ABB Procontic b
Programmable Control System

Programming

System-Specific programming units
07 PG 30
7.1

Multi-System programming units
07 PH 32
7.2

Multi-System programming software
907 PC 32
7.4
Regulations

Regulations concerning the setting up of installations

Apart from the basic "Regulations for the setting up of power units" VDE 0100 and for "The rating of creepage paths and air gaps" VDE 0110 the regulations "The equipment of power units with electrical components" VDE 0160 in connection with VDE 0660, part 500, have to be taken into due consideration. Further attention has to be paid to VDE 0113 in case of the control of working and processing machines. If operating elements are to be arranged near shock-hazzard parts with protection against electrical shock, VDE 0108, part 100, is relevant.

The user has to ensure that the units as well as the associated components have to be installed according to these regulations. Respectfully valid safety regulations, e.g. regulation for the prevention of accidents and the law concerning technical working material, are valid for machines and units connected as well.

ABB Procontic units have been built according to VDE regulation 0160. The protection against direct touching as demanded by chapter 5.5.1 of this VDE regulation has to be satisfied by the user, e.g. at installing of switch cabinet.

ABB Procontic units have been designed for operation according to insulation class A of VDE 0110. If considerable pollution is expected during operations, the units have to be installed in housings of the respective kind of protection.

* VDE stands for "Association of German Electrical Engineers".

Note: Please observe the national regulations for the installation of electrical equipments, which are valid in your country.

ABB Schalt- und Steuerungstechnik GmbH
System-Specific Programming Units

ABB Schalt- und Steuerungstechnik GmbH
Table of Contents

1  Programmer 07 PG 30  1-1
1.1 Outline  1-2
1.2 Extent of Supply  1-2
1.3 Technical Data  1-3
1.4 General Description of the Device  1-3
1.5 Operating Elements and Device Connections  1-3
1.6 The Buffered Program Storage  1-4

2  Possibilities of Use and Combination  2-1
2.1 07 PG 30 as an Independent Programmer  2-1
2.2 07 PG 30 with Printer  2-1
2.3 07 PG 30 with SIGMA*-tronic p  2-2
2.4 07 PG 30 with PROCONTIC b  2-2
2.5 07 PG 30 as a Test and Starting Unit  2-3
2.6 07 PG 30 as an EPROM Programmer  2-3
2.7 Serial interface of 07 TZ82 R201  2-3

3  Survey: Functions and Keys  3-1
3.1 Edit Instructions  3-1
3.2 $-Functions  3-1
3.3 Special Instructions, Function Keys  3-1
3.4 Instructions to Start a Program  3-1
3.5 Special Keys for Program Input  3-1

4  Switching on the Device and Program Input  4-1

5  Program and Operation Indications  5-1

6  Editing Functions  6-1
POINTER  6-1
INSERT  6-1
OVERWRITE  6-2
FIND  6-2
CHANGE  6-2
KILL  6-2
LIST  6-3

7  Program Start  7-1
7.1 Changing over between SIGMA*-tronic p and PROCONTIC b  7-1
7.2 Special Functions  7-1
TEST: Status Indication  7-1
RUN: Program Run with Manual Start  7-2
AUTORUN: Automatic Start with Voltage Connection  7-2
HALT: Program Stop  7-2
Changing over between RAM and EPROM Operations  7-2

8  $-Functions  8-1
$AUS: Switching off System Interface  8-1
$BAUD: Indicate Baud Rate Set  8-1
$B300: Transfer Speed  8-1
$B1200 (Baud Rate) for Printer or $B32400 for $TTY-Mode to Be Set (300 - 2400 Baud)  8-1

$CLEAR: Clear 07 PG 30 RAM  8-1
$FREE: Indicate Vacant Storage Location  8-1
$LIST: Print-out Overall Program  8-1
$PARITY: Search for Parity Errors  8-1
$PROM: Handle EPROM on a 07 PG 30 Socket  8-1
$PRxxxx: Set EPROM Type  8-2
$PR80  8-2
$PR82  8-2
$PR84R1  8-2
$PR84R2  8-2
$PRT16  8-2
$PRT32  8-2
$PRT32A  8-2
$PRT64  8-2
$PRT128  8-2
$PRT: Indicate EPROM Type Set  8-2
$RAMTEST: 07 PG 30 RAM Blank State Test  8-2
$REMOTE: Start Unit - Operation with Connected Programming and Test System (Personal Computer)  8-2
$SPS: Read out EPROM 07 ZE 82  8-3
$SPSV: Comparison of Programs 07 PG 30-RAM and 07 ZE 82-EPROM  8-3
$TTY: Terminal Operations  8-3
$PE: Search for End of Program (IPE)  8-3
$HEX: Hexadecimal Code  8-3
$OPTION: Functions "Further Loadings"  8-3
$DEUTSCH: Output of System Messages in German  8-4
$ENGLISH: Output of System Messages in English  8-4
$FRANCAIS: Output of System Messages in French  8-4
$MONITOR: Service Functions  8-4
$MONITOR  8-4
$XTTY  8-4
$XTTOFF  8-4

7.1 Extent of Language  9-1
Connection of the 07 PG 30 with PROCONTIC b  10-1

8.1 Accessories  11-1
07 SK 30 R2 Cable to Connect PROCONTIC b (07 ZE 82)  11-1
07 SK 31 Cable to Connect SIGMA*-tronic p  11-2
07 SK 32 Cable to Connect a Personal Computer  11-2
07 SK 33 Cable to Connect a Printer  11-2
07 SK 34 R2 Supply Cable  11-2

programmer 07 PG 30 / state 2.90  1
The programmer 07 PG 30 is a compact and sturdy built programming, test and EPROM programmer in a handy pocket size. It was designed as a programming and test device for the compact BBC minicontrol SIGMA-tronic p and the modular stored-program control PROCONTIC b (bit processing with 07 ZE 82) and serves for program input and changes as well as the testing of the program.

The serial interface installed allows the connection with a printer and thus offers the possibility to document and store programs.

Instead of a printer it is possible to connect the BBC programming and testing systems 907 PC 30 and 907 PC 31 operating on the basis of IBM-compatible personal computers. The device 07 PG 30 replaces the test and programming unit 07 IE 82 and serves as an EPROM programmer.

Of course, the device 07 PG 30 can also be used as an independent programming and EPROM loading device (stand-alone).

The program input is done by means of the "instruction list AWL". The input syntax and the other operating functions having proved reliable in the BBC programming systems 07 TD 12, 907 PC 30, and 907 PC 31, everybody knowing them can work with them on the device 07 PG 30. Someone not having this knowledge can familiarize himself with it quickly due to the simplicity of the language and the operating of the device.

The program entered or read in is stored in a battery-buffered RAM being able to replace the program storage of the connected control during testing operations.

Being used as an EPROM programmer EPROMs can be read, loaded, compared, checked and duplicated. The internal RAM-storage then serves as a temporary storage.

The programmer 07 PG 30 is supplied either by the power supply unit with plug available as an accessory or by the 24 V supply of the connected control.

The possibilities of the 07 PG 30 are listed in the following table:

**Editing functions**
- program instructions:
  - indicate
  - input
  - overwrite
  - insert
  - address
  - search
  - exchange

- program instructions, program records, complete programs:
  - erase
  - print out.

