OPERATING MODES

|T| "TEXT" or text memory as data register

Input and reading of texts in the text memory

Input of texts is effected on RAM 6116, 6264 or 8464 or on the buffered RAM modules PCA1.R92/95/96 which are plugged onto the right-hand text socket of the basic module.

The following 2 possibilities are available:

a) Using one of the PCA programming units, connected via the PGU connector.

b) With a peripheral unit with current loop interface, connected to the serial data interface (7 terminals on the right).

For detailed description refer to manual Software level 2.

Manual access to the text memory as data register (PAS 54)
(as of system program version V6.004)

As mentioned in connection with the instruction PAS 54, the text memory can also be used as data register. In order to understand the monitor function of the system program, the formats which are used for organizing the various registers must be brought to mind:

```
Counter register      : binary 16 bits
Text memory           : binary  8 bits or 16 bits
(as data register)      or BCD 8 bits
```

For manual access to the text memory as data register the operating mode selector switch must be set to "TEXT".

- Display of the text memory contents

    a) Immediate display of a character value of 8 bits (1 byte) in binary notation

    Upon actuation of key |A| , and subsequent input of the character number to be displayed (0...8191), the stored value (0...255) is displayed in the operand field in binary notation.

Input:	Display:	STEP	CODE	OPERAND
|A| 1931		1931	00	0 135

```
Character no. ──────────────┘       │     │   │
                                    │     │   │
Always 00 0 ────────────────────────┘     │   │
                                          │   │
Value of 8 bits ──────────────────────────┘   │
(1 byte) in binary
notation
```

		1930	00	0157
|-|				
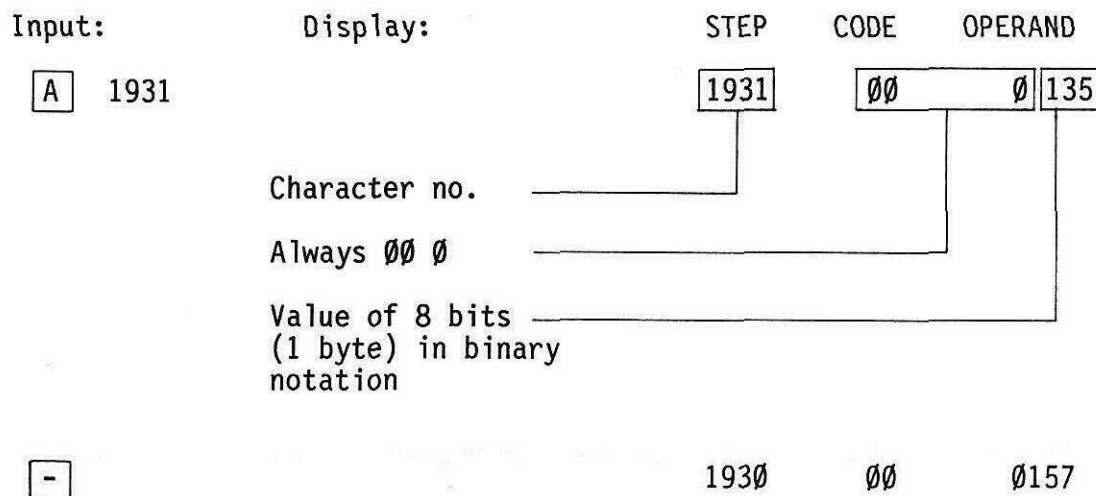
b) Display of the contents of 2 character no. (2 bytes = 16 bits) in binary notation

Actuating the key $\boxed{C}$ once has the result that in addition to the selected character number the value of the preceding character can also be combined to form a 16-bit value (2 bytes) in binary notation. Consequently values in the range 0...65'535 can be displayed in the CODE and OPERAND field.

