topline

Manual PCS 995

for the PCS 950/PCS 950c/ PCS 950e operating consoles

Intelligent soft-key actions ■
Logging • Statistics • Reporting ■
RecipeManager ■
Operating & Monitoring in Color ■

The operating consoles PCS*topline* offer the highest degree on perfection, unparalled in design and function. PCS*topline* keeps everything under control - from the PCS*mini* to the PCS*maxi*, with a superior operating culture and an unlimited setup freedom.

PCS, the first programmable operating console with a large selection of "ready-to-use" operating functions or operating tools which are simply selected via instructions. You can realize even the most unuasal operating requests at ease and in a minimum of time.

Today this way and tomorrow that way

One standard hardware for virtually thousands of different operating situations. Without extensive wiring and dozens of I/O points.

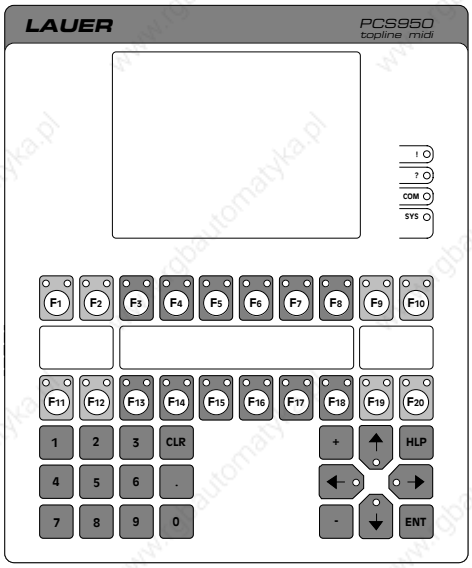
PCS for operating. What else?

reg 8035/0596 hs

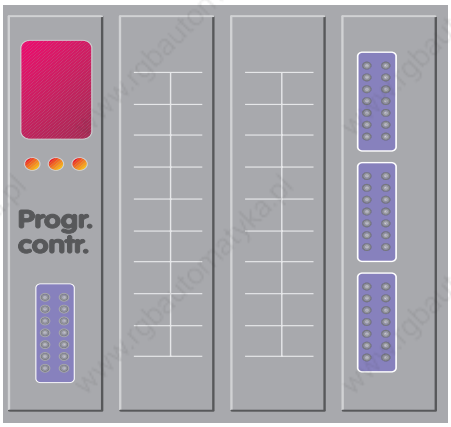
Print 06/05.96

- **Machine operation using 20 freely assignable keys;** these F01 to F20 labeled keys can be application specifically inscribed and are provided to the controller as status bits. In addition, situation-related soft-key actions can be assigned to these keys.
- **40 freely usable LEDs:** These can be assigned the indicating states »ON«, »DARK«, »FLASHING«, »INVERS FLASHING«. A green and a yellow LED is allocated to each function key.
- **Display of background bitmaps** separately for the areas status, working and soft-key.
- **Representation of any freely programmable characters** on the display.
- **Display of fixed texts with integrated variable values;** 9 variable formats are available for representation.
- **Setting up of several priority levels which can be changed related to situations;** This working-condition related management significantly offloads the programmable controller program.
- **Representation of the contents of a maximum of 214 programmable controller words as variables;** In addition, 55 internal (predefined) variables are available.
- **Modification of the contents of any word within the transfer area;** separate editors are available for every variable format.
- **Monitoring of rising or falling edges of a maximum of 1024 consecutive bits;** The assignment of texts, the management of 3 priority levels (information, warnings and faults) keeping the timely sequence as much as possible, organization of the FIRST MESSAGE, LAST MESSAGE, the individually settable clearing behavior are tasks which are managed by the PCS 950 by itself.
- **Logging of messages** with the CAME, WENT and ACKNOWLEDGED times is made by the PCS 950 itself. A logging memory is available for displayable (HISTORY) as well as for printable (MESSAGE PRINTER) texts.
- **Printing of shift-related or order-related pages** with any integrated internal or external variables.
- **Communication monitoring (wire-break, short circuit);** A very efficient data transfer is secured by the integrated priority management in connection with the intelligent package length optimization, the high thruput rate and the fault tolerance.
- **8 timers** with 8 daily repeating on-the-second ON/OFF switching points which can be freely edited as internal variables.
- **9 password levels** enable a differentiated access to operating texts and recipes. A 4-digit code number can be set-up for each password level in this way, your equipment can be protected against unauthorized operation.
- Using the **RecipeManager** 255 recipe blocks can be created with 255 forms each Recipe management is possible via PCS dialogs or under programmable controller control. Thus, the PCS relieves the programmable controller of product-related storage of equipment parameters

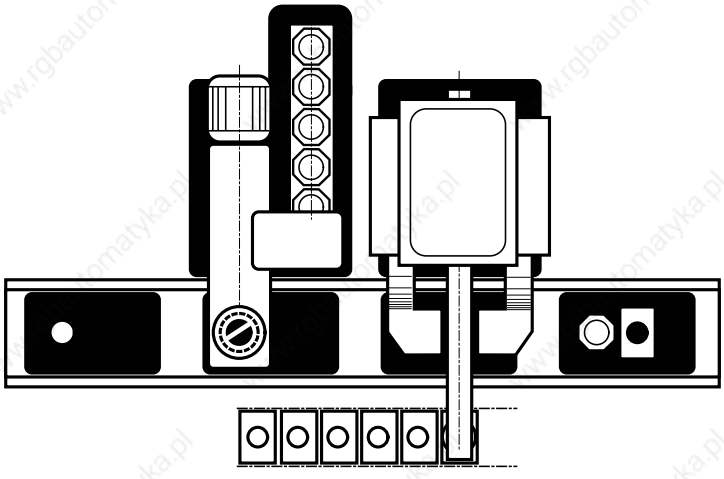
Operating & Observing

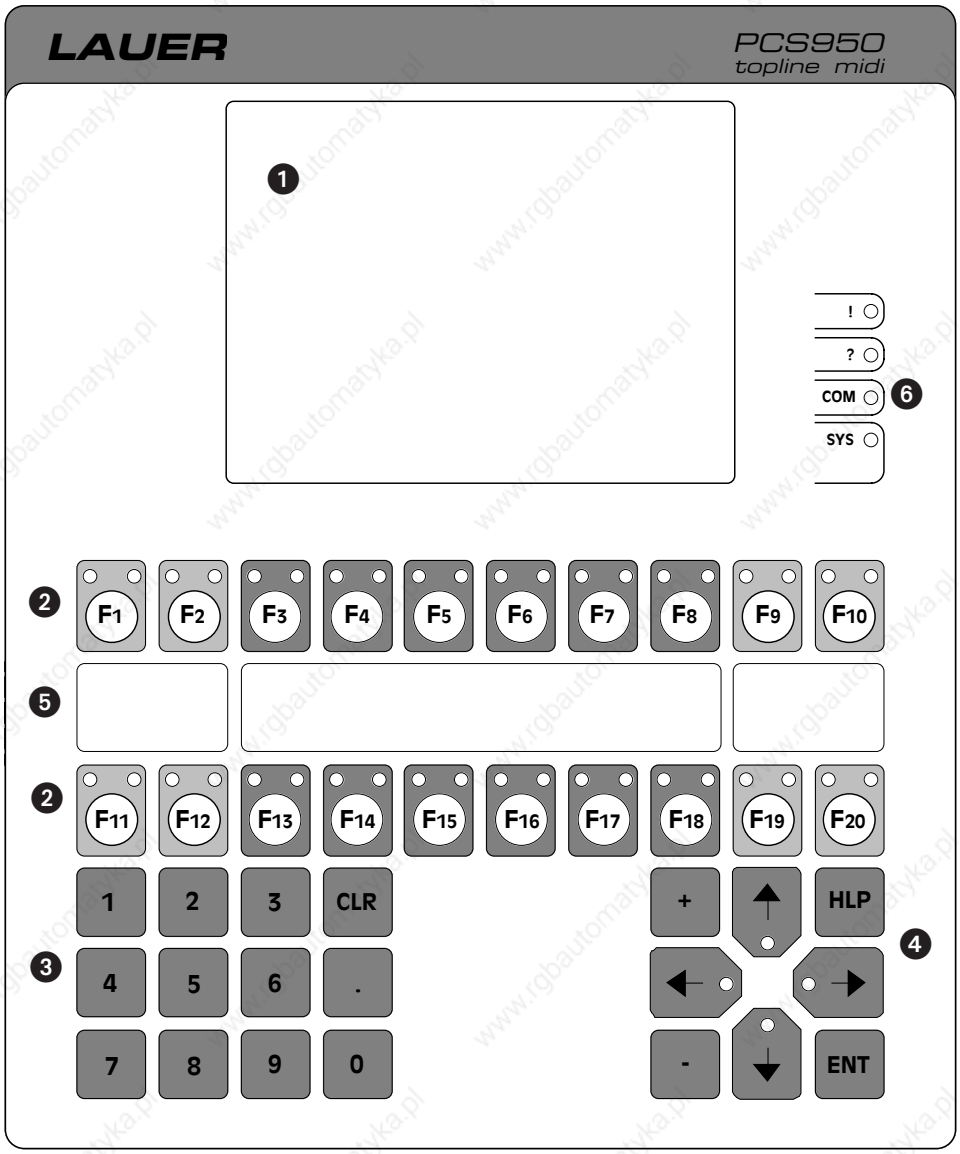


Controller



Machine





❶ Graphic LCD display, 320 x 240 pixels, 24 lines x 40 characters ❷ Function keys F1...F20, each containing two status LEDs ❸ Numeric keypad for preset values ❹ 8 control keys for menu operation and preset value input, cursor keys with LED ❺ Function key labeling 6 Important information about the PCS status

The modular operating console PCS 950 is equipped with a graphic LCD display and offers the maximum freedom of configuration.

The display enables any representation of information and variables by means of idle pages, operating pages, messages, help pages, status and softkey rows.

Via an internal bus, one module offering a wide range of functions can be plugged in at the rear. This is a prerequisite for an extremely flexible use of PCS*midi*.

PCS status (❻)

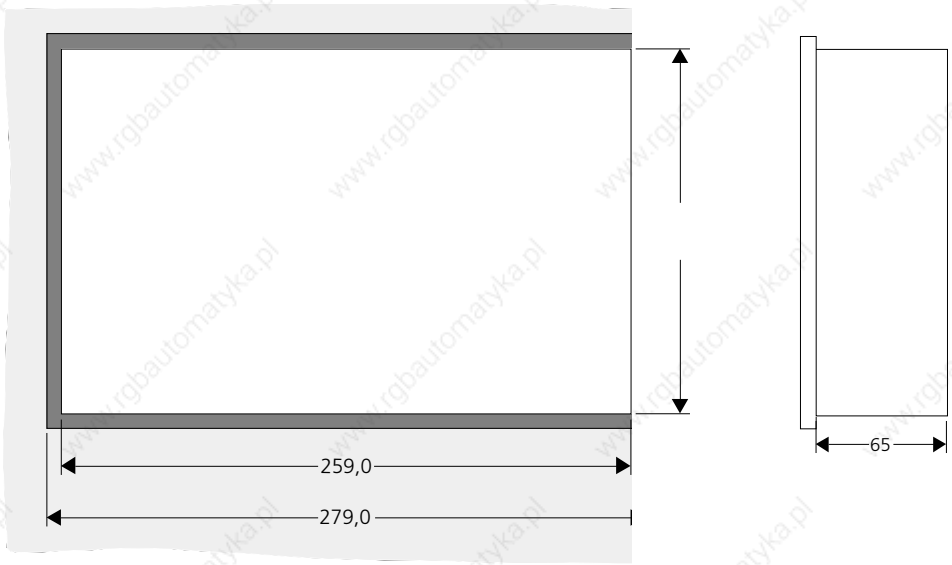
!	Message active
!	Priority > idle priority, currently inhibited
?	Operator prompt, input expected
COM	Communication not yet established
COM	Communication interrupted
SYS	Bios active

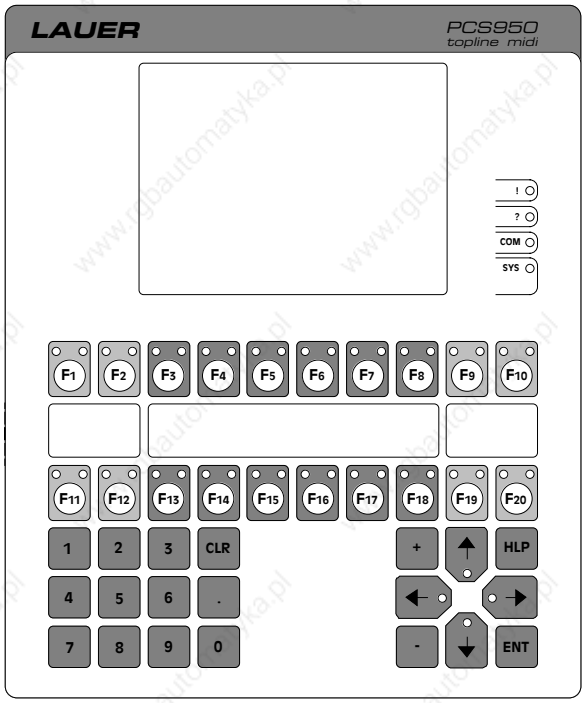
COM, **!** = LED continuously ON, **COM**, **!** = LED flashing

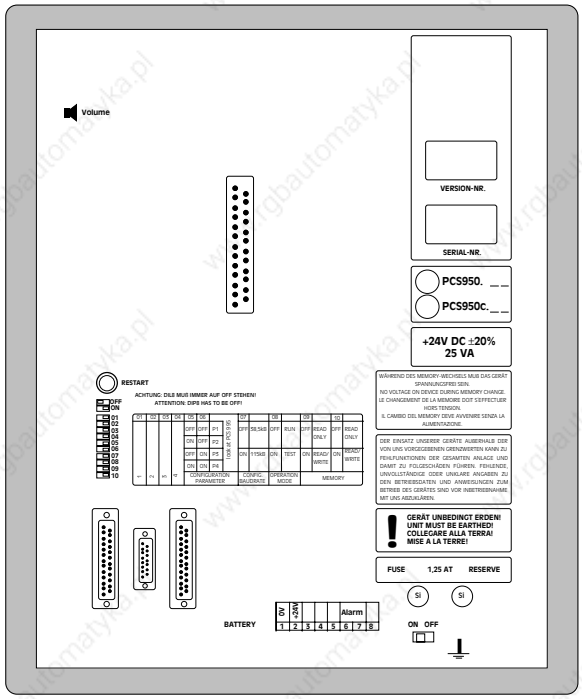
3 Specifications and dimensions

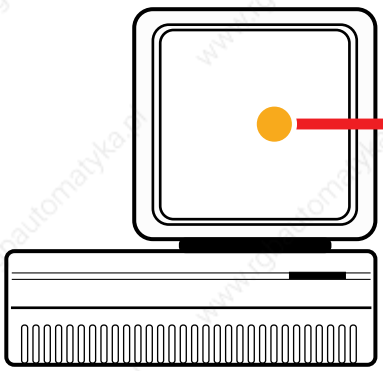
External dimensions	224 mm x 270 m, mounting depth without connector: 65 mm
Weight:	approx. 2000 g
Operating voltage:	+24 VDC ± 20%, protected against polarity reversal
Current consumption:	Iav @ 24 VDC 800 mA Imax @ 19 VDC (with cassettes max. 100 mA additionally) 1.0 A
Data storage:	flash EEPROM, min. 10000 write cycles
Noise immunity:	see manufacturer information
Protection class IEC 529:	rear: IP 20; front (after installation): IP 65
Humidity:	0...75%, exposure time of at least 48 hours
Vibration resistance:	3 g @ 50 and 100 Hz in all directions, min. 1 hour
Temperature:	storage: -25...+70 °C operation: 0...+50 °C
Front foil:	polyester
Pushbuttons:	mechanical with tactile touch
Display:	graphic LCD display with CFL background illumination, 24 lines x 40 characters, 5 x 8 matrix
Fuse:	1.25 A, small fuse, slow-blow, 1 spare fuse

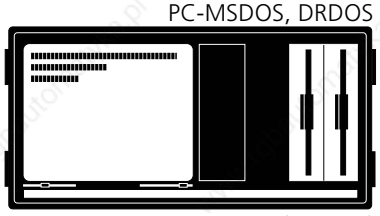
Warning!
The device is exclusively intended for being installed in another machine. Commissioning is prohibited, until conformity of the final product with the regulation 89/392/EWG has been ascertained.