**Test and EPROM loading functions**
- complete programs:
  - test (status indication)
  - take over from the programming systems 907 PC 30, 907 PC 31 or 07 TD 12
  - take over from the control
  - store temporarily
  - load RAM into EPROM
  - load EPROM into RAM
  - compare the contents EPROM/RAM

- further functions:
  - test and starting unit
  - RAM blank state test
  - EPROM blank state test
  - parity test
  - TTY function.
1.2 Delivery Unit

The delivery unit 07 PG 30 consists of the following parts:

programmer and testing device 07 PG 30
EPROM adapter
24-pole supply cable 07 SK 34

1.3 Technical Data

Programmer 07 PG 30:

| design:     | hand-hold / table-mounted device in pocket size, plastic casing with cover to protect the keyboard |
| measurements: | 145x132x36 mm |
| power supply: | 24 V-400 mA |
| admissible voltage difference: | +16.8 V-...+30 V |
| RAM buffering: | 4x1.5 V dry cell battery, size AA MIGNON |
| indication: | LCD, 16 characters, one line |
| character representation: | 5x7 point |
| program storage: | 4K-words, 16 bit each |
| keyboard: | ASCII keyboard with function keys, electronically unblocked |
| interfaces: | serial interface RS232C system interface 300 - 2400 Baud |
| transfer speed: | VDE 0160 |
| standard: | IP 20 according to DIN 40050, section 3 |
| kind of protection: | 0 ... +45°C |
| operating temperature: | -20 ... +70°C |
| storage temperature: | humidity class F according to DIN 40040 |
| humidity: | 0.7 kg |
| weight: | GJV3072102R1 |
| order number: |

Power Supply 07 NG 30:

| design:     | plug power pack |
| supply connection: | 220V +10%/-5%, 50 Hz |
| output voltage: | +18 V- +/-5%, 400 mA |
| type of construction: | according to VDE 0551 |
| protection class: | II according to VDE 0106 section 1 |
| weight: | 0.4 kg |
| order number: | GJV3072203R1 |

Accessories:

| 07 SK 30 cable for connection with PROCONTIC b (07 ZE 82) | GJR5241710R1 |
| 07 SK 31 cable for connection with SIGMA-tronic p | GJR5241720R1 |
| 07 SK 32 cable for connection with personal computer | GJR5241730R1 |
| 07 SK 33 cable for connection with a printer | GJR5241730R2 |

Spare Parts:

| 07 SK 34 supply cable for 07 PG 30 | GJV3072207R1 |
| 24-pole EPROM adapter | GJR5224213R1 |
1.4 General Description of the Device

The device 07 PG 30 has been designed for use in laboratories and offices as well as at the machine (program test). It is positioned in a handy plastic casing. The keyboard is protected by a cover. Inside the casing is coated with conductive material to protect against failures. A short instruction informing about the most important orders and giving information while operating the device can be found under the cover.

1.5 Operating Elements and Device Connections

1. System interface connection for SIGMA* -tronic p and PROCONTIC b (bit processing, 2 K-words)

2. connection for the covering of the cables attachable to interface 1

Attention!

Follow the instruction in paragraph 10 before the PROCONTIC b or the SIGMA* -tronic p cables being attached to or taken off the connection of the 07 PG 30.

3. Serial interface - RS 232 C

Hereby the connection of the programming systems 907 PC 30 or 907 PC 31 or 07 TD 12 or a printer (e.g. 07 DR 11) is made possible.

4. Power supply

Connection for power supply from the power supply unit 07 NG 30 or from a control with the supply cable 07 SK 34.

5. ON/OFF-switch

Switches the operating voltage on and off. The buffer voltage for the RAM is not influenced.

6. Programming equipment, 28-pole zero power base. The zero power base takes the EPROMs for loading, testing, comparing and reading purposes (if necessary by means of an EPROM adapter delivered with it).

7. LCD-Indication

The one-line indication consists of 16 positions. The characters are produced by means of a 5x7 point-matrix.

8. ASCII-keyboard with function keys.
1.6 The Buffered Program Storage

The control routine is filed in a battery-buffered RAM and remains stored even after switching off the device. The batteries have a service life of up to one year.

The battery box is below the LCD indication and accessible from the bottom of the device.

In case of not using the 07 PG 30 for a longer period it is recommendable to take out the batteries to save their capacities. Hereby a program possibly being in the RAM storage is erased.
2 Possibilities of Use and Combination

2.1 07 PG 30 as an Independent Programmer

In order to document programs the printer 07 DR 11 or any other printer with serial connections can be connected with the serial interface. The data are transferred by means of the printer cable 07 SK 33.

Data format: 8 data bits
1 stop bit
no parity
handshake XON/XOFF
transfer speed adjustable
between 300...2400 Baud

The transfer speed adjusted in the programmer can be interrogated by the function "$B$" and altered by "$Bxxxx$" (cf. paragraph 8).

After switching on the device there is always a transfer speed of 1200 Baud. In order to print out parts of the program the editing instruction "$L$" must be used (cf. paragraph 6).

The overall program can be printed out by using the function "$SLIST$" (cf. paragraph 8).

Setting for the serial interfaces of the printer 07 DR 11:

<table>
<thead>
<tr>
<th>SW 1</th>
<th>ON</th>
<th>OFF</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td>Japanese character set</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td>Japanese character set</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td>normal character set</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td>72 lines</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td>XON-XOFF telegram</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td>LF at DATA BUFFER full</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td>Print at CR, LF, VT, FF, US</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td>CR without LF</td>
</tr>
</tbody>
</table>

SW 2

<table>
<thead>
<tr>
<th>ON</th>
<th>OFF</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>PRINT 0 with (/)</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>full buffer magnitude</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>no function</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>no function</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>10 CPI</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>7 DATA bit</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>select by power ON</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>undirect print (diagram)</td>
</tr>
</tbody>
</table>

SW 21

<table>
<thead>
<tr>
<th>ON</th>
<th>OFF</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>1 stop bit</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>send data select</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>no parity</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>no parity</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>no function</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>8 bit</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>XON-XOFF telegram</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>XON-XOFF telegram</td>
</tr>
</tbody>
</table>

Together with the power supply unit 07 NG 30 resp. the supply cable 07 SK 34 the device 07 PG 30 is a completely independent programmer. In order to make programs for the control routines SIGMA*-ronic p and PROCONTIC the following is available for this stand-alone equipment:

- all editing functions
- all EPROM programming functions.

2.2 07 PG 30 with Printer
The BBC compact control SIGMA*-tronic p can be connected at the system interface. The data are exchanged via the cable 07 SK 31. The EPROM adapter of the cable is plugged at the SIGMA*-tronic p instead of the EPROM's 07 PR 82. In this configuration the program in the RAM storage of the 07 PG 30 can be depleted and tested. The buffered RAM of the 07 PG 30 then serves as the program storage of the control. All editing, testing and EPROM programming functions may be used.

The programmer is supplied with power via the SIGMA*-tronic p cable 07 SK 31 from the control.

Attention!

In this case, the power supply unit 07 NG 30 cannot be used to supply the 07 PG 30 and the control SIGMA*-tronic p.

2.4 07 PG 30 with PROCONTIC b

The PROCONTIC b cable 07 SK 30 helps to connect the system interface of the 07 PG 30 with the IE-interface of the 07 ZE 82 of the PROCONTIC b. This configuration allows the depletion of the program and the testing of the bit processing PROCONTIC b from the RAM storage of the 07 PG 30. All editing, testing and EPROM programming functions may be used.

As for the program storage the EPROM on the main frame can be chosen as can be the buffered RAM storage of the 07 PG 30 into which the program was read or typed.

Attention!

The instructions in paragraph 10 have to be followed by all means before connecting the 07 PG 30.
2.5 07 PG 30 as Test and Starting Unit

The device 07 PG 30 replaces the test and programming unit 07 IE 82 if combined with the program systems 907 PC 30 or 907 PC 31.

When connecting the 07 PG 30 with the Procontic b it is by all means necessary to pay attention to the instructions in paragraph 13. In addition to the connecting of the 07 PG 30 with the 07 ZE 82 the serial interface of the 07 PG 30 must be connected with the serial interface (COM1) of the programming systems 907 PC 30 or 907 PC 31 with the computer cable 07 SK 32.

By giving the instruction "$RE$ (cf. paragraph 10) the 07 PG 30 is switched into the operation "test and starting unit". By means of the orders "$SPPS$" and "$TEST$" the connected programming systems 907 PC 30 or 907 PC 31 can communicate with the 07 PG 30 (cf. the instructions "907 PC 31 Testing and Programming System for PROCONTIC b" and "907 PC 30 Testing and Programming System for PROCONTIC b").

If a program with 907 PC 30 is to be tested at the control system PROCONTIC b (bit processing), the procedure is as follows:

<table>
<thead>
<tr>
<th>step</th>
<th>907 PC 30</th>
<th>07 PG 30</th>
<th>meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>HALT</td>
<td>if 07 PG 30 not at HALT</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>$CL$</td>
<td>erase RAM storage</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>$RE$</td>
<td>set operation &quot;test and start&quot; transfer of the program from the PC to the 07 PG 30 end transfer mod start program start indication of status test cancel indication of status end REMOTE operation</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>$SPPS$</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>E</td>
<td>RUN RAM B2</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>TEST</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>ESC</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

In the RAM storage of the 07 PG 30 only the indicated range is overwritten by the program transferred by the programming and testing systems 907 PC 30 or 907 PC 31. The rest remains unchanged.