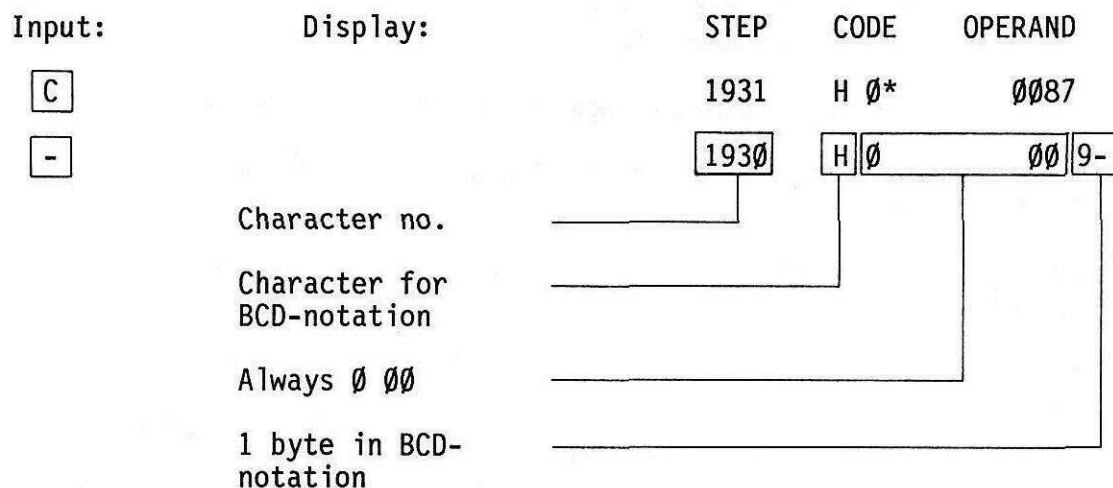| Input:	Display:	STEP	CODE	OPERAND
$\boxed{C}$		1931	E 3	4717

Character no. selected with key $\boxed{A}$

Character for 2 bytes

Value of the 2 characters 1930 and 1931 (2 bytes) in binary notation
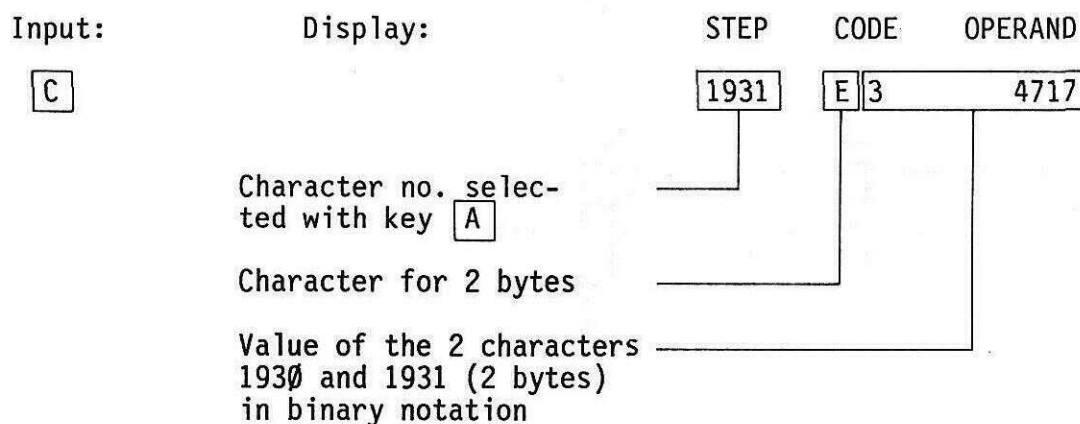
In this way, the contents of transferred counters can be displayed with their total capacity of 16 bits.

c) Display of 1 character no. (1 byte = 8 bits) in BCD-notation

By actuating key $\boxed{C}$ (convert) a second time, the bit pattern is displayed in BCD-notation.

Input:	Display:	STEP	CODE	OPERAND
$\boxed{C}$		1931	H 0*	0087
$\boxed{-}$		1930	H 0	00 9-

Character no.

Character for BCD-notation

Always 0 00

1 byte in BCD-notation

---

*) Character in the code applies to P05. For P10 refer to the table on the following page.