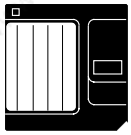










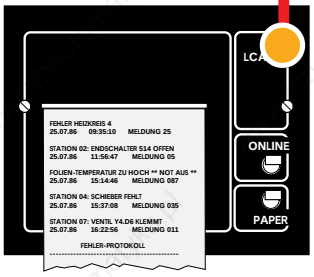


Programming with programming cable PCS 733

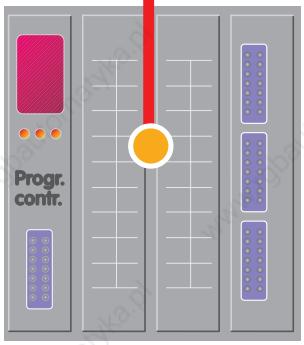
Simple configuration by means of the PCS*PRO*^{WIN} software

Communication with adapter cable PCS ...

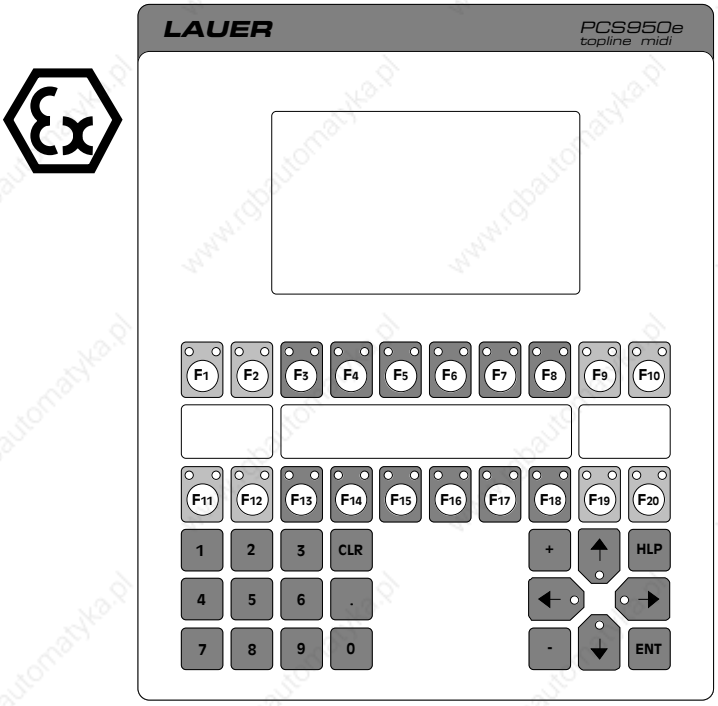
Printing with printer cable LCA 035/235



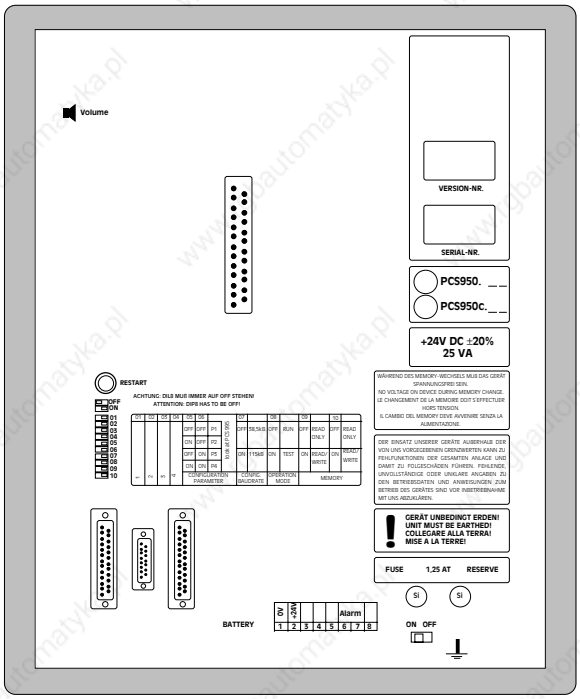
Industrial printer LCA 710



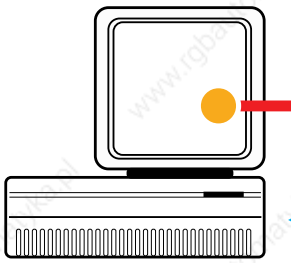
Ex area zone 1 and 2
Front end PCS 950e



Non-hazardous area
control unit ENT-DC-1.1-950

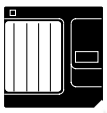


DATL-A data cable for
the connection of the
ENT-DC-1.1-950 control
unit



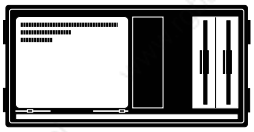
PC running MS-Windows

Programming
with the PCS 733
programming cable



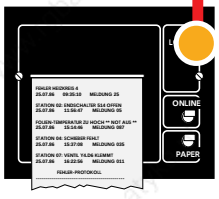
Configure simply via
PCS*PRO*^{WIN}

Communication
via PCS ...
adapter cable

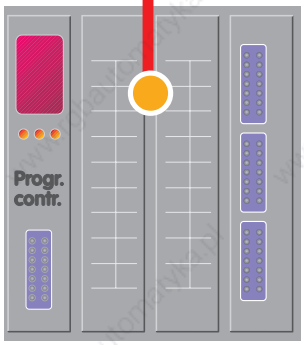


PG 7nn (MSDOS)

Printing
via the LCA 035/235
printer cable



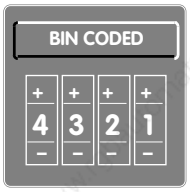
LCA 710 industrial printer



PCS*midi* represents a coherent operating concept for different programmable controller systems. The operating console PCS 950 offers a wide range of functions and tools for operating and observing.

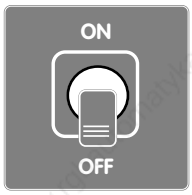
- *20 function keys (softkey functions also definable), each with two status LEDs (green/yellow - OFF, ON, FLASHING, INVERSE FLASHING)*
- *Any number of switches which can be labeled as desired (text or semi-graphics).*
- *Any number of selector switches which can be labeled as desired (text or semi-graphics), each with 256 switch positions.*
- *Key switch or code lock allow assignment of access rights.*
- *Date and time can be set from the PCS or (for synchronization) from the programmable controller.*
- *8 timers, each with 8 cams*
- *Numeric BCD/BIN preset value input via numeric keypad or IN/DEC keys. Up to 8 preset value variables per line.*
- *Simple input of ASCII preset values*
- *The bit pattern of a word (word variable) can be represented and modified in the PCS as desired*
- *Numeric display of binary actual values, optionally up to 5 digits (0..65.535) or 10 digits (0..4.294.967.295)*
- *Automatic conversion of the preset and actual values from BCD/BIN into the decimal format and vice versa with sign, limit values and scaling*
- *1024 message pages with text variables in 3 message priorities and with 5 delete modes*
- *256 pages with 8 variables per line are available for idle pages.*
- *Logging, machine report and output on the printer or the PC.*
- *127 operating pages*
- *RecipeManager - 255 recipe blocks with 255 recipe forms*
- *Analog input of preset values and analog display of actual values*
- *Up to 2 languages with different character sets can be configured (3 languages when using the additional cassette).*

Keys

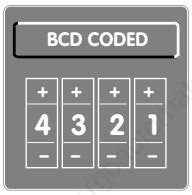


BINARY preset value input

Message texts



Switch



BCD preset value input

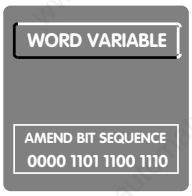
Operating and idle texts

Selector switch

ASCII preset value input

Help texts

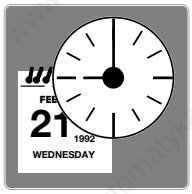
Code lock, key switch



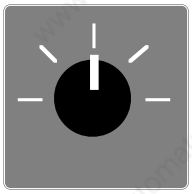
Change data word/flag

Logging, statistics, report


Graphics display



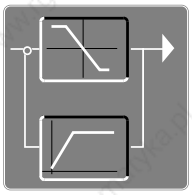
Date and time

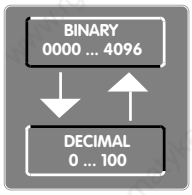

Analog preset value input

Preset value input via a menu

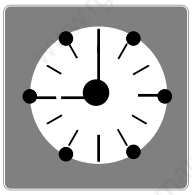
2 character sizes

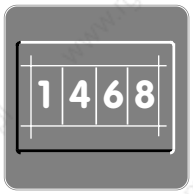
Several languages


Limit values, scaling



BIN/DEZ-conversion


Timer

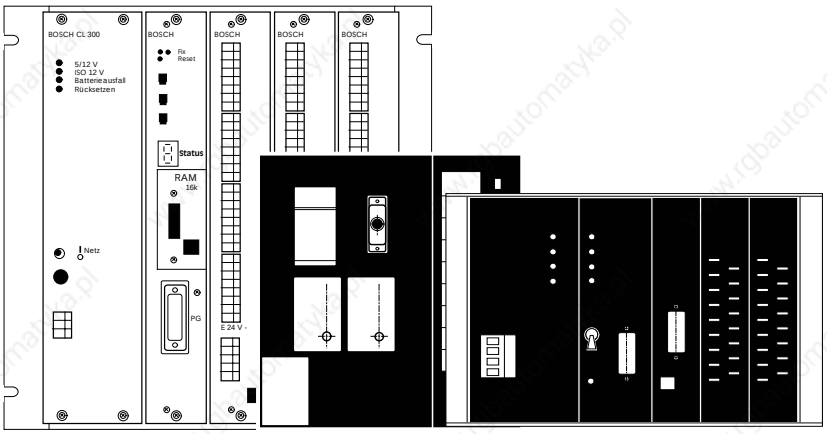


Numeric actual value

The electrical connection between a programmable controller of any type and the PCS is effected by a special adapter cable.

Data communication is based on a principle which can easily be understood:

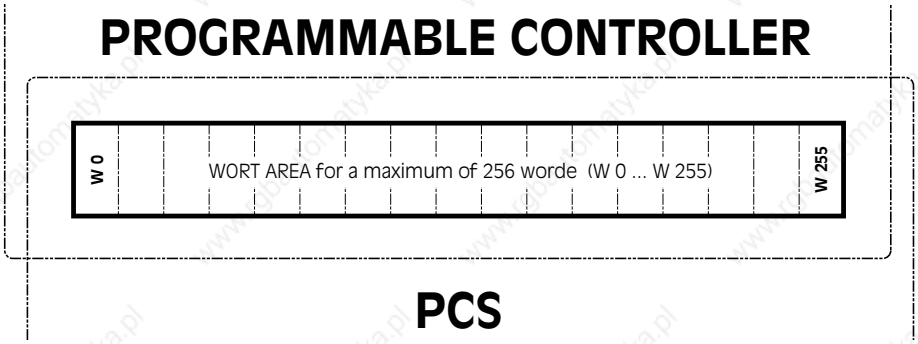
The PCS writes functions or preset values into previously defined programmable controller word areas. These functions or values are then read and interpreted by the programmable controller.

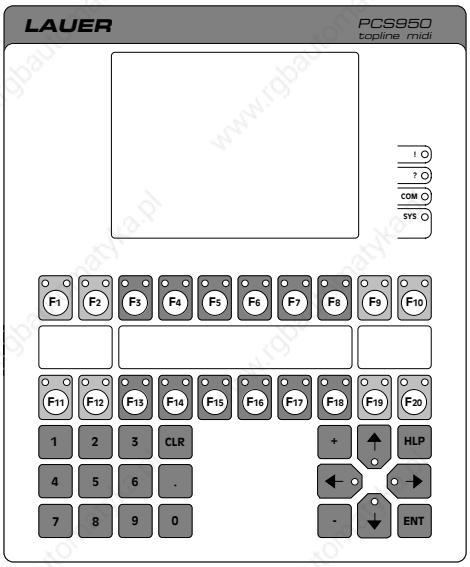


The programmable controller writes functions or actual values into previously defined word areas. These functions or values are then automatically read and interpreted by the PCS.

Depending on the programmable controller, a maximum of 256 words (with 16 bits each) or a total of 4096 I/O, are available for PCS/programmable controller communication.

The words from W 00 up to W 40 are permanently assigned in the PCS*midi* (see pages 8 to 15). Words 41 to 255 are available for any operating projects. The words can be assigned individually.





6 Data word and data bit assignment

Communication between PCS*midi* and a programmable controller of any type is effected by words (flags and others). A clearly defined task or function is assigned to each word W.

Data word	Function	PCS	programmable controller transmission direction
1. System area: W0..3:			
W0..2	Reserved for internal functions, not available to the user		
W3	Error word for communication (see driver manual PCS 91.xxx)		
2. Status area: PCS status (written into the programmable controller)			
KEYS:			
W4	Key bits F1...F8, F9...F10, Arrow Down, Arrow Up, Arrow Right, Arrow Left, -, +		▶▶▶▶▶
W5	F11...F20, CLR, ENTER, DIL 4-1, HELP, ·, 9..0, Reserve		▶▶▶▶▶
W6-W7	Reserved for additional keyboard (e.g. PCS 891)		▶▶▶▶▶
TIME AND DATE			
W9-12	Year, month, day, day of week, hour, minute, second		▶▶▶▶▶ ◀◀◀◀◀

W4

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	▲	▼	▶	◀	–	+

W5

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
F11	F12	F13	F14	F15	F16	F17	F18	F19	F20	CLR	ENT	DIL4	DIL3	DIL2	DIL1

W6

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
X	X	X	X	HLP	•	9	8	7	6	5	4	3	2	1	0

W7

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEYS - RESERVE								KEYS - RESERVE							

W8

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEYS - RESERVE								KEYS - RESERVE							

W9

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mi enium				Century				Decade				Year			

W10

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Month (tens)				Month (digits)				Day (tens)				Day (digits)			

W11

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Day of week (01...07)								Hour (00...23)							

W12

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Minute (00...59)								Seond (00...59)							

Data word	Function	PCS programmable controller transmission direction
PCS-STATUS		
W13-17	Acknowledgement bit, timer, (W14) number of old printer messages, (W15) number of new printer messages, (W16) priority status, (W17) priority number, text number on the display	▶▶▶▶▶
PRESET VALUE STATUS		
W18-19	Data word number, length, (W19) bit mask	▶▶▶▶▶

W13

15	14	13	12	11	10	9	8
Pr. Stop	Log Stop	Lifo	Hist. del.	Hist. arriv	Pr. bu. full	Hist. bu. full	Re-ser-ved

7	6	5	4	3	2	1	0
S7	S6	S5	S4	S3	S2	S1	S0

W14

15	14	13	12	11	10	9	8
Number of old printer messages (high byte)							

7	6	5	4	3	2	1	0
Number of old printer messages (low byte)							

W15

15	14	13	12	11	10	9	8
Number of new printer messages (high byte)							

7	6	5	4	3	2	1	0
Number of new printer messages (low byte)							

W16

15	14	13	12	11	10	9	8
		RESERVED					

7	6	5	4	3	2	1	0
Offl activ	Re-serve	Re-cipe-activ*)	Hist-ory	Fail-ures	War-nings	Infor-ma-tion	Menu

W17

15	14	13	12	11	10	9	8
8	4	2	1	X	X	512	256
Displayed priority				Text no. on displ. (high)			

7	6	5	4	3	2	1	0
128	64	32	16	8	4	2	1
Text number on display (low byte)							

W18

15	14	13	12	11	10	9	8
128	64	32	16	8	4	2	1
Data word preset value							

7	6	5	4	3	2	1	0
X	X	X	16	8	4	2	1
Preset value length byte							

W19

15	14	13	12	11	10	9	8
15	14	13	12	11	10	9	8
Bit mask high byte							

7	6	5	4	3	2	1	0
7	6	5	4	3	2	1	0
Bit mask low byte							

*) in preparation

Data word	Function	PCS	programmable controller transmission direction
3. Command area (read from the programmable controller)			
LED STATUS, DISPLAY And MEMORY MODE			
W20	LED driving, F1..F10, green		◀◀◀◀◀
W21	LED driving, F1..F10, flashing green		◀◀◀◀◀
W22, 23	LED driving, F1..F10, yellow, flashing yellow		◀◀◀◀◀
W24, 25	LED driving, F11..F10, green, flashing green		◀◀◀◀◀
W26, 27	LED driving, F11..F10, yellow, flashing yellow		◀◀◀◀◀
W28..33	Reserved for additional keyboard (e.g. PCS 891)		◀◀◀◀◀
W34	Bit map number for working area		◀◀◀◀◀
W35	Status page and softkey row number		◀◀◀◀◀