It is not possible to use the function "error search" with 907 PC 30 and 907 PC 31 in connection with the 07 PG 30.

**Note:**
In ONLINE working of 07 PG 30 with 907 PC 31 the error indication "error, PLC stopped" is indicated, although the PLC is running and the variables are actualized.

2.6 07 PG 30 as an EPROM programmer

With the EPROM programming device installed programs from the RAM storage of the 07 PG 30 can directly be transferred into an EPROM connected.

Programs with analog value processing and word processing cannot be entered and tested, as far as the 07 PC concerned. It is possible, however, to duplicate 4K-EPROMs 07 PR 84 R1 and 8K-EPROMs 07 PR 84 R2 by means of the 07 PG 30. The programs filed in these storage types can contain analog value and word processing. As the storage in the device only comprises 4K, 8K-words must be duplicated in two steps.

The necessary orders are described in paragraph 8.

2.7 Serial interface of 07 TZ 82 R201

If working with the 07 PG 30 in the serial interface on the 07 TZ 82 R201, the serial interface of 07 TZ 82 R201 has to be set on 7 bits, 2 stop bits and even parity.

Setting has to be done as follows:

1. Solder the battery on 07 TZ 82
2. Connect the 07 PG 30 to 07 TZ 82
3. Input of 07 PG 30 in $TTY$ mode (300 baud) $SP = 1$
4. RUN/STOP-key on the 07 ZE 84 from STOP to RUN or switch off and on the supply voltage (24 VDC) of the PLC.

The settings on the serial interface are stored, if the battery is soldered.
3 Survey: Functions and Keys

The carrying out of the following edit instructions and $-functions is done by operating the space key <BLANK>. If the instruction is fully indicated.

If the 07 PG 30 recognizes an edit instruction or a $-function uniquely on the basis of the characters input, the missing characters are supplemented automatically.

 Corrections by operating the <DEL>-key are possible before operating <BLANK>. The function of this key may differ in the various edit and $-functions.

### 3.1 Edit Instructions

- **POINTER** move the program pointer
- **INSERT** include statement
- **OVERWRITE** instruction to overwrite
- **FIND** search single instructions
- **CHANGE** exchange single instructions
- **KILL** erase single instructions or program spaces
- **LIST** print out program spaces

Instructions marked * need not be finished with <BLANK>.

### 3.2 $-Functions

- **$AUS** switching off the system interface
- **$Bxxxx** adjust transfer speed for printers or STTY-mode
- **$B** interrogate adjusted baud rate
- **$CLEAR** erase the program in the 07 PG 30
- **$FREE** RAM storage completely (4K-words)
- **$PRINT** indicates vacant memory space
- **$LIST** print-out of the overall program
- **$PROM** check overall program for parity errors
- **$PROM** (RAM/EPROM)
- **$PRxxx** select EPROM programming function
- **$PROM** set EPROM type
- **$SRT** interrogate EPROM type set
- **$SPAMTEST** blank state test of the 07 PG 30 RAM storage
- **$REMOTE** change over into the operation test and starting unit
- **$SPS** transfer contents EPROM 07 ZE 82 to 07 PG 30
- **$SPSV** comparison of contents EPROM 07 ZE 82 with RAM 07 PG 30
- **$TTY** terminal operation
- **$HEX** search for program end (IPE)
- **$OPTION** hexadecimal code of an instruction
- **$DEUTSCH** functions "further loadings"
- **$ENGLISH** output of system messages in German
- **$FRANCAIS** output of system messages in French
- **$MONITOR** service functions
- **$XTTY** $XTTOFF

### 3.3 Special Instructions and Function Keys

- **<SHIFT><CTRL><RESET>** resetting of the 07 PG 30 (initialization without loss of program), operate keys simultaneously
- **<ESC>** cancellation of function (with restrictions)
- **<SHIFT>** second highly positioned key function
- **<DEL>** erases the last character (depends on the respective input)
- **<BLANK>** instruction end (take over after complete input) and space key
- **<EPROM:RAM>** changing over from EPROM to RAM and vice versa, e.g., in the operation TEST
- **<CR>** CARRIAGE RETURN (used in the TTY-mode)
- **↓** beginning of following sentence
- **↑** previous beginning of sentence
- **←** previous instruction
- **→** following instructions

With the exception of <SHIFT>, <CTRL>, <RESET>, <ESC> and <EPROM:RAM> all keys have an automatic repetition function, if the key remains pressed.

### 3.4 Instructions to Start a Program

- **TEST** start of the program with status indication at the last place of the LCD indication (RAM or EPROM)
- **RUN** start of the program (RAM or EPROM)
- **AUTORUN** automatic start of the program
- **HALT** after voltage connection programm stop

### 3.5 Special Keys for Program Input

- **< >** IF
- **< & >** AND
- **< I >** OR
- **< - >** THEN
After connecting the supplying voltage the remark "wait" is indicated, then after a few seconds:

BBC 07 PG 30 B2 if PROCONTIC b was selected
BBC 07 PG 30 P if SIGMA-tronic p was selected.

This also appears after an initialization with the keys <SHIFT> <CTRL> <RESET>.

During this indication the keys <P> and <B> allow a preselection of the respective control system. Changing over at a later point of time is impossible.

Attention: It must be secured that the selection with the 07 PG 30 (B2 or P) corresponds exactly with the control system to be connected. Do not forget to pay attention to the instructions in paragraph 10.

A software reset (<SHIFT> <CTRL> <RESET>) might lead to program alterations.

Example: By means of the function SPRT 64 the address range from 0...4095 was released and a program covering this full address range was made or transferred.

By means of a software reset EPROM type 2732 appears automatically. After changing over with SPRT64 the program is overwritten with the contents of the EPROM of the 07 ZE 82 from address 2048 onwards with PROCONTIC b connected.

Input: Before the program input the previous indication (BBC 07 PG 30...) has to be acknowledged by means of the space key <BLANK>.

Indication: 0000 <contents of address 0> i
The input of the control routine is only possible by means of the keyboard already described.

The full PROCONTIC b extent of language is available for the input of the program. In order to deplete a program without errors only the extent of language described in paragraph 9 is necessary for the 07 PG 30.

Note:
1. Only one key may be operated at a time (exceptions <SHIFT>, <CTRL>).
2. All keys are electronically unblocked. The input speed of the characters should not be higher than the indication speed.
3. Effect of the key <SHIFT> without SHIFT: <-> & indication: &
   with SHIFT: <SHIFT> <-> & indication: -

Remark: Following this a difference is made between POINTER and CURSOR. The POINTER is always positioned at the address indicated. It can be moved along the complete program stored. The CURSOR only moves within a program instruction in the indication. It marks the input position and is indicated by "i".

Every input is checked for plausibility. In a specific position the 07 PG 30 only accepts characters allowed there.

Example: There must always be a combination character or a blank character <BLANK> before a variable. If the input is A without a combination character, it is interpreted by the 07 PG 30 as the function AUTORUN, not as the starting variable.

The input of the instruction = E..... is not possible.

Every instruction is finished by operating the space key <BLANK> and is then taken over in the RAM storage of the 07 PG 30. The pointer then goes over to the following address and the device expects the following entry.

As soon as an entry is recognized uniquely by the device, possibly missing characters are added automatically.

Example: input of the following program:

```
! E 0 0 . 0 1
& M 0 0 . 0 1
/ A 0 1 , 1 2
= S M 0 2 , 0 0
```

input: !EO<BLANK>1<BLANK>&MO<BLANK>
1<BLANK>A1<BLANK>12<BLANK>
=SM<BLANK>2<BLANK>
0<BLANK>

In case of an input error it is possible to erase instructions not yet finished with <BLANK> by operating the <DEL> key from right to left.

