface with the rho (system interface), and an external interface with the machine.

The pin assignment of the **system interface** with the rho is permanently designated by the rho operating system (refer to *Signal Description and Fault Messages*, part no. 1070 073028).

The **machine interface** is provided by the machine tool manufacturer by means of cable connections. The signal type is determined by the linking program.

Essentially, the command reservoir of the PIC encompasses the following functions:

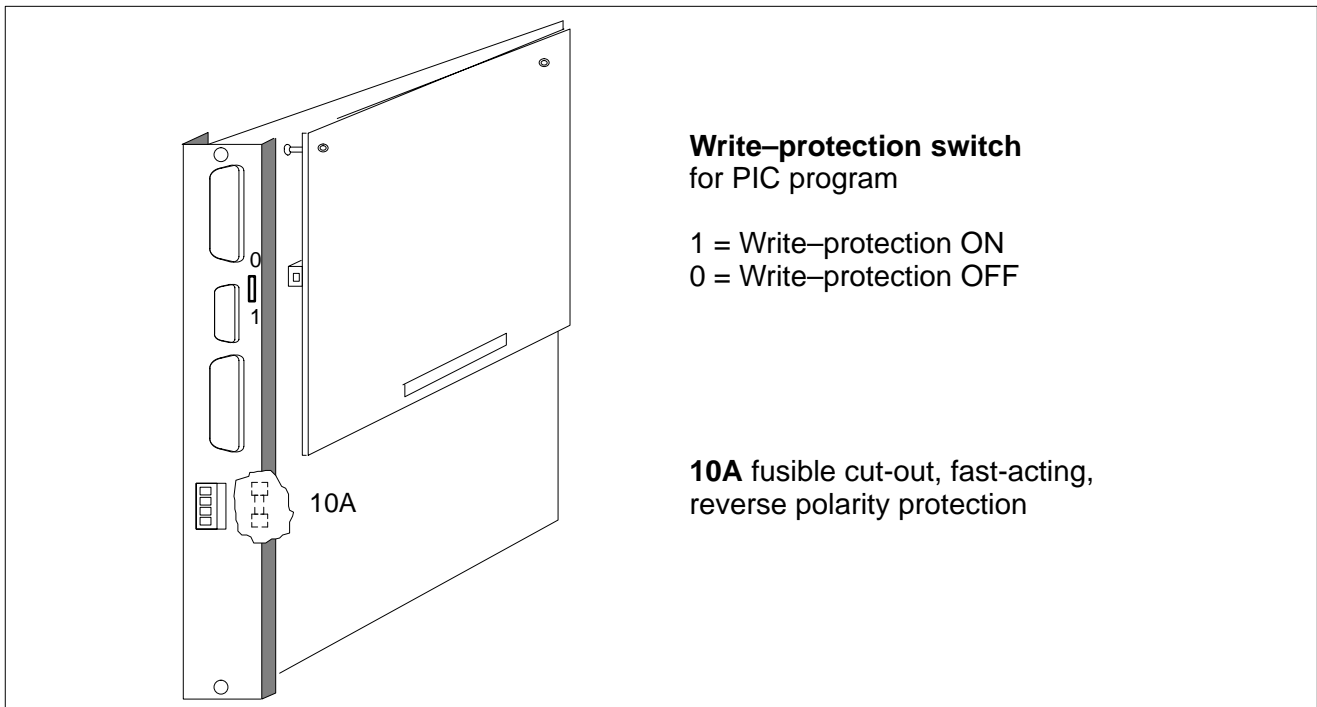
- binary links
- memory functions
- timer and counter function

Personal Notes:



2 PIC Versions

2.1 PIC250 module in rho 3.1, w/ 64 inputs, 40 outputs, 0.2 A



The PIC250 program does not influence the main RAM area because it is permanently stored in an EEPROM module.

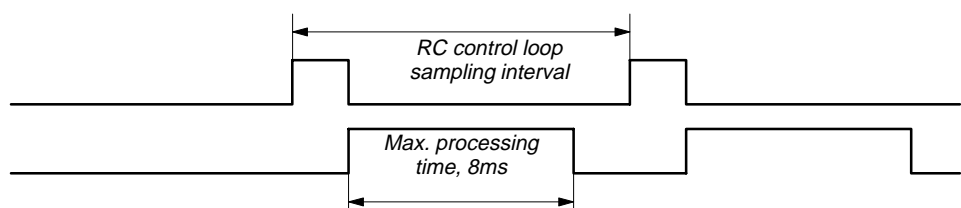
EEPROM = Electrically Erasable and Programmable Read-Only Memory module, combining the advantages of RAM and EPROM modules. A buffer/backup battery is therefore not required.

Prior to EEPROM programming, the write-protection must be disabled. On completion of programming, write-protection should again be enabled.

The program can be loaded, output and/or modified by means of **ROPS** on an MS-DOS computer.

The PIC250 module utilizes the clock timing-based process I/O image (= sampling intervals of RC control loops)

The maximum program length encompasses 8k instructions, with the internal processing in the PIC250 module being 8 ms.



2.2 PIC Input / Output Connections

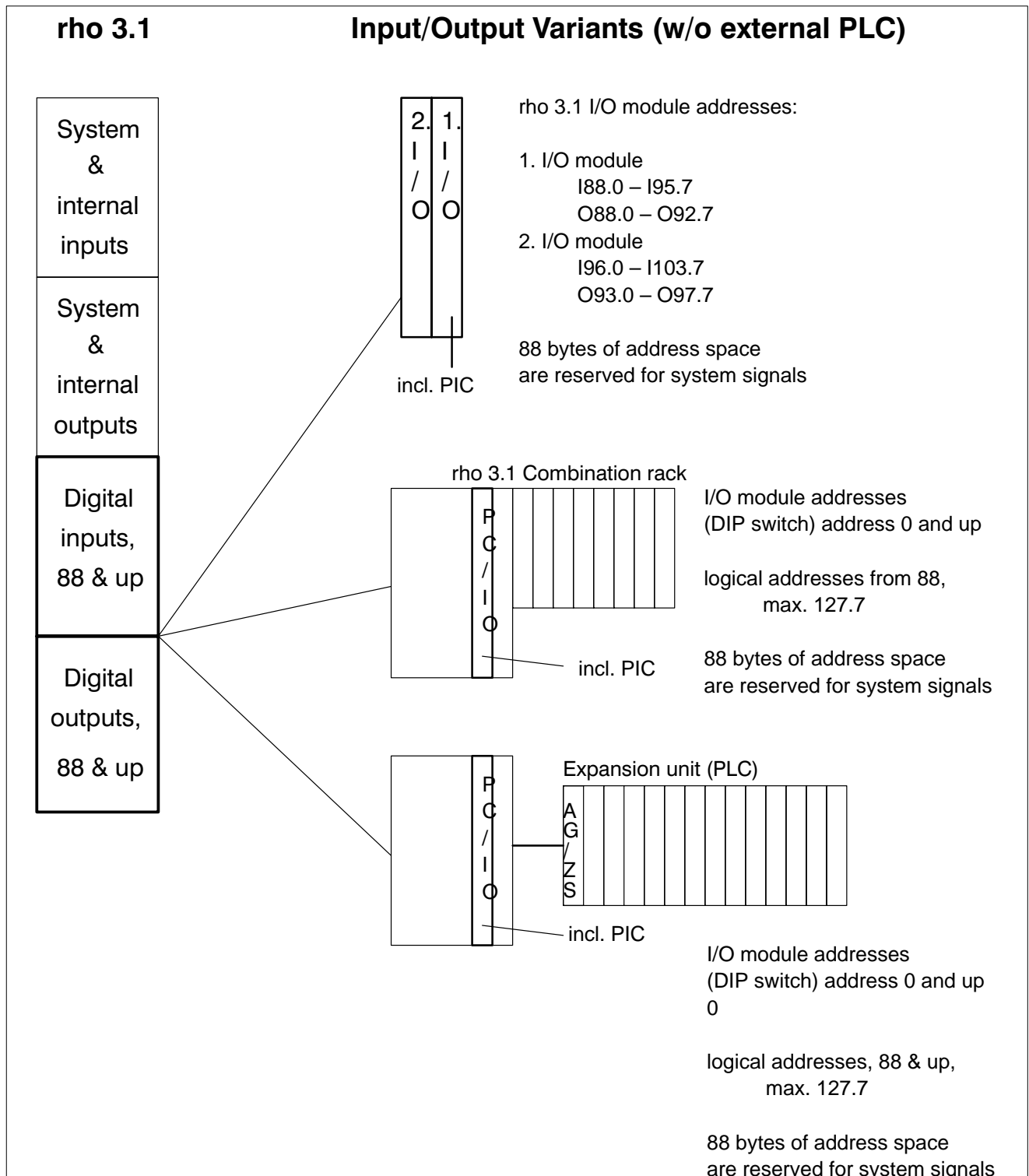


Fig. 2 Input / Output variants (without external PLC)

2.3 PIC Signals and direct signals

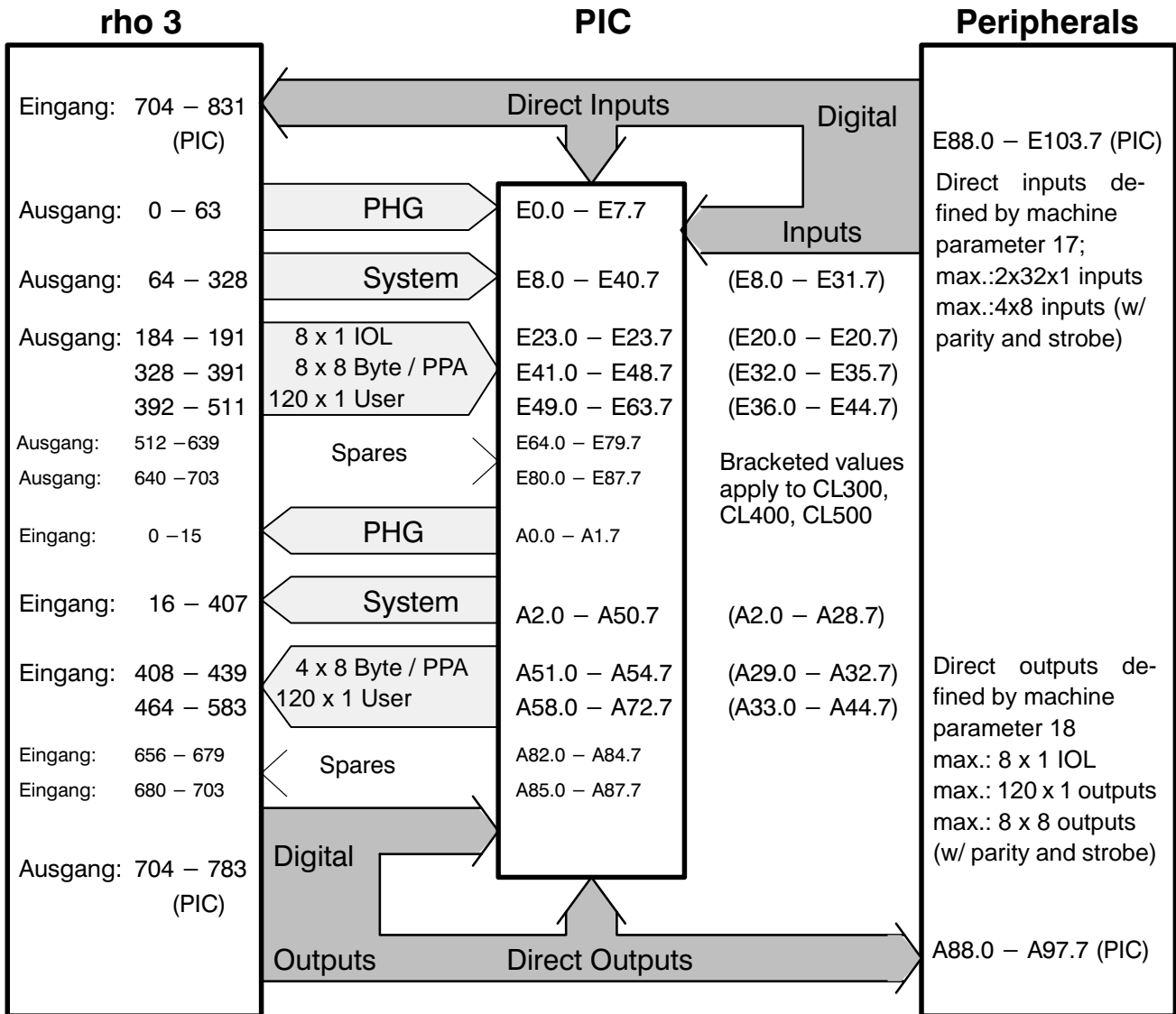


Fig. 3 PIC Input / Output signals

In the absence of extraordinary operating conditions, the peripheral signals are passed through to the rho robot control via the PIC (subject to possible interrogation). If the response time is insufficient, and signal interrogation is not required, the inputs and outputs can also be directly reported to the rho. See also the description of machine parameters 17 and 18 (1070 073026).

The user outputs that are controlled from within a BAPS process and merely passed through by the PIC, will retain their most recent status even if the process is aborted by means of a RESET command. Machine parameter 24 can be used to specify the range of user outputs (1 through 120) that is to be deleted in the case of a RESET.

2.4 Running SoftPIC on the rho 3.0

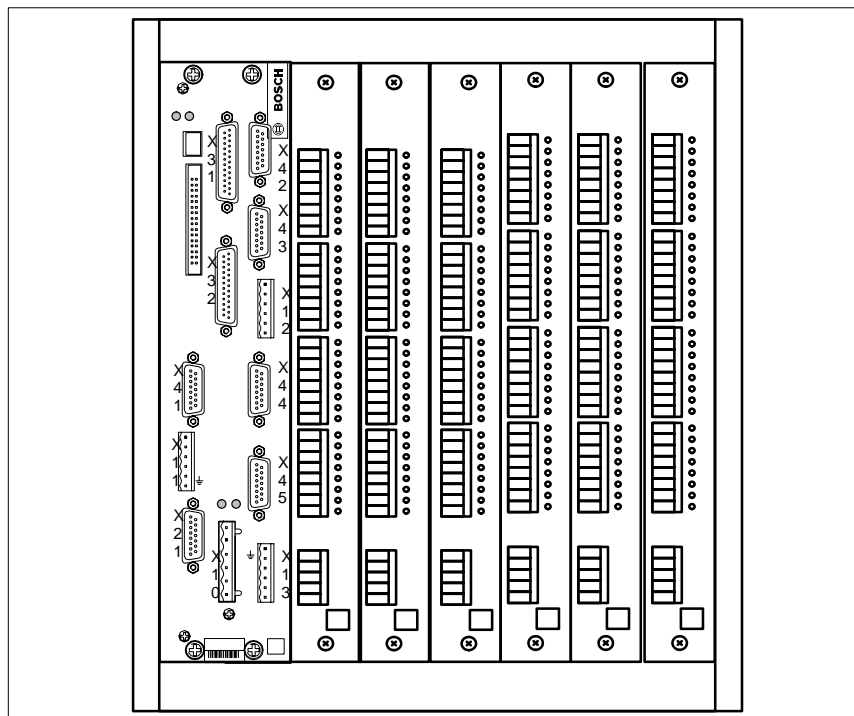


Fig. 4 SoftPIC for rho3.0

In the event that the rho 3.0 is operated in an EG2/K expansion rack as a stand-alone controller, it can be augmented with a **SoftPIC** for the purpose of peripherals signal interrogation. The **SoftPIC** comprises the **Software** implementation of a **Programmable Interface Controller**.

The extent of the command set of the SoftPIC is identical to that used by the PIC250 on the rho 3.1 robot control.

Because the processor of the rho 3.0 must also take care of the interrogation task, the processing time of the SoftPIC is added to the position feedback interval for the axes, i.e.,

$$\text{Position feedback interval} + \text{SoftPIC processing time} = \text{clock frequency (machine parameter P5)}$$

The SoftPIC requires approx. 3 ms to process 1 k of commands (processed command code).

Using the PHG hand-held programming unit, Mode 3.2.5 can be used to indicate the total machine processing time.

The module addresses for the I/O modules are set consecutively in binary code via DIP switch, starting at address 0. The PIC addresses for peripheral I/O's are located at address 88 and up. (See *Signal Descriptions and Fault Messages*, no. 1070 073028).



3 PIC Instructions

Programming of the PIC functions can be effected under the PROFI user interface in IL (instruction list) format, or under ROPS in BAPS-PIC.

3.1 Bit Operations

Operator (OP)	Operand (OPD)	Function
A	M, O, I	Boolean AND
AN	M, O, I	Boolean AND NOT
O	M, O, I	Boolean OR
ON	M, O, I	Boolean OR NOT
S	M, O,	Set (instruction)
R	M, O,	Reset
JPC	Address	Jump conditional
=	M, O,	Boolean EQUAL TO
NOP1		No operation
EP	Address	End of Program/ Concluding operation

3.2 Time / Counter Operations

Via marker fields (indirect operations)

OP	OPD	
S, R, = U/UN, O/ON	M	Times
S, R, = A/AN, O/ON	M	Counters

3.3 Operand Labelling

- M Marker, time start, time query, counter start, counter query
- I Input
- O Output

3.4 BAPS-PIC250 Instruction Set (Standard)

<p>Compiler instructions: Specifications for the compiler: PIC program or rho program, and/or version 100, 200, 220</p>	<pre>;; CONTROL = PIC250 ;; VERSION = 200 ;; MARKER_ANF = 512</pre>
<p>Program convention with the first and last instruction. The program name may be up to 8 characters long, and must correspond to the filename.</p>	<pre>PROGRAM filename : END</pre>
<p>All names of variables must be assigned to data types. The respective data types are PIC200 inputs, outputs or markers.</p>	<pre>INPUT: 1 = IN1, 2 = IN2 OUTPUT: 1 = OUT1</pre>
<p>The remanent markers start at address 192. The addresses below that number are reserved for times and counter functions.</p>	<pre>BINARY: 192 = MREM1, 193 = MREM2</pre>
<p>The non-remanent markers are reserved from address 512 onward (and/or from markER_ANF (>=512)onward) in the alphabetical order of the names of the variables.</p>	<pre>TEMP BINARY: MNREM1, MNREM2, MNREM3</pre>
<p>Logical operations, for example: mark1=in1 ANDin 2 ANDin 3 mark2in=1 ORin OR2 in 3 mark3in=3 AND NOTin OR2NOT in 3</p>	<pre>AND OR NOT</pre>
<p>Logical operations, for example: out1=mark1 out2=mark1 OR (mark2 AND mark3)</p>	<pre>= ()</pre>
<p>Conditional instruction: If the condition is satisfied, the control will execute instruction1, OR, if the condition is not met, instruction2 is executed.</p>	<pre>IF condition THEN instruction1 ELSE instruction2</pre>
<p>Forward jumps and backward jumps can be executed on markers (so-called labels) within a program.</p>	<pre>JUMP Label_1 : Label_1:</pre>
<p>Following a colon, commentary text reaching to the end of the line can be inserted.</p>	<pre>; text commentary</pre>



3.5 BAPS-PIC250 Instruction Set Extension in Version 200

Effective with ROPS version **W2C**, the BAPS-PIC compiler provides the option to translate not only existing PIC programs (BAPS-PIC 1.00, version 100), but also new PIC programs (BAPS-PIC 2.00, version 200).

The selection of the language version is controlled by the new **VERSION** compiler instruction. The default setting defines version 100, which means that existing PIC programs can be translated without modifications as was previously the case. The standard PIC program is located in the **BAPSPIC** subdirectory of the **ROPS3** directory.

3.5.1 New Program Structure (Version 200)

The syntax structure of the BAPS-PIC program was adapted to the structure of the BAPS language. This means that the instruction section of the program is separated from the declaration section by the keyword **BEGIN**, and the end of the program is identified by **PROGRAM_END**.

```
;;CONTROL = PIC250
;;VERSION = 200PROGRAM
Test
    ;Input / Output declarations
BEGIN
    ;...
    ; Instructions
    ;...
PROGRAM_END
```

3.5.2 Previous Program Structure (Version 100)

```
;;CONTROL = PIC250
;;VERSION = 100
PROGRAM Test
    ;Input / Output declarations
    ;...
    ; Instructions
    ;...
END
```

3.5.3 Combination Instruction

Through the use of the keywords BEGIN and ENDE, the combination instruction makes it possible to execute several instructions in all situations where only one instruction is permitted.

```

IF In1 THEN BEGIN
    Out1 = 1
    LED2 = 1
END
ELSE BEGIN
    Out1 = 0
    LED2 = 0
END
    
```

3.5.4 No-operation Instruction

In situations where previously an instruction was absolutely required, for example in the case of IF–THEN–ELSE decisions, the mandatory instruction can now be omitted.

```

If NOT Timeout THEN
    ;No-op instruction
ELSE
    JUMP ANS_END
; ...
ANS_END:
;No-op instruction
PROGRAM_END
    
```

3.5.5 INCLUDE Instruction

Effective with version 200, the file extension for insertion files can be freely selected. If only the filename is entered, the **.QLS** filename extension is entered by default. This makes it possible, for example, to save the signal declarations in a separate file.

```

;;INCLUDE Marker.TXT
;Inserts the file with filename Marker.TXT

;;INCLUDE Marker
;Inserts the file with filename Marker.QLS

;;INCLUDE Marker.
;Inserts the file with filename Marker. (no extension)
    
```




3.6 BAPS-PIC250 Instruction Set Extensions in Version 200 in conjunction with ROPS Version W3C

3.6.1 Channel Numbers for Temporary Markers

As was previously the case, markers can be defined by means of the keyword BINAER throughout the entire marker range (M0.0 – M126.7) 0 through 1015. However, they are remanent only in the range of 0 through 512. The markers (M127.0 – M127.7) 1016 through 1023 are internally assigned.