SAIA

Real BCD-bit patterns are displayed as decimals. If other characters are present as e.g. at character no. 1930, these are no BCD-bit patterns. In order to be able to interpret their values nevertheless, the following 7-segment characters are defined in the OPERAND:

Binary value	7-segment character P1Ø	P Ø5
1Ø		
11		
12		
13		
14		
15	blank	blank

- <u>Manual data inputs into the text memory</u>
  (limited RAM-memory in this area)

  Key [A] : subsequent input of the character no., at which the value is to
            be stored

  Key [E] : clears the old value and permits new input

  Key [C] : <u>before</u> key [E] means "convert"
            <u>after</u> key [E] means "clear"

  Key [+], [-], [A], [E]: cause storage of the value introduced

  Corresponding to the reading of data, 3 cases are distinguished for the manual input of data:

  a) <u>Input of a binary value of 1 byte (e.g. 48) at a character no.</u>
     <u>(e.g. 7436)</u>

Input:		Display:	STEP	CODE	OPERAND
[A]	7436		7436	ØØ	ØXXX
[E]		48	7436	ØØ	ØØ48
[+]			7437	ØØ	ØYYY

b) <u>Input of a binary value of 2 bytes (e.g. 1487) at character no. 7456</u>
<u>and 7457</u>

Input:			Display:		STEP	CODE	OPERAND
A	7457	1)			7457	ØØ	ØXXX
C		2)			7457	EY*	YYYY
E	1487	3)			7457	E1*	Ø487
C	Ø1487				7457	EØ*	1487
+					7459 4)	EZ*	ZZZZ

---

1) Always the higher address of a pair of 2 bytes is entered.

2) C before E results in the conversion to 2 bytes.

3) If values < 1Ø'ØØØ are entered, a Ø must be typed first.
   Correction with C .

4) The character no. is automatically increased by 2.

c) <u>Input of a BCD-value (e.g. 3Ø) at character no. 766Ø (in BCD-notation,</u>
<u>only values from Ø..99 ≙ 1 byte can be entered)</u>

Input:		Display:		STEP	CODE	OPERAND
A	766Ø			766Ø	ØØ	ØXXX
C				766Ø	EY*	YYYY
C				766Ø	HØ*	ØØZZ
E	3Ø			766Ø	HØ*	ØØ3Ø
+				7661	HØ*	ØØAB

---

*) Characters apply to PCA2.PØ5.

C 1 Selection of operating modes

Operating mode selector

PCA14

RUN
BREAK
STEP
MAN
TEXT
PROG.

V 1/1ø

PROG.