Every program must include a nested IPE.
The following table informs about the input position of the operators and operands and their allocation to SIGMA\textsuperscript{\textregistered}-tronic (1) and PROCONTEC b (2):

<table>
<thead>
<tr>
<th>position</th>
<th>operator</th>
<th>(1)</th>
<th>(2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>N O P O</td>
<td>IF</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>N O P 1</td>
<td>AND</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>!</td>
<td>OR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&amp;</td>
<td>THEN</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

instructions
N C T
SET
ERASE

operands
entry
output
marker (local)
marker (global)
step (local)
step (global)
time
counter

special instructions
end of program
beginning of block
end of block
negated
beginning of block with normalization
beginning of block with normalization negated
attributes
time constant
word constant
text constant
binary constant

* must be programmed, but does not have an effect
(1) = SIGMA\textsuperscript{\textregistered}-tronic p
(2) = PROCONTEC b

In order to enter the binary constants B0 and B1 the cursor must be moved to the right by means of the space key <BLANK> until a question mark "?" is indicated. After that it is possible to enter the binary constant #B...

When entering = RM ..... IM ..... and #S ..... a blank character <BLANK> must be entered before giving the group number.

In case of text constants for the output with the digital timing unit 07 TZ 82 the input of constants is different to the indication.

Example: input: #007<BLANK>
Indication: #*007

is added automatically.

This presentation does not influence the function.

Text constants occupy several program addresses. It can happen that the complete text constant is not indicated. In order to present the missing part of the text constants it is necessary to position the pointer with the function "PTRAxxxx" in the address range of the constants.

For the input of the above mentioned characters it may turn out to be necessary to position the cursor rightly by means of the space key <BLANK>.

Remark: Error or system messages must be acknowledged by all means by the space key <BLANK> before any further input.
The LCD indication serves for communication purposes between the user and the device (system messages) as well as program indications. Programs are indicated word by word. In case of instructions consisting of several words, the first appertaining address (for exceptions cf. paragraph 8 pointer) is always indicated.

Indication structure:

Example: The program instruction INE02, 04 at the address 0578 looks like when indicated:

position: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
indication: 0 5 7 8 ! N E 0 2 . 0 4 i

Position 16 has a double function:
in the editing mode:
i = INSERT
o = OVERWRITE

in the test mode:
0 = the variable indicated has the status 0
1 = the variable indicated has the status 1.

Operation indications:

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BBC</td>
<td>Hereby the device reports after the switching on or after the initialization</td>
</tr>
<tr>
<td>07P</td>
<td>The question mark appears if</td>
</tr>
<tr>
<td>?</td>
<td>- an entry not fitting the position has been attempted. It is not necessary to acknowledge. The input must go ahead correctly.</td>
</tr>
<tr>
<td>B2</td>
<td>- the device expects an entry, e.g. REALLY Y:N?</td>
</tr>
<tr>
<td>WAIT</td>
<td>This indicates that the user has to wait because the processing of the instruction takes &quot;a bit longer&quot; (a few seconds)</td>
</tr>
<tr>
<td>RAM full</td>
<td>indicated after entering an instruction at address 2047</td>
</tr>
<tr>
<td>already in HALT</td>
<td>the instruction HALT was given, but this operation had already been set</td>
</tr>
<tr>
<td>just HALT possible</td>
<td>an instruction was given, but, however, it is only possible in the operation HALT</td>
</tr>
</tbody>
</table>

Further system messages are mentioned in the descriptions of the single instructions.
By means of the functions described in this chapter programs can be made, changed and supplemented. The following instructions can only be entered with the cursor being in the left position (position 6).

**POINTER**

By means of the function "pointer" (program pointer) it is possible to move forward and backward in the program. The address the pointer is directed at can be read in the indication.

```
program storage (RAM)
0000 ....
0001 ....

Indication:
0002 .... ← pointer

2046 ....
2046 ....
2047 ....
```

Absolute pointer instructions:

**Input:** PO<BLANK>  
**output:** pointer origin (beginning of the program)  
**indication:** address 0000 and its contents

**Input:** PE<BLANK>  
**output:** pointer end (end of program)  
**indication:** contents of the first address after the last instructions

**Input:** PA<x>BLANK>  
**output:** pointer at address absolute address x and its contents

If instructions comprise more than one address, it is not automatically the first address of the combined instruction that is indicated, but only the part after the given address.

**Input:** PSA<x>BLANK>  
**output:** pointer at sentence absolute first instruction of the sentence x (beginning of sentence; instruction with operand "I")

This moving of the pointer is only possible as far as really program sentences are concerned. If a sentence to be positioned at does not exist, an "I" is indicated. After entering the character A the indication disappears for a short time while operating that function, but that does not have any influence on the effect.

Relative pointer movements:

**Input:** P<x>BLANK>  
**output:** pointer moving along +x addresses  
**indication:** present word address +x addresses, contents of this address

**Input:** P-x<BLANK>  
**output:** pointer moving along -x addresses  
**indication:** present word address -x addresses, contents of this address

In case of instructions comprising more than one address the recognition of the first appertaining address is not automatic. If the pointer is at the beginning of a sentence whose first instruction is a compound word the functions "pointer +x" and "pointer -x" cannot be used.

**Input:** PS<x>BLANK>  
**output:** pointer moving along +x sentence  
**indication:** present sentence address +x sentences, contents of this address

After entering the character x the indication disappears for a short time which does not have any influence on the effect. In case of a position at the beginning of a sentence beginning with a compound word the following beginning of a sentence is reached by PS2<BLANK>.

**Input:** PS-x<BLANK>  
**output:** pointer moving along -x sentences  
**indication:** present sentence address -x sentences, contents of this address

After entering the character x the indication disappears for a short time which on the other hand does not have any influence on the effect.

**Pointer movements by means of the arrow keys:**

- **Key:**  
  - **Function:** following beginning of sentence
  - **Remark:** If the pointer is moved sentence by sentence in programs containing instructions occupying more than one address (compound words), it can happen that the beginnings of sentences are not always recognized.

During certain operating conditions it is not excluded that the arrow keys (up and down) are read as "A" and "B".

- **Key:**  
  - **Function:** following instruction

In case of instructions comprising more than one word positioning always takes place at the first appertaining address.
INSERT

During the input of a program a difference must be made between the INSERT and the OVERWRITE modes.

Changing over:
Input: 1
Indication: INSERT (a few seconds)
At the last position (16) of the indication

A changing over is only possible in case of an instruction indicated or the cursor being at position 6 of the indication.

The INSERT mode helps to insert instructions or parts of a program into an existing program. This can take place until an instruction different to NOP0 or NOP1 reaches the last program address. After this the message “RAM FULL” appears.

Remark: It is impossible to give out an instruction to the last address of the selected range (SPRT16: 1023, SPRT32: 2047, SPRT64: 4095). To do so it is necessary to change over to the OVERWRITE mode.

OVERWRITE

The OVERWRITE mode helps to overwrite existing instructions with new ones.

Change over:
Input: 0
Indication: OVERWRITE (a few seconds)
At the last position (16) of the indication

Remark: If instructions occupying one address only are overwritten with instructions occupying more than one address (compound words), the necessary number of addresses is inserted; i.e. the program part following the present address is removed.

FIND

The instruction FIND helps to search for instructions in the program. The input of the instruction to be searched for can take place with or without operator. Every operating of the space key (BLANK) means the continuation of the search. An abort of the function is possible by operating the key ESC. The indication of the address with the instruction found last remains.

Exception: Special instructions must be given by operator.

Example a: The instruction M00.00 is wanted.

Input: F<BLANK><BLANK>M0<BLANK>0<BLANK>
Indication: address with the instruction M00.00 and the pertaining operator.

Input: <BLANK>
Indication: wait (a few seconds)
Indication: not found.

Input: <BLANK>
Indication: address with the instruction M00.00 with the pertaining operator

Here the instruction wanted existed only once in the program.

Example b: The instruction INE01.00 is wanted.

Input: F<BLANK><BLANK>INE1<BLANK>0<BLANK>
Indication: address with the instruction INE01.00 or:
Indication: not found.

Remark: Constants cannot be searched for with FIND. If after the occurrence of errors during the input of this function further entries turn out to be impossible, the O7 PG 30 must be initialized again (keys: <SHIFT>, <CTRL>, <RESET>). Errors in the indication may occur during the search for instructions occupying more than three addresses.

CHANGE

The instruction CHANGE causes an exchange of the given instruction - from position of pointer onwards - as often as the instruction to be changed can be found in the program.