W20

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	2	3	4	5	6	7	8	9	10	Reserved					
LED driving, green															

W21

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	2	3	4	5	6	7	8	9	10	Reserved					
LED driving, flashing green															

W22

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	2	3	4	5	6	7	8	9	10	Reserved					
LED driving, yellow															

W23

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	2	3	4	5	6	7	8	9	10	Reserved					
LED driving, flashing yellow															

W24

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
11	12	13	LED driving, green			17	18	19	20	Reserved					

W25

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
11	12	13	14	15	16	17	18	19	20	Reserved					
LED driving, flashing green															

W26

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
11	12	13	14	15	16	17	18	19	20	Reserved					
LED driving, yellow															

W27

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
11	12	13	14	15	16	17	18	19	20	Reserved					
LED driving, flashing yellow															

W28

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LED - RESERVED								RESERVED							

W29

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LED - RESERVED								RESERVED							

W30

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LED - RESERVED								RESERVED							

W31

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LED - RESERVED								RESERVED							

W32

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LED - RESERVED								RESERVED							

W33

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LED-RESERVED								RESERVED							

W34

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LED - RESERVED								Activate bitmap	Bit map number for working area						

W35

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Status page number								Softkey row number							

Data word	Function	PCS programmable controller transmission direction
COMMAND WORDS		
W36	Enable priorities + disable transmission + RecipeManager *) + activation bits	◀◀◀◀◀
W37	Printer driving, disable specific LEDs + message block transmission	◀◀◀◀◀
W38	Idle text number + operating text number	◀◀◀◀◀
W39	Print job	◀◀◀◀◀ ▶▶▶▶▶
W40	Operating printer text number	◀◀◀◀◀ ▶▶▶▶▶

*) in preparation

W36

15	14	13	12	11	10	9	8
OFF LINE	Re-ser-ved	En-able Re-cipe *)	Hist-ory	S	W	H	M
				Enable priorities			

7	6	5	4	3	2	1	0
Dis-able Comm C/D/E	Dis-able date	Dis-able time	Dis-able LED F-keys	Enable alarm output	Oper. hours coun-ter	Hist-ory Start	Sync Time

W37

15	14	13	12	11	10	9	8
Pr. stop Mess.	Log. stop	Lifo	Hist-ory delet.	Disab. beep	Disab. mess. LED	Disab HLP LED	Disab. Menu LED

7	6	5	4	3	2	1	0
MB7	MB6	MB5	MB4	MB3	MB2	MB1	MB0
Enable message block transfer							

W38

15	14	13	12	11	10	9	8
Stat/ Flash	64	32	16	8	4	2	1
	Idle text number (0...127)						

7	6	5	4	3	2	1	0
Pre-set -P	64	32	16	8	4	2	1
	Operating text number (1...127)						

W39

15	14	13	12	11	10	9	8
Print job / Pointer adjustment (high byte)							

7	6	5	4	3	2	1	0
Print job / Pointer adjustment (low byte)							

W40

15	14	13	12	11	10	9	8
OPER. PRINTER TEXT NUMBER (Print form) HIGH BYTE							

7	6	5	4	3	2	1	0
OPER. PRINTER TEXT NUMBER (Print form) LOW BYTE							

*) in preparation

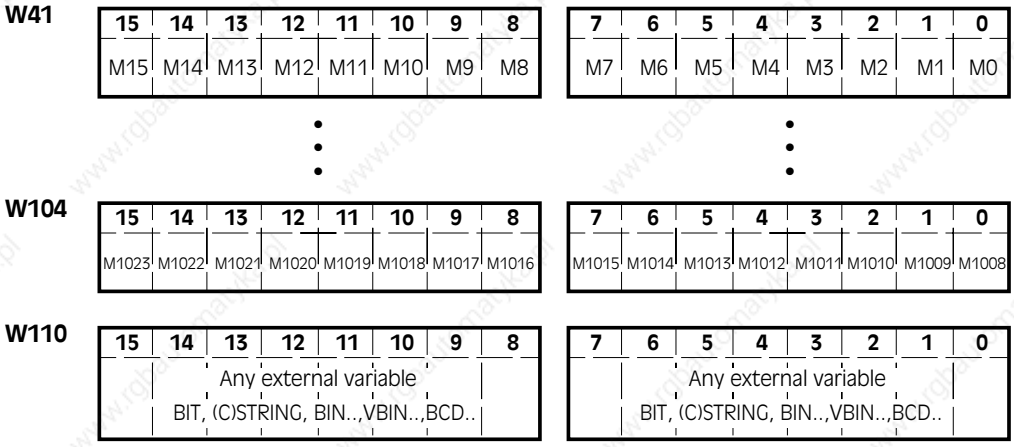
Data word	Function	PCS	programmable controller transmission direction
4. Message area: W41..110			
MESSAGE BLOCKS			
W41-48	Block 1		◀◀◀ ▶▶▶
W49-56	Block 2		◀◀◀ ▶▶▶
W57-64	Block 3		◀◀◀ ▶▶▶
W65-72	Block 4		◀◀◀ ▶▶▶
W73-80	Block 5		◀◀◀ ▶▶▶
W81-88	Block 6		◀◀◀ ▶▶▶
W89-96	Block 7		◀◀◀ ▶▶▶
W97-104	Block 8		◀◀◀ ▶▶▶

5. Expansion area: W105..W109

This area is reserved for possible extensions.

6. Variable area: W110..255

W110..255	Can be used for variables.		◀◀◀ ▶▶▶


W41
15 14 13 12 11 10 9 8
M15 M14 M13 M12 M11 M10 M9 M8
7 6 5 4 3 2 1 0
M7 M6 M5 M4 M3 M2 M1 M0
W104
15 14 13 12 11 10 9 8
M1023 M1022 M1021 M1020 M1019 M1018 M1017 M1016
7 6 5 4 3 2 1 0
M1015 M1014 M1013 M1012 M1011 M1010 M1009 M1008
W110
15 14 13 12 11 10 9 8
Any external variable
BIT, (C)STRING, BIN..,VBIN..,BCD..
7 6 5 4 3 2 1 0
Any external variable
BIT, (C)STRING, BIN..,VBIN..,BCD..

Internal variables

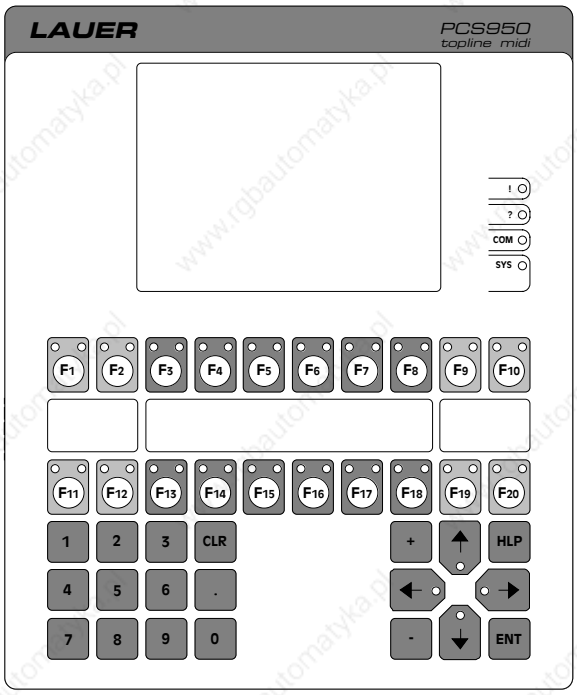
Firmware	*Prev. designat.*	*PCSPRO designat.*	*Type*	*Class*	*Length*			*Def. value*
[Z001]	ZP	[HINWEISE]	INT_BIN-2	ACTUAL	4			0
[Z002]	ZQ	[WARNUNGEN]	INT_BIN-2	ACTUAL	4			0
[Z003]	ZR	[STOERUNGEN]	INT_BIN-2	ACTUAL	4			0
[Z007]	ZX	[ERR_SCHITTST]	INT_BIN-2	ACTUAL	2			0
[Z008]	ZA	[TEXTNUMMER]	INT_BIN-2	ACTUAL	4			0
[Z009]	ZC	[ZEIT_MLD_KOMMT]	INT_ZEIT_MLD_KOMMT	ACTUAL	17			0
[Z010]	ZD	[ZEIT_MLD_GEHT]	INT_ZEIT_MLD_GEHT	ACTUAL	17			0
[Z011]	ZE	[ZEIT_MLD_QUITT]	INT_ZEIT_MLD_QUITT	ACTUAL	17			0
[Z012]	ZG	[UHR_STUNDEN]	INT_BIN0-2	PRESET	2			0
[Z013]	ZH	[UHR_MINUTEN]	INT_BIN0-2	PRESET	2			0
[Z014]	ZI	[UHR_SEKUNDEN]	INT_BIN0-2	PRESET	2			0
[Z015]	ZL	[DATUM_JAHR]	INT_BIN0-2	PRESET	2			0
[Z016]	ZK	[DATUM_MONAT]	INT_BIN0-2	PRESET	2			0
[Z017]	ZJ	[DATUM_TAG]	INT_BIN0-2	PRESET	2			0
[Z018]	ZN	[WOCHENTAG_IST]	INT_STRING	ACTUAL	x			0
[Z019]	ZO	[WOCHENTAG_SOLL]	INT_STRING	PRESET	x			0
[Z020]	ZY	[UHRZEIT]	INT_UHRZEIT	ACTUAL	8			0
[Z021]	ZZ	[DATUM]	INT_DATUM	ACTUAL	8			0
[Z022]		[ZEITSCHALTUHR]	INT_STRING	PRESET	16			0
[Z023]		[NOCKEN_NUMMER]	INT_BIN-2	PRESET	1			0
[Z027]		[BAUDRATE]	INT_STRING	PRESET	5			1
[Z028]		[PARITAET]	INT_STRING	PRESET	5			1
[Z029]		[DATENBIT]	INT_STRING	PRESET	1			0
[Z030]		[STOPBIT]	INT_STRING	PRESET	1			1
[Z031]		[RS232/TTY]	INT_STRING	PRESET	5			0
[Z032]		[HISTORYTEXTE]	INT_BIN-2	ACTUAL	4			0
[Z033]	x80	[MLDTXT_ZEILE1]	INT_MLDTXT_ZEILE	ACTUAL	40			0
[Z034]	x81	[MLDTXT_ZEILE2]	INT_MLDTXT_ZEILE	ACTUAL	40			0
[Z065]		[BETR_STD_IST]	INT_BIN-2	ACTUAL	10			0
[Z066]		[BETR_STD_SOLL]	INT_BIN-2	PRESET	10			0
[Z067]		[HISTORY_EINTR]	INT_BIN-2	ACTUAL	5			0
[Z068]		[DRUCKER_EINTR]	INT_BIN-2	ACTUAL	5			0
[Z069]		[ZSU_EIN_STUNDE]	INT_BIN0-2	PRESET	2			0
[Z070]		[ZSU_EIN_MINUTE]	INT_BIN0-2	PRESET	2			0
[Z071]		[ZSU_EIN_SEK]	INT_BIN0-2	PRESET	2			0
[Z072]		[ZSU_AUS_STUNDE]	INT_BIN0-2	PRESET	2			0
[Z073]		[ZSU_AUS_MINUTE]	INT_BIN0-2	PRESET	2			0
[Z074]		[ZSU_AUS_SEK]	INT_BIN0-2	PRESET	2			0
[Z075]		[DRUCKERTEXTE]	INT_BIN-2	ACTUAL	4			0
[Z076]		<TAB>	INT_DRUCKERSEQUENZ	ACTUAL	0	H09		
[Z077]		<ESC>	INT_DRUCKERSEQUENZ	ACTUAL	0	H1B		
[Z078]		<LF>	INT_DRUCKERSEQUENZ	ACTUAL	0	H0D	H0A	
[Z079]		<FF>	INT_DRUCKERSEQUENZ	ACTUAL	0	H0C		
[Z080]		<Fe+>	INT_DRUCKERSEQUENZ	ACTUAL	0	H1B	H45	
[Z081]		<Fe->	INT_DRUCKERSEQUENZ	ACTUAL	0	H1B	H46	
[Z082]		<Un+>	INT_DRUCKERSEQUENZ	ACTUAL	0	H1B	H2D	H31
[Z083]		<Un->	INT_DRUCKERSEQUENZ	ACTUAL	0	H1B	H2D	H30

External variables

The contents of the external or internal variables are stored in the PCS*midi* in words 41...255. The external variables are divided into the following variable formats:

1. **BIT and STRING variable**
2. **BCD and BIN variable**
3. **Word variable**
4. **ASCII variable**
5. **Timer variable**

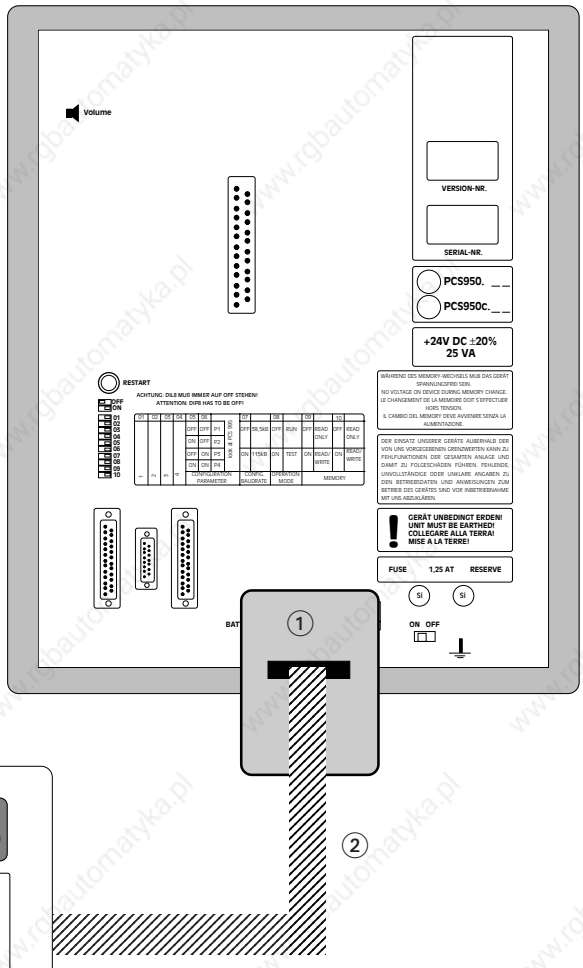
	Format	***Size***
1	*BIT variable*	*Max. length 80 characters*
1	*STRING variable*	*Max. length 80 characters*
1	*CSTRING variable*	*Max. length 80 characters*
2	*BCD-1 variable*	*Max. length 4 digits*
2	*BCD0-1 variable*	*Max. length 4 digits*
2	*BCD-2 variable*	*Max. length 8 digits*
2	*BCD0-2 variable*	*Max. length 8 digits*
2	*BIN-1, BIN-A variable*	*Max. length 16 bits/11 digits*
2	*BIN0-1, BIN0-A variable*	*Max. length 16 bits/11 digits*
2	*BIN-2, BIN-B variable*	*Max. length 32 bits/11 digits*
2	*BIN0-2, BIN0-B variable*	*Max. length 32 bits/11 digits*
2	*VBIN-1, VBIN-A variable*	*Max. length 16 bits/12 digits*
2	*VBIN0-1, VBIN0-A variable*	*Max. length 16 bits/12 digits*
2	*VBIN-2, VBIN-B variable*	*Max. length 32 bits/12 digits*
2	*VBIN0-2, VBIN0-B variable*	*Max. length 32 bits/12 digits*
3	*WORD variable (different respresentations* KM, KH, KY)	*KM: 17 digits* *KH: 4 digits* *KY: 7 digits*
4	*ASCII variable*	*Max. length 32 characters*
5	*Timer variable KT*	*Max. 40 characters*



The additional keyboard PCS 891 expands the function key area of the PCS operating console by adding 28 keys and 28 LEDs. Thus a total of 48 function keys and 38 LEDs are available. The 28 keys and 28 LEDs require 56 data bits which are assigned to words W7, 8 and 28 to 31 in the PCS 950. The function keys and LEDs of the PCS 891 correspond entirely to those of the operating console (except for softkey functionality).

To connect the additional keyboard PCS 891 to the PCS operating console, a cable and an adapter cassette are used. For PCS 891 installation, simply plug this cassette into the Memory Pack female connector.