PCA15

STEP   CODE OPERAND

ORUN   ACCU=1 O
OBREAK
OSTEP
OTEXT   PCA2.PØ5
OMAN
OPROG

Display of the
selected
operating mode

Keys for selecting
the operating mode

R		B	S
	T	M	P
7	8	9	A
4	5	6	+
1	2	3	−
0	L	E	C

SAIA

PROG
TEXT
MAN
STEP
BREAK
RUN

Operating mode
selector

PCA8.PØ1

For all other
PCA-programming aids
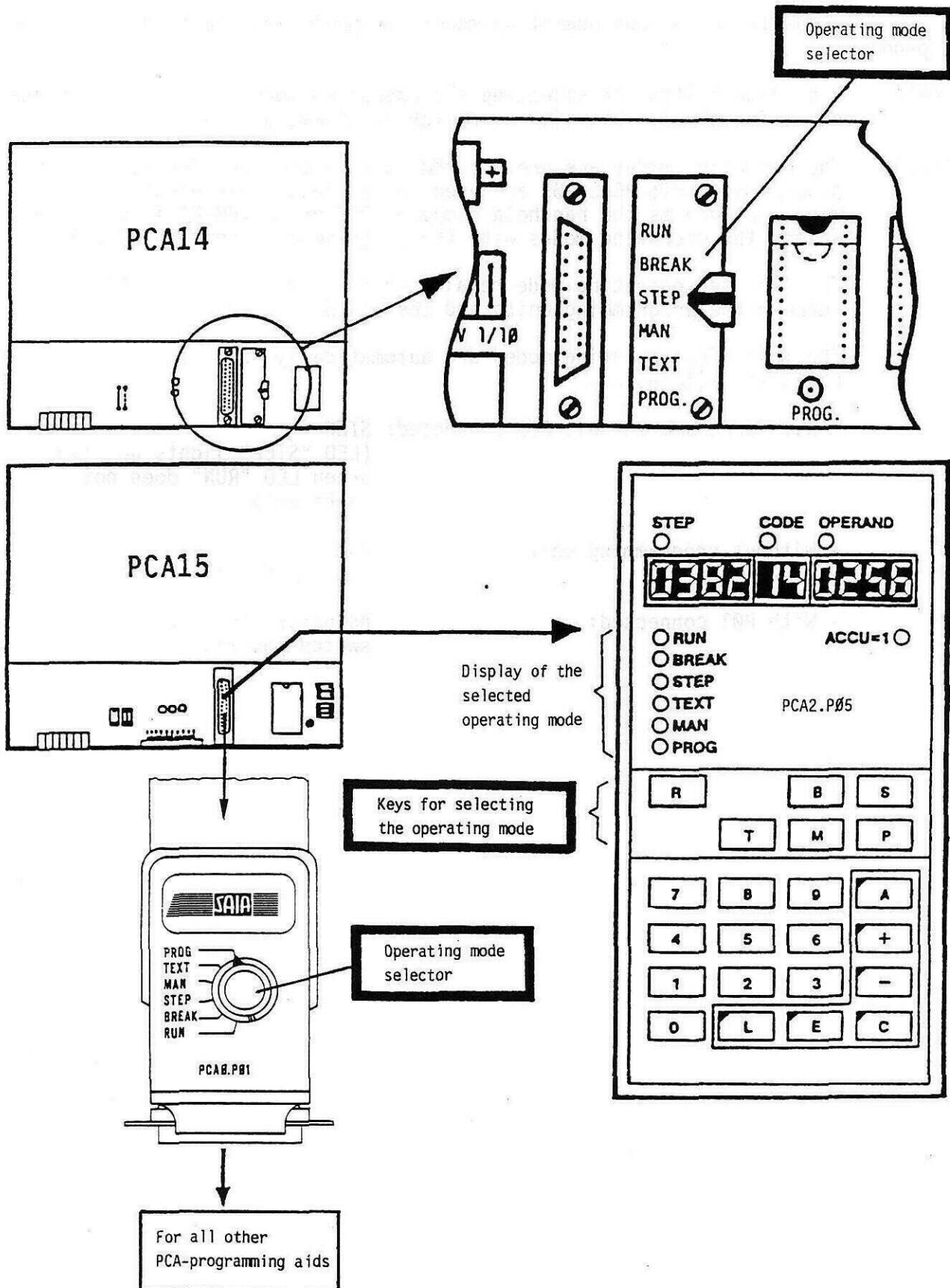
C 1.1 Operating modes, level 1H for PCA15 and PCA14

R	RUN	Normal program processing (lamp RUN on PCA15 lights up)
P	PROG	A user program can be loaded into a RAM memory (plugged onto the user plug-in socket of the PCA15).
M	MAN	Manual interrogation and setting of elements (inputs, outputs, flags, timers, counters)
S	STEP	Jump to a preselected step address (program line) of the user program and step-by-step operation
B	BREAK	Program processing up to a set "breakpoint" and subsequent step-by-step operation

C 1.2 Summary of operating modes

| R | **RUN** | <u>Normal program execution</u> |

The PCA1 is automatically in the RUN-mode when switching on if no programming unit is connected.

| P | **PROG** | <u>Programming</u> |

A program can be stored in a RAM-memory (on the user socket of the PCA1) or overwritten (corrected).

```
        Step              Code   Operand
A  x x x x        E   x x    x x x x

                  E   x x    x x x x    or  C  to delete a wrongly
                                                entered line

                  +    Terminates the input

Test program   +  +  or  -  -
```

| M | **MAN** ** | <u>Manual testing or setting of elements</u> |

(Elements = inputs, outputs, flags, counters, timers)

Testing:
```
        Step
A  x x x ────────────→ display of the logic state in the operand (∅/1)
    └─┬─┘
  Element address
```

Setting:
```
A  x x x  E  1  ←────→ or  ∅
  └─┬─┘
  Element address
```

| S | **STEP** | |

+ ──→ Display showing where the program is.