Input: C
Indication: CHANGE (at this point cancellation still possible with ESC)

Input: <BLANK>
Indication: from

Input: (former instruction)<BLANK>
Indication: in

Input: (new instruction)<BLANK>

The instructions can be entered with and without operator. If necessary the operator is to be replaced by a space character. Different entries (1st instruction with, 2nd instruction without operator are not allowed).

Example a: E00.00 is to be replaced by E01.00.

Input: C<BLANK>E2<BLANK>E0<BLANK>0<BLANK>
<BLANK>E1<BLANK>0<BLANK>

Example b: IE02.00 is to be replaced by IE03.00.

Input: C<BLANK>IE2<BLANK>0<BLANK>IE3
<BLANK>0<BLANK>

Remark: The exchange can take some time if simple instructions (one address) occurring in the text frequently are replaced by compound words (several addresses).

It is impossible to use the function CHANGE for instructions occupying three or more addresses. Before exchanging single word instructions with compound word instructions the availability of sufficient NOP0 after the end of the program must be checked. Otherwise a number of changes can have taken place before the indication of the message “RAM FULL”. An unknown rest would remain unprocessed in this case.

The existence of the instruction to be changed in the text is not checked. Changes take place up to the Internal program delimiter (can be traced by means of PTR E)
The instruction KILL leads to the erasure of an address range given in the form of absolute addresses.

In case of the INSERT mode being set the following program is shifted to the beginning with as many words as have been erased. In the OVERWRITE mode the instructions erased are replaced by NOPs.

Input: \(<K> <\text{BLANK}>\)  
indication: KILL  
input: (initial address)  
(final address)  
input: <\text{BLANK}>  
indication: final current program address  
(cancellation possible by <ESC>)  
(cancellation possible by <ESC>)

The contents of the absolute addresses 4 to 25 are to be erased.

Input: \(<K> <\text{BLANK}>\)  
indication: KILL  
input: 0004, 0025  
input: <\text{BLANK}>  
If just 25 is entered instead of 0025, the finish must be <\text{BLANK}> <\text{BLANK}>.

Remark:
Before calling the KILL function it must be checked if initial and final addresses are not in the range of instructions having more than one address.

Having entered ",," the first address cannot be erased by the DEL-key. Having fully entered the initial and final addresses the function can be cancelled by means of ESC.

The instruction LIST helps to print out parts of the program on a printer connected. The part of the program is given in the form of the absolute initial and final addresses.

Input: \(<L> <\text{BLANK}>\)  
indication: LIST  
input: (initial address)  
(final address)  
input: <\text{BLANK}>  
indication: final address entered after the end of the print-out  
(cancellation possible by ESC)  
(cancellation possible by ESC)

When entering the final address missing addresses can be replaced by <\text{BLANK}>. The print-out procedure cannot be cancelled.

Remark: Please pay attention to the initial and final addresses not being in the range of instructions occupying several addresses. The initial address of instructions of more than one word are not searched for automatically.
The remarks made in paragraph 10 have to be paid attention to by all means before connecting the 07 PG 30 to the SIGMA-tronic b or PROCONTIC b.

The start of a program can happen in various ways. Two conditions must always be fulfilled: the correct control type must be set and the correct type of storage the program shall be depleted from must be called.

The following combinations are possible:

- **program storage**
- **control routine**
  - RAM in the 07 PG 30
  - PROCONTIC b
  - (bit processing)
  - EPROM in the 07 ZE 82
  - PROCONTIC b
  - (bit processing)
  - RAM in the 07 PG 30
  - SIGMA-tronic p

### 7.1 Change-over between SIGMA-tronic p and PROCONTIC b

A change-over is impossible before the initialization (\textless SHIFT\textgreater, \textless CTRL\textgreater, \textless RESET\textgreater) or the switching on of the device.

Selection PROCONTIC b:
- **Input:** B
- **Indication:** BBC 07 PG 30 B2

Selection SIGMA-tronic p:
- **Input:** P
- **Indication:** BBC 07 PG 30 P

### 7.2 Special Functions

#### TEST

By means of the function TEST the depletion of the program out of the selected type of storage begins and the status of the indicated instruction at the last position (16) of the indication panel is indicated.

- **Input:** T
- **Indication:** TEST RAM B2?

  - **TEST RAM B2?**
    - if PROCONTIC b and RAM are selected
    - TEST EPROM B2?
      - if PROCONTIC b and EPROM selected (change-over by means of key EPROM RAM possible)
      - TEST P?
        - if SIGMA-tronic p is selected

- **Input:** Y or \textless BLANK\textgreater
- **Indication:** test run started
- **Input:** N
- **Indication:** cancellation of test function.

If there is no connection to SPS or if the connection is broken, no SPS is indicated. Acknowledgement by \textless BLANK\textgreater.

In the operation TEST the following other functions are possible:

- **Pointer A**
  - **Input:** PA \textless BLANK\textgreater
  - **Indication:** PA
  - **Input:** \textless XXX\textgreater
  - **Indication:** Address \textless XXX\textgreater and its contents.

- **Pointer +1**
  - **Input:** P+
  - **Indication:** current word address +1 and its contents.

- **Pointer -1**
  - **Input:** P-
  - **Indication:** current word address -1 and its contents.

- **Pointer S+1**
  - **Input:** PS+
  - **Indication:** beginning of the following sentence.

- **Pointer S-1**
  - **Input:** PS
  - **Indication:** beginning of the previous sentence.

- **Pointer End**
  - **Input:** PE \textless BLANK\textgreater
  - **Indication:** address after the last program address.

- **Pointer Beginning**
  - **Input:** PB \textless BLANK\textgreater
  - **Indication:** 0000 contents.

  - **Input:** FND, SPE, SFREE, SHEX, RUN, HALT
  - **Indication:** (description cf. paragraphs 9 and 10).

The contents of the RAM storage in the 07 PG 30 remain in the operation TEST EPROM B2. The contents of the EPROMs are indicated on the 07 ZE 82.

Cancellation of the function TEST:
- **Input:** R
- **Indication:** RUN RAM B2?
- **Input:** \textless BLANK\textgreater
- **Indication:** (contents of 07 PG 30-RAM)
- **Input:** H
- **Indication:** HALT?
- **Input:** \textless BLANK\textgreater
- **Indication:** (the operating control is halted)
- **Input:** \textless BLANK\textgreater
- **Indication:** (contents of 07 PG 30-RAM)

**program:** 07 PG 30  sum: 5.56
If a wrong key is pressed during the TEST operation a question mark (?) appears at position 6 in the indication panel. The status indication remains on the indication panel but is not updated any more.

Remark with respect to the 907 PC 30 resp. 907 PC 31:
If HALT is entered in the 07 PG 30, it takes some 10 seconds until 907 PC 30 resp. 907 PC 31 bring out a respective message if the functions TEST or ONLINE are set there.

No status is indicated for the program instruction = MA.

**RUN**

The program depletion of the control is started manually by means of the function RUN, however without status indication.

Input: R

Indication: RUN RAM B2?

RUN EPROM B2?

RUN P?

If the control has not been connected or if the connection is disturbed, "no SPS" is indicated. Acknowledge this message by <BLANK>.

If the operation RUN was set, the control is influenced after a software reset (keys <SHIFT>, <CTRL>, <RESET>) as follows:

The working program is stopped. All outputs are set back, indication of the title line BBC 07 PG 30.

When selecting the functions "RUN... or "AUTORUN..." a preselected EPROM type is automatically changed into SPRT 32.

When selecting the functions "RUN... or "AUTORUN..." a preselected EPROM type is automatically changed into SPRT 32.

**AUTORUN**

If the operation AUTORUN was set before the supply voltage was switched off the program starts automatically after switching the voltage on again.

Input: A

Indication: AUTORUN RAM B2?

AUTORUN EPROM B2

AUTORUN P?

Remark:
If an EPROM is plugged onto the main frame 07 ZE 82 in the operation AUTORUN RAM B2, the message "R2PROM out" appears. The possibility of an EPROM being on the 07 ZE 82 is only checked during the initialization of the device (07 PG 30 switched on off: keys <SHIFT>, <CTRL>, <RESET> ).