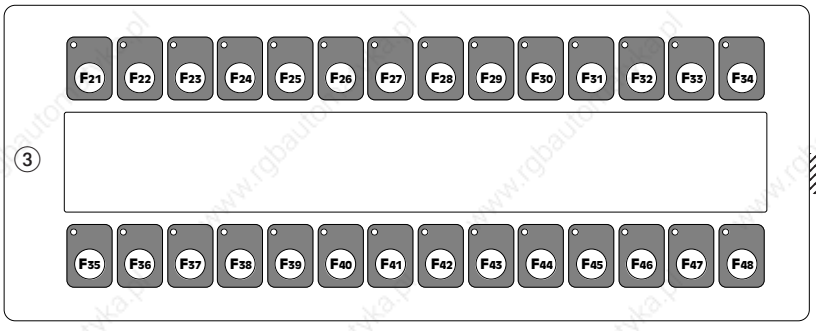
The adapter cassette of the PCS 891 also contains the EEPROM (memory capacity of 64 kBytes corresponds to the Memory Pack PCS 802).



① = Adapter and EEPROM cassette of the PCS 891

② = Cable (part of the PCS 891) for operating console connection, length: approx. 300mm

③ = Additional keyboard PCS 891



Operation of the additional keyboard requires:

- Memory Pack PCS 891
- Connection cable between Memory Pack and the additional keyboard
- Additional keyboard

Attention!
Connection of the additional keyboard is only allowed after removing power of the PCS 950. Removing the additional keyboard or switching ON or OFF the power supply voltage of the additional keyboard is not allowed during operation.

Assignment of the additional control elements (LEDs and keys)

The PCS 950 words mentioned above are valid. The additional keyboard requires the following words: W 7, 8 and 28, 29, 30, 31 (reserved for the additional keyboard).

LED { W 28 / W 29

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
L21	L22	L23	L24	L25	L26	L27	L28	L29	L30	L31	L32	L33	L34	—	—
L35	L36	L37	L38	L39	L40	L41	L42	L43	L44	L45	L46	L47	L48	—	—

LED-flashing { W 30 / W 31

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
L21	L22	L23	L24	L25	L26	L27	L28	L29	L30	L31	L32	L33	L34	—	—
L35	L36	L37	L38	L39	L40	L41	L42	L43	L44	L45	L46	L47	L48	—	—

F-keys { W 7 / W 8

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
F21	F22	F23	F24	F25	F26	F27	F28	F29	F30	F31	F32	F33	F34	—	—
F35	F36	F37	F38	F39	F40	F41	F42	F43	F44	F45	F46	F47	F48	—	Status

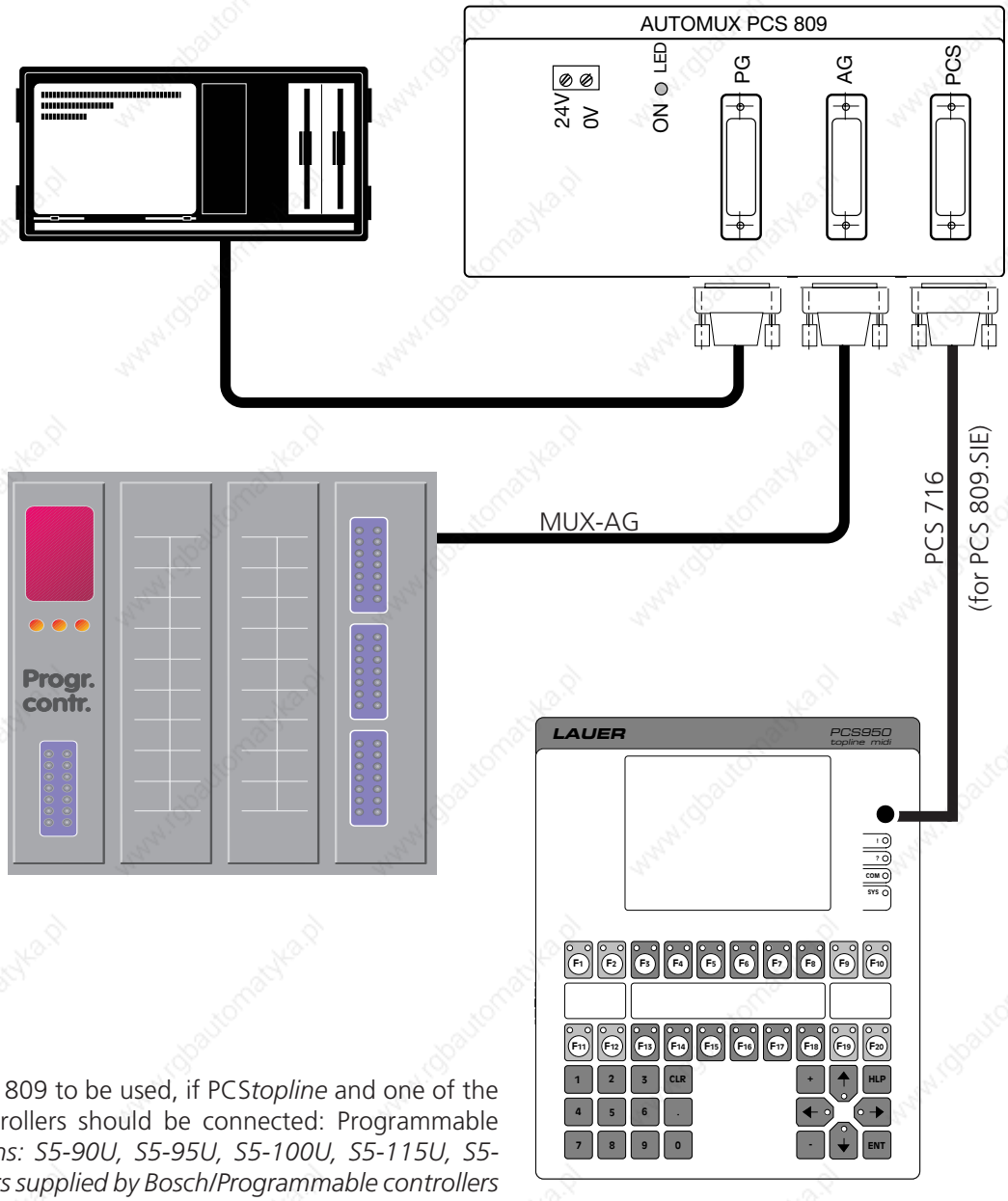
0 = keyboard not connected
1 = keyboard connected

If communication between the PCS and the Siemens S5 is established via the L1 standard protocol or the AS511 protocol, always one programmer interface is occupied.

Since small-sized programmable controller systems are equipped with only one programmer interface, limitations have to be considered during startup, i.e. the programmer and the PCS cannot be used simultaneously.

Automux PCS 809 is able to cope with these limitations. The PCS 809 expands the interface between the programmable controller and the programmer so that the controller can be operated simultaneously by the programmer and the PCS. Switching occurs automatically in the MUX.

The PCS 809 is designed as a startup tool. After commissioning, the PCS operator console is connected directly to the programmer interface of the programmable controller.



We recommend Automux PCS 809 to be used, if PCS*topline* and one of the following programmable controllers should be connected: Programmable controllers supplied by *Siemens: S5-90U, S5-95U, S5-100U, S5-115U, S5-135U/Programmable controllers supplied by Bosch/Programmable controllers supplied by Mitsubishi* and others. Automux PCS 809 is supplied with the adapter cable MUX /AG.

11 Quality and support

Quality is the most important factor in our company. From the electronic component to the manufactured device, quality is completely tested by qualified personal.

For this purpose, national and international test standards (ISO, TÜV, VDE, CE, Germanischer Lloyd) are applied. Each PCS is tested to 100% at different temperatures (5 ... 55°C) and test voltages (19 ... 33 VDC) and submitted to a permanent test under worst case conditions during 48 hours. This assures a maximum of quality!

Our products are not only characterized by a maximum economy and reliability, but also by a comprehensive and complete service.

- Qualified application support by qualified sales engineers.
- Our support is available to you every day by word and deed. Use our direct info line, if you have questions concerning the PCS*topline*

  Tel ***(+49)(0)7022 / 9660 220 + 221 + 222 + 223***
  Fax ***(+49)(0)7022 / 9660 224***
  Mailbox ***(+49)(0)7022 / 9660 225***

- Intensive and practice-orientated training for our products. Either in our training center or, after agreement, in your company.
- You do not only receive demo devices, but you are also supported during your first application by our specialists.
- Current information about our products by *„lauer aktuell"*
- Update service for our software

From advice to user support, from hotline to service, from manual to training - a comprehensive individual service is guaranteed.

PCS 950

TECHNICAL MANUAL

OPERATING CONSOLE PCS 950, PCS 950c, PCS 950e

PART 1: DESCRIPTION OF THE OPERATING CONSOLE

11.01.1995
Version 1.0

TABLE OF CONTENTS

1 GENERAL NOTES

1.1 STRUCTURE OF THE MANUALS

The first part of this manual describes the control elements, the connections and the basic I/O system (BIOS). The functionality currently available is explained in part 2. Since this functionality is only determined by PCS*PRO*^{WIN}, this description may be incomplete in some points. If necessary, consult the PCS*PRO*^{WIN} help system.

In any case, only the PCS*PRO*^{WIN} software can be used for data record creation. This software also combines the data record with the firmware, the driver and an additional functionality (optional). When configuring the transmission, you can select whether data record 1 or 2 is loaded into the internal Flash-EEPROM.

Warning!
Creation of a data record is only possible by means of the PCS*PRO*^{WIN} software. Other software packages are inadmissible and may cause malfunctions in the PCS and in the programmable controller.

All drivers are only used for interfacing the different programmable controller systems and are independent of the functionality given by the firmware. Information about the specific drivers and representation of the driving area within the programmable controller are described in the separate PCS 91.xxx manuals (for example PCS 91.SIE explaining interfacing the PCS with the programmable controllers of the Siemens company). These manuals describe all possibilities that exist to interface the PCS with the products of the corresponding manufacturer.

Warning!
Use only the drivers specified for the programmable controller. Other drivers may cause malfunctions in the PCS and in the programmable controller.

1.2 COMMISSIONING

The description of the commissioning procedure you are just reading refers to those facts which have to be observed when using the PCS 950. Commissioning of the programmable controller being used is described in the corresponding manuals of the programmable controller manufacturer.

The commissioning procedure is described below:

- Switch off the equipment or machine
- Set the DIL switches 1 ... 10 located on the rear of the device. The necessary instructions can be found in the next chapter and in the corresponding driver manual.
- Connect the supply voltage to the PCS 950. The supply voltage connections 1 (0V) and 2 (+24V) are screw terminals accepting wires up to 2 mm^2. The current consumption and the supply voltage limits are indicated in the section „Specifications“.

Warning!
The protective conductor and 0V of the supply voltage are separated in the device. The protective conductor is connected to the enclosure, to the noise filter and to the interface enclosures. The enclosure must be grounded to avoid noise in the best way. The grounding wire (4mm²) must be as short as possible. Additionally, 0V must be neutralized near the power supply (according to VDE regulations).

- Set the PCS 950 parameters using the parameterization software PCS*PRO*^{WIN} supplied by Systeme Lauer.

Warning!
Creation of a data record is only possible by means of the PCS*PRO*^{WIN} software. Other software packages are inadmissible and may cause malfunctions in the PCS and in the programmable controller.

Warning!
Use only the drivers specified for the programmable controller. Other drivers may cause malfunctions in the PCS and in the programmable controller.

- Set the PCS 950 parameters using the parameterization software PCS*PRO*^{WIN} supplied by Systeme Lauer.

Warning!
Malfunctions may occur in the PCS and in the programmable controller, if they are not correctly configured. Check the correct functioning of the PCS and the programmable controller.

- The functions of the LEDs of the PCS 950 are explained in section 3.2 and in the corresponding driver manual.

1.3 MEMORY MAPPING

The PCS 950 memory contains among others the following memory areas:

- **EPROM**

  The PCS 950 features a fixed EPROM area (BIOS) which only contains a boot strap program and the required display and keyboard handling programs. Corresponding messages are displayed, if invalid data is present in other parts of the memory.

- **EEPROM**

  A memory of 2 * 8 kByte for individual recipes is available here. The content of this memory is managed by the RecipeManager *) of the firmware.

- **FLASH-EEPROM**

  This memory area contains 256 kByte used for the firmware, 2 x 128 kByte for data records and 2 x 8 kByte for the driver. This memory area is completely electrically erasable. The content of this memory determines the entire functionality of the PCS 950.

- **ADDITIONAL CASSETTE**

  The additional cassette always contains a Flash-EEPROM. The capacity of this memory depends on the cassette type being used. Normally, this cassette contains another, alternative data record which can be activated by a menu contained in the BIOS. Additionally, this cassette enables firmware, data and drivers to be transferred. For this purpose, any internal memory area can be copied onto the cassette. Depending on the cassette capacity, several cassettes may be required for this purpose.

- **BATTERY-BACKED RAM**

  This internal memory (2 x 128 kByte) contains all non-volatile data. This area is only managed by the firmware.

*) in preparation

2 CONTROL ELEMENTS

2.1 DIL SWITCHES

10 DIL switches numbered from 1 to 10 are located at the rear:

DIL 1 to 4 = Programmable controller bits. These switches are freely abailable to the firmware

DIL 5, DIL 6 = Configuration parameter (driver)
e.g. baud rate, interface selection

OFF	OFF	Configuration 1
ON	OFF	Configuration 2
OFF	ON	Configuration 3
ON	ON	Configuration 4

For details see driver manual PCS 091.x

DIL 7 = Transmission baud rate with PCS*PRO*[WIN]
ON = 115,0 kBaud
OFF = 38,5 kBaud

DIL 8 = Operation Mode
ON = Stop, service program expected
OFF = Run, normal operation

Warning!
!! This switch must be set to OFF during operation, otherwise malfunctions may occur in the PCS and in the programmable controller !!

DIL 9, 10 = Write protection of the internal Flash-EEPROM
ON = EEPROM may be overwritten
OFF = EEPROM write-protected

Warning!
!! These switches must be set to OFF during operation, otherwise malfunctions may occur in the PCS and in the programmable controller !!

*) Switch no. 7 determines the baud rate of the transmission initiated by the PCS 950. PCS*PRO*[WIN] detects this baud rate automatically.

DIL switches 9 and 10 must be set to OFF after programming, otherwise data storage may not be guaranteed in all cases. Under normal conditions (including switching on and off at any time), - will not occur.

2.2 LED DISPLAYS

All LED displays are assigned 4 states: OFF, ON, FLASHING and INVERSE FLASHING. FLASHING consists of 75% light phase and 25% dark phase and INVERSE FLASHING consists of 75% dark phase und 25% light phase. The upper 4 LEDs indicate the operating states of the PCS. All LEDs, except SYS, are managed by the firmware.

ATTENTION	OPERATOR PROMPT	COMMUNICATION ERROR	OPERATING SYSTEM
!	?	COM	SYS

■ **SYS LED**

This LED lights as soon as the PCS 950 is working based with BIOS routines. This occurs, if there is no firmware, after starting the OFFLINE menu for data record switching, or when copying cassettes with <HELP+CLR>, or after starting the transmission with PCS*PRO*^{WIN}.

■ **OPERATOR PROMPT (?)**

See the functional description.

■ **ATTENTION (!)**

See the functional description.

■ **COMMUNICATION ERRORS (COM)**

ON: Communication has not been started after applying power.

FLASHING: Communication with the programmable controller has been interrupted!

For details, see the driver description.

2.3 RESET PUSHBUTTON AND ON/OFF SWITCH

The reset pushbutton is located above the 10 DIL switches. It initiates a software reset which is not required under normal conditions. The ON/OFF switch is required for cassette replacement.

Attention!
Before replacing the cassette, the PCS must be switched off!

2.4 KEYS

The PCS 950 is equipped with 20 function keys, 10 control keys and 10 numeric keys. The key functions are determined by the firmware.

3 CONNECTIONS

3.1 SUPPLY VOLTAGE

The supply voltage connections 1 (0V) and 2 (24V) are screw terminals accepting wires up to 2 mm². The current consumption and the supply voltage limits are indicated in the section »SPECIFICATIONS«.

Warning!
The protective conductor and 0V of the supply voltage are separated in teh device. The protective conductor is connected to the enclosure, to the noise filter and to pin 1 of the serial interfaces. The enclosure must be grounded to avoid noise in the best way. The grounding wire (4 mm²) must be as short as possible. Additionally, 0V must be neutralized near the power supply (according to VDE regulations).

The alarm output is designed as relay contact (NO) between pins 6 and 7. Only low voltage (24V) and a maximum current of 0.5A may be applied to this contact. Internal protection is realized by a PTC resistor which protects the contact in a limited manner.

Contact driving is determined by the firmware.