Effective with version 2.20, a channel number is permitted as a part of the declaration of non-remanent markers. The address assignment is effected relative to the start of the non-remanent markers in the alphabetical order of the names of the variables; the numbering starts at 0.

Although the addresses of the non-remanent markers normally start at 512 (M64.0), thus the starting address can be shifted upward by means of MARKER_ANF compiler instruction.

In addition, byte and word-specific marker assignments are now permitted.

Example:

```
TEMP BINARY : 1 = mark1      —> M64.1
TEMP BYTE   : TempByte
TEMP WORD   : TempWord
```

If addresses are assigned twice, the BASP-PIC compiler returns an appropriate warning. In the course of automatic addressing in the alphabetical order of the names of their variables, markers that are already assigned are skipped by the compiler. In the above example, the compiler 'searches' the non-remanent marker field for an 8 and 16-bit gap for the *TempByte* and *TempWort* variable, respectively.

3.6.2 Declaration of Constants

Effective with BAPS-PIC, version 2.20, it is now possible to utilize constants and/or calculated constant expressions.

Constants are defined with the **CONSTANT** keyword in conjunction with the variables of a program within the declaration section. The declaration section may contain several constant definitions. The BAPS-PIC compiler replaces each constant in the source text with the associated value.

Syntax :

```
CONSTANT : Name = Constant_Expression
           ( , Name = Constant_Expression ).
```

The declared constant can also be calculated by means of a term. The following operators are permissible for calculated constant expressions:

- NOT
- MOD
- +
- –
- *
- /
- ()

Example:

CONSTANT:

ONE	= 1,
ON	= 1, ; logical TRUE!
OFF	= NOT ON, ; logical FALSE
BLACK	= ONE,
RED	= BLACK + ONE,
GOLD	= BLACK + 2 * ONE,
FOUR	= 40 / 10 * (2 – 1),
FIVE	= 50 MOD 15

Calculated constants

Due to the new features in version 220 it is now possible to specify constants (e.g. address specifications for channel definition) also in the form of 'constant expressions' (i.e., calculated constants).

Syntax: Address = Constant_expression.

Example:

CONSTANT:

```
START_INX = 100,
U_GRENZE = 0,
O_GRENZE = 15
```

; ...

ARRAY[U_GRENZE .. O_GRENZE] BINARY :

```
( START_INX + 0 .. START_INX + 15 ) = BinArray
```

BYTE: Number

; ...

```
Number = O_GRENZE – U_GRENZE + 1
```



Constants in various expressions

With the introduction of the compiler version 2.20, constants can now be specified with an almost arbitrary number numerical base. This presents advantages, for example, when mapping binary masks.

As a numerical base, the value range of 2 to 32 ([2 .. 36]) is permitted.

Syntax : **CONSTANT** : **Num_basis # Num_value**

Examples:

```

CONSTANT:MaskAnd      = 2#11111111,      ; binary
                    MaskOr      = 2#00000000,    ; binary
                    MaskOct     = 8#1234567,    ; octal
                    MaskHex     = 16#FF        ; hexadecimal

```

3.6.3 Standard Constants

The following constants are available as a default:

Constant	Description	Value
VERSION	current version number	e.g. 220
MINBYTE	lowest byte value	0
MAXBYTE	highest byte value	255
MINWORD	lowest word value	0
MAXWORD	highest word value	65535

3.6.4 New Standard Types

In the compiler version 2.20, the BAPS-PIC programming language was extended bay adding the following data types:

- **BYTE** (8 Bit)
- **WORD** (16 Bit)

Accordingly, the BINARY, BYTE and WORD standard types are now available.

Example:

```

INPUT BYTE      : 136 = Out1_8PHG      ; RC output 1 - 8 of PG
                    ; (I17.0 .. I17.7)
INPUT WORD     : 224 = Inpos_1_16A    ; INPOS axes 1 through 16
                    ; (I28.0..I29.7)
INPUT WORD     : 152 = Cod_Error     ; coded fault return
                    ; (I19.0..I20.7)

```

The address specifications for the **BYTE** and **WORD** data types refer to the consecutive numbering as per signal description. Starting with these numbers, the number of bits corresponding to the bit width of the data type is used.

3.6.5 ARRAY Declaration

To improve the structuring of variables, the ARRAY data type was introduced. This makes it possible to address associated variables under a common name.

The individual signals are accessed via an index. The index must be constant, i.e. calculable.

In the event that the addresses are present in one sequence (*ByteFeld* in the following example), the array can be accessed as a whole. If the addresses are not available in one sequence, the array can be accessed element-by-element only.

Example:

```

ARRAY [ 1 .. 32 ] BINARY :
    (16..23, 64..71, 112..119, 160..167) = t_start_m    ; Start Timer [n]
;
    (M2.0–M2.7, M8.0–M8.7, M14.0–M14.7, M20.0–M20.7)
; ...
ARRAY [ 0 .. 3 ] BYTE :           200    = ByteArray
;
                                     (M25.0 – 7, M26.0 – 7, M27.0 – 7, M28.0 – 7)
ARRAY [ 1 .. 6 ] INPUT :
    (42, 30, 28, 26, 24, 17)      = JOG_P_PHG
;
    (E5.2, E3.6, E3.4, E3.2, E3.0, E2.1 als JOG_P_PHG [1] .. [6] -switch)
; ...
ByteArray = 0                      ; delete entire array!
t_start_m[32] = 1                   ; start of time 32 !
; ...
    
```

Non-remanent marker arrays can be specified by using the reserved **TEMP** word.

Arrays containing inputs and outputs are also possible!

Example:

```

ARRAY [ 0 .. 3 ] TEMP BINARY :    47    =    TempArray
ARRAY [ 0 .. 2 ] OUTPUT BYTE :    1     =    OutArray
ARRAY [ 0 ..11 ] INPUT WORD :     1     =    InArray
    
```

In the event that the addresses are present in one sequence, the field can be accessed as a whole. If the addresses are not available in one sequence, the array can be accessed element-by-element only.

Example:

```

; RC output: selected with group 1–8
INPUT ARRAY [0..7] BINARY : (168 .. 175) = AnwGrp_1_8
BYTE : ByteVar
; ...
IF AnwGrp_1_8 [ 3 ] THEN ...           ; 'Element-specific' access
ByteVar = AnwGrp_1_8                   ; 'Array-specific access
    
```



3.6.6 Introduction of Byte and/or Word Instructions

For the new types named **BYTE**, **WORD** and **ARRAY**, the following operations are permitted:

- Logical OR link OR
- Logical AND link AND
- Negation NOT
- Equality =
- Assignment =
- Multiple assignment :=

Example:

```
ARRAY[1..4]      OUTPUT BYTE :      702 = OutByte
                  TEMP BYTE :      ByteMarker,
                  EINGANG BYTE   : 136 = Aus1_8PHG ; RC Output 1 – 8 from PG
                                          ; (I17.0 .. I17.7)
;...
OutByte[1] := ByteMarker := Byte2      ; Multiple assignment
OutByte[1] = ByteMarker OR Aus1_8PHG
; ...
```

When assigning constants to **BYTE**, **WORD** and/or file variables, the compiler verifies that the constant can be completely stored within the variable. For example, the assignment of **259** to a variable of the **BYTE** type (**ByteMarker** in this example) is not permissible.

3.6.7 Multiple Assignment

Effective with version 2.20, a sequence of assignments taking the form:

```
Out1 = 1
Out2 = 1
Out3 = 1
; ...
OutN = 1
```

can be replaced by a "multiple assignment" as follows:"

Out1 := Out2 := Out3 := ... OutN := 1

The multiple-assignment operator **:=** must be used in order to prevent ambiguities (e.g. confusion with the equal-to operator).

3.6.8 Compare Operators

The only available compare operation are possible to establish equality or inequality (by negating the entire compare expression). The permitted operators for compare operations can be variables as well as constants!

Example:

```

ARRAY [1..4]    OUTPUT BYTE : 702 = OutByte
                TEMP  BYTE :   ByteMarker
INPUT BYTE     : 136 = Out1_8PHG      ; RC Output 1 – 8 from PG
                                           ; (I17.0 .. I17.7)
; ...
IF Out1_8PHG = ByteMarker THEN BEGIN
; ...
    END
; ...
; ELSE is more efficient here!
; ...
IF NOT ( Out1_8PHG = ByteMarker ) THEN BEGIN
; ...
    END
; ...

```

3.6.9 Line Numbers for Lines of Commentary

Effective with version 2.20, commentary from the .QLS file can be transferred to the .P20 file. This can be of advantage when testing PIC programs by means of the PIC test system.

The new function is activated by the compiler instruction **;;COMMENT+**. It is again disabled with **;;COMMENT-**. The compiler instruction can appear in the source text as often as required. The default setting is **;;COMMENT-**!



3.6.10 Backward Jump Warning

The BAPS-PIC program allows the programming of backward jumps, which may cause endless loops in the program. For this reason the BAPS-PIC compiler issues an appropriate warning if it detects a programmed backward jump. If required, this warning can be suppressed with the compiler instruction `;;WARNING -`.

3.6.11 Language-dependent Code

The BAPS-PIC compiler was enhanced to the effect that now, dependent upon the version of source language (German or English) used, the P20 file will be generated in the respective language. This enhancement is independent of the version number stated in the source file.

Comparison of language versions:

German-language version	English	Function
U	A	Boolean AND
O	O	Boolean OR
UN	AN	Boolean AND NOT
ON	ON	Boolean OR NOT
S	S	Set
R	R	Reset
SPB	JC	conditional jump
=	=	Assignment
NOPI	NOPI	No-operation instruction
PE	EP	End of program
M	M	Marker
E	I	Input
A	O	Output

Personal Notes:



4 PIC Program Generation

4.1 Using BAPS to generate a PIC Program

A word processing utility program (e.g. Editor) is used to write a **BAPS-PIC source program**. The resulting filename extension will be **.QLS**.

As the next step, the **compiler** will be used to translate the source program into an executable object file.

Any errors that may occur in the process are logged and stored in the **.ERR** file.

Subsequent to an error-free compiler run the system generates a **.P2O** and a **.P2X** file. The **.P2O** file is required for monitoring operation only, and must therefore not be loaded. The **.P2X** file comprises the hex file of the PIC250 program. This file is loaded into the robot control (PIC250 card).

Provided that the **.P2O** on the PC and the **.P2X** file in the rho robot control feature the same name, the monitor integrated in PIC test system will be able to indicate the signal statuses.

The PIC source program features the same language symbols as an RC source program performing comparable functions.

Refer to Chapter 3 for information on the language reservoir (keywords).

In line 1 of the sequential program, the destination control is declared by a compiler instruction (identified by a double colon). The syntax of this instruction is as follows:

```
;;CONTROL = PIC250
```

In the event that the extended instruction set is to be used, this is specified by the following instruction:

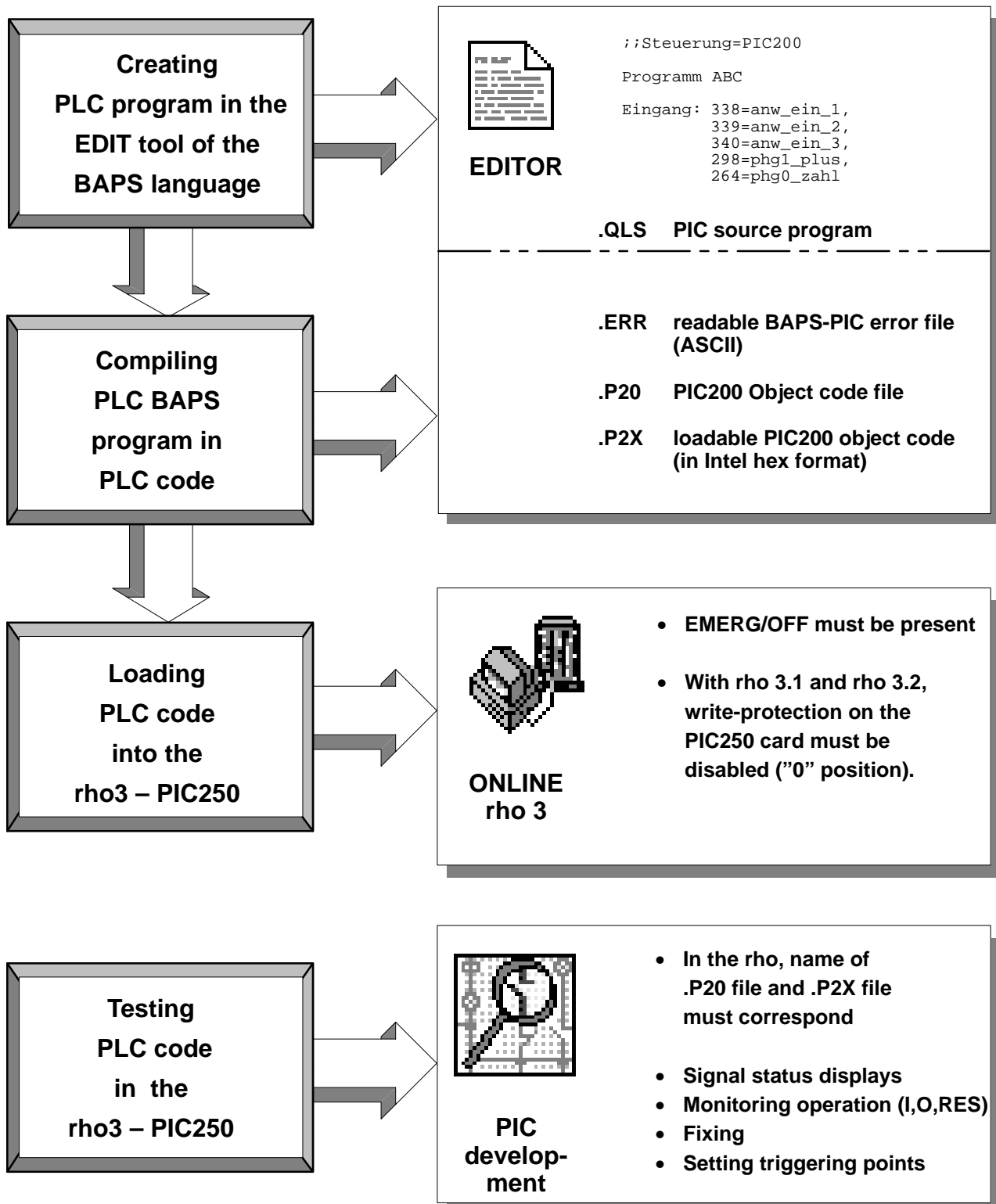
```
;;VERSION = 200
```

The markers up to 192 are occupied by the max. 32 times and counters of the PIC. The times and counters are expandable in multiples of 8 (8, 16, 24, 32). In the event that not all of the 32 times and counters are used, the associated marker addresses will be freely assignable.

Language symbols may be written in both upper-case and lower-case expressions. Any variables that are employed can consist of a maximum of **12 characters**. (Creating the PIC program in PROF1 will reduce this length to 8 characters.) The first character must be a **letter**. The underscore character **"_"** is the **only** permitted **special character**.

Compiler instructions, such as CONTROL and VERSION, must precede the PROGRAM instruction. Program lines containing compiler instructions must not contain any other characters in addition to the instruction.

4.2 Programming Tools





4.3 Editor



Editor

Choosing (double-clicking) the Editor symbol invokes an editing function that can be used for creating or modifying any type of ASCII file. For PIC programs, the **.QLS** filename extension must be used.

Several files of any size can be edited simultaneously.

The Editor features its own Help system with the following contents:

- Editor operation
- BAPS2 syntax
- BAPSPIC syntax

Using the **Tool** menu, several applications can be directly invoked via menu command. Various ROPS3 tools, such as **BAPS Development**, **PIC Development** and **ONLINE**, have already been integrated.

At this point, there exists also the option to integrate special program calls (refer to Editor Help function for details).

The working directory for the Editor is registered in the EDITOR.INI file which is located in the ROPS3 directory (refer to Editor Help function for details).

The **Tool** menu option also provides the command for activating the **BAPS-PIC compiler** which will check and verify the code entries.

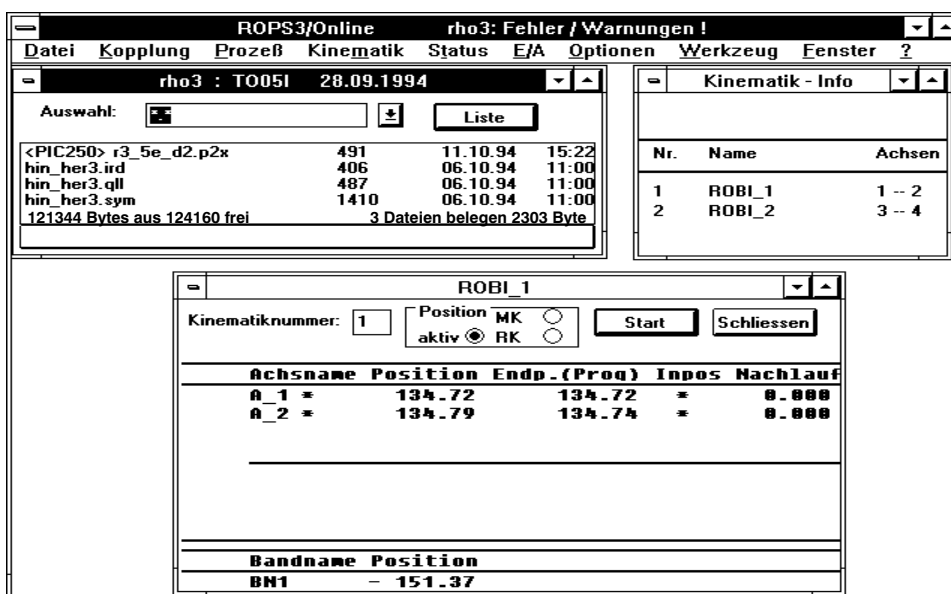
In the event that an error occurs, an error file named FILENAME.**ERR** is created. It lists the faulty source program lines and error messages. If an .ERR file is generated, no new .P20 and .PX2 files will be created. Any files with these filename extensions that originate from earlier compiler runs will be retained unchanged.

- 'FILENAME.**P2X**' – Control-executable program.
- 'FILENAME.**P2O**' – Program file indicated by the programming unit in monitoring operation.

4.4 ONLINE Connectivity Application



Provides functions for file handling with the rho 3, for visualizing various control statuses, and for process handling.



The following functions are available:

- Lists of PC files
- File handling of rho 3 files (List, Copy, Delete and Rename functions)
- Prozess handling (List, Status, Select, Start and Stop functions)
- Kinematics information, axis display
- Display of tool in current use
- Display of RK system
- Display of axis positions in UK (only with option bit set)
- Basic status information and error indication
- Display of RC I/O's, digital I/O's, SK I/O's and CAN-EIA Bus 1 through 4
- Set RC outputs 10.1 through 10.8
- Home position
- Options (function keys, setup, etc.)
- Tools, i.e., user-defined integration of applications
- Miscellaneous Microsoft Windows functions (window displays, etc.)

To download the PIC program into the rho robot control, proceed as follows: Enable Emergency-OFF on the rho itself, or stop PIC via the PHG hand-held programming unit (refer to PHG instructions).

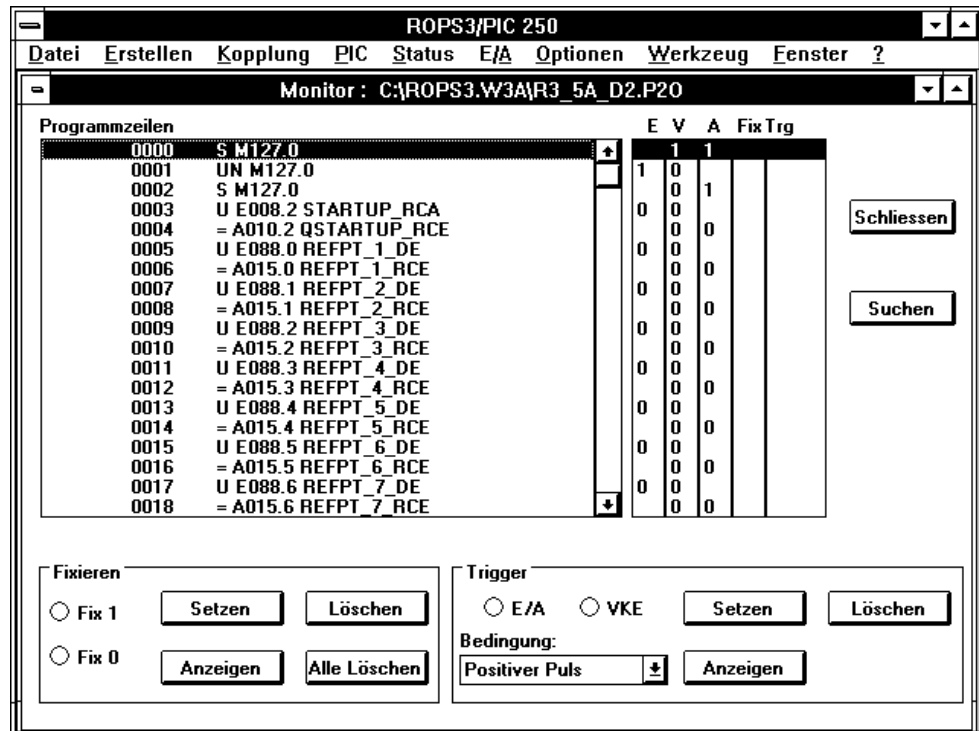
F2 Select PC file list
Mark desired FILENAME.PX2 file

F5 Select PC – RC transfer mode

Select **Überschreiben** command to start the data transfer.