When selecting the functions "RUN... or "AUTORUN..." a preselected EPROM type is automatically changed into SPRT 32.

If the operation AUTORUN was set, the control is influenced after a software reset (keys <SHIFT>, <CTRL>, <RESET>) as follows:

control continues indication of the current address.

**HALT**

Using the instruction HALT leads to the stopping of a program started with TEST, RUN or AUTORUN.

Input: H

Indication: HALT?

Input: Y or <BLANK> the program is stopped

Input: N the program continues.

If the operation HALT was already set, the message "HALT already" is indicated.

After this function being carried out the green light diode on the 07 ZE 82 is not extinguished, but the control is stopped.

In case a digital timing unit 07 TZ 82 in the unit PROCONTIC b the red light diode of the 07 TZ 82 changes to red continuous light after calling this function: "error 4" (no PE in the program) is indicated via the serial interface.

**Change over between RAM and EPROM operations**

The change over of the program storage is only possible in the course of the operation HALT; it materializes after one of the staring functions TEST, RUN or AUTORUN.

Example:

Input: H

Indication: HALT?

Input: Y or <BLANK> the program is stopped

Input: T indication: TEST EPROM B2?

Input: key EPROM:RAM indication: TEST RAM B2?

Input: Y or <BLANK> program from the RAM storage is started.
$\text{SAUS}$

This function helps to switch the serial interface of the 07 PG 30 in such a way that the PROCONTIC b is not influenced by the plugging of the PROCONTIC b cable 07 SK 30. This function must always be called before the switching of the connection (read the remarks in Annexure A).

Input: $\text{SA<BLANK>}$
indication: plug B2:
connect working with the control possible.

$\text{SBAUD}$

This function helps to set the data transfer speed (Baud rate) to a printer connected or for a transfer in the $\text{TTY}$ mode.

Input: $\text{SB3<BLANK>}$ 300 Baud
$\text{SB12<BLANK>}$ 1200 Baud
$\text{SB24<BLANK>}$ 2400 Baud.

The transfer speed set can be interrogated as follows:

input: $\text{SB<BLANK>}$
indication: xxxx BAUD $(xxxx = \text{Baud rate set})$.

The value 1200 Baud is set automatically after the device switched on or after initialization (SHIFT, CTRL, RESET).

$\text{SCLEAR}$

This function erases the complete program storage (4K words RAM).

Input: $\text{SCl<BLANK>}$
indication: REALLY Y/N?
in put: Y
indication: 0000
input: N

the complete program is erased irretrievably NOP0 program and pointer position remain unchanged.

$\text{SFREE}$

This function indicates the storage still vacant.

Input: $\text{SFR<BLANK>}$
indication: vacancy for xxxx words.

xxxx = number of vacant storage addresses existing after the last program instruction. NOP0 and NOP1 are recognized as vacant storage spaces. Addresses occupied by NOP0 or NOP1 within a program remain unconsidered.

$\text{S\text{LIST}}$

The function $\text{S\text{LIST}}$ is used if the overall program is to be printed out by a printer connected.

Input: $\text{SL<BLANK>}$
indication: LIST in operation.
Cancellation of the printing procedure by operating key <ESC>.

$\text{SPARITY}$

After calling this function the program is checked for parity faults. According to the setting the check takes place in the RAM storage or in the EPROM on the EPROM frame of the 07 PG 30.

Input: $\text{SP<BLANK>}$
indication: PARITY xxxx
PARITY RAM
xxxx = EPROM type set
if RAM is selected, change over possible
input: <BLANK>
indication: parity OK.
ERROR in: yyyy
yyyy = address where a parity error was found.

Acknowledgement of the last indication by <BLANK>. The parity check always begins at address 0000 and ends with the first parity error. After the detection of the first error the call of this function must be repeated until the indication "parity OK" appears.

$\text{SPROM}$

By this function a data communication between the RAM storage and the EPROM plugged onto the EPROM socket of the 07 PG 30 becomes possible.

Input: $\text{SPRO}$
indication: SPROM(L, S, T, V; E)
input: L<BLANK>
S<BLANK>
T<BLANK>
V<BLANK>
E<BLANK>
indication: SPROMLxxxx
SPROMSxxxx comparison OK
SPROMTxxxx ERROR AT: yyyy
PROM vacant
SPROMVxxxx ERROR AT: yyyy
comparison OK
xxxx = EPROM type set.

Acknowledgement of the last message by <BLANK>.
EPROM-types longer than 4K-words must be processed in two steps. By preselecting the EPROM type the indication mentioned below appears automatically.

Indication: 0000-4095 Y/N?
input: N
indication: 4096-8191 Y/N?
input: Y the 4K-range selected is processed.

The function is impossible in the operations TEST, RUN and AUTORUN. The correct EPROM type must be set in advance by all means (cf. the following paragraph).

$$PRxxx$$

Function to select the various EPROM types. The following types can be chosen among:

input: $PR0<BLANK>
$PR82<BLANK>
$PR84R1<BLANK>
$PR84R2<BLANK>
$PR2<BLANK>
$PR32<BLANK>
$PR64<BLANK>
$PR128<BLANK>

ABB type 07 PR 80 (2716)
ABB type 07 PR 82 (2732
ABB type 07 PR 84 R1
ABB type 07 PR 84 R2
2716
2732
2754, 2764
2718

After switching on or after initialization the EPROM type 07 PR 82 (2732) is set automatically.

Interrogation of the EPROM type set is possible as follows:

input: $PR<BLANK>
indication: TYPE 27xxx.

It is not the BBC type specification that is indicated, but always 27xxx (xxx = cf. above).

In order to apply the EPROM type 07 PR 82 it is necessary to plug on the EPROM adapter (GJR52242131R1) delivered together with it as an accessory (cf. remark on the inside of the keyboard cover).

$$RAMTEST$$

The buffered program storage (RAM) undergoes a blank state test.

Input: $RA<BLANK>
indication: RAM vacant
ERROR AT: xxxx
RAM storage not vacant at address xxxx

$$REMOTE$$

This instruction helps the 07 PG 30 to change over into the test and start mode. Program transfer and program test then take place by means of the programming and testing systems 907 PC 30 or 907 PC 31 via the serial interface. This system operates on the basis of IBM-compatible personal computers.

The 07 PG 30 is connected with the computer cable 07 SK 32. In this configuration the 07 PG 30 replaces the test and programming unit 07 IE 82.

Input: $RE<BLANK>
indication: REMOTE
data exchange with the personal computer (PC) possible.

In the REMOTE operation the following entries are possible:

input: $R<BLANK>
indication: RUN RAM B2?
RUN EPROM B2?
RUN P?

input: $A<BLANK>
indication: AUTORUN RAM B2?
AUTORUN EPROM B2?
AUTORUN P?

input: $H
indication: HALT?
input: $E<BLANK>
indication: REMOTE end Y/N?

The program depletion is stopped, a respective error message appears on the screen of the PC after a few seconds.

No valid status is transmitted for the program instruction =MA by the 07 PG 30 to the personal computer (907 PC 30) or the 07 TD 12 connected. There the status 1 is always indicated for =MA during test operations. In case of error messages or change of operation of the 07 PG 30 during the remote operation the programming and testing systems 907 PC 30 or 07 TD 12 produce the message "control does not report".

$$SPS$$ (for PROCONTIC b2 only)

This function helps to transfer the contents of the EPROMs on the main frame of the PROCONTIC b 07ZE 82 to the RAM storage of the 07 PG 30.

Input: $SP<BLANK>
indication: wait (short time)
indication: contents of the first program address after IPE.

Care!
If the PLC is not connected to the 07 PG 30, and the function $SPS is called, the program stored in the 07 PG 33 is deleted.

$$SPSY$$ (for PROCONTIC b2 only)

This function compares the contents of the EPROMs on the main frame 07ZE 82 with the contents of the RAM storage of the 07 PG 30.
**STTY**

In this operation the 07 PG 30 is used as a terminal. Thus data can be exchanged with other data sources in half-duplex or full-duplex operations via the serial interface (terminal operation).

**S**

Input: \$T<BLANK>
indication: HALF-DUPLEX Y/N?
input:  Y
    half-duplex operation
or:   N
    full-duplex operation
indication: _
    cursor only.

Now any signs may be entered. Signs received are shown on the indication panel. The indication is erased by the sign following <CR> (input or reception).