Warning!
The supply voltage as well as the input and output voltages and the voltages at the other interfaces must be functional extra-low voltages with protective isolation (transformer according to VDE 0551).

3.2 SERIAL INTERFACES

The PCS 950 is equipped with a „combined" COM interface and an RS232/TTY interface PRN. Either an RS 232 (V24) or, alternatively, a TTY (current-loop interface), active or passive, is available via the 25-pole JD female connector. An RS 422 or, alternatively, an RS 485 interface is available via the 15-pole JD connector. Please refer also to the descriptions in the PCS 91.x driver manuals.

- PRO/COM (RS232/TTY) and COM (RS485/RS422)

  The software is used to configure this interface as RS232, TTY, RS485, or RS422 interface. For a RS232/ TTY connection, the 25-pole SUB-D female connector and for a RS485 and RS422 the 15-pole SUB-D female connector is used. These two female connectors should NOT be used simultaneously. This interface is activated by the loaded driver and must be configured appropriately within PCSPROWIN. Normally, 4 configurations can be programmed. In this case, the DIL switches 5 and 6 are used for interface selection (see the driver manual PCS 91.x).

- PRO/PRN (RS232/TTY)

  This interface can be configured as RS232 or TTY interface. It is activated and parameterized by the firmware. For parameter settings, an internal variable is required. These values are stored in the non-volatile RAM.

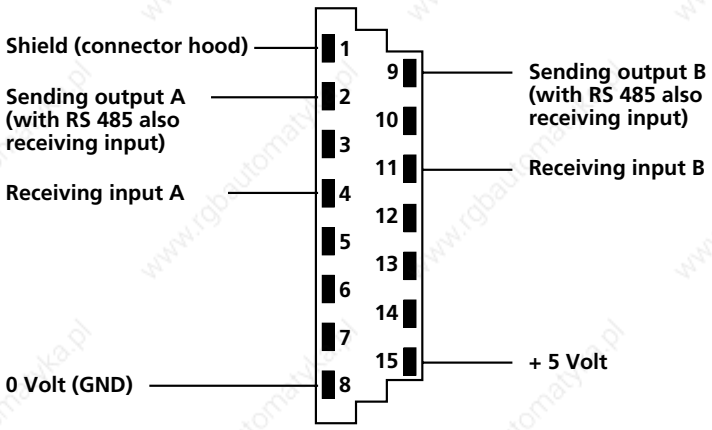
3.2.1 CONFIGURATION/PROGRAMMING

The RS 232 interfaces enable configuring the PCS 950 with a PC/programmer using the connection cable PCS 733 (configuration cable) and PCS*PRO*^{WIN}. Configuration start and programming are detected via the DSR input. The PCS is thus ready for program transmission. Please notice that for programing the DIL switches 9 and 10 must be set to ON so that the EEPROM is able to store data. Detection takes place on both interface sides, independently of the configuration previously set. A programmable controller simulation with PCS*PRO*^{WIN} is only possible on the PRO/COM interface.

Attention!
The level at DSR (Pin 6) is determined by the PC output DTR (25 -pole: Pin 20; 9-pole: Pin 4). Since the level of this pin is not defined after booting the PC/ programmer or after exiting a program, it is possible that the PCS is in configuration mode (only if the configuration cable PCS 733 is plugged in). In this case, the PCS program is stopped and the SYS LED is ON. If any communication with the programmable controller was established, it is now aborted. In this case, you must disconnect the PCS 733 cable. The PCS*PRO* software normally sets the correct level at this PIN.

3.2.2 COMMUNICATION ASSIGNMENT RS 485/RS 422

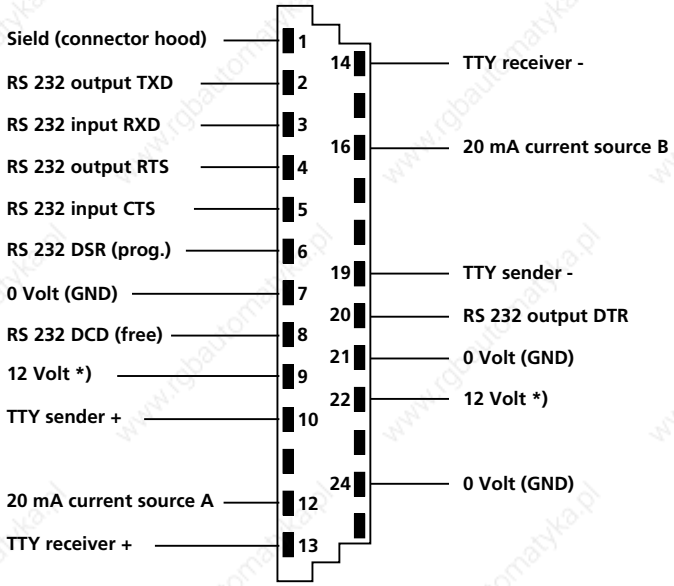
(rear view on the connector)



Warning!
Depending on your driver and the programmable controller you must use a special communication cable. Moreover, DIL switches 5 and 6 must be set according to the specified driver parameters. Otherwise malfunctions may occur in the PCS and the programmable controller. See the corresponding PCS91.x driver manual.

3.2.3 CONNECTOR ASSIGNMENT RS 232/TTY

(rear view on the female connector)



*) 12V max. 150mA

A total of 4 separated current loop sources are available for TTY (2 (A+B) per interface).

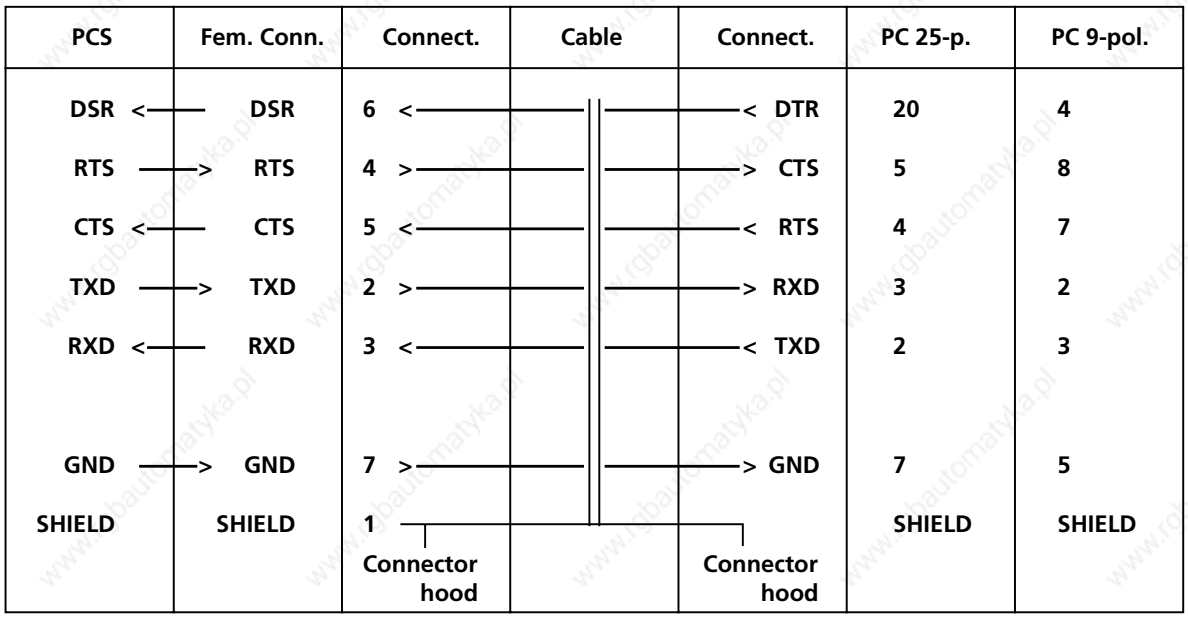
Warning!
If external current loop sources are used, the maximum e.m.f. may not exceed 15 V. Furthermore, real current sources with a maximum of 22 mA are required. Otherwise malfunctions may occur in the PCS and in the programmable controller!

3.3 CONFIGURATION CABLE PCS 733

The cable described below is required for configuration or data record transmission (driver, functions, variables, texts and menus).

You can also use this cable for programmable controller simulation on the PC.

Connection PC/Programmer - PCS 950, PCS 900, PCS 090, PCS 095 or PCS 9000:



3.4 CASSETTE CONNECTION

The 32-pole female connector allows the following cassettes to be used:

- PCS 802 Memory cassette with 64 kByte of Flash-EEPROM
- PCS 803 Memory cassette with 32 kByte of Flash-EEPROM (additionally 6 analog outputs and 6 digital inputs)
- PCS 804 Interbus-S connection with 64 kByte of Flash-EEPROM
- PCS 805 AEG bit bus cassette with 64 kByte of Flash-EEPROM
- PCS 806 Memory cassette with 128 kByte of Flash-EEPROM
- PCS 807 Profibus-DP connection with 128 kByte of Flash-EEPROM *)
- PCS 808 Arcnet connection with 128 kByte of Flash-EEPROM *)
- PCS 891 Memory cassette with 64 kByte of Flash-EEPROM and additional keyboard

*) in preparation

4 BIOS

4.1 DATA RECORD SELECTION

The non-volatile memory stores data used to specify, if the PCS 950 is operated with the internal data record 1 or 2 or with the external data record (on cassette). If this value is not specified in a clearly manner, a warning message is output before the firmware is started (this message has to be confirmed with <ENTER>) or you are prompted to select <1>, <2>, <3> or <4>. The system is then restarted and data are checked again.

4.2 SELECTION OF THE COPY FUNCTION

To select the copy function, press the <CLR> key while pressing and holding the <HELP> key. If the copy function is enabled, the keys are used in the following way:

<ENTER>
This is the confirmation key used to activate the selected function.

<+>, <-> and <.>
These keys are required to select the source or destination memory area.

When using the copy function, please observe the following points:

- When copying data onto cassette(s), ALL assigned blocks have to be copied. Since only deleted blocks can be overwritten by programmed data when reading data into the internal Flash-EEPROM , several cassettes may be required. If the data record exceeds 64 kByte, 4 PCS 802 cassettes are necessary for example.
- Currently, the individual blocks are stored on the following addresses:

CPU 1

Block 4, 6 and 7	firmware CPU 1
Block 5	driver 1 and 2
Block 8 and 9	external cassette slot
Block C and D	data record 1
Block E and F	data record 2

CPU 2

Block 4-7	firmware CPU 2
Block C and D	bitmap data record 1
Block E and F	bitmap data record 2

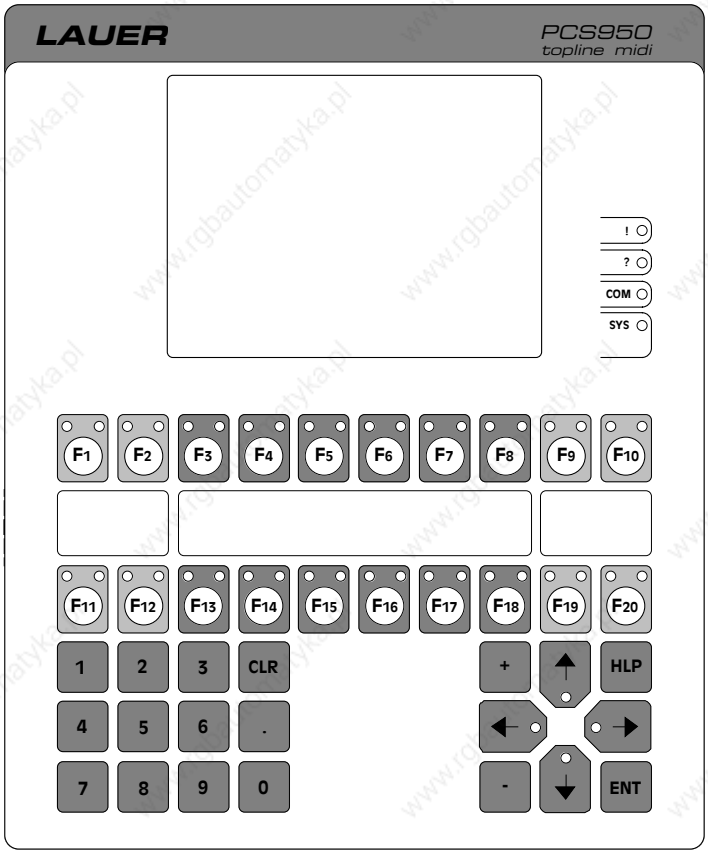
5 SPECIFICATIONS

5.1 SPECIFICATIONS OF THE PCS 950

Dimensions:	front panel cutout: (wxh) 204 +1 mm * 259 +1 mm Clearance in the control cabinet for mounting hardware: (wxh) 224 mm x 279 mm Mounting depth without connector: 65 mm, with SUB-D female connector and cable 110 mm External dimensions (wxh) 224 mm x 270 mm
Weight:	2000 g
Supply voltage:	+24VDC ± 20%, protected against polarity reversal
Current consumption:	Iav = 800 mA at 24V Imax = 1.0 mA at 19V (with cassettes max. 100 mA additionally)
Noise immunity:	see manufacturer information
Enclosure type:	according to IEC 529: rear: IP 20 front: IP 65
Humidity:	0..75%, continuous test of 48 hours
Vibration resistance:	3g @ 50Hz in all directions, min. 5 hours 3g @ 100Hz in all directions, min. 1 hour
Temperature:	storage: -25..+70°C operation: 0..+50°C
Data storage:	Flash-EEPROM, min. 10000 write cycles
Front foil:	polyester
Dimensions of the insert foil:	210 +0 -0.4 mm * 26,0 +0 -0.4 mm * 0.1 mm
Pushbuttons:	mechanical with tactile touch
Display:	LCD CFL display 320 x 240 pixels, 24 x 40 characters, 5 x 8 matrix, small character set 12 x 20 characters, 10 x 16 matrix, large character set
Fuse:	1.25 A, small fuse, slow-blow, 1 spare fuse

Warning!
The device is exclusively intended for being installed in another machine. Commissioning is prohibited, until conformity of the final product with the regulation 89/392/EWG has been ascertained.

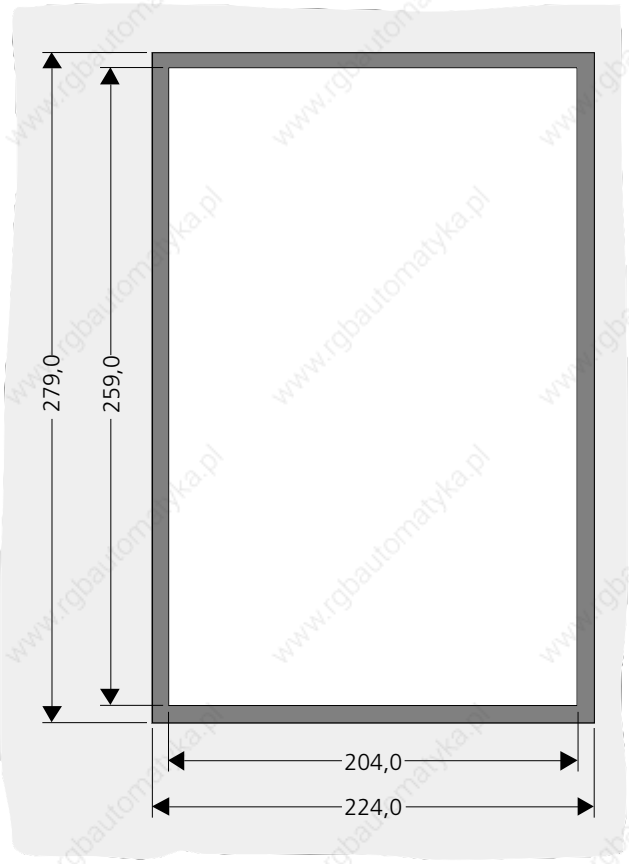
5.2 DIMENSIONS OF THE PCS 950



Front view

Mounting dimensions

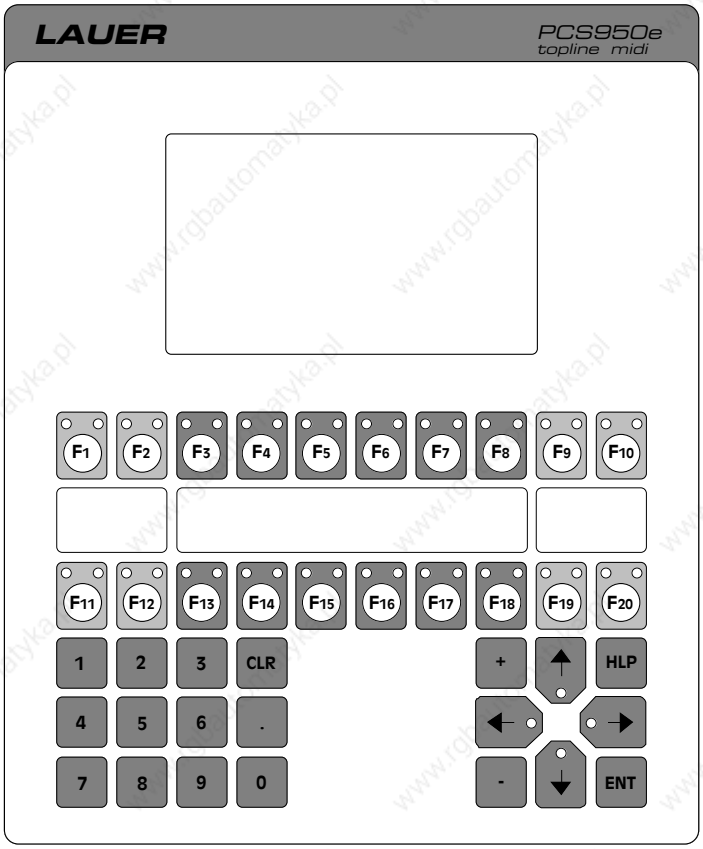
Front panel cutout:	204^{+1} mm x 259^{+1} mm
Clearance in the control cabinet for mounting hardware:	224 mm x 279 mm
External dimensions	224 x 270 mm
Mounting depth	65 mm