During the programming process, the respective active line of code is displayed on the PHG (EEPROM or FLASH version).

Subsequent to controller start-up, Emergency-OFF must again be released.

**4.5 PIC250 Development System**

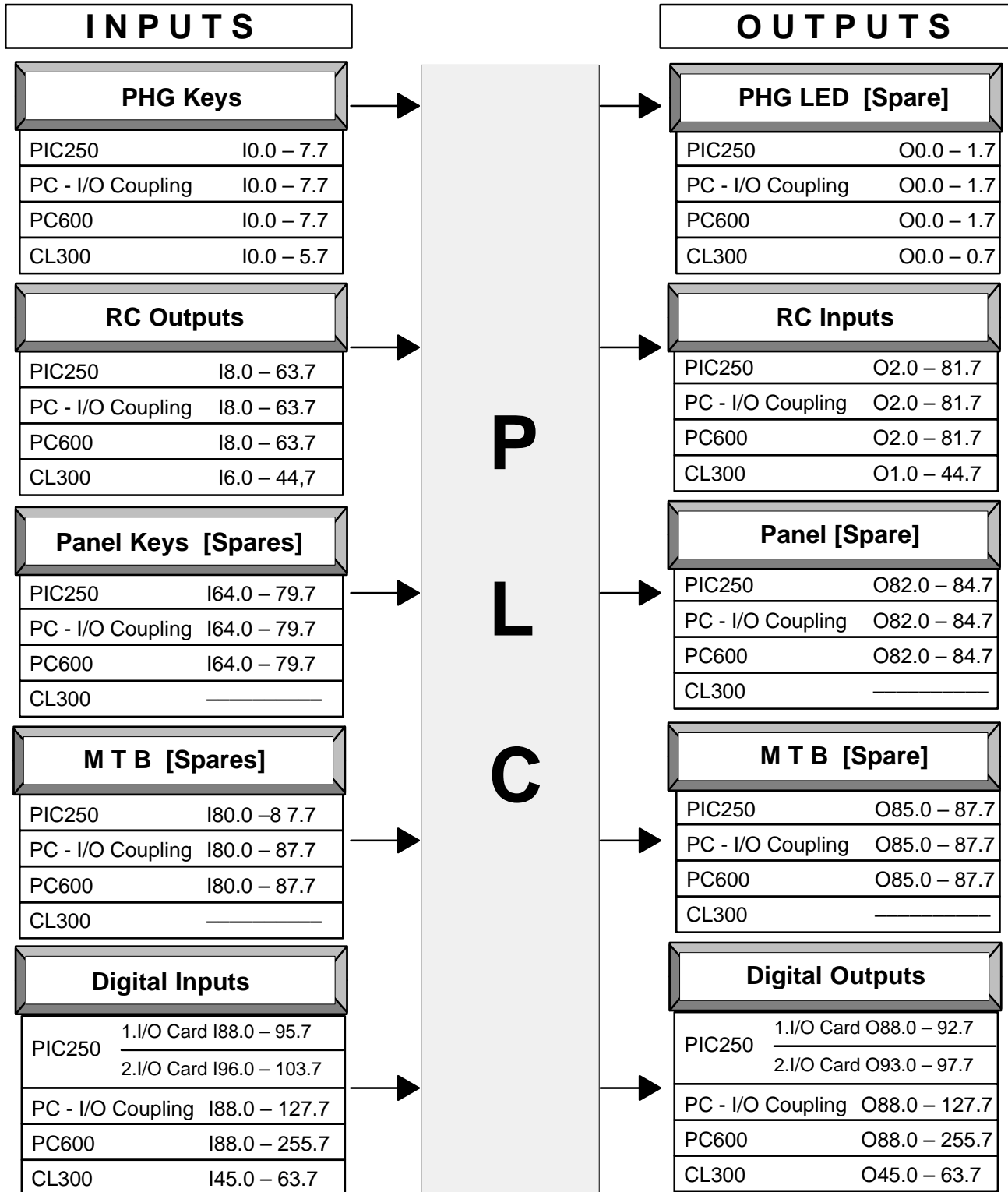
The application comprises a development and testing tool for PIC250 programs.

The following functions are available to the user:

- Source program compilation
- File printing and display
- Search function in display window
- Invoking the Editor
- **File conversion .P2X \longleftrightarrow .P20**
- Saving and storing programs
- **PIC-Monitor**, i.e., testing of sequential programs
- Basic status information and error indication
- **Times and counter statuses**
- **Status markers**
- **Long-time triggering / Long-time forcing**
- **Displays: RC I/O's, digital I/O's, servo card I/O's and CAN-EIA Bus 1-4**
- Setting RC outputs 10.1 through 10.8
- Home position
- Options (function keys, Setup, etc.)
- Tools, i.e., user-defined integration of applications)
- Comprehensive Help function

Personal Notes:

5 PIC, CL300 & CL400 Address Ranges



5.1 Marker Assignment

M0.0	<p>Used for max. 32 times and 32 counters</p> <p>For example, if only 8 times and counters are used, the markers, from M6.0 (BINARY: 48) upward are also free for use.</p> <p>Refer to Chapter 8, <i>Signal Description and Error Messages</i>, no. 1070073028</p>	BINARY:	0
.		.	
.		.	
.		.	
.		.	
M23.7			191

M24.0	Remanent markers	BINARY:	192
.		.	
.		.	
M63.7	Fixed assignment as per BINARY no.		511

M64.0	Non- remanent markers	TEMP BINARY:	MERK1, MERK2,
.	<p>The assignment occurs in alphabetical order from M64.0 and up. Effective with v. 2.20, a fixed address can be assigned with TEMP BINARY also.</p>	.	
.		.	
.		MERKn.	
M126.7		0 = ErstMerk	

M127.0	Internally assigned for intermediate results, etc.	
.		
.		
M127.7		



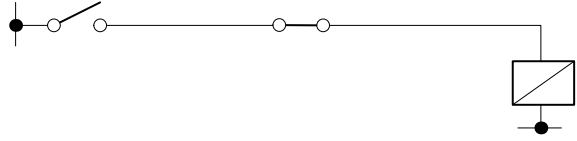
6 Examples

6.1 Programming Example – AND

AND Link

Problem:

IN1	IN2	OUT1	Symbol addresses
I0.0	I0.1	A0.0	Absolute addresses



Description:

Output OUT1 becomes logical "1" only if input IN1 is "1" and input IN2 is "0".
In the case of all other signal combinations OUT1 is "0."

PLC Solution:

PI	OP	OPD	Symbol
0	U	I0.0	IN1
1	AN	I0.0	IN 2
2	=	A0.0	OUT1

BAPS Solution:

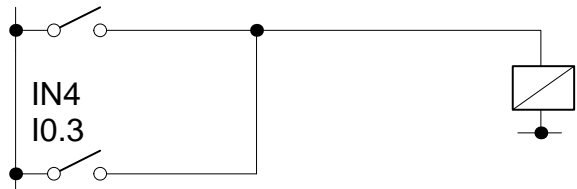
OUT1 = IN1 AND NOT IN2

6.2 Programming Example – OR

OR Link

Problem:

IN3	OUT2	Symbol addresses
I0.2	O0.1	Absolute addresses



Description:

Output OUT2 goes HIGH if one of the two inputs is HIGH (1).
 Output OUT2 goes LOW if neither of the two inputs is HIGH.

PLC Solution:

PI	OP	OPD	Symbol
3	A	I0.2	IN3
4	O	I0.3	IN4
5	=	O0.1	OUT2

BAPS Solution:

OUT2 = IN3 OR IN4



6.3 Programming Example – NOT

NOT Link

Problem:

OUT2	OUT3	Symbol addresses
O0.1	O0.2	Absolute addresses

The diagram shows a single rungs of a ladder logic network. On the left, there is a normally open contact (represented by two small circles connected by a horizontal line) labeled OUT2. A horizontal line extends from this contact to the right, where it connects to a coil (represented by a rectangle with a diagonal slash) labeled OUT3. The coil is connected to a common rail (ground symbol) at the bottom.

Description:

Output OUT3 goes to "1" if Output OUT2 is "0".
Output OUT3 goes to "0" if Output OUT2 is "1"

PLC Solution:

PI	OP	OPD	Symbol
6	AN	A0.1	OUT2
7	=	A0.2	OUT3

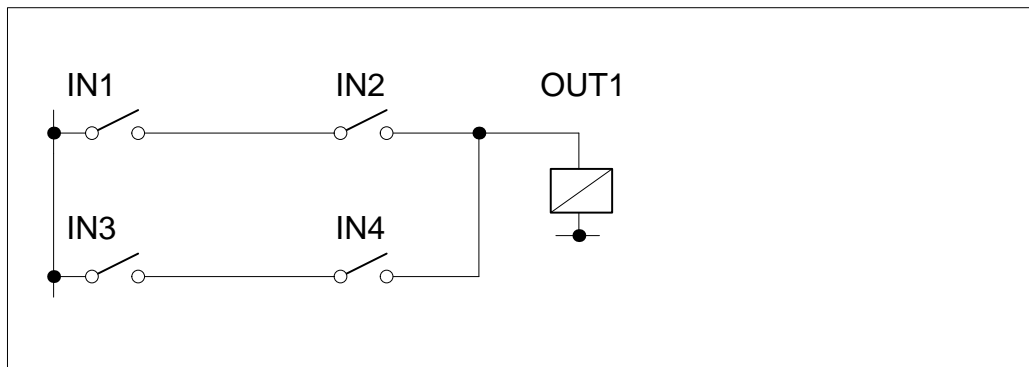
BAPS Solution:

OUT3 = NOT OUT2

6.4 Programming Example – AND before OR

AND before OR

Problem:



Description:

The PLC processes **AND** instructions before **OR** instructions. If OR is to be processed before AND, multi-level links must be converted to markers by storing intermediate results.

PLC Solution:

PI	OP	OPD	Symbol
1	A	I0.0	IN1
2	A	I0.1	IN2
3	O	I0.2	IN3
4	A	I0.3	IN4
5	=	O0.0	OUT1

BAPS Solution:

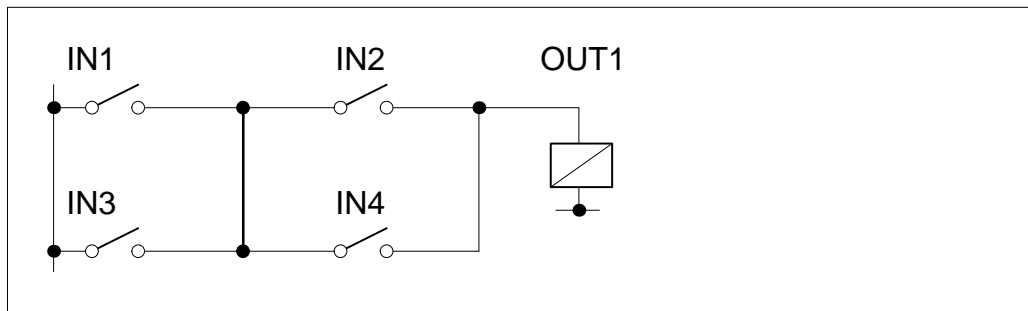
OUT1 = IN1 AND IN2 OR IN3 AND IN4



6.5 Programming Example – OR before AND

OR before AND

Problem:



Description:

The PLC processes **AND** instructions before **OR** instructions. If **OR** is to be processed **before AND**, multi-level links must be converted to markers by storing intermediate results.

PLC Solution:

PI	OP	OPD	Symbol
1	A	I0.0	IN1
2	O	I0.2	IN3
3	=	M0.0	Mark1
4	A	I0.1	IN2
5	O	I0.3	IN4
6	=	M0.1	Mark2
7	A	M0.0	Mark1
8	A	M0.1	Mark2
9	=	O0.0	OUT1

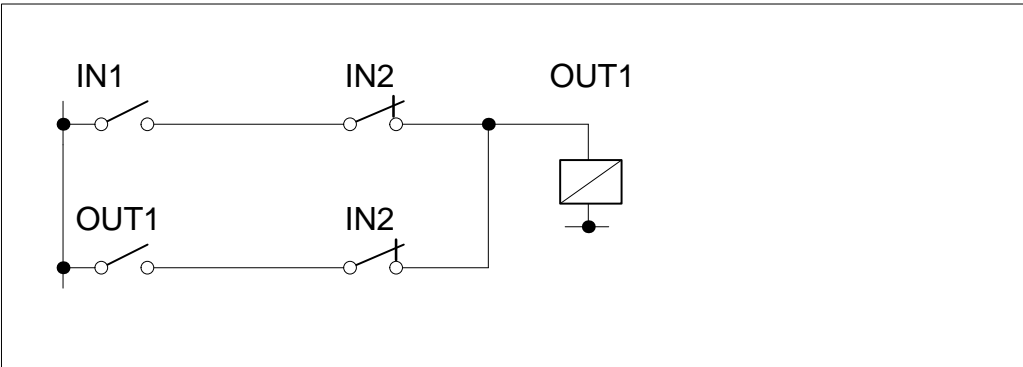
BAPS Solution:

Mark1 = IN1 OR IN3
Mark2 = IN2 OR IN4
OUT1 = Mark1 AND Mark2
 or
OUT1= (IN1 OR IN3) AND (IN2 OR IN4)

6.6 Programming Example – Set / Reset

SET / RESET w/ Save Function

Problem:



Description:

With Set condition met and Reset condition not met, OUT1 will be set, and remains on 'log 1'.
If Reset condition is met, OUT1 is reset.

PLC Solution:

PI	OP	OPD	Symbol
41			
42	A	I0.0	IN1
43	S	O1.0	OUT1
44	A	I0.1	IN2
45	S	O1.0	OUT1
46			

BAPS Solution:

```

IF IN1 THEN OUT1 = 1
IF IN2 THEN OUT1 = 0
    
```



6.7 Programming Example – Conditional Jump

CONDITIONAL JUMP

Problem:

**Conditional jumping (skipping) over program sections.
Jump condition is RES=1**

Description:

When jump condition is met, jump is executed to symbolic END address.
If RES=0, the jump is not executed, and OUT1 is set, provided that IN2 is available.

PLC Solution:

PI	OP	OPD	Symbol
41			
42	A	I0.0	IN1
43	JPC	46	END
44	A	I0.1	IN2
45	S	O1.0	OUT1
46	END		
47	A	I0.0	IN1
48	A	I0.1	IN2
49	=	O1.1	OUT2

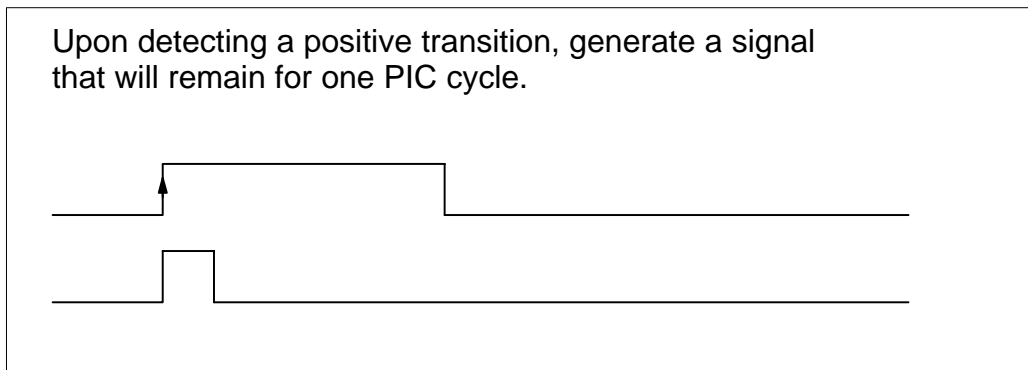
BAPS Solution:

IF IN1 THEN JUMP END
IF IN2 THEN OUT1 = 1
END:
OUT2 = IN1 AND IN2

6.8 Programming Example – Positive Transition

POSITIVE TRANSITION

Problem:



PLC Solution:

PI	OP	OPD	Symbol
1	A	I0.0	Pushbutton IN1
2	AN	M43.0	Auxiliary marker HM1
3	=	M43.1	Pos. transition marker
4	A	M0.0	Pushbutton IN1
5	=	M43.0	Auxiliary marker1

BAPS Solution:

PFL = IN1 AND NOT HM1

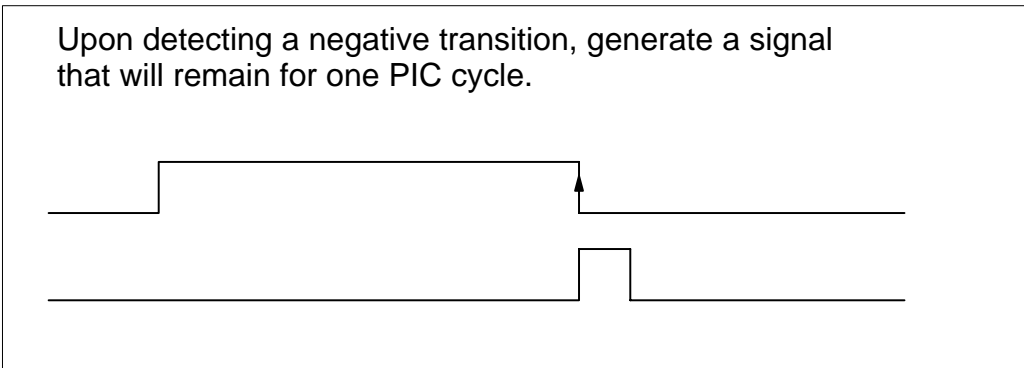
HM1 = IN1



6.9 Programming Example – Negative Transition

NEGATIVE TRANSITION

Problem:



PLC Solution:

PI	OP	OPD	Symbol
1	AN	I0.0	Pushbutton IN1
2	A	M43.2	Auxiliary marker HM2
3	=	M43.3	Transit marker, pos.
4	A	I0.0	Pushbutton IN1
5	=	M43.2	Auxiliary marker HM2

BAPS Solution:

```

NFL = NOT IN1 AND HM2
HM2 = IN1

```

6.10 Programming Example – TIME (Raising Delay)

TIME Function

Description:

The time values can be defined via the PHG (Mode 3.2.2.2) and/or specified via machine parameter 13. **The time is started by setting the Start marker.** (Time starts counting upon negative transition at the starting input.)

The query for the two time outputs is similar to that for an input:
 The first marker bit (TA_n) **goes HIGH as soon as the time activation is detected, and remains at 1 as long as the time is elapsing.**
 This behaviour corresponds to that of a *drop-off delay*.

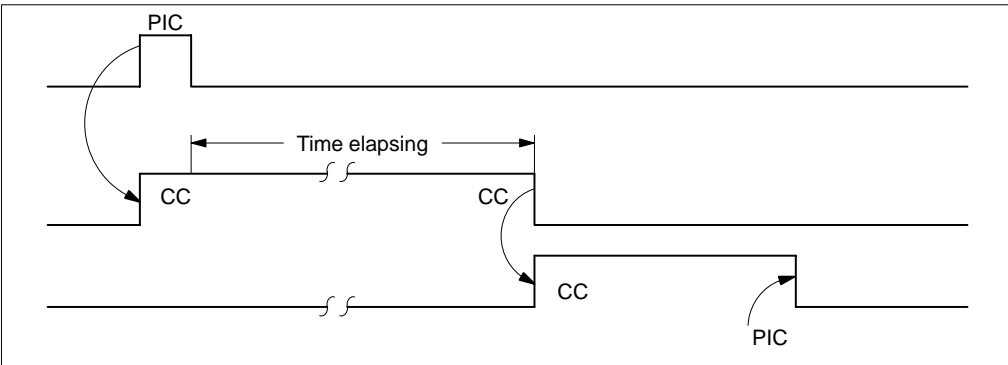
The second marker bit **goes HIGH when the time has expired (pick-up delay).** It can be reset by the PIC. It is automatically reset with every time start.