In case of having changed over to the English or French language before calling this function, "yes" resp. "oui" have to be entered instead of J.

A two-fold operation of the key ESC cancels this function.

**Remark concerning working with the digital timing unit 07 TZ 82 R201:** Time values can be read out resp. new setpoints can be given via the serial interface of the 07 PG 30 and the serial interface of the 07 TZ 82. The transfer can be done with capital letters only. Therefore the SHIFT key has to be operated during the input via the 07 PG 30.

The data exchange with the digital timing unit 07 TZ 82 R101 is not possible.

Setting of the serial interface of the 07 PG 30:

- 7 data bits
- 2 stop bits
- even parity

In case of the digital timing unit 07 TZ 82 R201 from position b the above mentioned data format must be adjusted by means of the 07 PG 30.

Input in the STTY mode: \$T = 1 <CR>

After a stop and further start of the control (run/stop switch) or in case of a connected buffer battery after connecting/disconnecting the power supply the setting of the protocol in the 07 TZ 82 applies (for further information cf. system description Proconic b, chapter 2, section 8.4)

**SPE**

This function searches either in the RAM storage or in the EPROM on the socket of the 07 PG 30 from the address 0000 for the first program end (IPE).

Input: \$SPE
indication: PE RAM
    PE yyyy
    if the RAM storage was selected
    if EPROM was selected
    yyyy = EPROM type)

input: key EPROM RAM
    if a different storage is wanted
input: <BLANK>
indication: xxxx \& PE
    xxxx = appertaining address
    not found
    no IPE there.

**$HEX**

This function helps to present the contents of the program address in a hexadecimal way. The presentation always refers to the current indication only.

Example:
indication: 0234 \& PE
input: \$H<BLANK>
    0234 h0B80 h = hexadecimal code
    <BLANK>
    indication: 0234 \& PE

The input of programs can take place in hexadecimal presentation:

Example: The instruction RA03.02 is to be entered in the hexadecimal code.
Input: <BLANK> 3032 <BLANK>.

After entering hexadecimal bit patterns misleading interpretations might occur. Therefore the checking of the correctness of the input is by all means necessary.

**SOPTION**

By means of this function further abilities of a specific EPROM can be read into the EPROM socket installed in the device. The appertaining EPROM has a specific identification code being interrogated by the 07 PG 30. If this identification code is not contained in the EPROM plugged in, the message "wrong version" appears. The description of the additional functions cannot be found in this manual.

The EPROM with the options is not ready for delivery at present.

**SDEUTSCH, ENGLISH, FRANCAIS**

The output of system messages in German, English, and French is the effect of the use of these functions. The language German is automatically set after switching on the device resp. after the initialization (SHIFT, CTRL, RESET). The input of the functions is identical in all three languages.

Input: \$EN<BLANK>
    indication of the messages in English
    SFRA<BLANK>
    SD<BLANK>
    indication of the messages in French
    indication of the messages in German.

**$MUNITOR, $MONITOT, $SXTY, $SXTOFF**

The use of these functions is restricted to BBC service personnel only.
9 Extent of Language of the 07 PG 30

Cf. software description of PROCONTIC b as well.

Operators:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IF</td>
<td>beginning of sentence</td>
</tr>
<tr>
<td>AND</td>
<td></td>
</tr>
<tr>
<td>OR</td>
<td></td>
</tr>
<tr>
<td>THEN</td>
<td>allocation</td>
</tr>
<tr>
<td>NOT</td>
<td>negation</td>
</tr>
<tr>
<td>SET</td>
<td></td>
</tr>
<tr>
<td>RESET</td>
<td></td>
</tr>
</tbody>
</table>

Operands:

| Inputs: | E 00.00 ....... E 15.15 |
| Outputs: | A 00.00 ....... A 15.15 |
| Marker: | M 00.00 ....... M 07.15 |
|         | M 00.00 ....... M 63.15 with 07 AS 82 only |
| Steps:  | S 00.00 ....... S 07.15 only with 07 AS 82 |
|         | S' 00.00 ....... S' 63.15 or 07 RK 80 |
| Timings: | T 00.00 ....... T 03.15 with 07 AS 82 only |
| Counter: | Z 00.00 ....... Z 00.15 with timing units only |
|         | Z 00.00 ....... Z 00.15 with 07 TZ 82 only |

= MA 
= special instructions:

= M A 
= beginning of subprogram

= NMA 
= MC 
= NMC 
= end of subprogram

! M E 
! P E 
= end of program

Attributes:

c.b change of format with 07 TZ 82 only

time constants:
#0, 0 0 E 0 .... #0, 9 9 E 3
#0, 0 0 0 E 0 .... #0, 9 9 9 E 5 with 07 TZ 82 only

word constants:
#W 0 0 0 0 0 ... #W 3 2 7 6 7

text constants:
#Text ASCII characters max. 10

Binary constants:
#B0
#B1

text elements:
= BS00 F000 appertaining element bit B00 with 07 TZ 82 only

cycle supervising element:
BS00 F000 appertaining element bit B64 with 07 TZ 82 only
Connection of the 07 PG 30 with the PROCONTIC b

The specifications of the respective system descriptions must be followed strictly as far as the supply voltages are concerned.

Attention!

Please do not touch the contacts of the system plug for the connection of the controls SIGMA :tronic p and PROCONTIC b (electrostatically endangered elements).

Procedure for the connection of the 07PG30 with the PROCONTIC b (07 ZE 82):

1. Connection with an operating control PROCONTIC b 07 ZE 82 by means of the cable 07 SK 30 (program depletion of the EPROM on the 07 ZE 82).

The following operations are possible:

a) passive listening in to comprehend the status (TEST)
b) transfer of the program from EPROM 07 ZE 82 in the 07 PG 30 RAM storage (program depletion of the PROCONTIC b is not disrupted).
c) active influencing of the control (program depletion stopped, program depletion of the 07 PG 30 RAM storage).

The following procedure must be taken in the above mentioned cases a), b) and c):

- connect supply voltage to 07 PG 30
  a) with the cable 07 SK 34, +24 V directly from the PROCONTIC b or
  b) with the power supply unit 07 NG 30

- switch on 07 PG 30, switch at position ON

- call $-function SANS
  This function switches the system interface of the 07 PG 30 that way that the PROCONTIC (07 ZE 82) is not influenced when plugging the PROCONTIC b cable. The 07 PG 30 automatically changes over to the operation HALT (NOT PROCONTIC b2-HALT!).

- connection to the 07 ZE 82 by means of PROCONTIC b cable 07 SK 30

- use functions wanted.

Procedure when switching off:

Attention!

The following procedure is not allowed:

- Switching off the supply voltage of the 07 PG 30 (switch to OFF) as long as the PROCONTIC b cable 07 SK 30 is still plugged.

- Removing the PROCONTIC b cable 07 SK 30 before the program depletion of the EPROM of the 07 ZE 82 is continued by a respective instruction. The system interface is neutral then.

Switching off according to 1a) and 1b):

- Remove PROCONTIC b cable 07 SK 30. PROCONTIC b continues without disturbances, switch off 07 PG 30 (switch to OFF)

Switching off according to 1c):

- Call function "RUNEPROM B2".
  This initializes the PROCONTIC b. All outputs, markers and
  steps occurring in the EPROM program are reset.
  The program depletion of the EPROM 07 ZE 82 is started.

- Remove PROCONTIC b cable 07 SK 30, switch off 07 PG 30 (switch to OFF)

2. The 07 PG 30 is already connected to a PROCONTIC b control:

- voltage supply for the 07 PG 30 via:
  07 SK 34 (24 V directly from the PROCONTIC b)
  07 SK 30 (PROCONTIC b cable)
  07 NG 30 (must be switched on power network sided together with the PROCONTIC b)

- PROCONTIC b cable 07 SK 30 is plugged in

- ON/OFF switch of the 07 PG 30 at ON.

Common switching of the supply voltage of the PROCONTIC b and the 07 PG 30:

If the operations "AUTORUN RAM B2" or "AUTORUN EPROM B2" are preselected, the program depletion starts automatically.