5.3 PCS 950e FRONT END SPECIFICATIONS

Dimensions:	Front panel cutout (w x h):	204 +1 mm x 259 +1 mm
	Clearance in the steel cabinet for mounting parts (w x h):	224 mm x 279 mm
	Installation depth w/o connector:	65 mm, with 8-pin round connector and 130 mm cable
	Outside dimensions (w x h):	224 mm x 270 mm
	Dimensions of the insertion foil:	210 + 0 -0.4 mm x 26.0 + 0 -0.4 mm x 0.1 mm
Weight:	2000 g	
Noise immunity:	refer to the manufacturer declaration	
Protection class:	according to IEC 529:	rear side: IP 20 front sidee: IP 65
Humidity:	0..75%, 48 hours continous test	
Vibrations resistance:	3g @ 50Hz in all directions, min. 5 hours 3g @i 100Hz in all directions, min. 1 hours	
Temperature:	storage: -20..+70 °C operation: 0..+50 °C	
Front foil:	polyester	
Keys:	mechanical with tactile touch	
Display:	LC display 240 x 128 pixels	
Character set:	16 x 40 characters, 6 x 8 matrix, small character set 8 x 20 characters, 12 x 16 matrix, large character set	

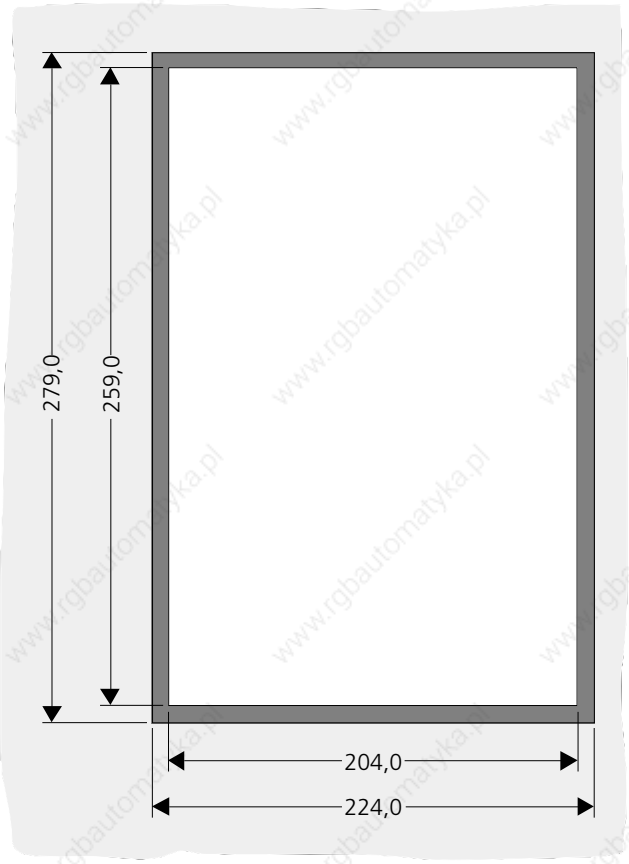
5.4 PCS 950e FRONT END DIMENSIONS



Front view

Mounting dimensions

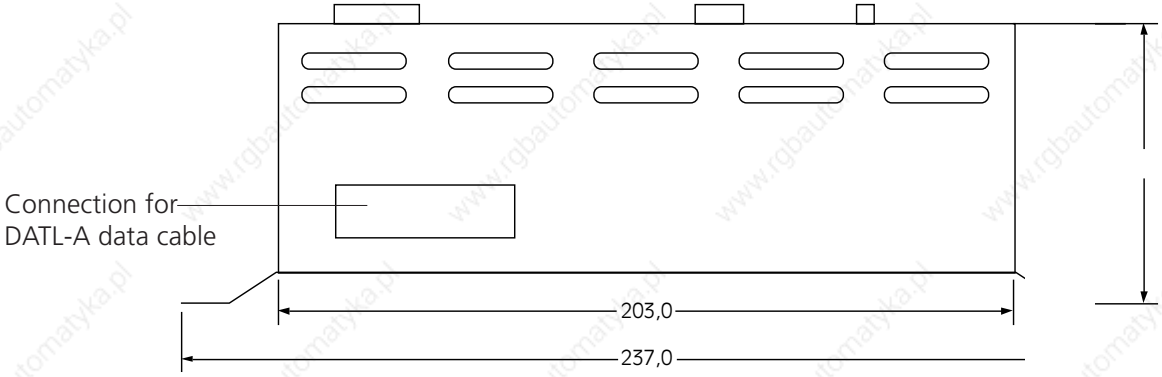
Front panel cutout:	204^{+1} mm x 259^{+1} mm
Clearance in the switching cabinet for mounting parts:	224 mm x 279 mm
Outside dimensions:	224 x 270 mm
Installations depth w/o connectors:	65 mm

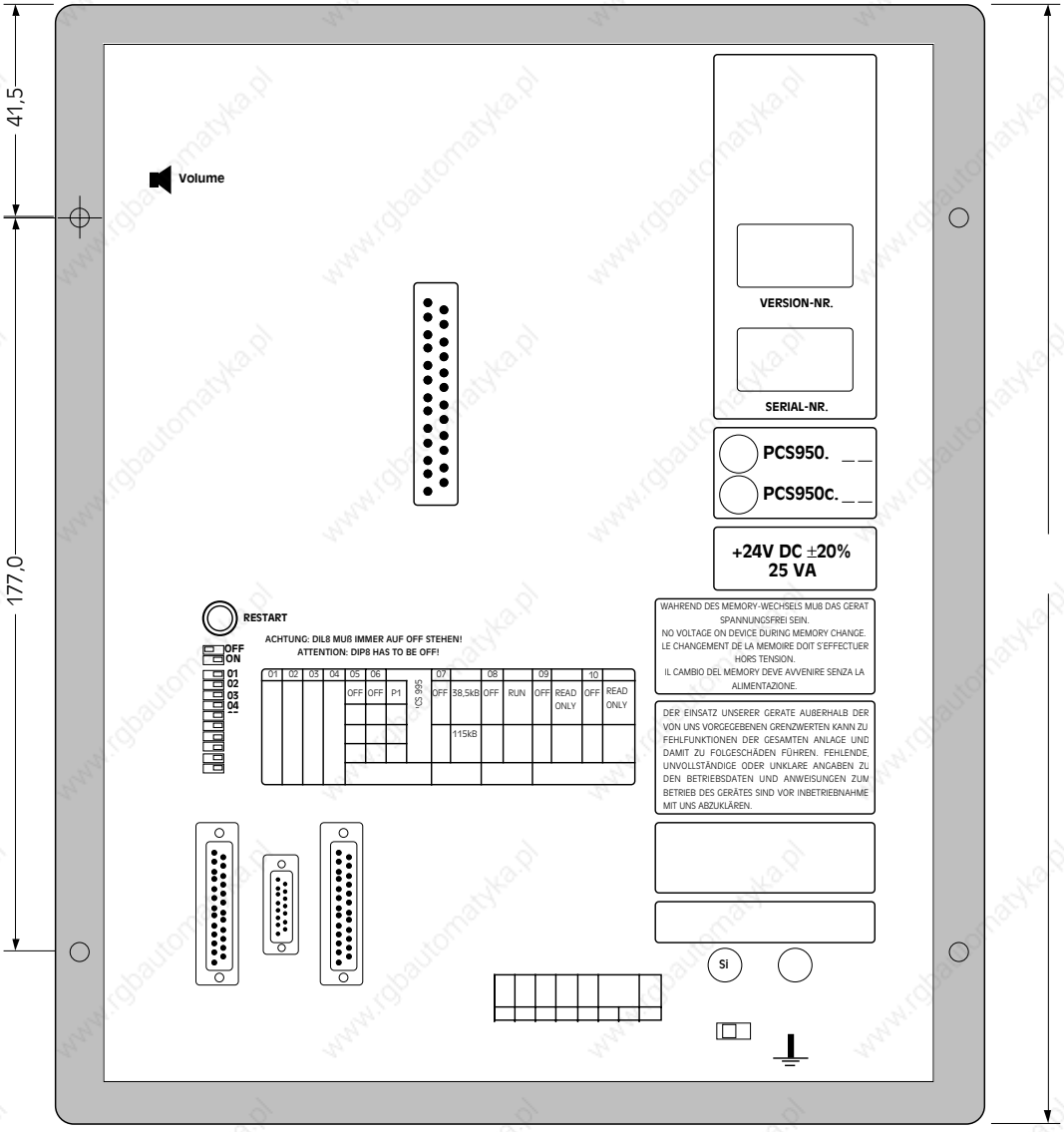


5.5 PCS 950e ENT-DC-1.1-950 CONTROL UNIT SPECIFICATIONS

Hazardous-duty type:	EEx ib IIC PTB no. EX-92.C.2046. x
Dimensions:	outside dimensions (w x h x d): 237 mm x 260 mm x 81 mm
Weight:	2500 g
Operating voltage:	+24VDC ± 20%, reverse voltage protected
Current consumption:	I = 1400 mA @ 24 Volt Imax = 1700 mA @19 Volt (max. 100 mA additionally with cassette)
Noise immunity:	refer to the manufacturer declaration
Protection class:	according to IEC 529: IP 20
Humidity:	0..75%, 48 hours continous test
Temperature:	storage: -20..+70 °C operation: 0..+50 °C
Data storage:	flash EEPROM, min. 10000 write cycles
Fusing:	controller: replacable from the outside, 1.25 A, miniature fuse, slow-blow type, 1 spare fuse Control unit: internal, on request

5.6 PCS 950e CONTROL UNIT DIMENSIONS





5.7 MAINTENANCE

Warning!
Static charge of the front panel is possible. Clean only with a moist cloth.

The front foil is made of polyester. So there is a risk of static charge. For this reason, the front panel may only be cleaned with a moist cloth.

This is especially important, when using the PCS 950 in an Ex area.

Note!
The back-up battery for the internal RAM should be replaced every 5 years. During replacement the device should be switched on to avoid loss of log memory and message printer data.

The PCS 950 does not require any regular maintenance, but the back-up battery for the internal RAM should be replaced every 5 years. This battery can be ordered at Systeme Lauer as spare part.

Order designation for the replacement battery: PCS 010

Warning!
The LCD display contains poisonous substances. Do not touch the display, if it is damaged.

5.8 USING THE PCS 950 IN AN EX AREA

Warning!
The devices can only be pre-setup by Systeme Lauer for use in Ex area 1 or 2. Depending on the application, the device must be installed according to VDE 0165 or VDE 170/171.

The PCS 950 can be pre-setup for use in an Ex area.

This must be specified when ordering the device. A subsequent release or certified declaration by the manufacturer is not possible. The devices can be pre-setup for use in Ex area 1 or 2.

An overpressure encapsulation with a low-pressure system is available. This means that a difference in atmospheric pressure of 2 - 4 mbar exists between the interior space and the outside of the front. Higher pressures may cause damages to the display.

The devices are only pre-setup for use in Ex area 1 or 2. This means that the devices must be installed according to VDE 0165 or VDE 170/171, depending on the application. For installation of the devices in encapsulated enclosures with pressure protection - including test certifications which may be required - Systeme Lauer informs the user about the cooperating companies on request.

For use of the devices in Ex area 2, please refer to the specifications of the manufacturer and an explanatory memorandum published by Systeme Lauer. The specifications of the manufacturer may be used as basic documentation for the certification of the device in Ex area 2.

5.9 APPLICATION OF THE PCS 950e IN THE EX AREA

System concept

Functionality, the PCS 950e is identically to the PCS 950/PCS 950c. The display is different with it´s LC display having 240 x 128 pixels (PCS 950: graphical LC display with 320 x 240 pixels, PCS 950c: color graphic LC display with 320 x 240 pixels). The system is made up by two individual components which are connected by an intrinsically safe data cable.

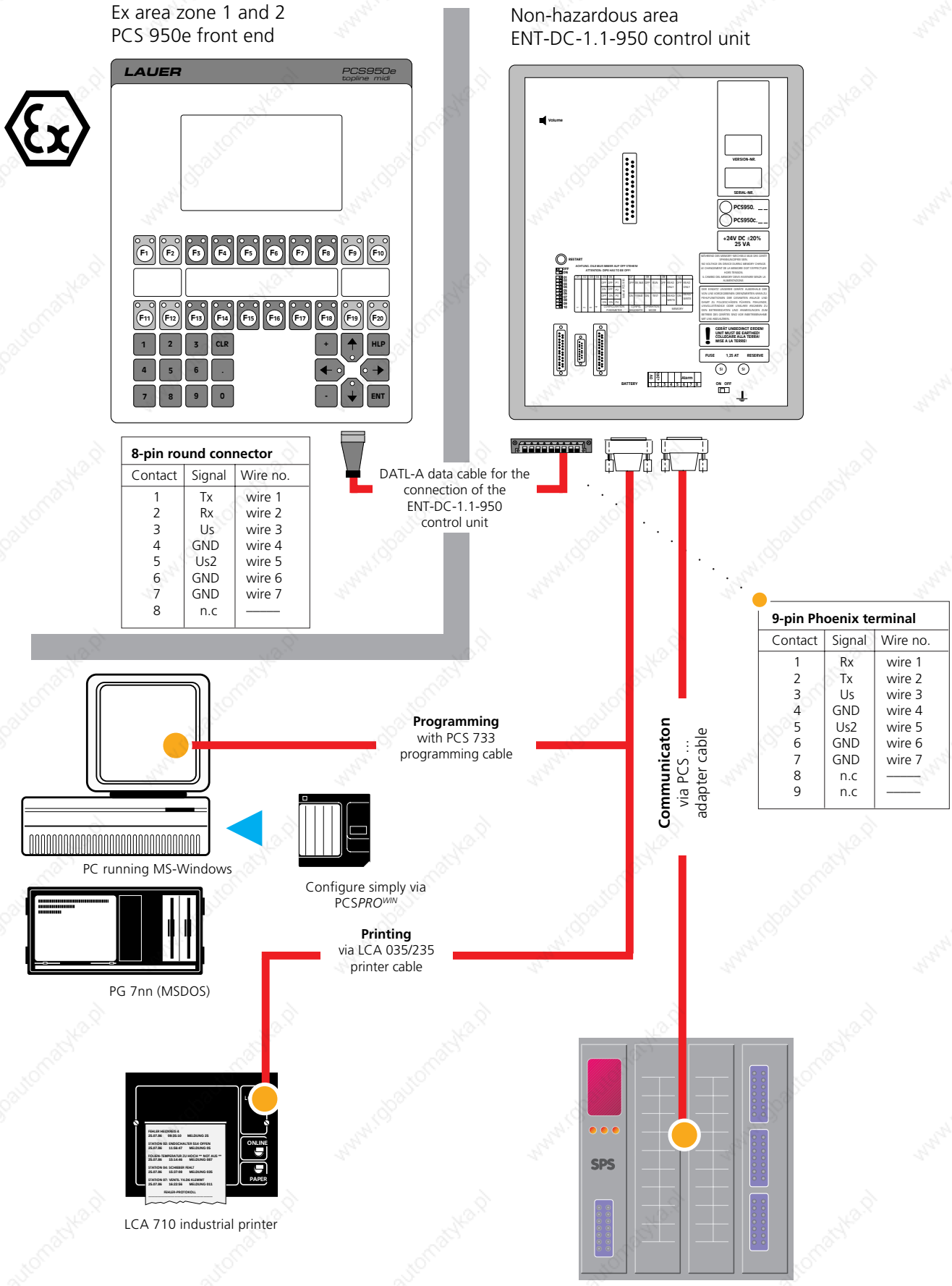
The intrinsically safe PCS 950e front end with protection class EEx ib IIC T4 for zone 1 and 2 is used in the hazardous area for display and operator guidance. The device size and device dimensions are identically to the PCS 950/PCS 950c. All software functions (variables, messages, etc.) are supported as for the standard device.