Diagram:

M Time n start (ST_n)

M Time n active (TA_n)

M Time expired (TE_n)



BAPS Solution:

```

;Raising delay

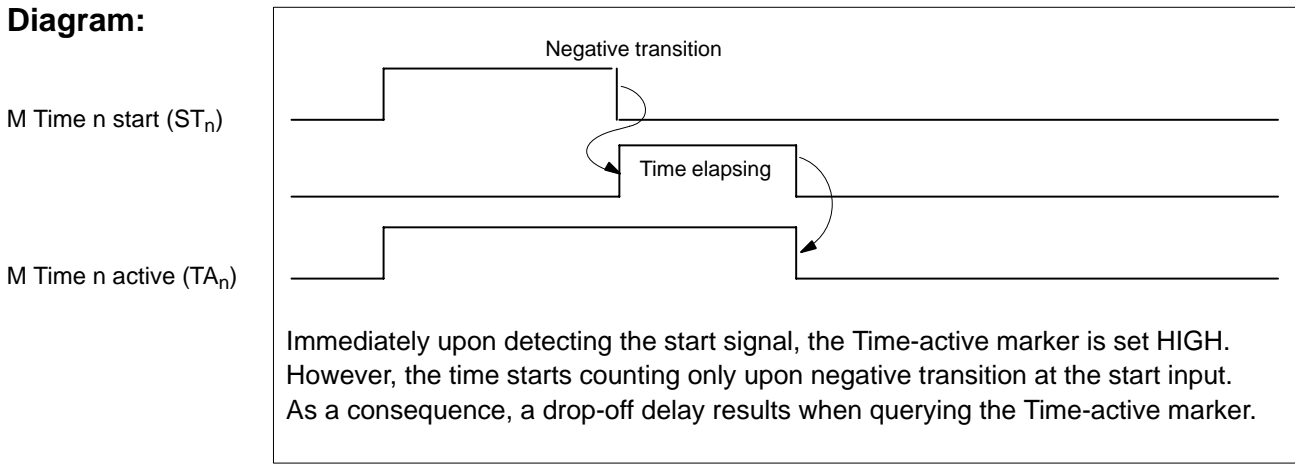
PFL = IN9 AND NOT HM1      ; Positive transition
HM1 = IN9

T0_START = PFL              ; Time start
OUT3 = T0_AKTIV_M          ; Time active
OUT4 = T0_ENDE_M           ; Time expired
    
```

6.11 Programming Example – TIME (Falling Delay)

TIME Function

Diagram:



PLC Solution:

PI	OP	OPD	Symbol
1	A	I0.1	Pushbutton IN2
2	=	M2.1	START MARKER TIME 1
3	A	M4.1	TIME ACTIVE MARKER
4	=	O0.2	Auxiliary marker HM2

BAPS Solution:

```

;Drop-off delay

T1_START = IN7           ; Time start
OUT3 = T1_AKTIV_M       ; Time elapsing
    
```

6.12 Programming Example – Counter

COUNTER Function

Description:

Number 32 counters (configurable in steps: 8, 16, 24, 32)
 Range 1 through 65535
 Direction of count decreasing

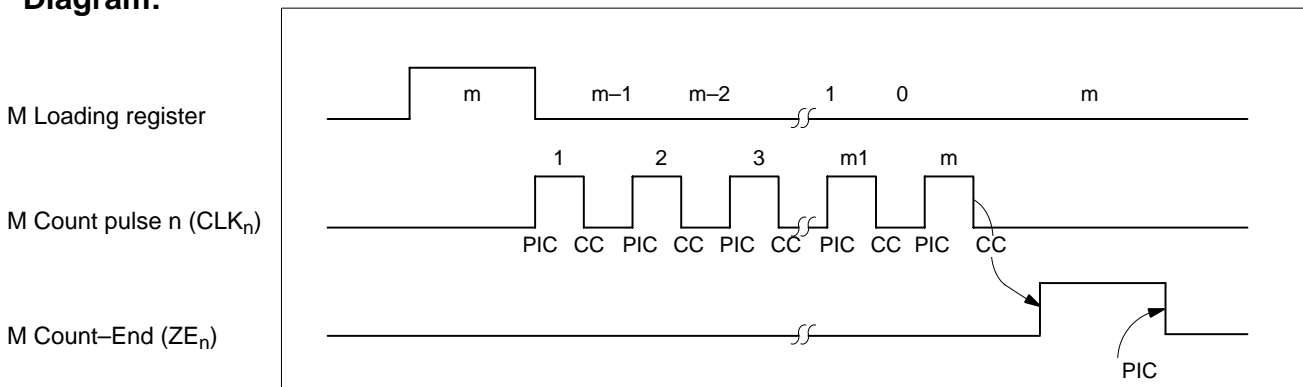
The nominal counter values can be defined via the PHG (Mode 3.2.2.3) or in machine parameter 14.

The starting value is reduced by the count of 1 per each counting pulse (*Bit from Marker 0*).

If the counter = 0, and an additional counting pulse is detected, the corresponding marker bit ZE_n from byte 1 is set.

Counter signals may be present for one PIC cycle only. If the signal lasts longer, the counter is decremented by 1 for each PIC cycle.

Diagram:



m = Value of the counter loading register

The counting pulse is generated by the PLC and deleted by the CC.

With each CLK_n (*byte 0*) counting pulse, the starting value of the counter is decremented by 1.

The counting bit is set by the RC when the counter has reached the value "0", and one additional counting pulse has been received. The bit is reset by the PIC when the counter is again started.

**PLC Solution:**

PI	OP	OPD	Symbol
	A	I1.2	RESET input
	=	M5.0	Counter Reset
	A	I1.3	Counter input
	AN	M35.0	Auxiliary marker 1
	=	M0.0	Counter – 1
	A	I1.3	Input
	=	M35.0	Auxiliary marker 1
	A	M1.0	Counter done
	AN	M35.1	Auxiliary marker 2
	S	O40.0	Counter expired
	A	M1.0	Counter done
	=	M35.1	Auxiliary marker 2
	A	M5.0	Counter Reset
	R	O40.0	Counter expired

BAPS Solution:

```
Z0_RESET_M = IN10
```

```
Z0_DEKREM_M = IN11 AND NOT Z0_FLANK1_M
```

```
Z0_FLANK1_M = IN11
```

```
IF Z0_ENDE_M AND NOT Z0_FLANK2_M THEN OUT8 = 1
```

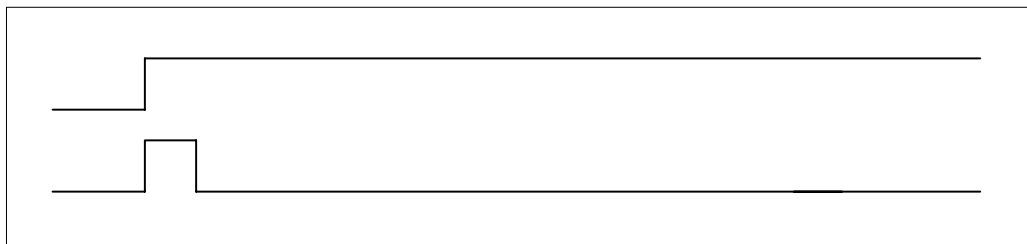
```
Z0_FLANK2_M = Z0_ENDE_M
```

```
IF Z0_RESET_M THEN OUT8 = 0
```

6.13 Programming Example – Trigger Pulse

TRIGGER PULSE for one PIC Cycle

Problem:



PLC Solution:

PI	OP	OPD	Symbol
1	AN	M110.0	Non-remanent aux. marker
2	=	M110.1	Trigger pulse
3	S	M110.0	Non-remanent aux. marker

BAPS Solution:

```

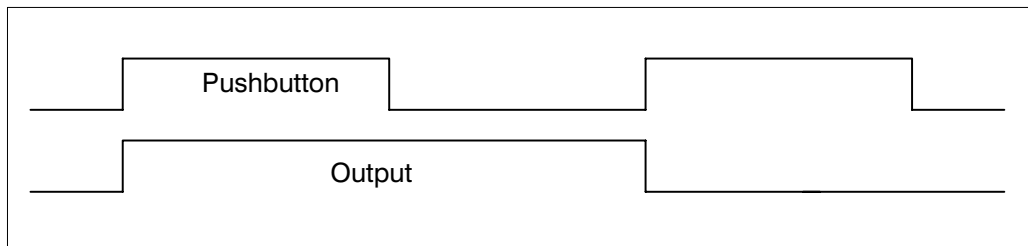
RI = NOT HM_RI
IF NOT HM_RI THEN HM_RI = 1
IF IN3 THEN OUT3 = 1
IF (IN4 OR RI) THEN OUT3 = 0
    
```

The trigger pulse is used for standardizing signals during system start-up (Power-ON status).

6.14 Programming Example – Cycling ON/OFF w/ Pushbutton

Cycling ON/OFF w/ Pushbutton

Problem:



PLC Solution:

PI	OP	OPD	Symbol
1	A	I0.7	Input, switched
2	AN	M32.0	Auxiliary marker HM
3	=	M32.1	Transition marker FLM
4	A	I0.7	Input, switched
5	=	M32.0	Auxiliary marker HM
6	A	M32.1	Transition marker FLM
7	SPB	10	Jump to Destination
8	AN	O32.6	Output, switched
9	=	O32.6	Output, switched
10			Jump destination

BAPS Solution:

FLM = IN3 AND NOT HM

HM = IN3

IF NOT FLM THEN SPRUNG WEITER

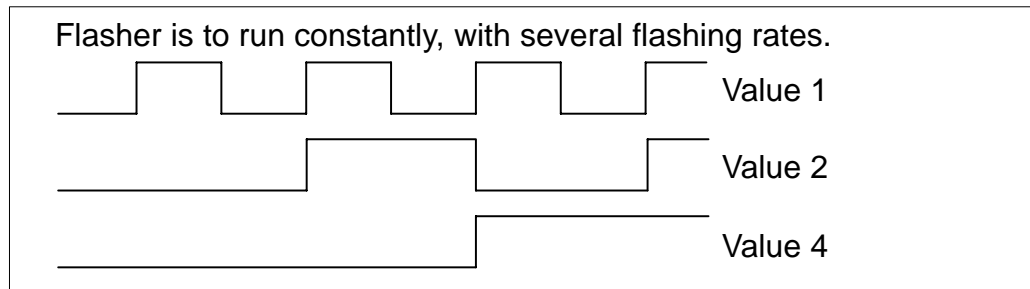
OUT3 = NOT OUT3

WEITER:

6.15 Programming Example – Flasher

FLASHER

Problem:



PLC Solution:

PA	OP	OPD	Symbol
	AN	M4.2	Time active
	AN	M64.0	Transition marker HM
	=	M2.2	Time start
	AN	M4.2	Time active
	=	M64.0	Transition marker HM
	AN	M2.2	Time start
	JPC	End	End_flasher
	AN	M65.1	Value1
	=	M65.1	Value1
	JPC	End	End_flasher
	AN	M65.2	Value2
	=	M65.2	Value2
	JPC	End	End_flasher
	AN	M65.3	Value4
	=	M65.3	Value4
		End	End_flasher destination

**FLASHER (cont'd)**

PI	OP	OPD	Symbol
	A	M65.1	Value1
	=	O0.1	
	A	M65.2	Value2
	=	O0.2	
	A	M65.3	Value4
	=	O0.3	

BAPS Solution:

```
T2_START_M = NOT T2_AKTIV_M AND NOT HM
HM = NOT T2_AKTIV_M
IF NOT T2_START_M THEN JUMP END
W1 = NOT W1IF
W1 THEN JUMP END
W2 = NOT W2IF
W2 THEN JUMP END
W4 = NOT W4
END:

OUT1 = W1
OUT2 = W2
OUT3 = W3
```

6.16 Programming Example – Byte / Word Instructions

BYTE / WORD Instructions**Function:**

To facilitate programming procedures, Byte or Word instructions can be defined.

For reasons of compatibility, the compiler converts the Byte and/or Word instructions into PIC250 code.

The converted instructions are displayed via the PIC monitor.

For linking operations the following instructions are available:

Log. AND	AND	;11001010 AND 11110000 = 11000000
Log. OR	OR	;11001010 OR 11110000 = 11111010
Negation	NOT	;NOT 11001010 = 00110101
Equality	=	;Byte comparison (bit-by-bit)
Assignment	=	;Source = destination
Multiple assignment	:=	;Source = Dest1, Source = Dest2, etc.

;If only one bit is specified with assignment or multiple assignment, all bits of the byte / word will be set to this status.

Defining Variables and Constants:

CONSTANT:	BYTEMASK= 2#00001111,	;Byte in bin. format
	WORD0FFF = 16#0FFF	;Word in hex format
BINARY:	192=BINMARK1,193=BINMARK2	;M24.0 .1
WORD:	256=MWORD1, 272=MWORD2	;M32/33,M34/35
OUTPUT WORD:	728=DIGOUT	;O91.0–92.7
OUTPUT BYTE:	408 = GANZ1	;O51.0–51.7
OUTPUT:	560=E97, 561=E98,562=E99	;O70.0–70.2

Program:

;Value assignments:

DIGOUT = 1	;Set all 16 outputs
GANZ1 = BYTEMASK AND GANZ1	;Mask bits 4–7
GANZ1 = BYTEMASK OR GANZ1	;Set bits 0 through 3

;Multiple assignment:

BINMARK1 := BINMARK2 := 0	;BINMARK 1 & 2 = 0
E97 := E98 := E99 := 1	;RC Inputs 97-99 = 1

;Write 0FFFH to MWORD1, MWORD2 and DIGOUT

MWORD2 := DIGOUT := WORD0FFF

MWORD1 := MWORD2 := 1 ;all bits, M32 through 35

= 1

6.17 Programming Example – Fields

FIELDS

Function:

It is possible to define fields of bit, byte or word variables
 The field index provides a default range for the processing variable.
 In the event that the field components are arranged in linear succession, the specification of one start address will suffice.

In the case of irregular succession of field components, all addresses must be specified in ascending order.
 Access to the field components is effected by means of the name of the variables, followed by their respective indexes enclosed in square brackets, e.g. **Var [1]**.

Definition:

```

ARRAY [ 1 .. 32 ] BINARY :
    (16..23, 64..71, 112..119, 160..167) = t_start_m ;Start Timer [n]
    ;M2.0–M2.7, M8.0–M8.7, M14.0–M14.7, M20.0–M20.7
ARRAY [ 0 .. 3 ] BYTE :      200 = ByteArray
    ;(M25 – M28)
ARRAY [ 1 .. 6 ] INPUT :
    (42, 30, 28, 26, 24, 17) = JOG_P_PHG
    ;E5.2, E3.6, E3.4, E3.2, E3.0, E2.1 als JOG_P_PHG [1]..[6] –Taste
    
```

Program:

```

t_start_m[1] := t_start_m[5] := 1           ;Start time1 and time2

ByteArray = 0           ;Delete entire byte field (M25.0 – M28.7)

;Passing-through PHG keys
JOG_M_RCE[1] = JOG_M_PHG[1] AND NOT JOG_P_PHG[1]
.
JOG_M_RCE[6] = JOG_M_PHG[6] AND NOT JOG_P_PHG[6]

JOG_P_RCE[1] = JOG_P_PHG[1] AND NOT JOG_M_PHG[1]
.
JOG_P_RCE[6] = JOG_P_PHG[6] AND NOT JOG_M_PHG[6]
    
```

6.18 Programming Example – Compare Operations

COMPARE Operations**Function:**

Available compare operations are limited to detecting equality or inequality (by negating the entire compare expression).
The operators for compare operations can be variables as well as constants!

Definition:

```
CONSTANT:  MAX = 128,  
            MIN = 4  
INPUT BYTE: 136 = A1_8PHG      ;RC output 1 – 8 from PG  
                                     ;(E17.0 .. E1 7.7)  
TEMP WORD:  MWORT             ;Marker word  
OUTPUT:     704 = A1          ;Digital output 1
```

Program:

```
IF A1_8PHG = MAX THEN A1 = 0  
IF A1_8PHG = MIN THEN A1 = 1  
  
IF NOT (MWORD = MAX) THEN  
  BEGIN  
  .  
  END
```



7 PIC & PROFI Software

7.1 Loading and Saving with PROFI Software

The PROFI software can also be used to create PIC programs. It should be stated, however, that the PROFI environment does not provide ONLINE service or monitoring operation for PIC programs under PROFI.

Subsequent to editing a **.P2T** source file, the PIC programs can only be "assigned." This means that subsequently to an error-free assignment, they are stored, complete with absolute addresses, in a **.P2O** file.

For loading and unloading, two batch files are available. Their names are **LOAD_PIC.BAT** and **SAVE_PIC.BAT**. They are located in the **[ROPS3]** directory. To facilitate handling of these batch files, they should be integrated into the PROFI environment in the form of **applications**.

7.2 Installing the Batch Files

The following section describes the required installation procedure for **LOAD_PIC.BAT**. The integration of **SAVE_PIC.BAT** follows a similar pattern. Effective with PROFI software version 3.0, the .P20 file format was changed. The new file format will be recognized by **LOAD_PIC.BAT** and appropriately converted.

When working with the PROFI software, the Help function can be invoked at any time by pressing the **<F10> function key**.

To install the batch file, proceed as follows:

- ★ Start the PROFI software.
- ★ From the main menu, press the **<F7> function key** to select the **Appl.** (Application) menu command.
- ★ Press the **<F2> function key** to select the **Change** menu command.
- ★ A submenu opens in which up to nine application programs can be installed. To select an available program slot, the corresponding function key must be pressed.
- ★ As the next step, the system expects a user-defined title to be entered, e.g. "Load rho3." Type in the title, and confirm the entry by pressing **<Enter>**.
- ★ Now the program name can be entered. For batch files, type the following:
C:\COMMAND.COM
To confirm your entry, press **<Enter>**.
- ★ Next, the parameter will be specified.
Type: **/c C:\ROPS3\LOAD_PIC.BAT %1 %2**
To confirm your entry, press **<Enter>**.
- ★ As a last step, press **<A>** to activate **Pause**.
- ★ To confirm your entry, press **<Enter>**.
- ★ This concludes the installation.



7.3 Operation from within the Main Menu

- ★ To select the **Appl.** (Application) command, press **<F7>**.
- ★ To select **Execute**, press **<F1>**.
- ★ To select the **Load rho3** title, move the cursor to the title, and press the function key to the right of the title (**F1 – F9**).
- ★ Select **Parameter %1** DOS filename (*without filename extension*).
To confirm your entry, press **<Enter>**.

In the case of filenames following DOS conventions, a pathname can be specified; if no directory is specified, the current path will be used.

The specification of a filename extension will cause the batch process to be aborted!

- ★ Select **Parameter %2** rho filename (*without filename extension*).
To confirm your entry, press **<Enter>**.

Pathname specifications are not permissible on the rho robot control!

The batch process will now be started and executed. The results will be displayed on the monitor.

To return to PROFI, press **<ESC>**.



Selecting the Save rho3 option will cause the parameters %1 (*DOS filename*) and %2 (*rho filename*) to be switched in accordance with the change in transfer direction, i.e., Parameter %1 represents the rho filename, and Parameter %2 the DOS filename.

7.4 Standard PIC Programs for PROFI Software

Effective with PROFI software version **W3B**, the PROFI programs are installed in their own subdirectories.

The PIC program, identified by filename **R3_2D.P2O**, is located in the [ROPS3\r3_2.200] directory. The associated symbol file is named **R3_.S2S**.

The standard PLC programs for the **CL300**, **CL400** and **CL500** are located in the subdirectories named **r3_3.300**, **r3_4.400** und **r3_5.500**.

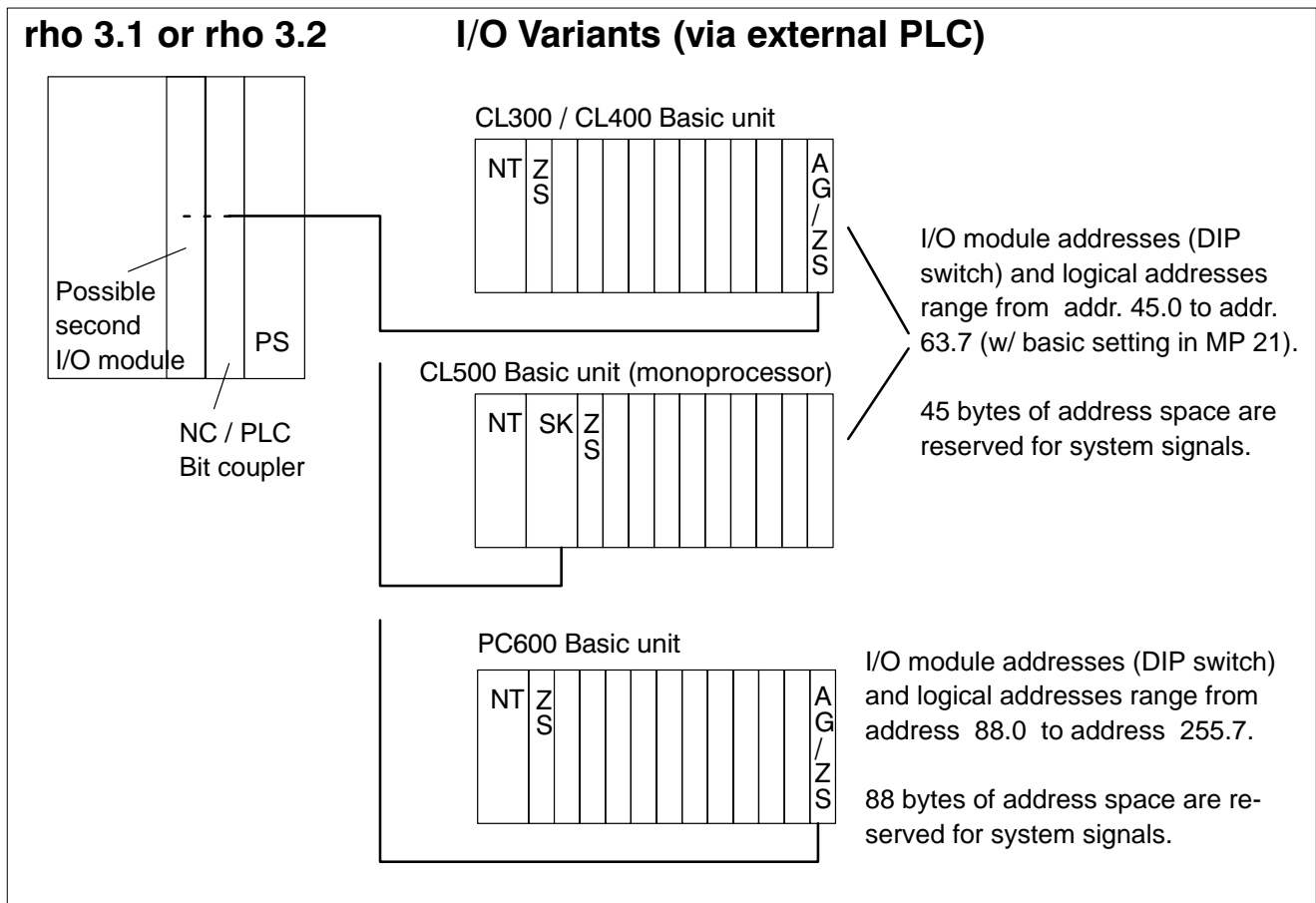
The standard programs for programming in BAPSPIC are located in the directory named **ROPS3\BAPSPIC**.