Procedure for the switching off:

the configuration remains:

- the supply voltage is switched off
- 07 PG 30 is taken out of the configuration; for procedure of 1.
11 Accessories

11.1 07 SK 30 R2 Cable to Connect PROCONTIC b
(07 ZE 82)

Table of connections:

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A6</td>
</tr>
<tr>
<td>2</td>
<td>A3</td>
</tr>
<tr>
<td>3</td>
<td>A5</td>
</tr>
<tr>
<td>4</td>
<td>B2</td>
</tr>
<tr>
<td>5</td>
<td>C5</td>
</tr>
<tr>
<td>6</td>
<td>A1</td>
</tr>
<tr>
<td>7</td>
<td>B5</td>
</tr>
<tr>
<td>8</td>
<td>A1</td>
</tr>
<tr>
<td>9</td>
<td>C7</td>
</tr>
<tr>
<td>10</td>
<td>A1</td>
</tr>
<tr>
<td>11</td>
<td>B1</td>
</tr>
<tr>
<td>12</td>
<td>A1</td>
</tr>
<tr>
<td>13</td>
<td>C4</td>
</tr>
<tr>
<td>14</td>
<td>A1</td>
</tr>
<tr>
<td>15</td>
<td>C5</td>
</tr>
<tr>
<td>16</td>
<td>A1</td>
</tr>
<tr>
<td>17</td>
<td>C2</td>
</tr>
<tr>
<td>18</td>
<td>A1</td>
</tr>
<tr>
<td>19</td>
<td>C3</td>
</tr>
<tr>
<td>20</td>
<td>A1</td>
</tr>
<tr>
<td>21</td>
<td>B6</td>
</tr>
<tr>
<td>22</td>
<td>A1</td>
</tr>
<tr>
<td>23</td>
<td>A2</td>
</tr>
<tr>
<td>24</td>
<td>A1</td>
</tr>
<tr>
<td>25</td>
<td>A1,C1</td>
</tr>
<tr>
<td>26</td>
<td>-</td>
</tr>
<tr>
<td>27</td>
<td>-</td>
</tr>
<tr>
<td>28</td>
<td>-</td>
</tr>
<tr>
<td>29</td>
<td>-</td>
</tr>
<tr>
<td>30</td>
<td>-</td>
</tr>
<tr>
<td>31</td>
<td>-</td>
</tr>
<tr>
<td>32</td>
<td>B6</td>
</tr>
<tr>
<td>33</td>
<td>A7</td>
</tr>
<tr>
<td>34</td>
<td>A1,C8</td>
</tr>
<tr>
<td>35</td>
<td>-</td>
</tr>
<tr>
<td>36</td>
<td>-</td>
</tr>
<tr>
<td>37</td>
<td>-</td>
</tr>
<tr>
<td>38</td>
<td>-</td>
</tr>
<tr>
<td>39</td>
<td>-</td>
</tr>
<tr>
<td>40</td>
<td>-</td>
</tr>
</tbody>
</table>

11.2 07 SK 31 Cable to Connect SIGMA- tronic p

On the side of SIGMA- tronic p this cable consists of an EPROM adapter. The adapter contains active elements. This is the reason why it cannot be described more detailed.
11.3 07 SK 32 Cable to Connect a Personal Computer

On the side of the 07 PG 30 this cable consists of a 3-pole plug switch of 3.5 mm and on the side of the PC of a 25-pole standard plug (MIN-D socket).

11.4 07 SK 33 Cable to Connect a Printer

The cable consists of a 3-pole plug switch of 3.5 mm on the side of the 07 PG 30 and of a 25-pole standard plug (MIN-D plugs).

11.5 07 SK 34 R2 Supply Cable

The cable consists of a 2-pole plug switch on the side of the 07 PG 30 and a 2-pole system plug on the side of the 24 V.
The unit 07 PH 32 is a commercially available IBM compatible personal computer with the following system data:

- 80C286 processor, 12 or 8 MHz
- 640 kbytes RAM
- 1 hard disk drive, 20 Mbytes
- 1 3 1/2" disk drive, 1.44 Mbytes
- 2 RS-232-C serial interfaces
- 1 parallel interface (centronics)
- MS-DOS operating system V3.31
- Supertwist screen, 9" diagonals
- RGB interface
- Connection for an external bulk storage unit
- Keyboard with 80 keys (including cursor keys and function keys)
- Mouse for a fast cursor movement
- Battery operation (approx. 3.5 hours)
- Power supply unit
- Adapter 9-polar to 25-polar
General

The 907 PC 32 programming and test software (order number GJP202300R102) for ABB Procontic T300, ABB Procontic T200, ABB Procontic b, ABB Procontic CS31, ABB Procontic K200 and SIGMA*tronic p is supplied together with a detailed set of operating instructions.

This software was designed for the 07 PH 32 IBM AT compatible personal computer. An extensively automatic installation program loads the 907 PC 32 software package into this unit or into another IBM AT compatible personal computer.

The 907 PC 32 programming and test software allows a simple and economical programming of PLC programs in:

- Function block diagrams (FBD)
- Ladder diagrams (LD)
- Instruction list (IL)

The program can be entered in symbols as well as absolutely. The PLC program is extended by symbolic codes, texts and comments. Help and error messages, which can be recalled at any time, as well as a syntax test facilitates the program entry. The compilation as a FBD or as a LD is carried out in a common editor. The elements can be mixed from FBD and LD and can be combined with each other.

Features

The scope of the listed features depends on the capabilities of the separate controls.

Menu guidance

- Modern, easy-to-follow menus in the pop-up menu technique
- Representation in colour (not for 07 PH 32)
- Fast selection of the menu points with the mouse or keyboard
- Recalling external programmes on the DOS level directly from the menu (DOS shell)

Path data

- Entry of a file name with the respective DOS path
- Displaying the configuration overview by means of the file directory

Password protection

- Access justification in several privileged levels

Modularisation (not for ABB Procontic K200)

- Handling large projects
- Dividing the projects into logical structures
- Division into program and variable modules

FBD/LD Editor

- Single editor for the programming with graphic symbols as a function block diagram and as a ladder diagram
- Combination of ladder diagram networks with elements of the function block diagram

Comfortable IL editor

- Representation with symbols and text in various forms
- Cursor control with the mouse
- Selection of connection elements via a selection menu with the mouse

Variable editor

- Complete list of all entered variables
- Ordering according to absolute or symbolic variables
- Acceptance and handing over of variable lists from and to any data processing systems
- Making variable lists available for certain CAD/CAE systems

Text Editor

- Entry of any ASCII files, max. 255 characters per line

Comments

- Verbal description of networks or program segments

Segment plans

- Division of the programs into segment plans
- Simple management by segment plan names and segment plan numbers
Functional scope

An extensive number of commands is available to compile programs:

- Syntax test of all variables
- Block commands - marking
  - deleting
  - moving
  - copying
  - saving
  - loading
  - printing
  - deleting unused variables
- Search commands - search the set number
  - search the word number
  - search the variable
  - search the symbol
  - search the command
  - search the line number
  - repeatation
  - search for the segment plan
  - search for the VE
  - search for the free connection
- Searching and replacing
- Insertion
- Deletion

ONLINE functions

Numerous ONLINE functions support the user when booting up, i.e., e.g.:

- Status display in
  - Function block diagram
  - Ladder diagram
  - Instruction list
  - Variable list
- Program
  - Transmission
  - Start
  - Abortion
  - Stop
  - Continuation
  - Status
- Single cycle on/off
- Single step on/off
- Breakpoint - setting it
  - displaying it
  - deleting it
- Triggering - by time
- Overwriting
- Typing
- Forcing
- Altering
  - Time and counter nominal values
  - Variable addresses
  - Operators
  - Operand codes
  - Program sections to a limited extent

Selected variables can also be summarised in ONLINE lists and their status displayed on the screen.

Program documentation

The automatic program documentation includes the output of the following lists by the printer:

- Function block diagram
- Instruction list
- CE library
- Logic plan
- Ladder diagram
- Variable list
- Cross reference list
- Comment list
- ONLINE list
- Text page
- Data area
- Modular list
- Gesamt variable list
- Entire reference list
- System configuration

The output is adapted to any printer.

Printing format editor

A special printing format editor allows the respective list to be extended by individual headers and footers. Certain data can automatically as well using the headers or footers, i.e., e.g., the name of the project file, date and time.