Using the intrinsically safe DATL-A7-4 data cable, the PCS 950e front end is connected to the ENT-DC-1.1-950 (power supply with controller) that is installed in the non-hazardous area. This data cable is used for the voltage supply and data transfer to the PCS 950e front end. No other cabling is required in the hazardous area.

A programmable controller is connected to the ENT-DC-1.1-950 that is also used for the programming of the system. Here, all interfacing options (interfaces, module slots) of the PCS 950/PCS 950c are available. All cables for connecting programmable controller systems (PCS 7xx) and cassettes (PCS 80x) can be used.
The commissioning instructions described in section 1.2 also apply to the ENT-DC-1.1-950.

The PCS*PRO*^{WIN} software is also used for the configuration of the system. All performance features of the PCS 950/PCS 950c are integrated into the PCS 950e and are fully supported by PCS*PRO*^{WIN}.

5.10 SYSTEM SETUP OF THE PCS 950e



5.11 PCS 950e DECLARATION OF CONFORMATY

6 INDEX

7 IMPORTANT USER INFORMATION

7.1 IDEOGRAMS AND SYMBOLS

The following symbols and ideograms are used in this manual:

Danger!
Directly imminent danger which causes death and most serious injuries.

Warning!
Possibly dangerous situation which can cause death and most serious injuries.

Caution!
Possibly dangerous situation which can cause light and less serious injuries.

Attention!
Possibly harmful situation which can cause damage to the product or its environment.

Mechanical pressure causes damage to the product.

Information concerning safety when using the devices in an Ex area.

Information and notes which must be additionally observed.

7.2 SAFETY RELATED INFORMATION

- The device may only be connected to the systems specified by Systeme Lauer.
- The device meets the current technical state of the art.
- Only trained and qualified persons who have familiarized themselves with the product are allowed to install and operate the device.
- The responsibility of persons operating the device must be clearly determined in order to avoid undefined competencies.
- The relevant safety regulations and standards must be observed.
- Opening the device is not allowed. Systeme Lauer is not responsible for resulting damages.
- Before commissioning the device, this instruction manual must be read thoroughly.
- Modifications of or changes to the design of the device are not allowed. Systeme Lauer is not responsible for resulting damages.
- The supply voltage of the device must be within the range specified in the section „Specifications". Systeme Lauer is not responsible for damages resulting from non-compliance to this requirement.
- The latest manuals and documentations are valid.

The specifications published by Systeme Lauer were determined with our methods and facilities; characteristics are only guaranteed in this respect. The user is responsible for testing and determining the suitability for the specific application or for use under actual conditions. Systeme Lauer does not assume any warranty for this.

Modifications reserved

PCS 950

TECHNICAL MANUAL

OPERATING CONSOLE PCS 950, PCS 950c, PCS 950e

PART 2: FUNCTIONALITY OF THE OPERATING CONSOLE

11.01.1995
Version 1.0

TABLE OF CONTENTS

1 FUNCTION

The actual functionality is determined by the firmware which PCS*PRO*^{WIN} transfers into the PCS 950, together with the corresponding data record and the specific driver for the programmable controller. The extension of the DOS file is .FRM. Since the functionality is extended in an upward compatible way, this manual may not represent the current state. The help system integrated in PCS*PRO*^{WIN}, however, always describes the latest version.

Warning!
Use only the PCS*PRO*^{WIN} software to create projects. Other software packages may cause malfunctions in the PCS and in the programmable controller !

1.1 APPLICATION RANGE

The operating console PCS 950 allows the user to perform the following tasks in a simple way:

- **1. Machine operation via 20 freely assignable pushbuttons:** These keys - numbered from F01 to F20 - can be freely labeled and are available in the controller as status bits. Furthermore, each key can be assigned softkey actions which depend on the specific situation.
- **2. 40 freely assignable LEDs:** The following states can be assigned to these LEDs: »ON«, »OFF«, »FLASHING« and »INVERSE FLASHING«. Each function key is equipped with a green and a yellow LED.
- **3. Representation of background bitmaps** which are separated for the status area, working area and softkey area.
- **4. Representation of freely programmable characters** on the display
- **5. Representation of a fixed text combined with variable values:** 9 variable formats are available for this purpose.
- **6. Assignment of several priority levels which can be changed according to the situation:** This practical management reduces the programmable controller program efficiently.
- **7. Representation of the cnontents of up to 214 programmable controller words as variables:** Additionally, 81 internal (pre-defined) variables are available.
- **8. Modification of the contents of any words within the transfer area:** The system contains individual editors for each variable format.
- **9. Monitoring of up to 1024 consecutive buts to detect rising and falling edges:** The following tasks are automatically performed by the PCS 950: Assignment to texts, management of three priority levels (information messages, warning messages and failure messages), nearly complete observance of timing requirements, organization of FIRST MESSAGE, LAST MESSAGE and setting of specific deletion procedures.
- **10. Logging of messages** (CAME, GONE and ACKNOWLEDGED) is automatically performed by the PCS 950. A log buffer is not only available for display texts (HISTORY) but also for printable texts (MESSAGE PRINTER).
- **11. Printing of shift or job related pages** with any external and internal variables.
- **12. Communication monitoring (BROKEN WIRE, SHORT CIRCUIT):** The integrated priority management combined with an intelligent packet length optimization, the high data throughput rate, and fault tolerance of the logs enable an extremely efficient data transmission.
- **13. Eight timers** with 8 ON/OFF switching instants repeated every day and with seconds precision, which can be freely edited as internal variable.

1.2 CONTROL ELEMENTS AND DISPLAYS

1.2.1 LED DISPLAYS

All LED displays are assigned 4 states: OFF, ON, FLASHING and INVERSE FLASHING. FLASHING consists of a 75% light phase and a 25% dark phase, whereas INVERSE FLASHING consists of a 75% dark phase and a 25% light phase.

The function keys are equipped with 20 green and 20 yellow LEDs which can be freely activated by the programmable controller. They are controlled by the LED STATUS words W20 - W27.

The upper 4 LEDs indicate the operating states of the PCS. All LEDS, except SYS, are controlled by the firmware.

FAILURE	OPERATOR PROMPT	COMMUNICATION ERROR	OPERATING SYSTEM
!	?	COM	SYS

■ **OPERATOR PROMPT (?)**

ON: The PCS is waiting for key activation (acknowledgement or deletion of messages, preset value input, exit of an operating page).

FLASHING: If a message is displayed to with deletion procedure 4 has been assigned, this LED flashes as far as the corresponding message bit is set to logical 1 (the message cannot be deleted). If the message bit is set to 0, the LED is permanently ON and the message can be acknowledged with <CLR>. If the <HLP> key has been pressed and a help text has been programmed for the currently activated priority, this LED and the (!) LED are flashing alternately.

■ **OPERATING PAGES, INFORMATION, WARNING MESSAGE, FAILURE (!)**

ON: An INFORMATION MESSAGE, WARNING MESSAGE or a FAILURE is displayed.

FLASHING: An OPERATING PAGE, a WARNING MESSAGE, an INFORMATION MESSAGE or a FAILURE has been activated, but is (currently) not displayed due to an enabled priority locking in command word A (W36; bit 8..11). If the <HLP> key has been pressed and a help text has been programmed for the currently activated priority, this LED (!) and the (?) LED are flashing alternately.

■ **COMMUNICATION ERROR (COM)**

ON: Communication has not been started since powering up.

FLASHING: Communication with the programmable controller has been interrupted!

If communication is established, this LED is OFF. If communication is aborted (after being started) the audible error message is enabled for a short time and this LED flashes.

Warning!
Check reaction/action of the programmable controller.
After programmable controller restart following a communication failure, the desired reaction/action of the programmable controller has to be checked.

■ **ARROW KEY LEDs DURING DISPLAY OF OPERATING PAGES**

In this mode, the (!) LED is OFF or FLASHING. The arrow key LEDs can be locked via bit 8 of command word B (W37).

ON: This <arrow> key enables access to other preset values which can be edited.

■ **ARROW KEY LEDs DURING DISPLAY OF MESSAGES AND HISTORY TEXTS**

In this mode, the (!) LED is constantly ON. The arrow key LEDs can be locked via bit 10 of command word B (W37). If the

<Arrow Up> LED is ON:

The message located above can be activated.

<Arrow Down> LED is ON:

The message located below can be activated.

<Arrow Left> LED is ON:

The topmost message can be activated.

<Arrow Right> LED is ON:

The bottommost message can be activated.

■ **ARROW KEY LEDs DURING DISPLAY OF HELP TEXTS**

In this mode, the (!) LED and the (?) LED are flashing alternately. The arrow key LEDs can be locked via bit 9 of the command word B (W37). If the

<Arrow Up> LED is ON: The main page of this help text can be activated.

<Arrow Down> LED is ON: The following pages of this help text can be displayed.

1.2.2 CHARACTER REPRESENTATION ON THE DISPLAY

During operation, 24 lines with 40 characters each (small character set) are available without limitations concerning the status area, working area and softkey area (no automatic display). 2 character sets (small/large) are selectable. Each character set includes the characters H10 to HFF. Please observe the following assignment:

■ **H00 to H0F (0 to 15)**

These characters are internally required and cannot be used in texts.

■ **H10 to H7F (16 to 127)**

These characters correspond to IBM code page 437 (Western Europe) and are permanently assigned to the BIOS. They can be used in any way.

■ **H80 to HFF (128 to 255)**

These characters can be modified as desired within PCS*PRO*^{WIN} and can be used in texts in any way. The most common code pages are supplied with PCS*PRO*^{WIN}:

850	Romanic character set
865	Norway
860	Portugal
852	Hungary
866	Cyrillic (Eastern Europe)

The 7-bit character sets of the PCS 950 (Cyrillic, Serbo-Croatian and Katakana) are additionally available in the area between 128 and 255 of PCS*PRO*^{WIN}.

Flashing of individual characters (-> preset value input) is controlled by the PCS itself. If the idle text priority is enabled, operating texts can be represented as completely flashing texts by setting bit 15 of command word C (W38) to logical 1.

1.2.3 COLOR DISPLAY (PCS 950c only)

With the PCS 950c, texts and variables can be displayed in colors. For the representation in color, attributes can be assigned to texts and variables using PCS*PRO*^{WIN}. These attributes reference one of 32 possible color pallet entries. The entry determines the foreground and background color of the representation.

Black, red, green, yellow, blue, cyan, magenta and white can be selected for the foreground and background color.

The standard pallet features the following entries:

Pallet entry	Foreground	Background	Pallet entry	Foreground	Background
0	black	white	16	white	black
1	red	white	17	white	red
2	green	white	18	white	green
3	blue	white	19	white	blue
4	magenta	white	20	white	magenta
5	black	red	21	red	black
6	yellow	red	22	red	yellow
7	black	green	23	green	black
8	blue	green	24	green	blue
9	red	green	25	green	red
10	yellow	blue	26	blue	yellow
11	red	blue	27	blue	red
12	black	blue	28	blue	black
13	cyan	black	29	black	cyan
14	magenta	black	30	black	magenta
15	yellow	black	31	black	yellow

1.2.4 KEYS

The following key types are used: function keys, numeric keys and control keys. All keys are also available in the programmable controller as key bits. As long as a key is pressed, the corresponding bit of the word area is set to logical 1. »Pressing« a key generates a short audible signal called keyboard click. Due to their „REPEAT" functions, some keys generate repeated audible signals.

The function keys <F1> to <F20> are transferred into the programmable controller. Depending on the specific situation, softkey actions can be defined for each function key.

Depending on the displayed priority, the numeric keys and the control keys are used for internal PCS functions. Accessing these keys in the programmable controller is therefore limited.

- **IDLE PRIORITY (0)**
  If this priority is enabled, only the <HLP> key is used for internal functions.
- **OPERATING PRIORITY (1)**
  In this priority, the numeric keys <0..9> and the control keys <+>, <->, <.>, <Arrows>, <CLR>, <ENTER> and <HLP> are used for internal functions.

- **MESSAGE PRIORITIES (2, 3, 4)**

  Depending on PCS programming (deletion procedure, number of message text lines, message help text) internal functions are assigned to the <Arrow> keys and to the <CLR> and <HLP> keys.

- **HISTORY (5)**

  Internal functions are assigned to the <HLP> and to the <Arrow> keys.

- **OFFLINE MENU (8)**

  The <ENTER> key is used for a special function. All other keys (except the F keys) allow you to exit the OFFLINE menu.

- **HELP KEY**

  This key is used in all priorities for HELP text display. If these texts contain several lines, use the <Arrow Up/ Down> keys to page through the text.

Important!
Pressing inadmissible keys generates an audible error message, if the IDLE PRIORITY is not enabled. If this is not desired, bit 11 of word 37 (log.1) can be used for suppression!

1.2.4.1 KEY EVALUATION IN THE PROGRAMMABLE CONTROLLER

To evaluate the control and numeric keys in the programmable controller, please observe the following aspects:

- Please note that if the current priority available in word 17 is evaluated, transmission of this PCS status requires more time than transmitting keys. For this reason, a consistent transmission of the PCS status and the keys cannot be guaranteed. Normally, data are consistent after two transmission cycles, but this cannot be assured.

Warning!
Data consistency is not guaranteed. The user has to assure consistency by creating an appropriate programmable controller program. Otherwise malfunctions may occur!!

A consistency is only guaranteed, if all priorities are locked (set bits 8-15 of word 36 to zero) and if bits 12-15 of word 17 are set to 0. If a help text has been defined for the idle text, the <HELP> and the <Arrow Up/ Down> keys can be used to trigger internal functions.

1.2.5 AUDIBLE SIGNAL

3 audible signals are generated:

- a short keyboard click when »pressing« a key
- a „REPEAT" click which is generated when you press and hold a key to which a „REPEAT" function has been assigned
- an audible error message (duration of 0.5 seconds), if an inadmissible key is pressed

The volume of the audible signal can be set by a potentiometer located at the rear of the PCS 950.

If the audible signal is not desired, you can suppress it by setting bit 11 of word 37 to logical 1.

2 PARAMETER SETTING IN THE PCS 950

2.1 OVERVIEW

The PCS 950 is operating simultaneously on a maximum of 8 priority levels, but only one priority can be represented at a time. This priority controls character representation on the display and scanning of the keyboard. The different priorities can be enabled or disabled, independently of the representation. Activation of priorities is marked in the PCS status word 16 by setting the corresponding bit(s) (0-7) to logical 1. This enables the programmable controller program to react correspondingly. Enabling and disabling of priorities is explained in a separate section. Normally, the highest activated priority is displayed, unless it is locked (suppressed) by the programmable controller program (word 36, bits 8-15). If a priority has been suppressed, the next less significant priority is checked. The last significant priority can never be suppressed and is always active. The following levels are currently assigned:

- IDLE PRIORITY (0)
- OPERATING PRIORITY (1)
- INFORMATION PRIORITY (2)
- WARNING PRIORITY (3)
- FAILURE PRIORITY (4)
- HISTORY PRIORITY (5)
- RECIPE MANAGER *) (6)
- ABORT OF COMMUNICATION (7)
- OFFLINE PRIORITY (8)

*) in preparation

The currently represented priority is reported to the programmable controller as a numerical value in bits 12-15 of word 17.

Warning!
Check the functioning of the PCS after parameter setting. All functions for which parameters have been set must be checked. Otherwise malfunctions may occur in the programmable controller!!

2.1.1 VARIABLES

Variables are place holders used in an text. A maximum of 8 variables can be defined per text line. These variables are normally stored in the programmable controller starting with word 110. If the entire message bits are not required, the free area may also be used for variables (see also message bits...). According to the type, variables are represented in the PCS as texts or numbers. The following variable types are used: ACTUAL VALUES, PRESET VALUES and PRESET-P VALUES.

ACTUAL VALUES are variables that cannot be modified by the PCS. The corresponding values can only be displayed.