8 Bit Coupler

8.1 Bit Coupling – PIC Implementation in External PLC

The bit coupler card provides a bus connection between the rho 3 controller and the PLC. It is used for a cyclical exchange of interface signals between the rho 3 and the PLC. The clock rate is provided by the I/O cycle of the PLC). The data exchange must be completed with a timing range of $3\text{ ms} < x < 250\text{ ms}$. In the event of a shortfall of the time threshold, the rho 3 will return a time error message indicating "PLC program runtime < 2 ms."

If an objective is the sparse use of binary I/O addresses, the bit coupler can also be addressed to the extended input/output of the PLC.



The bit coupling arrangement with the PC600 basic unit provides the same address range as the PIC250 module.

In the case of the CL300, CL400 and CL500 controllers, the reduced address range imposes the restriction of a reduced interface with the rho 3 (refer to *Signal Description*, no.1070 073028).

Any connection between the **CL500** and the rho via bit coupler is permitted **in monoprocessor operation** only.

8.2 Settings on the rho 3

All settings are effected through the modification of machine parameters.

I/O module configuration MP 20:

- 0: Automatic module identification, or
- 2: PC600 bit coupling or CL400 coupling with extended interface
- 12: CL300 / 400 / 500 bit coupling
- 13: CL300 / 400 / 500 bit coupling
+ 2nd I/O module for direct I/O via MP 17 and/or 18
- 14: PC600 bit coupling
+ 2nd I/O module for direct I/O via MP 17 and/or 18
- 15: CL300 / 400 / 500 bit coupling + 2nd I/O module (via PLC)
- 16: PC600 bit coupling + 2nd I/O module (via PLC)
- 17: rho3.0 – rack unit in PLC
- 18: rho3.0 – Stand-alone unit
- 19: rho3.0 – rack unit in PLC (with extended interface for CL400)

re: 13, 14 The signals of the additional I/O module can be processed in the rho only if they were declared as direct I/O in machine parameter 17 and/or 18.

re: 15, 16 The signals of the additional I/O module (8 bytes inputs and 5 bytes outputs) are placed directly behind the address range of the bit coupler. (Example: CL300: I45 – I52, and O45 – O49.) If only one 32I / 24O module is used, the range of 8 input and 5 output bytes must still be reserved.

PLC Bit coupling address range MPP 21:

This is the address specification for the standard interface of the rho 3 in the PLC. This range cannot be used by application signals (external I/O).

In normal circumstances the range encompasses 45 bytes (reduced CL300 interface) and the address ranges for inputs and outputs are using the same addresses. The input and output ranges can be declared separately. In the event that an additional number of peripheral signals is needed, the final address of the rho3 interface can be reduced. This will also reduce the number of user inputs/outputs of the rho3 interface.

The following settings are available:

		Default setting
Start address	Outputs:	0
End address	Outputs:	44
Start address	Inputs:	0
End address	Inputs:	44



Examples with additional I/O module in the rack of the rho 3
(configuration 15 and/or 16)

	CL300, 400, 500	PC600
Start address outputs (bit coupler):	0	0
End address outputs (bit coupler):	44	87
Start address inputs (bit coupler):	0	0
End address inputs (bit coupler):	44	87
Start address outputs (2nd I/O module):	45	88
End address outputs (2nd I/O module):	49	92
Start address inputs (2nd I/O module):	45	88
End address inputs (2nd I/O module):	52	95

8.3 PLC Settings

The bit coupler card is connected by means of a cable with the PLC expansion module in the CL300, 400, and 500 controllers. Addressing uses the I/O cycle of the PLC. No hardware settings are required.

8.4 Bit Coupler in I/O Field

In the I/O field of the PLC, the bit coupler occupies the input and output addresses as specified in MP 21. These addresses are no longer available for digital I/O signals.

The inputs and outputs are **automatically** exchanged during the I/O cycle.

Order number for bit coupler in I/O field:	050 100
Order number for bit coupler in I/O field, incl. CAN:	050 110

8.5 Bit Coupler in EI/EO Fields

In the EI/EO field of the PLC, the bit coupler occupies the input/output addresses as specified in MP 21.

The inputs and outputs are **not automatically** exchanged in the course of the I/O cycle, but must be loaded from the PLC with the **EZAZLOAD.PxO** pörogram module. In the PLC, the interface signals of the rho3 are located within the marker range (default configuration: inputs M46 and up, outputs M0 and up). The marker addresses for input and output range are defined by parameters P0 and P1 of the EZAZLOAD program module.

P0	Ind. address for rho3 interface inputs (copied from EI2 through EI46)
P1	Ind. address for rho3 interface outputs (copied to EO2 through EO46)

Addresses EI1.0 through EI1.7 are reserved for peripheral interrupts. For this reason, the bit coupler addresses start with EI2 and EO2.

Effective with ROPS version W3B, the program module is located in the controller-specific directory named `.vr3_x.x00` (x= 3, 4, 5 for CL300, CL400 or CL500).

Order number for bit coupler in EI/EO field: 050 105

With the EI/EO variant, the use of two bit couplers in the PLC is possible:

1st rho3	bit coupler in I/O field
2nd rho3	bit coupler in E/EO field

or

1st rho3	bit coupler in EI/EO field
2nd rho3	bit coupler in EI/EO field

8.6 Special Considerations – Bit Coupler in the CL400

The CL400 executes the I/O cycle only up to the last address up to which module were identified in the course of system start-up. However, due to the address specification via machine parameter, the bit coupler is not automatically recognized. For this reason, the **OB2** must be activated and modified as follows:

Entry 2: Bit 2 = 1 I/O state to be executed as per nominal assignment list.

The nominal assignment is defined from entry 33 and/or 49 upward:

Entries, 33 and up: 1 bit per each assigned input byte.

Entries, 49 and up: 1 bit per each assigned output byte

As a consequence of the extremely short command processing time of the CL400, it may in certain circumstances become necessary to extend the cycle time of the CL400 to a minimum of **3 ms**. This can be effected by invoking the **ZEITPB0** module in the OB5, OB7 and OB1 modules.

CL400 with PC600 interface:

Similar to the PC600, the CL400 features an address range of 255 bytes for inputs and outputs. However, the automatic exchange during the I/O cycle is limited to 128 input and 64 output bytes. The I/O's situated above this range must be separately addressed within the application program via indirect addressing.

Effective with ROPS version W3C, a standard program for passing through the expanded address range is available.



9 rho 3.0 <--> PLC Interface

9.1 Introduction

In the event that the rho 3.0 is **operated as a rack unit in the PLC**, the data exchange between the rho 3.0 and the PLC is handled by a communications module. The **KOMFIFO** communications module is available for the CL300, CL400 and CL500 controls. Adapted to the respective control, and effective with ROPS version W3B, this communications module is located in the control-specific directories named **lr3_x.x00** (where x= 3, 4, 5 for CL300, CL400 and CL500, respectively).

The data can be saved on the PLC within marker, data module, data buffer or data field areas.

Accordingly, the rho 3.0 will require the reduced CL300 interface assignment (46 bytes). (Refer also to *Signal Description*, no. 1070 073028).

The communications module occupies, starting with the module address specified on the rho 3.0, **4 bytes each in the (EI) and (EO) range** of the PLC.

EI: Extended input, EO : Extended output

The data exchange must be completed within a time interval of 3 ms < x < 250 ms, otherwise the PLC will return a rho 3.0 PLC timeout error.

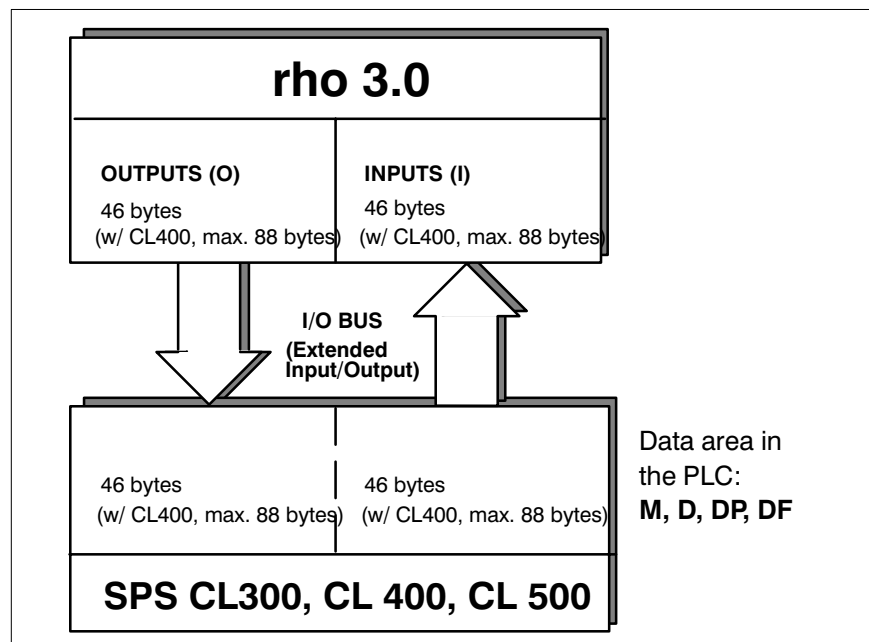


Fig. 5 rho3.0 and SPS Interface



The ROPS software packet contains standard interface programs for use as a basis for user-defined applications.

9.2 KOMFIFO Module for rho 3.0 / PLC Communications

The **KOMFIFO** module, in conjunction with the **DBLOAD** module, effects the data transfer between the rho 3.0 and the PLC via a data module. The respective modules are invoked directly by the rho application program.

Beginning with software version TO06, the KOMFIFO and DBLOAD modules are provided on the ROPS diskettes. Subsequent to the completed software installation, they can be accessed in the ROPS3 directory.

9.2.1 KOMFIFO Module Parameters:

P0	W	START Input No data transfer will take place as long as P0=0. Transfer starts when P0=1 and rho 3.0 Ready.
P1	W	Centralized / Decentralized Application 0: centralized (in basic or expansion unit), 1: decentralized (via RM 4 module)
P2	W	Module address (as per DIP switch setting on rho 3.0) with decentralized arrangement: Busmaster address
P3	W	Busmaster FIFO number (decentralized structure only) From Busmaster address assignment list
P4	W	FIFO buffer depth (decentralized structure only) Block length of data to be transferred (max. 48 bytes)
P5	W	Data volume/count with rho 3.0, fixed at 46 bytes selectable from TO07 and up: 88 bytes for CL400 interface
P6	W	Status and acknowledge marker <i>Bit 0</i> is set to 1 subsequent to successful data transfer. If no transfer is active, and/or at the start of transfer, the status marker is reset. <i>Bit 10</i> indicates if rho 3.0 module is ready. Bit 10 = 1 : Module ready Bit 10 = 0 : Module not ready

9.2.2 DBLOAD Module Parameters:

P0	W	Indirect address for Receive data rho outputs <=> PLC inputs Example: P0 WK12EH = M 46
P1	W	Indirect address for Send data rho outputs <=> PLC outputs Example: P1 W K100H = M 0

The data exchange is handled via a data module. At the time that DBLOAD and KOMFIFO are invoked, this data module must already be opened.



9.2.3 Exchanging Interface Signals with the PLC

- The rho 3.0 occupies 4 bytes in the EI/EO range on the I/O bus. These addresses are used for the data exchange with the PLC.
- Memory capacity must be provided in the PLC for the interface signals. The defined memory ranges can be marker, data module, data buffer or data field ranges.



In the event that a data module is utilized, it must already be opened when the module is invoked.

- The number of transferred bytes must be set to 46 bytes (standard interface). Beginning with version TO07, 88 bytes can also be transferred (expanded interface).

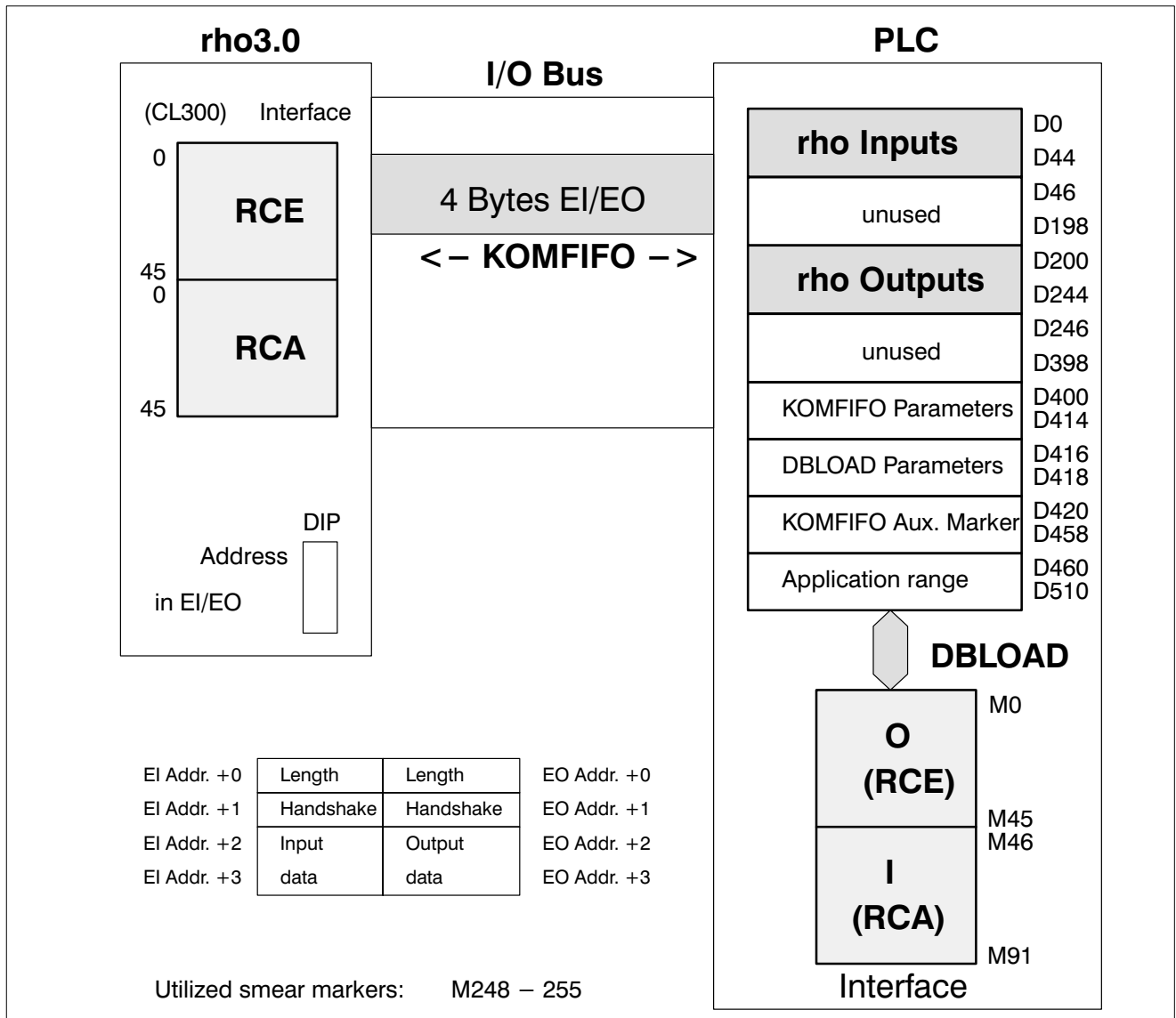


Fig. 6 Data Transfer via KOMFIFO Module

9.2.4 Example of a rho 3.0 <--> SPS Data Exchange

Program in OM1 of the PLC

```

CM      -RHO3DAT ; Data module for rho inputs / outputs

CM      -DBLOAD,2; reads I/O data in RHO3DAT DM
P0  W   K12EH   ; Index address for PLC input field (M46)
P1  W   K100H   ; Index address for PLC output field (M0)

CM      -KOMFIFO,7 ; Communications module for rho 3.0
P0  W   K1       ; Start inputs
P1  W   K0       ; 0 = centralized, 1 = decentralized
P2  W   K0       ; Modul address (DIP switch, rho 3.0)
P3  W   K0       ; Busmaster FIFO no. (if decentralized)
P4  W   K48      ; FIFO depth (max. 48) only when decentralized
P5  W   K46      ; Number of data bytes (46=CL300 interface)
P6  W   D412     ; Status and acknowledge word

CM      -R3_ANW   ; rho 3.0 Application program

```

The rho 3.0 <--> PLC interface occupies 46 bytes of input and output range. For this data, operands **M**, **D**, **DP** or **DF** define a mapping range. This range first maps the rho 3.0 inputs and then the rho 3.0 outputs.



With the use of the ranges D, DP or DF as interface memory, no bit-wise access to the interface signals. Bit signals must be queried via a calculating register.

Example for CL400, CL500:

Querying interface signal:

```

CM      DM5       ; Activate data module
L       D0,0      ; Load data register 0
A       O.3       ; Query bit 3
.
=       M5.7

```

Setting interface signal:

```

L       D40,B     ; Load data register 0
A       I2.5
.
=OM     B.3       ; Set bit 3 in register
T       B,D40

```




9.3 Start-up Characteristics with Several rho 3.0 in the PLC

The data exchange between the rho 3.0 and the ZS central processing unit can commence only after both controls have completed their start-up sequences. The rho 3.0 (and other intelligent peripheral bus modules, such as CC10.3, Typ 1 osa or the BM DP Busmaster) use the **I/OINIT** bus signal to inform the ZS400 that the module is in the start-up phase.

If the I/OINIT signal is active, the PLC waits until the I/OINIT signals of all modules go LOW. This causes the PLC to start up, and the communication is started. In the case of the rho version TO07I, this means that the rho that first completes its start-up sequence will enter into a timeout because it initially failed to receive a response from the PLC.

In the TO07I version, to cause the rho to reinitiate communications automatically, the following program part (e.g. at the end of the KOMFIFO module) must be extended thus:

```
AN      SM20.0
JPC     -end
L      W   K0,O
T      W   A,D424
```

Beginning with rho version TO0J, the occurrence of the timeout error during start-up is delayed for 80 ms. Subsequent to the first transfer (KOMFIFO), the previous timeout value is again reset.

In the case of decentralized operation of the rho 3.0, the PLC timeout error also occurs in the case of different start-up characteristics even in the TO08E version.

Beginning with version 2.3 of the ZS400 operating system, the I/OINIT signal can be disabled in the OM2 during the start-up phase.

With the I/OINIT signal disabled, the PLC starts immediately. The rho controls are serviced immediately after their start-up sequence (KOMFIFO).

To release the rho-specific signals, bit 10 (RC Ready) can be used by P6 (KOMFIFO). See also *rho 3.0 Connectivity Requirements and Project Planning Information*, no.1070 073 350).

9.4 rho 3.0 in Decentralized Application

Effective with the TO08E version, the operating system supports the decentralized operation of the rho 3.0 in an expansion unit that is connected via PROFIBUS. The KOMFIFO module (interconnecting the rho 3.0 and the PLC) was expanded accordingly.

On the PLC side, the required support is provided by the following modules:

BM DP12 Busmaster in the higher-level PLC

RM4 DP12 Remote module in the decentralized expansion unit

In the directory with the pathname `\ROPS3\DEZENTRL.C00`, two example PLC programs (standard 46-byte interface and expanded 88-byte interface) are available.