<u>Jump to the preselected step address of the user program</u>

A 139 + ──→ Program jumps to step 139, then

+ + ... step-by-step execution of the program with the result of the logic operation being checkable ☀ ACC = 1*. Switching to RUN is always possible.

In case of parallel programs, <u>only the activated parallel program</u> is executed in the STEP-mode.

| B | **BREAK** | <u>Interruption of the program run and subsequent step-by-step-operation</u> |

+ ──→ Display showing where the program is

+ + ... step-by-step execution of the program with the result of the logic operation being checkable ☀ ACC = 1*. Switching to RUN is always possible.

In case of parallel programs, all programs are executed simultaneously (as in the RUN-mode).

<u>Setting of a breakpoint</u>

A 82∅ + ──→ Program runs up to step 82∅, then

+ + ... step-by-step operation skipping the "criterial" point.

---

*) ACC = accumulator is used to indicate the result of the logic combination. If ACC = 1 (conditions of the logic combination fulfilled = 1), the following switching instructions are executed.

**) If the address of a timer or counter is preceded by a 3 (e.g. 326∅ for counter 26∅), the value of this register can be read or entered manually with E value + .

C 1.3 Detailed description of the operating modes

[R] RUN   <u>Normal program processing</u>
                  The PCA15 is automatically in the RUN-mode when switching on
                  if no programming unit is connected. For PCA14, the sliding
                  switch must be in position RUN.

[P] PROG   <u>Programming</u>
                  A program can be stored in a RAM memory (on the user plug-in
                  socket of the PCA1) or overwritten (corrected).

                            STEP   CODE   OPERAND
                        [A] xxxx   [E] xx   xxxx

                                  [E] xx   xxxx

                                [C] Deletes a wrongly entered line

                                [+] Terminates the input

                              [+][+] or [-][-] to display the program

[M] MAN   <u>Manual testing or setting of elements</u>
                  (Elements = inputs, outputs, flags, counters, timers)

                              STEP[1]   OPERAND

Testing: [A] xxx $\emptyset$/1 ——► display of the logic
                                            state

Setting: [A] xxx   [E] ——► [1] or [$\emptyset$]

---

1) STEP = Element address
   If the address of a timer or counter is preceded by a 3 (e.g. 326$\emptyset$ for
   counter 26$\emptyset$), the value of this register can be read or entered manually
   with:

   [A] 3xxx [E] value [+], [-], [A], [E]

   For an example see following page.

Continued from footnote 1)

Example: Input of values 23419 or 127 into counters 290 or 291.

Input:		Display:	STEP	CODE	OPERAND
A 3290			3290	ØY	YYYY

```
              Units
              Ten-thousands
              Always Ø
```

Input:		Display:	STEP	CODE	OPERAND
A	3290		3290	ØY	YYYY
E	23419		3290	Ø2	3419
E	127		3291	Ø1	ØØ27

Correction before storing

C			3291	ØØ	ØØØØ
Ø	127*		3291	ØØ*	Ø127
+					

* Values <1Ø'ØØØ have to be preceded by a leading Ø.

| S | STEP | | + | ───► Display showing where the program is.

Jump to the preselected step address of the user program

| A | 139 | + | ───► Program jumps to step 139

| + | | + | ... step-by-step processing of the program with the linkage result being checkable: ☀ ACCU = 1 [2)]

Switching to RUN is always possible.
In case of parallel program, only the activated parallel program is processed in the STEP mode.

| B | BREAK | Interruption of the program run and subsequent step-by-step operation

| + | ───► Display showing where the program is.

| + | | + | ... step-by-step execution of the program with the linkage result being checkable: ☀ ACCU = 1 [2)]

Switching to RUN is always possible.
In case of parallel programs, all programs are processed simultaneously (as in the RUN-mode).