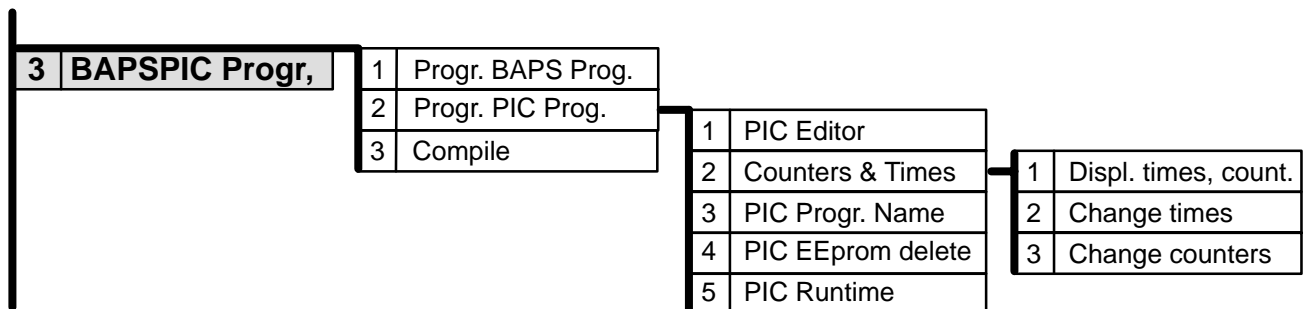
As the data transfer is restricted to a maximum of 48 bytes per transmission, the use of the expanded interface will require transmission in block mode.

Another example in the directory named `\ROPS3\ZEN_DEZ.C00` describes the mixed operation including one rho 3.0 in centralized, and one rho 3.0 in decentralized operation.

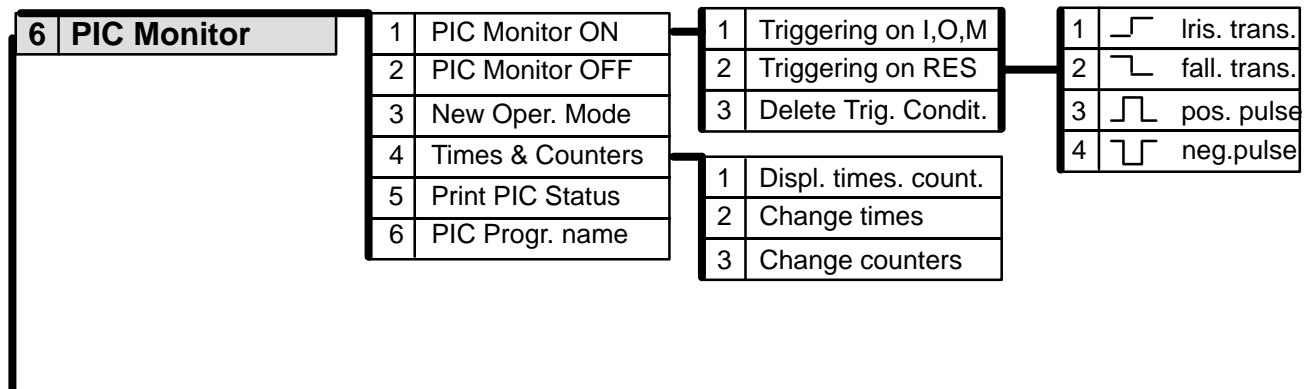


10 Menu Tree / Interface Programs

10.1 Excerpt, PHG Menu Tree



Attention: Mode 6 is not supported with SoftPIC!



Delete OIC on rho 3.0 Stop PIC Program

Deadman+ Alt + Mode + 3	Deadman + Alt + Mode + 4
-------------------------	--------------------------

10.2 Table of available Interface Programs

Control	Programming	File designations	Function
PIC 250 for rho 3.1 and rho 3.2 and/or SOFTPIC for rho 3.0	BAPSPIC German in directory named: ROPS.W3x\BAPSPIC	R3_xy_D.QLS R3S_xy_D.INC Kxy_PHGD.INC R3_xy_D.P2O R3_xy_D.P2X	Source program Symbol file PHG2000 Symbols (from version O07) Object file File to be loaded
	BAPSPIC English in directory named: ROPS.W3x\BAPSPIC	R3_xy_E.QLS R3S_xy_E.INC Kxy_PHGE.INC R3_xy_E.P2O R3_xy_E.P2X	Source program Symbol file PHG2000 Symbols (from version TO07) Object file File to be loaded
	PROFI (IL) for PIC in directory: ROPS.W3x\R3_2.200	R3_xy.P2O R3_xy.P2X R3_xy.S2S	Object file File to be loaded Symbol file
New with version W3D	BAPSPIC2 German in directory named: ROPS.W3x\BAPSPIC from BAPSPIC 2.20	R3_xy2D.QLS R3S_xy2D.INC Kxy_PHGD.INC MF.INC R3_xy2D.P2O R3_xy2D.P2X MF.QLS MF.P2O MF.P2X	Source program Symbol file PHG2000 Symbols (from version TO07) Symbols multi-function I/O Object file File to be loaded Source program multi-function I/O Object file multi-function I/O File to be loaded multi-function I/O
	BAPSPIC2 English in directory named: ROPS.W3x\BAPSPIC from BAPSPIC 2.20	R3_xy2E.QLS R3S_xy2E.INC Kxy_PHGE.INC R3_xy2E.P2O R3_xy2E.P2X	Source program Symbol file PHG2000 Symbols (from version TO07) Object file File to be loaded
CL300			
CL 300 for rho 3.1 and rho 3.2 bit coupling	PROFI (IL) for CL300 in directory named: ROPS.W3x\R3_3.300	R3_xy.P3O R31_OB1.P3O ZEITPB0.P3O EZAZLOAD.P3O R31_xy.P3A R31_xy.S3S	Standard interface module for all rho 3.0 Organization module for rho 3.1 / 3.2 Time module for cycle time adaptation Connection via extended field Only when bit coupling, via extended field File to be loaded Symbol file
CL 300 for rho 3.0 in rackmount version	PROFI (IL) for CL300 in directory named: ROPS.W3x\R3_3.300	R3_xy.P3O R30_OB1.P3O KOMFIFO.P3L DBLOAD.P3O R30_xy.P3A R30_xy.S3S	Standard interface module for all rho 3.0 Organization module for rho 3.0 Function module for rho 3.0 connection Load Data module for rho 3.0 connection File to be loaded Symbol file



Control	Programming	File designations	Function
CL400			
CL 400 for rho 3.1 and rho 3.2 bit coupling	PROFI (IL) for CL400 in directory named: ROPS.W3x\R3_4.C00	R3_xy.PCO R31_OB1.PCO OB2.PCO R31_OB5.PCO ZEITPB0.PCO R31_xy.PCA R31_xy.SCS	Standard interface module for all rho 3.0 Organization module for rho 3.1/3.2 System settings for CL400 Start-up module for run-up after Power-ON Time module for cycle time adaptation File to be loaded Symbol file
CL 400 for rho 3.1 bit coupling with extended Interface	PROFI (IL) for CL400 in directory named: ROPS.W3x\R3_6.C00	R3_xy.PCO R31OB1_6.PCO OB2.PCO A_LOAD_6.PCO R31_OB5.PCO ZEITPB0.PCO R31_xy_6.PCA R31_xy_6.SCS	Standard interface module for all rho 3.0 Organization module for rho 3.1 extd interface System settings for CL400 Connection Bitkoppler im E/A–Feld Start-up module for run-up after Power-ON Time module for cycle time adaptation File to be loaded, extended interface Symbol file, extended interface
CL 400 for rho 3.1 and rho 3.2 bit coupling via extended I/O range	PROFI (IL) for CL400 in directory named: ROPS.W3x\R3_4.C00	R3_xy.PCO R31Z_OB1.PCO EZAZLOAD.PCO OB2.PCO R31_OB5.PCO ZEITPB0.PCO R31Z_xy.PCA R31Z_xy.SCS	Standard interface module for all rho 3.0 Organization module for rho 3.1/3.2 Connection via EI/EO extension System settings for CL400 Start-up module for run-up after Power-ON Time module for cycle time adaptation File to be loaded Symbol file
CL 400 for rho 3.1 bit coupling via ext. I/O range with externed interface	PROFI (IL) for CL400 in directory named: ROPS.W3x\R3_6.C00	R3_xy.PCO R31OB1Z6.PCO EZAZLD_6.PCO OB2.PCO R31_OB5.PCO ZEITPB0.PCO R31Zxy_6.PCA R31Zxy_6.SCS	Standard interface module for all rho 3.0 Organization module for rho 3.1/3.2 Connection via EI/EO extension System settings for CL400 Start-up module for run-up after Power-ON Time module for cycle time adaptation File to be loaded, extended interface Symbol file, extended interface
CL 400 for rho 3.0 rackmount version	PROFI (IL) for CL400 in directory named: ROPS.W3x\R3_4.C00	R3_xy.PCO R30_OB1.PCO KOMFIFO.PCL DBLOAD.PCO ZEITPB0.PCO R30_xy.PCA R30_xy.SCS	Standard interface module for all rho 3.0 Organization module for rho 3.0 Function module for rho 3.0 connection Load data module for rho 3.0 connection Time module for cycle time adaptation File to be loaded Symbol file
CL 400 for rho 3.0 rackmount version w/o M0..M90 Marker assignment	PROFI (IL) for CL400 in directory named: ROPS.W3x\R3_4.C00	R3_xy_DB.PCO R30D_OB1.PCO KOMFIFO.PCL ZEITPB0.PCO R30_xy.PCA R30_xy.SCS	Standard interface module for all rho 3.0 without marker assignment Organization module for rho 3.0 without marker assignment Function module for rho 3.0 connection Time module for cycle time adaptation File to be loaded Symbol file without marker assignment

Control	Programming	File designations	Function
CL 400 for rho 3.0 rackmount version with extended Interface	PROFI (IL) for CL400 in directory named: ROPS.W3x\R3_6.C00	R3_xy.PCO R30OB1_6.PCO KOMFIF0.PCL DBLOAD_6.PCO ZEITPB0.PCO R30_xy_6.PCA R30_xy_6.SCS	Standard interface module for all rho 3.0 Organization module for rho 3.0 Function module for rho 3.0 connection Load data module for rho 3.0 connection Time module for cycle time adaptation File to be loaded, extended interface Symbol file, extended interface
CL 400 for rho 3.0 rackmount version with extended Interface w/o marker assignment	PROFI (IL) for CL400 in directory named: ROPS.W3x\R3_6.C00	R3_xy6DB.PCO R30D_OB1.PCO KOMFIFO.PCL ZEITPB0.PCO R30xy6DB.PCA R30xy6DB.SCS	Standard interface module for all rho 3.0 without marker assignment Organization module for rho 3.0 without marker assignment Function module for rho 3.0 connection Time module for cycle time adaptation File to be loaded Symbol file without marker assignment
CL500			
CL 500 for rho 3.1 and rho 3.2 bit coupling	PROFI (IL) for CL500 in directory named: ROPS.W3x\R3_5.500	R3_xy.P5O R31_OB1.P5O EZAZLOAD.P5O ZEITPB0.P5O R31_xy.P5A R31_xy.S5S	Standard interface module for all rho 3.0 Organization module for rho 3.1/3.2 Only with use of bit coupler in EI/EO extension Time module for cycle time adaptation File to be loaded Symbol file
CL 500 for rho 3.0 rackmount version	PROFI (IL) for CL500	R3_xy.P5O R30_OB1.P5O KOMFIFO.P5O DBLOAD.P5O ZEITPB0.P5O R30_xy.P5A R30_xy.S5S	Standard interface module for all rho 3.0 Organization module for rho 3.0 Function module for rho 3.0 connection Load data module for rho 3.0 connection Time module for cycle time adaptation File to be loaded Symbol file
CL 500 for 2 rho 3.0 rackmount versions	PROFI (IL) for CL500	R3_xy.P5O 2R3_xy.P5O 2R3_OB1.P5O KOMFIFO.P5O DBLOAD.P5O 2R3_xy.P5A 2R3_xy.S5S	Standard interface module for 1. rho3 Standard interface module for 2. rho3 Organization module for rho 3.0 Function module for rho 3.0 connection Load data module for rho 3.0 connection File to be loaded Symbol file



Decentralized operation: New with rho3.0, T008E version			
CL 400 for rho 3.0 rackmount version, decentralized operation w/ Profibus DP	PROFI (IL) for CL400 in directory named: ROPS.W3x \DEZENTRL.C00	R3_xy.PCO R30OB1D4.PCO KOMFIFO.PCL DBLOAD_6.PCO ZEITPB0.PCO R30_xyD4.PCA R30_xyD4.SCS	Standard interface module for all rho 3.0 Organization module for rho 3.0 Function module for rho 3.0 connection Load data module for rho 3.0 connection Time module for cycle time adaptation File to be loaded Symbol file
CL 400 for rho 3.0 rackmount version with extended interface, decentralized operation w/ Profibus DP	PROFI (IL) for CL400 in directory named: ROPS.W3x \DEZENTRL.C00	R3_xy.PCO R30OB1_D.PCO KOMFIFO.PCL FIFODM1.PCL DBLOAD_D.PCO ZEITPB0.PCO R30_xy_D.PCA R30_xy_D.SCS	Standard interface module for all rho 3.0 Organization module for rho 3.0 Function module for rho 3.0 connection Function module for rho 3.0 connection Datenbaustein laden for rho 3.0 connection Time module for cycle time adaptation File to be loaded, extended interface Symbol file, extended interface
CL 400 for rho 3.0 rackmount version, decentralized and centralized operation.	PROFI (IL) for CL400 in directory named: ROPS.W3x \ZEN_DEZ.C00	R3_xy.PCO 2R3_xy.PCO OB1ZENDE.PCO KOMFIFO.PCL FIFODM1.PCL DBLOAD.PCO ZEITPB0.PCO ZEN_DEZ.PCA ZEN_DEZ.SCS	Standard interface module for 1. rho3 Standard interface module for 2. rho3 Organization module for rho 3.0 Function module for rho 3.0 connection Function module for rho 3.0 connection Load data module for rho 3.0 connection Time module for cycle time adaptation File to be loaded, extended interface Symbol file, extended interface

x : Software version number currently **6**

y : Software index (successive letter code) used for indexing



PCL software programs for the CL400, enabling the use of the PC600 interface for rho3.1 and rho3.0, are available effective with ROPS software version W3C.

Personal Notes:



A Appendix

A.1 Abbreviations

ESC	Electrostatically sensitive components.
ESD	Electrostatic Discharge Abbreviation occurring in many terms referring to electrostatic discharges, e.g. ESD protection, ESD hazard, etc.
PE	Protective Earth

A.2 SPS-Terminologie Deutsch/Englisch

Operanden/Operands

Deutsch		Englisch	
AST	Anwender-Stack	AST	Application stack
AWP	Anwenderprogrammzähler	UPP	User program pointer counter
A	Ausgang	O	Output
AZ	Ausgangszusatzfeld	EO	Extended output
D	Datum	D	Data
DB	Datenbaustein	DM	Data module
DF	Datenfeld	DF	Data field
DP	Datenpuffer	DB	Data buffer
E	Eingang	I	Input
EZ	Eingangszusatzfeld	EI	Extended input
F	Fehler	E	Error
FI	FIFO-Operand (Warteschlange)	FI	FIFO operand
IA	Interface-Ausgang	IO	Interface output
IE	Interface-Eingang	II	Interface input
K	Konstante	K	Constant
KD	Doppelwort-Konstante	KD	Constant double word
KF	Gleitkomma-Konstante	KF	Constant floating point
KME	Koordinierungsmerker einfach	CFS	Coordination flag single
KMP	Koordinierungsmerker permanent	CFP	Coordination flag permanent
Kx.y	Zeitkonstante	Kx.y	Constant of time
M	Merker	M	Marker
'nr'	Parameternummer	'nr'	Number as parameter
P	Parameter	P	Parameter
PI	Peripherieinterrupt	PI	Peripheral interrupt
S	Systembereich	S	System range
SI	Systeminterrupt	SI	System interrupt
SM	Sondermerker	SM	Special marker
T	Zeit	T	Time
TI	Zeitinterrupt	TI	Time interrupt
Z	Zähler	C	Counter
\$	direkte Adreßeingabe für die Befehle L und T	\$	Operand absolute
-xx	Symbolischer Operand	-xx	Symbol

**Befehle/Instructions**

Deutsch		Englisch	
ADC	Addition mit Carry	ADC	Addition with carry
ADD	Addition	ADD	Addition
AF	Alarm freigeben	AE	Alarm enable
AS	Alarm sperren	AD	Alarm disable
BA	Baustein aufruf unbedingt	CM	Call module
BAAG	Baustein aufruf arithmetisch größer, AG=1	CMAG	Call module arithmetical greater, AG=1
BAB	Baustein aufruf bedingt, VKE=1	CMC	Call module conditional, RES=1
BAC	Baustein aufruf Carry, C=1	CMCY	Call module carry, C=1
BACN	Baustein aufruf Carry nicht, C=0	CMCN	Call module carry not, C=0
BACZ	Baustein aufruf Carry oder Null, C=1 oder Z=1	CMCZ	Call module carry or zero, C=1 or Z=1
BAI	Baustein aufruf invers, VKE=0	CMCI	Call module conditional invers, RES=0
BALG	Baustein aufruf logisch größer, LG=1	CMLG	Call module logical greater, LG=1
BAM	Baustein aufruf Minus, N=1	CMM	Call module minus, N=1
BAMZ	Baustein aufruf Minus oder Null, N=1 oder Z=1	CMMZ	Call module minus or zero, N=1 or Z=1
BAN	Baustein aufruf nicht Null, Z=0	CMN	Call module not zero, Z=0
BAO	Baustein aufruf Overflow, O=1	CMO	Call module overflow, O=1
BAON	Baustein aufruf Overflow nicht, O=0	CMON	Call module overflow not, O=0
BAP	Baustein aufruf Plus, N=0	CMP	Call module plus, N=0
BAX	Baustein aufruf im zweiten Segment	CMX	Call module into second segment
BAZ	Baustein aufruf Null, Z=1	CMZ	Call module zero, Z=1
BE	Baustein ende unbedingt	EM	End of module
BEAG	Baustein ende arithmetisch größer, AG=1	EMAG	End of module arithmetical greater, AG=1
BEB	Baustein ende bedingt, VKE=1	EMC	End of module conditional, RES=1
BEC	Baustein ende Carry, C=1	EMCY	End of module carry, C=1
BECN	Baustein ende Carry nicht, C=0	EMCN	End of module carry not, C=0
BECZ	Baustein ende Carry oder Null, C=1 oder Z=1	EMCZ	End of module carry zero, C=1 or Z=1
BEI	Baustein ende invers, VKE=0	EMI	End of module invers, RES=0
BELG	Baustein ende logisch größer, LG=1	EMLG	End of module logical greater, LG=1
BEM	Baustein ende Minus, N=1	EMM	End of module minus, N=1
BEMZ	Baustein ende Minus oder Null, N=1 oder Z=1	EMMZ	End of module minus Zero, N=1 or Z=1
BEN	Baustein ende nicht Null, Z=0	EMN	End of module not zero, Z=0

Deutsch		Englisch	
BEO	Bausteinende Overflow, O=1	EMO	End of module overflow, O=1
BEON	Bausteinende Overflow nicht, O=0	EMON	End of module overflow Not, O=0
BEP	Bausteinende Plus, N=0	EMP	End of module plus, N=0
BEZ	Bausteinende Null, Z=1	EMZ	End of module zero, Z=1
BID	Wandlung Binär in Dezimal	BID	Binary to decimal conversion
BLA	Blockanfang	SBL	Start of block
BLAA	Blockanfang absolut	SBLA	Start of block absolute
BLE	Blockende	EBL	End of block
BX	2. Datenbaustein aufruf	CX	2nd call data module
BXB	2. Datenbaustein aufruf bedingt, VKE=1	CXC	2nd call data module conditional, RES=1
BXI	2. Datenbaustein aufruf bedingt invers VKE=0	CXCI	2nd call data module conditional invers, RES=0
CH	Tausche unbedingt	CH	Change
CHAG	Tausche arithmetisch größer, AG=1	CHAG	Change arithmetical greater, AG=1
CHB	Tausche bedingt, VKE=1	CHC	Change conditional, RES=1
CHC	Tausche Carry, C=1	CHCY	Change carry, C=1
CHCN	Tausche Carry nicht, C=0	CHCN	Change carry not, C=0
CHCZ	Tausche Carry oder Null, C=1 oder Z=1	CHCZ	Change carry or zero, C=1 or Z=1
CHI	Tausche bedingt invers, VKE=0	CHCI	Change conditional invers, RES=0
CHLG	Tausche logisch größer, LG=1	CHLG	Change logical greater LG=1
CHM	Tausche Minus, N=1	CHM	Change minus, N=1
CHMZ	Tausche Minus oder Null, N=1 oder Z=1	CHMZ	Change minus or zero, N=1 or Z=1
CHN	Tausche nicht Null, Z=0	CHN	Change not zero, Z=0
CHO	Tausche Overflow, O=1	CHO	Change overflow, O=1
CHON	Tausche Overflow nicht, O=0	CHON	Change overflow not, O=0
CHP	Tausche Plus, N=0	CHP	Change plus, N=0
CHZ	Tausche Null, Z=1	CHZ	Change zero, Z=1
CLSB	Lösche Systembefehle	CLSI	Clear system instruction
CMP	Zweier-Komplement	TC	Tow's complement
DBA	Baustein aufruf registerindirekt	DCM	Dynamical call module
DEB	Wandlung Dezimal in Binär	DEB	Decimal to binary conversion
DEC	Dekrement	DEC	Decrement
DEF	Definition	DEF	Define
DEFW	Definition Wort	DEFW	Define word
DI	Sperrern Interruptgruppe	DAI	Disable all interrupts
DIV	Division	DIV	Division



Deutsch		Englisch	
DX		DX	
EI	Freigeben Interruptgruppe	EAI	Enable all interrupts
ERE	Anwenderereignis erreicht	EVA	Event achieved
ERH	Anwenderereignis anfordern im Hintergrund	EVB	Event instruction background
ERS	Anwenderereignis anfordern im Hintergrund mit Systeminterrupt	EVS	Event with system interrupt
ERU	Anwenderereignis anfordern unmittelbar	EVD	Event instruction directly
EXC	Tausche Registerinhalt	EXC	Exchange
FF	Feld freigeben	FR	Field release
FS	Feld schützen	FS	Field save
G	Größer	GT	Greater than
GG	Größer oder gleich	GTE	Greater than or equal
GL	Gleich	EQ	Equal
HLT	Halt	HLT	Halt
IF	Interrupt freigeben	EI	Enable interrupt
INC	Inkrement	INC	Increment
IR	Interrupt rücksetzen (löschen)	RI	Reset interrupt
IS	Interrupt sperren	DI	Disable interrupt
K	Kleiner	LT	Less than
KG	Kleiner oder gleich	LTE	Less than or equal
KL	Kleiner	LT	Less than
L	Laden	L	Load
LABB	Laden Inhalt des Abbildbereiches	LIMR	Load image range
LAH	Laden absolut adressiert im Hintergrund	LAB	Load absolut range in background
LAS	Laden absolut adressiert im Hintergrund mit Systeminterrupt	LAS	LAB with system interrupt
LAU	Laden absolut adressiert unmittelbar	LAD	Load absolut range directly
LFH	Laden feldadressiert im Hintergrund	LFB	Load field in background
LFI	Laden aus FIFO-Speicher	LFI	Load from FIFO
LFS	Laden feldadressiert im Hintergrund mit Systeminterrupt	LFS	LFB with system interrupt
LFU	Laden feldadressiert unmittelbar	LFD	Load field directly
LI	Laden Interruptregister der Interruptgruppe	LAI	Load all interrupts
LM	Laden der Interruptmaske	LIM	Load interrupt mask

Deutsch		Englisch	
LMB	Laden des Inhalts des Memorybereiches	LMB	Load memory band
LMBX	LMB im zweiten Segment	LMBX	LMB into second segment
LO	Leer Oder, entspricht: O(LO	Empty logical or, O=(
LPB	Laden Peripherie Bus	LPB	Load periphery bus
LPC	Laden Programmzähler	LPC	Load program counter
LSP	Laden Stack Pointer	LSP	Load stack pointer
LUZ	Laden Uhrzeit zyklisch	LCC	Load clock cyclical
LUZS	Laden Uhrzeit zyklisch mit Systeminterrupt	LCCS	LCC with system interrupt
LZS	Laden Zeit-Sollwert	LNT	Load normalize time
MUL	Multiplikation	MUL	Multiplication
N	Einer-Komplement	N	Negation, one's complement
NOP0	Leeranweisung 0, 0000H	NOP0	No operation, 0000H
NOP1	Leeranweisung 1, FFFFH	NOP1	No operation, FFFFH
O	Oder	O	Or
ON	Oder nicht	ON	Or not
O(Oder Klammer auf	O(Empty logical or, O(
P	Prüfe Bit	TST	Test
PE	Programmende	EP	End of program
Pi	Parameterfestlegung bei parametrisierten Bausteinanruf, i='nr'	Pi	Parameter line, i='nr'
PN	Prüfe negiert Bit	TSTZ	Test on zero
POP	Transferiere vom Stack	POP	Transfer out from stack
PSi	Parameterfestlegung bei Systembefehlen, i='nr'	PSi	Parameter line of system instructions, i='nr'
PUSH	Lade auf Stack	PUSH	Load into stack
R	Rücksetzen	R	Reset
RC	Rücksetze Carry Flag	RCY	Reset carry
RCL	Rotieren links durch Carry	RCL	Rotate through carry left
RCR	Rotieren rechts durch Carry	RCR	Rotate through carry right
RFI	Rücksetzen FIFO (Lösche FIFO)	RFI	Reset FIFO
RI	Rücksetzen der Interruptregister der Interruptgruppe	RAI	Reset all interrupts
ROL	Rotieren links	ROL	Rotate left
ROM	Rücksetzen ohne Monitoranzeige	RWM	Reset without monitoring
ROR	Rotiere rechts	ROR	Rotate right
RT	Rücksetzen Zeit	RT	Reset time



Deutsch		Englisch	
RZ	Rücksetzen Zähler	RC	Reset counter
S	Setzen	S	Set
SA	Starte Zeit als Ausschaltverzögerung	SF	Start time as falling delay
SAR	Schiebe arithmetisch rechts	SAR	Shift arithmetical to right
SBB	Subtraktion mit borgen	SBB	Subtraction with borrow
SC	Setze Carry Flag	SCY	Set carry
SE	Starte Zeit als Einschaltverzögerung	SR	Start time as raising delay
SI	Starte Zeit als Impuls	SP	Start time as puls
SINT	Sende Interrupt	SINT	Send interrupt
SLL	Schiebe logisch links	SLL	Shift logical to left
SLR	Schiebe logisch rechts	SLR	Shift logical to right
SOM	Setzen ohne Monitoranzeige	SWM	Set without monitoring
SP	Sprung unbedingt	JP	Jump
SPAG	Sprung arithmetisch größer, AG=1	JPAG	Jump arithmetical greater, AG=1
SPB	Sprung bedingt, VKE=1	JPC	Jump conditional, RES=1
SPC	Sprung Carry, C=1	JPCY	Jump carry, C=1
SPCN	Sprung Carry nicht, C=0	JPCN	Jump carry not
SPCZ	Sprung Carry oder Null, C=1 oder Z=1	JPCZ	Jump carry or zero, C=1 or Z=1
SPI	Sprung bedingt invers, VKE=0	JPCI	Jump conditional invers, RES=0
SPLG	Sprung logisch größer, LG=1	JPLG	Jump logical greater, LG=1
SPM	Sprung Minus, N=1	JPM	Jump minus, N=1
SPMZ	Sprung Minus oder Null, N=1 oder Z=1	JPMZ	Jump minus or zero, N=1 or Z=1
SPN	Sprung nicht Null, Z=0	JPN	Jump not zero, Z=0
SPO	Sprung Overflow, O=1	JPO	Jump overflow, O=1
SPON	Sprung Overflow nicht, O=0	JPON	Jump overflow not, O=0
SPP	Sprung Plus, N=0	JPP	Jump plus, N=0
SPZ	Sprung Null, Z=1	JPZ	Jump zero, Z=1
SS	Starte Zeit als speichernde Einschaltverzögerung	SRE	Start time as raising delay extended
SUB	Subtraktion	SUB	Subtraction
SV	Starte Zeit als verlängerter Impuls	SPE	Start puls extended
SWAP	Vertausche Hi-/Lo-Byte im Register	SWAP	Interchange operand bytes
SYN	Synchronisationspunkt erreicht	SYN	Synchronisation point achieved
SZ	Setze Zähler	SC	Set counter
T	Transfer	T	Transfer
TABB	Transferiere in den Abbildbereich	TIMR	Transfer image range

Deutsch		Englisch	
TAH	Transfer absolut adressiert im Hintergrund	TAB	Transfer absolut range in background
TAS	Transfer absolut adressiert im Hintergrund mit Systeminterrupt	TAS	TAB with system interrupt
TAU	Transfer absolut adressiert unmittelbar	TAD	Transfer absolut range directly
TDEC	Zeit dekrementieren	TDEC	Time decrement
TFH	Transfer feldadressiert im Hintergrund	TFB	Transfer field in background
TFI	Transfer in FIFO-Speicher	TFI	Transfer FIFO
TFS	Transfer feldadressiert im Hintergrund mit Systeminterrupt	TFS	TFB with system interrupt
TFU	Transfer feldadressiert unmittelbar	TFD	Transfer field directly
TH	Zeit halt	TH	Timer halt
TM	Transfer der Interruptmaske	TIM	Transfer interrupt mask
TMB	Transfer in Memory-Bereich	TMB	Transfer memory band
TMBX	TMB im zweiten Segment	TMBX	TMB into second segment
TPB	Transfer Peripherie Bus	TPB	Transfer periphery bus
TSP	Transferier Stack Pointer	TSP	Transfer stack pointer
U	Und	A	And
UG	Ungleich	NEQ	Not equal
UN	Und nicht	AN	And not
VGL	Vergleichen logisch	CPL	Compare logical
VGLA	Vergleichen logisch und arithmetisch	CPLA	Compare logical and arithmetical
WE	Wecken	AB	Alarm bell request
WES	Wecken mit Systeminterrupt	ABS	AB with system interrupt
WEZ	Wecken zyklisch	ABC	Alarm bell request cyclical
WEZS	Wecken zyklisch mit Systeminterrupt	ABCS	ABC with system interrupt
XO	Exklusiv Oder	XO	Exclusive or
XON	Exklusiv Oder nicht	XON	Exclusive or not
ZR	Zähle rückwärts	CD	Count down
ZV	Zähle vorwärts	CU	Count up
=	Zuweisung	=	Equal-to sign
=OM	Zuweisung ohne Monitoranzeige	=WM	Equal without monitoring
*	Hilfsmarke setzen	*	Set help label
(Klammer auf	(Left bracket
)	Klammer zu)	Right bracket
)N	Klammer zu negiert)N	Right bracket with negation

**Bausteine/Moduls**

Deutsch		Englisch	
ASS	Assemblerbaustein	ASS	Assembler module
DB	Datenbaustein	DM	Data module
FB	Funktionsbaustein	FM	Function module
OB	Organisationsbaustein	OM	Organisation module
PB	Programmbaustein	PM	Program module
ZB	Zusatzbaustein	EM	Extended module

Sonstige Software-Begriffe/Other software notions

Deutsch		Englisch	
AWL	Anweisungsliste	IL	Instruction list
FUP	Funktionsplan	FUD	Function diagram
KPL	Kontaktplan	LD	Ladder diagram
OKN	Operandenkennzeichen	OID	Operand identifier
OPD	Operand	OPD	Operand
OPE	Operandenergänzung	OPA	Operand attribute
OPR	Operator	OPR	Operator
OPT	Operationsteil	OPP	Operation part
PA	Programmanweisung	PI	Program instruction
PAE	Parameterergänzung	PAA	Parameter attribute
PAR	Parameter	PAR	Parameter
PZ	Programmzweig	RG	Programm rung
Q	Quelloperand	SRC	Source operand
WSB	Weiterschaltbedingung		Step-on condition
Z	Zieloperand	DEST	Destination operand

A.3 BAPS-Terminologie

Standardfunktionen

Deutsch	Englisch
@	@
ABBRUCH	BREAK
ALLE	EVERY
ALLE_AUS	EVERY_END
ALTERNATIV	ALTERNATIVE
ANFANG	BEGIN
ANSONSTEN	DEFAULT
AUSGANG	OUTPUT
BAND	BELT
BEI	WHEN
BEI_AUS	WHEN_END
BELEGE	RESERVE
BEVOR	BEFORE
BINAER	BINARY
BIS	UNTIL
BISS_DASS	UNTIL
CIRCA	APPROX
DANN	THEN
DATEI	FILE
DEF	DEF
DEZ	REAL
EINGANG	INPUT
ENDE	END
EXAKT	EXACT
EXKLUSIV	EXCLUSIVE
EXKLUSIV_ENDE	EXCLUSIVE_END
EXTERN	EXTERNAL
FAHRE	MOVE
FALLS	CASE
FALLS_ENDE	CASE_END
FEHLER	ERROR
FELD	ARRAY
FREIGABE	RELEASE
FUER	FOR

Deutsch	Englisch
GANZ	INTEGER
GLEICH	EQUAL
GLOBAL	PUBLIC
GRENZE_AUS	LIMIT_OFF
HALT	HALT
INTERRUPT	INTERRUPT
JOG_AUS	JOG_OFF
JOG_EIN	JOG_ON
KIN_FORTSTAR	KIN_START
KIN_STOP	KIN_STOP
KONSTANTE	CONST
KREIS	CIRCULAR
LESE	READ
LESE_ANFANG	READ_BEGIN
LINEAR	LINEAR
MACHE	DO
MAL	TIMES
MAX_ZEIT	MAX_TIME
MIN_ZEIT	MIN_TIME
MIT	WITH
MK_PUNKT	JC_POINT
MOD	MOD
NACH	TO
NACHDEM	AFTER
NICHT	NOT
ODER	OR
PARALLEL	PARALLEL
PARALLEL_ENDE	PARALLEL_END
PAUSE	PAUSE
PERMANENT	PERMANENT
PRIO	PRIO
PROGRAMM	PROGRAM
PROGRAMM_ENDE	PROGRAM_END
PROGR_SLOPE	PROGR_SLOPE



Deutsch	Englisch
PTP	PTP
PUNKT	POINT
RAHMEN	FRAME
REF_PKT	REF_PNT
RELATIV	RELATIVE
RELATIV_ENDE	RELATIVE_END
RK_RAHMEN	WC_FRAME
RSPRUNG	RETURN
SATZ_SLOPE	BLOCK_SLOPE
SCHLIESSE	CLOSE
SCHREIBE	WRITE
SCHREIBE_ANF	WRITE_BEGIN
SCHREIBE_END	WRITE_END
SCHRITTWEITE	STEP
SEMAPHOR	SEMAPHORE
SONST	ELSE
SOWIE	ALSO
SPIEGLE	MIRROR
SPIEGLE_AUS	MIRROR_OFF
SPRUNG	JUMP
SPZ_FKT	SPC_FCT
START	START
STOP	STOP
SYNC	SYNC
SYNCHRON	SYNCHRON
SYNCHRON_END	SYNCHRON_END
TEMP	TEMP
TEXT	TEXT
TYP	TYPE
UEBER	VIA
UND	AND
UP	SUBROUTINE
UP_ENDE	SUB_END
VERBUND	RECORD
VERBUND_ENDE	RECORD_END
VERSCHIEBE	MOVE_REL

Deutsch	Englisch
VORSCHUB_HAL	FEED_HOLD
VORSCHUB_STA	FEED_START
WAEHREND	WHILE
WARTE	WAIT
WDH	REPEAT
WDH_ENDE	REPEAT_END
WENN	IF
WERKZEUG	TOOL
WERT	VALUE
ZEICHEN	CHAR
ZUORDNE	ASSIGN

Übersetzeranweisungen

Deutsch	Englisch
ACHSNAMEN	JC_NAMES
ANTRIEBSART	DRIVE_TYPE
DATEI_FEHLER	FILE_ERROR
EINFUEGE	INCLUDE
INT	INT
KINEMATIK	KINEMATICS
KOORDINATEN	WC_NAMES
POSE	POSE
PROZESS_ART	PROCESS_KIND
SER_EA_STOP	SER_IO_STOP
STEUERUNG	CONTROL
TESTINFO	DEBUGINFO
WARNUNG	WARNING

Standardvariable

Deutsch	Englisch
@IPOS	@POS
@MPOS	@MPOS
A	A
A_PTP	A_PTP
AFAKTOR	AFACTOR
AFEST	AFIX
AFEST_PTP	AFIX_PTP

Deutsch	Englisch
DFAKTOR	DFACTOR
G_VFAKTOR	G_VFAKTOR
GRENZE_MAX	LIMIT_MAX
GRENZE_MIN	LIMIT_MIN
HBG	MCP
INPOS	INPOS
IPOS	POS
MP	MP
ORIMATRIX	ORIMATRIX
OVERRIDE	OVERRIDE
PHG	PHG
R	R
R_PTP	R_PTP
RADIUS	RADIUS
RK_SYSTEM	WC_SYSTEM
SKALA	SCALE
STATUS	STATUS
T	T
TFEST	TFIX
TTY	TTY
V	V
V24_1	V24_1
V24_2	V24_2
V24_3	V24_3
V24_4	V24_4
V_JOG	V_JOG
V_ORI	V_ORI
V_PTYP	V_PTP
VFAKTOR	VFACTOR
VFEST	VFIX
VFEST_ORI	VFIX_ORI
VFEST_PTP	VFIX_PTP
WZ_SYS	TOOL_SYS

Standardfunktionen

Deutsch	Englisch
ABS	ABS
ATAN	ATAN
BNR_DATEI	BNR_FILE
CHR	CHR
COS	COS
DATEI_ENDE	END_OF_FILE
DATUM	DATE
GANZ_ZFELD	INT_ASC
GANZTEIL	TRUNC
INDEX_MAX	INDEX_MAX
KOMMANDO	COMMAND
MK	JC
ORD	ORD
ORI	ORI
PROZ_PRIO	PROC_PRIO
PROZ_ZUSTAND	PROC_STATUS
RK	WC
RK_RECHNUNG	WC_COMPUTATION
RUNDUNG	ROUND
SIN	SIN
SYS_ZEIT	SYS_TIME
UNTERBRECHE	BREAK
WURZEL	SQRT
ZEILEN_ENDE	LINE_END
ZFELD_GANZ	ASC_INT
ZUSTAND	CONDITION

Standardkonstanten

Deutsch	Englisch
RK_UR	WC_UR



A.4 Safety instructions

A.4.1 Dansk

Sikkerhedshenvisningerne i denne brugsanvisning



Disse symboler anvendes i den foreliggende brugsanvisning i følgende tilfælde:



FORSIGTIG

Dette symbol benyttes, hvis der skal advares mod **farlig elektrisk spænding**. Hvis advarslen ikke følges nøjagtigt eller ignoreres kan det medføre **personskader**.



FORSIGTIG

Dette symbol benyttes, hvis en unøjagtig eller manglende overholdelse af anvisningerne kan medføre beskadigelser af **personer**.



VIGTIGT

Dette symbol benyttes, hvis en unøjagtig eller manglende overholdelse af anvisningerne kan medføre beskadigelser af **apparater eller filer**.



Dette symbol benyttes for at gøre Dem opmærksom på noget særligt.



FORSIGTIG

0.1
Risiko for personer og ting!
Prøv hvert nyt program, inden De tager et anlæg i drift!



VIGTIGT

0.2
Risiko for modulet!
Modulet må ikke sættes i eller trækkes ud af stikket, når der er tændt for styringen! Modulet kan blive ødelagt. Der skal først slukkes for styringens netdelmodul, den eksterne spændingsforsyning og signalspændingen eller disse skal trækkes ud af stikket, inden modulet må sættes i eller trækkes ud af stikket!



VIGTIGT

0.3
Risiko for modulet!
Ved omgang med modulet skal alle forholdsregler til ESD-beskyttelse iagttages!
Undgå elektrostatisk udladninger!

Sikkerhedshenvisninger på styrekomponenterne

På styrekomponenterne selv kan der være anbragt følgende advarsler og henvisninger, som skal gøre Dem opmærksom på bestemte ting:



Advarsel mod farlig elektrisk spænding!



Advarsel mod farer fra batterier!



Elektrostatisk udsatte komponenter!



Træk netstikket ud, inden De åbner!



Bolt kun til tilslutning af jordledningen PE!



Tilslutning kun for funktionsjording, fremmedspændingsfattig jord!



Kun til tilslutning af en afskærmningsledning!

A.4.2 Deutsch

Sicherheitshinweise in dieser Gebrauchsanweisung



Diese Symbole werden in dieser Gebrauchsanweisung unter den folgenden Bedingungen verwendet.



VORSICHT

Dieses Symbol wird benutzt, wenn vor einer **gefährlichen elektrischen Spannung** gewarnt werden soll. Durch ungenaues Befolgen oder Nichtbefolgen dieser Anweisung kann es zu **Personenschäden** kommen.



VORSICHT

Dieses Symbol wird benutzt, wenn es durch ungenaues Befolgen oder Nichtbefolgen von Anweisungen zu **Personenschäden** kommen kann.



ACHTUNG

Dieses Symbol wird benutzt, wenn es durch ungenaues Befolgen oder Nichtbefolgen von Anweisungen zu **Beschädigungen von Geräten oder Dateien** kommen kann.



Dieses Symbol wird benutzt, wenn Sie auf etwas Besonderes aufmerksam gemacht werden sollen.



VORSICHT

0.1

Gefahr für Personen und Sachen!

Testen Sie jedes neue Programm bevor Sie eine Anlage in Betrieb nehmen!



ACHTUNG

0.2

Gefahr für die Baugruppe!

Baugruppe nicht bei eingeschalteter Steuerung stecken oder ziehen! Baugruppe kann zerstört werden. Zuerst Netzteilbaugruppe der Steuerung, externe Spannungsversorgung und Signalspannung ausschalten oder abziehen und erst dann Baugruppe stecken oder ziehen!



ACHTUNG

0.3

Gefahr für die Baugruppe!

Beim Umgang mit der Baugruppe müssen alle Vorkehrungen zum ESD-Schutz eingehalten werden! Elektrostatische Entladungen vermeiden!