Setting of a "breakpoint"

| A | 820 | + | ───► Program runs up to step 820 in slow RUN operation

| + | | + | ... step-by-step operation over the "critical" point

---

2) ACCU (= accumulator) is used to indicate the status of the logic combination.
When LED lights up, the ACCU = 1 (conditions of the logic combination fulfilled, linkage result = 1), and the following switching instructions are executed.

C 2 Further operating modes (only PCA14)

| M | "MAN"

Manual access to the software date-time

In case you use the date-time module E4Ø refer to chapter B 1, where reading
and writing to the hardware date-time will be explained (black box routines,
Software manual).

All programming units allow direct access to the software date-time (reading
and writing).

Contrary to the buffered hardware date-time (PCA1.E4Ø module), the software
date-time works only as long as voltage is applied to the PLC. Therefore, all
values have to be introduced each time the PCA14 is started up. The date-time
has a max. deviation factor of 3s/day.
The following table shows the signification and the numerical range for the
addresses 4ØØØ...4ØØ7.

Address	Meaning	Numerical range
4ØØØ	Week of the year	1...53
4ØØ1	Day of the week	1...7
4ØØ2	Year (1989=89)	Ø...99
4ØØ3	Month	1...12
4ØØ4	Day of the month (Feb = 28)*	1...31
4ØØ5	Hours	1...23
4ØØ6	Minutes	1...59
4ØØ7	Seconds	Ø...59

*) Contrary to the hardware date-time R27, the software date-time does not
   take the leap years into account (February = 28 days).

You may enter a maximum of 2 digits which appear in the operand (see examples
on the following page).

- Examples: Input for Thursday, June 2nd, 89, 1Øh 12min 45s

Input: Display: STEP   CODE   OPERAND

A	4ØØØ		4ØØØ	ØØ	ØØYY
E	22*		4ØØØ	ØØ	ØØ22
E	4*		4ØØ1	ØØ	ØØØ4
E	89		4ØØ2	ØØ	ØØ89
E	6*		4ØØ3	ØØ	ØØØ6
E	2*		4ØØ4	ØØ	ØØØ2
E	1Ø		4ØØ5	ØØ	ØØ1Ø
E	12		4ØØ6	ØØ	ØØ12
E	45		4ØØ7	ØØ	ØØ45
+					

After entering the seconds (4ØØ7), key + is depressed, provided that the input corresponds to the actual time. Do not press key E again, since otherwise the input of the calendar week is erased.

- Display:

Input: Display:

A	4ØØØ		4ØØØ	ØØ	ØØ22	22. week of the year
+			4ØØ1	ØØ	ØØØ4	Thursday
+			4ØØ2	ØØ	ØØ89	1989
+			4ØØ3	ØØ	ØØØ6	June
+			4ØØ4	ØØ	ØØØ2	2
+			4ØØ5	ØØ	ØØ1Ø	1Øh
+			4ØØ6	ØØ	ØØ12	12min
+			4ØØ7	ØØ	ØØ45	45s
						46s
						47s
						..
						..

* Calendar week and day of the week must correspond to month and date!

Setting of operating modes

A PLC can operate in various operating modes for preparing, testing and editing a program.

PCA14:   A sliding switch for selecting the operating modes is provided on the operating panel. This sliding switch is always active.

PCA15:   The operating modes are preselected outside the PLC. The keys of the programming unit PCA2.PØ5 are used, or in case a different programming aid such as the handheld computer P18 or an IBM-PC is used, to select the operating modes with the programming interface PCAØ.PØ1.

The selected operating mode remains active, when the connection between the programming units and the PCA15 is ended.

The following operating modes are automatically selected when the PCA15 is switched on.

- With programming unit PØ5 connected: STEP
(LED "STEP" lights up, the green LED "RUN" does not light up!)

- Without programming unit: RUN
(LED "RUN" lights up)

- With PØ1 connected: According to selector switch position