Sicherheitshinweise an den Steuerungskomponenten

An den Steuerungskomponenten selbst können folgende Warnungen und Hinweise angebracht sein, die Sie auf bestimmte Dinge aufmerksam machen sollen:



Warnung vor gefährlicher elektrischer Spannung!



Warnung vor Gefahren durch Batterien!



Elektrostatisch gefährdete Bauelemente!



Vor dem Öffnen Netzstecker ziehen!



Bolzen nur für Anschluß des Schutzleiters PE!



Anschluß nur für Funktionserde, fremdspannungsarme Erde!



Nur für Anschluß eines Schirmleiters!

**A.4.3** Ελληνικά

Υποδείξεις ασφαλείας στις παρούσες οδηγίες χρήσεως



Τα σύμβολα αυτά στις παρούσες οδηγίες χρήσεως χρησιμοποιούνται υπό τους ακόλουθους όρους:

**ΚΙΝΔΥΝΟΣ**

Αυτό το σύμβολο χρησιμοποιείται για να σας προειδοποιήσει από επικίνδυνη ηλεκτρική τάση. Αν δεν τηρούνται με ακρίβεια ή δεν τηρούνται καθόλου οι οδηγίες μπορεί να προκληθούν σωματικές βλάβες.

**ΚΙΝΔΥΝΟΣ**

Το σύμβολο αυτό χρησιμοποιείται, όταν μπορεί να προκληθούν σωματικές βλάβες, αν δεν τηρούνται με ακρίβεια ή δεν τηρούνται καθόλου οδηγίες.

**ΠΡΟΣΟΧΗ**

Το σύμβολο αυτό χρησιμοποιείται, όταν μπορεί να προκληθούν ζημιές σε συσκευές ή σε αρχεία, αν δεν τηρούνται με ακρίβεια ή δεν τηρούνται καθόλου οδηγίες.



Το σύμβολο αυτό χρησιμοποιείται, όταν θα πρέπει να επιστηθεί η προσοχή σας σε κάτι το σημαντικό.

**ΚΙΝΔΥΝΟΣ**

0.1

Κίνδυνος για πρόσωπα και αντικείμενα!

Δοκιμάστε κάθε καινούριο πρόγραμμα πριν θέσετε μια εγκατάσταση σε λειτουργία!

**ΠΡΟΣΟΧΗ**

0.2

Κίνδυνος για το στοιχείο κατασκευής!

Μην αφαιρείτε ή τοποθετείτε το στοιχείο κατασκευής σε κύκλωμα που είναι σε λειτουργία! Το στοιχείο κατασκευής μπορεί να καταστραφεί. Πρώτα αφαιρείτε ή αποσυνδέετε το στοιχείο κατασκευής της ρύθμισης του ηλεκτρικού κυκλώματος, κατόπιν την παροχή τάσης και την τάση σήματος και μετά τοποθετείτε ή αφαιρείτε το στοιχείο κατασκευής.

**ΠΡΟΣΟΧΗ**

0.3

Κίνδυνος για το στοιχείο κατασκευής!

Όταν έχετε στα χέρια σας το στοιχείο κατασκευής πρέπει να τηρείτε όλα τα μέτρα για την ηλεκτροστατική προστασία! Αποφεύγετε ηλεκτροστατικές εκφορτίσεις!

Υποδείξεις ασφαλείας σε εξαρτήματα ρύθμισης και ελέγχου

Τα εξαρτήματα ρύθμισης και ελέγχου μπορεί να φέρουν τις ακόλουθες προειδοποιήσεις και υποδείξεις, που επιστούν την προσοχή σας σε ορισμένα πράγματα:



Προειδοποίηση σχετικά με επικίνδυνη τάση ηλεκτρικού ρεύματος!



Προειδοποίηση σχετικά με κινδύνους, που προέρχονται από μπαταρίες!



Στοιχεία κατασκευής, για τα οποία υπάρχει ηλεκτροστατικός κίνδυνος!



Πριν από το άνοιγμα βγάλτε το φως από την πρίζα!



Πείροι μόνο για σύνδεση προστατευτικού αγωγού (γείωσης) PE!



Σύνδεση για γείωση λειτουργίας, γείωση για άσχετο ασθενές ρεύμα!



Μόνο για σύνδεση θωρακισμένου αγωγού!

A.4.4 English

Safety instructions in this manual



These symbols are used throughout this manual subject to the following conditions.



DANGER

This symbol is used to warn of the presence of **dangerous electrical current**. Insufficient or lacking compliance with these instructions can result in **personal injury**.



DANGER

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



CAUTION

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



This symbol is used to inform the user of special features.



DANGER

0.1
 Danger to persons and equipment!
 New programs must be tested before a system is put into operation!



CAUTION

0.2
 Danger to the module!
 Do not insert or remove module when the control is switched on! This can destroy the module. Switch off or remove control power supply module, external power supply and signal voltage before inserting or removing the module!



CAUTION

0.3
 Danger for the module!
 When handling the module, follow all precautions for e.s.d. protection! Avoid electrostatic discharges!

Safety instructions on the control components

The following warnings and notices may be indicated on the control components themselves and have the following meaning:



Danger: High voltage!



Danger: Battery acid!



Electrostatically-sensitive components!



Disconnect at mains before opening!



Pin for connecting PE conductor only!



This connection for functional earthing or low-noise earth only!



For screened conductor only!

A.4.5 Español

Indicaciones de seguridad en estas instrucciones de empleo



Estos símbolos se utilizan en estas instrucciones de empleo bajo las siguientes condiciones.



PRECAUCION

Este símbolo se utiliza para advertir de una **tensión eléctrica peligrosa**. La ejecución inexacta o la no ejecución de esta instrucción puede provocar **daños a personas**.



PRECAUCION

Este símbolo se utiliza cuando por una ejecución inexacta o la no ejecución de instrucciones pueden llegar a producirse **daños a personas**.



ATENCION

Este símbolo se utiliza cuando por la ejecución inexacta o la no ejecución de instrucciones pueden llegar a producirse **daños en equipos o ficheros**.



Este símbolo se utiliza cuando se le debe llamar al usuario la atención respecto a algo especial.



PRECAUCION

0.1

¡Peligro para personas y bienes materiales!
¡Compruebe cada nuevo programa antes de poner en funcionamiento una instalación!



ATENCION

0.2

¡Peligro para el módulo!
¡No enchufe ni extraiga el módulo cuando el control está conectado! Puede destruirse el módulo. ¡Desconecte o desenchufe primero el módulo de fuente de alimentación del control, la alimentación de tensión externa y la tensión de señalización y sólo después enchufe o extraiga el módulo!



ATENCION

0.3

¡Peligro para el módulo!
¡Observe en la manipulación del módulo todas las precauciones en cuanto a la protección ESD! ¡Evite descargas estáticas!

Indicaciones de seguridad en los componentes de control

Los componentes de control mismos pueden estar marcados por las siguientes advertencias e indicaciones que le deben llamar la atención al usuario:



¡Advertencia ante tensión eléctrica peligrosa!



¡Advertencia ante riesgos por baterías!



¡Elementos constructivos con riesgos de descargas electrostáticas!



¡Antes de abrir, desenchufar el conector de la red!



¡Perno sólo para la conexión del conductor protector PE!



¡Conexión sólo para toma de tierra de funcionamiento, tierra de poca tensión externa!



¡Sólo para la conexión de un conector blindado!

A.4.6 Français

Directives de sécurité relatives au présent mode d'emploi



Ces symboles sont utilisés dans les conditions suivantes:



Ce symbole est utilisé lorsque l'on veut mettre en garde contre une **tension électrique dangereuse**. Risque de **dommage corporel** si les consignes données ne sont pas respectées ou lorsqu'elles sont mal respectées.



Ce symbole est utilisé s'il y a un risque de **dommage corporel** si les consignes données ne sont pas respectées ou lorsqu'elles sont mal respectées.



Ce symbole est utilisé s'il y a un risque de dommage matériel ou risque de destruction de fichier si les consignes données ne sont pas respectées ou lorsqu'elles sont mal respectées.



Ce symbole est utilisé lorsqu'il s'agit d'attirer votre attention sur un point particulier.



0.1
Risque pour les personnes et le matériel !
Testez chaque nouveau programme avant de mettre une installation en service!



0.2

Risque pour l'unité !

Ne branchez ou ne débranchez pas l'unité lorsque la commande est activée ! Risque de destruction de l'unité. Avant de brancher ou de débrancher l'unité, coupez ou déconnectez d'abord le bloc d'alimentation de la commande, l'alimentation en courant électrique externe et la tension de signal !



0.3

Risque pour l'unité !

Respectez toutes les mesures de protection ESD lors du maniement de l'unité ! Evitez les décharges électrostatiques !

Mesures de sécurité relatives aux dispositifs de commande

Les pictogrammes et messages d'avertissement suivants peuvent se trouver sur les éléments de commande afin d'attirer votre attention sur certains points:



Présence de tension électrique dangereuse



Danger lié à la présence de batteries



Modules sensibles à l'électricité statique



Enlever la fiche secteur avant l'ouverture



Uniquement pour le raccordement de la terre PE !



Uniquement pour le raccordement à la terre, terre sans bruit !



Uniquement pour le raccordement d'un câble blindé

A.4.7 Italiano

Avvertenze per la sicurezza in queste istruzioni per l'uso



Questi simboli vengono impiegati in queste istruzioni per l'uso nelle seguenti condizioni.



PERICOLO

Questo simbolo viene impiegato per segnalare la presenza di **tensioni elettriche pericolose**. La mancata osservanza, anche parziale, di queste istruzioni può provocare danni alle **persone**.



PERICOLO

Questo simbolo viene impiegato qualora l'osservanza imprecisa o la mancata osservanza delle istruzioni possono provocare danni alle **persone**.



ATTENZIONE

Questo simbolo viene impiegato qualora l'osservanza imprecisa o la mancata osservanza delle istruzioni può provocare danni agli **apparecchi o ai file**.



Questo simbolo viene impiegato quando si voglia richiamare l'attenzione su qualcosa di particolare.



PERICOLO

0.1

Pericolo per persone ed oggetti!

Provare ogni nuovo programma prima di mettere in funzione l'impianto!



ATTENZIONE

0.2

Pericolo per il modulo!

Non innestare o rimuovere il modulo quando il comando è acceso! Il modulo potrebbe venire distrutto. Spegnere prima il modulo d'alimentazione del comando, l'alimentazione esterna di tensione e la tensione del segnale e solo successivamente innestare o rimuovere il modulo!



ATTENZIONE

0.3

Pericolo per i moduli!

Durante operazioni con i moduli rispettare tutte le misure di protezione ESD! Evitare scariche elettrostatiche!

Avvertenze per la sicurezza sui componenti di comando

Sui componenti di comando stessi possono essere applicate le seguenti targhette di avvertimento e di avvertenza, che richiamano l'attenzione su particolari pericoli:



Avvertimento per tensione elettrica pericolosa!



Avvertimento per pericoli dovuti alle batterie!



Elementi costruttivi danneggiabili da cariche elettrostatiche!



Sfilare la spina dalla rete prima di aprire!



Perno solo per il collegamento del conduttore di protezione PE!



Collegamento per messa a terra funzionale,
terra senza rumore!



Solo per il collegamento di un conduttore
schermato!



A.4.8 Nederlands

Veiligheidsrichtlijnen in deze gebruiksaanwijzing



Deze symbolen worden in deze gebruiksaanwijzing onder de volgende voorwaarden gebruikt.



ATTENTIE

Dit symbool wordt gebruikt, als de aandacht op een **gevaarlijke elektrische spanning** gevestigd moet worden. Wordt deze aanwijzing niet precies gevolgd of zelfs genegeerd, dan is **lichamelijk letsel** niet uitgesloten.



ATTENTIE

Dit symbool wordt gebruikt wanneer door onnauwkeurige of niet-naleving van aanwijzingen **schade aan personen** kan worden berokkend.



LET OP

Dit symbool wordt gebruikt wanneer door onnauwkeurige of niet-naleving van aanwijzingen **schade aan toestellen of bestanden** kan worden berokkend.



Dit symbool wordt gebruikt wanneer wij u op iets bijzonders willen attent maken.



ATTENTIE

0.1
Gevaar voor lichamelijk letsel en materiële schade!
Test elk nieuw programma voor u een installatie opstart!



LET OP

0.2
Gevaar voor de module!
Als de besturing ingeschakeld is, de module niet inste-
ken of uittrekken! De module kan hierdoor kapot gaan.
De module van het netdeel van de besturing, de ex-
terne spanningstoevoer en de signaalspanning uit-
schakelen of aftrekken en pas dan de module inste-
ken of uittrekken.



LET OP

0.3
Gevaar voor de module!
In de omgang met de module alle voorschriften m.b.t.
de ESD-beveiliging in acht nemen! Elektrostatische
ontladingen vermijden!

Veiligheidsaanwijzingen bij de besturingscomponenten

Aan de besturingscomponenten zelf kunnen de vol-
gende waarschuwingen en richtlijnen aangebracht
zijn. Zij zijn bedoeld om u op bepaalde zaken te atten-
deren:



Waarschuwing voor gevaarlijke elektrische spanning.



Waarschuwing voor gevaar veroorzaakt door akku's.



Elektrostatisch gevoelige componenten.



Trek de stekker uit alvorens te openen.



Bouten alleen voor aansluiting van de veiligheidsaarding PE.



Aansluiting uitsluitend voor functionele, spanningsarme aarde!



Alleen voor aansluiting van een afgeschermde kabel.



A.4.9 Português

Instruções de segurança contidas nas presentes instruções de serviço



Estes símbolos são utilizados nas presentes instruções de serviço nos seguintes casos:



CUIDADO

Este símbolo é utilizado para indicar uma **tensão eléctrica perigosa**. Em caso de não observância ou observância incorrecta desta instrução, existe **perigo de ferimento de pessoas**.



CUIDADO

Este símbolo é utilizado quando existe o **perigo de ferimento de pessoas** por observância incorrecta ou não observância das instruções.



ATENÇÃO

Este símbolo é utilizado quando existe o perigo de danificação de aparelhos ou ficheiros por observância incorrecta ou não observância das instruções.



Este símbolo é utilizado para chamar a atenção para algo de especial.



CUIDADO

0.1

Perigos de ferimentos de pessoas e de danos materiais!

Antes de colocar uma instalação em funcionamento há que experimentar sempre qualquer programa novo!



ATENÇÃO

0.2

Perigo para o módulo!

Não retire ou introduza o módulo quando o comando estiver ligado! O módulo poderá ser danificado. Primeiro desligue ou retire o módulo de alimentação do comando, o cabo alimentador da rede e a tensão de sinal, e em seguida, poderá introduzir ou retirar o módulo!



ATENÇÃO

0.3

Perigo para o módulo!

Na utilização do módulo, respeitar todas as prescrições para a protecção do ESD! Evitar descargas electrostáticas!

Instruções de segurança nos componentes de comando

Nos próprios componentes de comando podem estar afixados os avisos ou as instruções seguidamente descritos para chamar à atenção para determinados pontos.



Aviso referente a uma tensão eléctrica perigosa!



Aviso referente a perigos relacionados com baterias!



Módulos em perigo electrostático!



Antes de abrir tirar o cabo alimentador da rede!



Borne apenas para ligação do condutor de protecção à massa PE!



Ligação apenas para ligação à terra funcional, terra com baixa tensão externa!



Só para ligação de um condutor blindado!



A.4.10 Suomi

Tämän käyttöohjeen turvallisuusohjeet



Näitä symboleja käytetään tässä käyttöohjeessa seuraavasti.



VAROITUS

Tätä symbolia käytetään, kun varoitetaan **vaarallisesta sähköjännitteestä**. Seurauksena voi olla **henkilövahinko**, jos ohjetta ei seurata tai sitä ei seurata tarkkaan.



VAROITUS

Tätä symbolia käytetään, jos ohjeiden noudattamatta jättäminen voi johtaa **henkilövahinkoihin**.



HUOMIO

Tätä symbolia käytetään, jos ohjeiden noudattamatta jättäminen tai niiden epätarkka seuraaminen voi johtaa **laitteiden tai tiedostojen vahingoittumiseen**.



Tätä symbolia käytetään, kun halutaan kiinnittää lukijan huomio johonkin erikoisseikkaan.



VAROITUS

0.1

Henkilö- ja tavaravahinkovaara!

Testaa jokainen uusi ohjelma, ennen laitteiston käyttöönottoa!



HUOMIO

0.2

Rakennesaryhmä voi vioittua!

Älä liitä tai irrota rakennesaryhmää ohjauksen ollessa päällekytkettynä! Rakennesaryhmä voi tuhoutua. Kytke ensin ohjauksen verkko-osarakenneryhmä, ulkoinen jännitteentulo ja signaalijännite pois päältä tai irrota ne ja liitä tai irrota rakennesaryhmä vasta sitten!



HUOMIO

0.3

Rakennesaryhmä voi vioittua!

Rakennesaryhmän kanssa toimittaessa on kaikkia ESD-suojaan liittyviä toimenpiteitä noudatettava! Elektrostaattista latausta on vältettävä!

Ohjauskomponenttien turvallisuusohjeet

Ohjauskomponentteihin voi olla merkittynä seuraavat varoitukset ja ohjeet, joiden tarkoitus on kiinnittää käyttäjän huomio tiettyihin seikkoihin:



Varoitus, vaarallinen sähköjännite!



Varoitus, akkujen aiheuttamat vaarat!



Sähköstaattisesti vaarannetut rakennesosat!



Vedä verkkopistoke irti pistorasiasta ennen avaamista!



Pultti vain suojajohtimen PE liitännälle!



Liitäntä häiriösuojattuun erilliseen suoja- maadoituspisteeseen!



Vain suojajohtimen liitäntää varten!

A.4.11 Svenska

Säkerhetsanvisningar i denna driftsinstruktion



Dessa symboler används i denna driftsinstruktion för följande förutsättningar.



VARNING

Denna symbol används, vid varning för **farlig elektrisk spänning**. Om denna anvisning inte exakt följs eller inte följs alls kan det medföra **personskador**.



VARNING

Denna symbol används, när **personer kan skadas** om anvisningar inte exakt följs eller inte följs alls.



OBS

Denna symbol används, när **apparater eller filer kan skadas** om anvisningar inte exakt följs eller inte följs alls.



Denna symbol används, när Ni skall göras uppmärksam på något särskilt.



VARNING

0.1

Fara för person- och sakskador!

Prova varje nytt program innan Ni tar en anläggning i drift!



OBS

0.2

Fara för en komponentgrupp!

Stick inte in och drag inte heller ur en komponentgrupp när styrningen är tillkopplad! Komponentgruppen kan förstöras. Frånkoppla eller drag först ur styrningens nätdelskomponentgrupp, extern spänningsförsörjning och signalspänningen och stick in eller drag först därefter ut komponentgruppen!



OBS

0.3

Fara för en komponentgrupp!

Vid arbeten med komponentgruppen skall alla åtgärder för ESD-skydd innehållas! Statiska urladdningar skall undvikas!

Säkerhetsanvisningar på styrningskomponenterna

På styrningskomponenterna kan följande varningar och anvisningar vara placerade, som vill göra Er uppmärksam på vissa saker:



Varning för farlig elektrisk spänning!



Varning för faror genom batterier!



Komponenter som kan skadas av elektrostatisk urladdning!



Drag ur kontakten innan öppning!



Bultar endast för anslutning av skyddsledaren PE!



Anslutning endast för funktionsjordning,
jordning med låg interferens!



Endast för anslutning av en avskärningsle-
dare!

Notes:

Bosch-Automationstechnik

Robert Bosch GmbH
Geschäftsbereich
Automationstechnik
Industriehydraulik
Postfach 30 02 40
D-70442 Stuttgart
Telefax (07 11) 8 11-18 57

Robert Bosch GmbH
Geschäftsbereich
Automationstechnik
Fahrzeughydraulik
Postfach 30 02 40
D-70442 Stuttgart
Telefax (07 11) 8 11-17 98

Robert Bosch GmbH
Geschäftsbereich
Automationstechnik
Pneumatik
Postfach 30 02 40
D-70442 Stuttgart
Telefax (07 11) 8 11-89 17

Robert Bosch GmbH
Geschäftsbereich
Automationstechnik
Montagetchnik
Postfach 30 02 07
D-70442 Stuttgart
Telefax (07 11) 8 11-77 12

Robert Bosch GmbH
Geschäftsbereich
Automationstechnik
Antriebs- und Steuerungstechnik
Postfach 11 62
D-64701 Erbach
Telefax (0 60 62) 78-4 28

Robert Bosch GmbH
Geschäftsbereich
Automationstechnik
Schraub- und Einpreßsysteme
Postfach 11 61
D-71534 Murrhardt
Telefax (0 71 92) 22-1 81

Robert Bosch GmbH
Geschäftsbereich
Automationstechnik
Entgrattechnik
Postfach 30 02 07
D-70442 Stuttgart
Telefax (07 11) 8 11-34 75

Technische Änderungen vorbehalten

Ihr Ansprechpartner

BOSCH



Robert Bosch GmbH
Geschäftsbereich
Automationstechnik
Antriebs- und Steuerungstechnik
Postfach 11 62
D-64701 Erbach
Telefax (0 60 62) 78-4 28