PRESET and PRESET-P VALUES are variables that are not only displayed but can also be modified by the PCS. The modified, stored value can be found on the word address specified during variable definition. The suffix (-P) is used as a key function. This allows for example to limit preset value access to certain operator groups. If an operator access to the PRESET-P variables is disabled (bit 7 of word 38 set to logical 0), these variables are represented in the same way as ACTUAL values and cannot be modified.

2.1.2 TEXTS

The following text types are used: PRINTABLE TEXTS und DISPLAY TEXTS. Idle texts, operating texts, message texts, history texts, status texts and help texts can be displayed, whereas the message printer texts and operating printer texts are printable.

- Idle texts are used with priority 0.
- Operating texts are used with priority 1.
- Depending on the created program, message texts are used in information texts (priority 2), warning texts (priority 3) or failure texts (priority 4). Their length may not exceed 2 lines (small character set) or one line (large character set).
- History texts are stored in the history buffer (priority 5) and may only contain internal variables for the text number, time COME, GONE and ACKNOWLEDGED, identification number, current date/time, counter for operating hours and the history text number (current value).
- Message printer texts are stored in the printer message buffer and are assigned to the message texts.
- Operating printer texts are selected by numbers and are not stored but immediately transferred to the printer.
- If defined, help texts are displayed for the priorities 0 - 5 when pressing the <HLP> key. Releasing the key clears the text from the display. A help text consists of up to 32 lines of text.
- If defined, status texts are visualized in the upper display area, independently of the priority.

2.1.2.1 STATUS PAGE

A status page (0..21 lines) can be permanently visualized in the upper area of the display. The variables and texts in this area are actual values that can only be displayed and not be edited. The number of the status page to be displayed is specified in bit 8..15 of word W35. A background bitmap can be assigned to each status page and is displayed in this area. A status page can only be displayed, if it has been previously defined in PCS*PRO*^{WIN}.

2.1.2.2 SOFTKEY ROW

Up to 256 softkey rows can be defined in the PCS 950. Within each softkey row, an internal or external softkey function can be assigned to each function key. The additional texts for the softkeys are displayed in the lower area of the display. The number of the softkey row is specified in bit 0..7 of word W35. If you want to use a softkey row, you must define it in PCS*PRO*^{WIN}.

2.1.2.2.1 EXTERNAL SOFTKEY FUNCTIONS

The softkey functions 1...255 are designated as external softkey functions. For each function, up to 8 different „actions" can be defined. These actions consist of data word write/read operations in the programmable controller. This enables another background bitmap to be activated by a softkey function for example.

2.1.2.2.2 INTERNAL SOFTKEY FUNCTIONS

The softkey functions starting with 255 are reserved for device-dependent internal actions. The following functions are currently defined:

Function	Designation	Function
256	LOGIN	Loading the password dialogue for activating of a new access level.
257	LOGOUT	Exiting the current access level.

2.1.2.3 WORKING AREA

Priority-related pages (e.g. idle texts, operating texts, message texts, history texts and help texts) are visualized in the center display area between the status page and the soft key row. The variables and texts contained in this area can be edited by the user, unless this is locked by the programmable controller. A background bitmap which is displayed in this area can be assigned to each page.

The elements which can be used on status and operating pages are described in the following sections.

2.2 VARIABLES

Variables can be used in all texts. Beginning with this location, the PCS reserves space for the variable. Format and length of the variable are interpreted from the variable description. A maximum of 8 variables per text line are allowed. During text creation, PCS*PRO*^{WIN} automatically considers the additional variable length in each line.

There is a difference between INTERNAL and EXTERNAL variables. The source values of the EXTERNAL variables are located in the programmable controller. A corresponding description must be created for these variables. The description of the external variables is stored in the PCS during configuration. Some internal variables require that the start values or the expressions (depending on the language!) of the variable descriptions must be completed.

The variable types (V)BIN(0)-1,A also enable scaling, i.e. a specified value range (source) in the programmable controller is transferred to another representation range (destination) in the PCS (limitation: the multiplier must be positive !).

For all BIN (binary) variables, the number of pre-decimal and decimal point places and the limit values (minimum and maximum values) can be programmed as constants.

BCD(0)-1,2 enable a minimum and maximum value and the number of digits to be defined.

Each variable can be defined as ACTUAL, PRESET or PRESET-P value.

2.2.1 EXTERNAL VARIABLE FORMATS

An overview is given below before each type is described in detail.

- 1. BIT

  A string (expression) is assigned to the two states of a bit in the programmable controller. The string is freely selectable, but its length may not exceed that of a display line (40 characters). The string may not contain any variable. The longer expression determines the space to be reserved. The BIT variable is immediately written into the programmable controller after modification.

- 2. STRING

  A string (expression) can be assigned to each value of the least significant byte of a programmable controller word. So the maximum number of expressions is 256. The maximum length of an expression may not exceed that of a display line (40 characters). The space to be reserved is determined by the longest expression. The string may not contain any other variable.

- 3. CSTRING

  A string (expression) can be assigned to each value of the least significant byte of a programmable controller word. So the maximum number of expressions is 256. The maximum length of an expression may not exceed that of a display line (40 characters). The space to be reserved is determined by the longest expression. The string may not contain any other variable. The difference between the CSTRING variable and the STRING variable is that the CSTRING variable is immediately written into the programmable controller after modification.

■ 4. BCD

Values with a selectable number of digits are displayed. For these numbers, the BCD format is required in the programmable controller. The decimal point cannot be displayed. Unused leading digits are ignored when reading the ACTUAL value and set to zero when writing the PRESET(-P) value. The following variable formats are admissible:

BCD-1: Selectable number of digits between 1 and 4 (maximum). This variable requires a word in the programmable controller.

BCD0-1: Just as BCD-1, but leading zeros are displayed instead of blanks.

BCD-2: Selectable number of digits between 1 and 8 (maximum). This variable requires a double word in the programmable controller.

BCD0-2: Just as BCD-2, but leading zeros are displayed instead of blanks.

■ 5. BIN

The 16-bit value of a word or the 32-bit value of a double word in the programmable controller are represented in the fixed-point format as unsigned number. The variable requires a maximum of 11 digits (including decimal point). Selection of the number of pre-decimal and decimal point places enables display of the decimal point. In this case, the display space required for the decimal point has to be considered. If 16-bit variables are used, scaling is possible, i.e. conversion of the programmable controller range into that of the PCS and, vice versa, conversion of the PCS range into that of the programmable controller. In the programmable controller, the range from $0 to $FFFF (16-bit variables) and from $0 to $FFFFFFFF (32-bit variables) can be displayed. In the PCS, the value range from 0 to 4 294 967 295 (maximum) can be displayed. The following variable formats are admissible:

BIN-1: This variable requires one word in the programmable controller. The number of pre-decimal point places can be set to a value between 1 und 10 (maximum) and the number of decimal point places to a value between 0 (without decimal point) and 9 (maximum). If decimal point places are specified, the variable requires an additional character for the decimal point representation. If the minimum value of the programmable controller and that of the PCS or the maximum value of the programmable controller and that of the PCS are different, a scaled BIN variable is used. In this case, entry of pre-decimal point places is separated from the entry of decimal point places, if these are specified. After pressing the <.> key, decimal point places can be entered. This kind of number entry is also called calculator entry.

BIN0-1: Just as BIN-1, but leading zeros are displayed instead of blanks.

BIN-A: Just as BIN-1, but the value is not entered as on a calculator, but by „shifting“ beyond the decimal point (from right to left).

BIN0-A: Just as BIN-1, but the value is not entered as on a calculator, but by „shifting“ beyond the decimal point (from right to left). Additionally, leading zeros are displayed instead of blanks.

BIN-2: This variable requires one double word in the programmable controller. The number of pre-decimal point places can be set to a value between 1 und 10 (maximum) and the number of decimal point places to a value between 0 (without decimal point) and 9 (maximum). If decimal point places are specified, the variable requires an additional character for the decimal point representation. When using this variable, entry of pre-decimal point places is separated from the entry of decimal point places, if these are specified. After pressing the <.> key, decimal point places can be entered. This kind of number entry is also called calculator entry.

BIN0-2: Just as BIN-2, but leading zeros are displayed instead of blanks.

BIN-B: Just as BIN-2, but the value is not entered as on a calculator, but by „shifting“ beyond the decimal point (from right to left).

BIN0-B: Just as BIN-2, but the value is not entered as on a calculator, but by „shifting“ beyond the decimal point (from right to left). Additionally, leading zeros are displayed instead of blanks.

■ 6. VBIN

The 16-bit value of a word or the 32-bit value of a double word in the programmable controller are represented in the fixed-point format as signed number. The variable requires a maximum of 12 digits (always with sign and optionally with decimal point). Selection of the number of pre-decimal and decimal point places enables display of the decimal point. In this case, the display space required for the decimal point and the sign has to be considered. If 16-bit variables are used, scaling is possible, i.e. conversion of the programmable controller value range into that of the PCS and, vice versa, conversion of the PCS range into that of the programmable controller. In the programmable controller, the range from $80000 to $7FFF (16-bit variables) and from $80000000 to $7FFFFFFF (32-bit variables) can be displayed. In the PCS, the value range from -2 147 483 648 to +2 147 483 647 (maximum) can be displayed. The sign can be changed using the <+> or <-> key. The following variable formats are admissible:

VBIN-1: This variable requires one word in the programmable controller. The number of pre-decimal point places can be set to a value between 1 und 10 (maximum) and the number of decimal point places to a value between 0 (without decimal point) and 9 (maximum). If decimal point places are specified, the variable requires an additional character for the decimal point representation. If the minimum value of the programmable controller and that of the PCS or the maximum value of the programmable controller and that of the PCS are different, a scaled VBIN variable is used. In this case, entry of pre-decimal point places is separated from the entry of decimal point places, if these are specified. After pressing the <.> key, decimal point places can be entered. This kind of number entry is also called calculator entry.

VBIN0-1: Just as VBIN-1, but leading zeros are displayed instead of blanks.

VBIN-A: Just as VBIN-1, but the value is not entered as on a calculator, but by „shifting“ beyond the decimal point (from right to left).

VBIN0-A: Just as VBIN-1, but the value is not entered as on a calculator, but by „shifting“ beyond the decimal point (from right to left). Additionally, leading zeros are displayed instead of blanks.

VBIN-2: This variable requires one double word in the programmable controller. The number of pre-decimal point places can be set to a value between 1 und 10 (maximum) and the number of decimal point places to a value between 0 (without decimal point) and 9 (maximum). If decimal point places are specified, the variable requires an additional character for the decimal point representation. When using this variable, entry of pre-decimal point places is separated from the entry of decimal point places, if these are specified. After pressing the <.> key, decimal point places can be entered. This kind of number entry is also called calculator entry.

VBIN0-2: Just as VBIN-2, but leading zeros are displayed instead of blanks.

VBIN-B: Just as VBIN-2, but the value is not entered as on a calculator, but by „shifting“ beyond the decimal point (from right to left).

VBIN0-B: Just as VBIN-2, but the value is not entered as on a calculator, but by „shifting“ beyond the decimal point (from right to left). Additionally, leading zeros are displayed instead of blanks.

■ 7. WORD

The 16-bit value of a word in the programmable controller is represented in the bit format. The <+> and Btt<-> keys are used to position the cursor on the individual bits. A single bit is reset by pressing the <0> key and set by pressing the <1> key. This data format permanently requires 17 characters per line. A blank inserted between the HIGH and the LOW byte is used for better representation.

The WORD variable of the PCS 950 is used to represent the content of a 16-bit word in different formats:

KM - bit-by-bit representation of a word, e.g. ‘10001001 10101011’ (see word variable)

KH - hexadecimal representation of a word, e.g. ‘89AB’ (for entry see ASCII variable)

KY - byte-by-byte decimal representation, e.g. ‘137 171’ (for entry see binary variable)

■ 9. ASCII

Up to 32 characters (16 words) can be read from the programmable controller and represented as ASCII characters or changed. Pressing the <+> or <-> key causes display of the ASCII character with the next higher/lower ASCII code. The <.> key is used to move the cursor one digit to the right. If the last character has been entered, pressing the <.> key repositions the cursor on the first character. Numbers can directly be entered.

■ 10. TIMER

The TIMER variable format is used to specify a 3-digit time value and to select the time base from 4 possible values.

The TIMER variable reads/writes the content from/into a 16-bit word in the following format:

'00dd cccc bbbb aaaa'

aaaa = BCD coded number D1 (0..9) of the time value

bbbb = BCD coded number D2 (0..9) of the time value

cccc = BCD coded number D3 (0..9) of the time value

dd = Time base value (0..3)

Content of the word '2 1 0 0' - time value 100 corresponds to 100 seconds

I

Time base 2 (corresponds to * 1s)

2.2.1.1 BIT VARIABLE FORMAT

A single bit of a specified word can be represented as actual value or set/reset as preset value using the <+>/ <-> key. The modification is effective immediately after pressing the key. The other bits of the corresponding word are not affected during storing.

EXAMPLE:

Suppose, you have assigned a BIT variable to word 130 as PRESET value using PCSPROWIN and selected bit 15 as bit number. The string (expressions) assigned to the logical bit state 0 is „CLOSED“ and the string assigned to the logical bit state 1 is „OPEN“. Summary:

Word number:	130
Class:	PRESET
Variable format:	BIT
Bit position:	15
Expression 0 (AP0):	CLOSED
Expression 1 (AP1):	OPEN

The variable is inserted into the operating text 0 in the following way:

If bit 30.15 = 0, the following is displayed after selecting operating text 0:

Ist das Bit 30.15 = 1, so erscheint bei angewähltem Bedientext 0 im Display:

If this operating text is located within an operating page, use the <+> and <-> keys to switch between these two states.

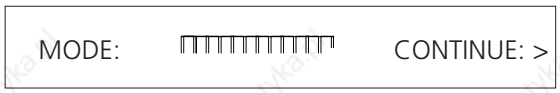
2.2.1.2 STRING VARIABLE FORMAT

When specifying the actual value, up to 256 texts can be assigned to the content of the less significant byte of a word. To define the preset value on the operating page, the value of the less significant byte is incremented using the <+> key and decremented using the <-> key. In this case, no intermediate values are written into the programmable controller.

EXAMPLE: Suppose, you have assigned a STRING variable to word 131 as PRESET-P value using PCS*PRO*^{WIN}. The terms „SERVICE", „CONFIGURATION" and „AUTOMATIC" have been assigned to the strings (expressions) 0..2. Summary:

Word number:	131
Class:	PRESET-P
Variable format:	STRING
Expression 0 (AP0):	SERVICE
Expression 1 (AP1):	CONFIGURATION
Expression 2 (AP2):	AUTOMATIK

The variable is inserted into the operating text 15 in the following way:



If the less significant byte of word 131 is set to 1, the following is displayed after selecting operating text 15:

If the variable is used in an operating text, the value in word 131 can be decremented up to the value of zero using the <-> key and incremented up to 2 using the <+> key. It must be noticed, however, that a modified value is only stored in the word after pressing the <ENT> key or after exiting the variable field. If the value has to be transferred immediately into the programmable controller, please refer to the description of the CSTRING variable format.

NOTES

1. The bits of the most significant byte of word 131 are ignored during reading and reset to 0 when stored in the programmable controller. This enables detection of changes by the programmable controller program.

*2. If the previous value is not changed, no data are stored (this is also valid for bits 8..15).

3. A maximum of 256 expressions (including 0) are admissible.

4. The limit depends on the number of programmed expressions; the minimum value is always 0.

5. A minimum of 3 expressions must be specified, otherwise the variable has to be declared as BIT.

*6. After editing has been started, exiting the entry field after specifying a value exceeding the limits is not possible.

7. Restoring the previous value is always possible by pressing the <CLR> key.

* These points are only valid, if the operating page options correspond to the default setting!

2.2.1.3 CSTRING VARIABLE FORMAT

Concerning the actual value, this variable type corresponds to the STRING type.

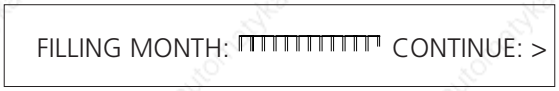
If the variable is used on an operating page, the value on the specified word address can be decremented up to 0 using the <-> key and incremented using the <+> key. In contrast to STRING, a modified value is immediately written into the programmable controller (i.e. after each modification).

EXAMPLE:

Using PCS*PRO*^{WIN}, you assigned a CSTRING variable to word 132 as PRESET value. The following terms are assigned to the strings (expressions) 0..11: „JANUARY“, „FEBRUARY“, „MARCH“, „APRIL“, „MAY“ up to „DECEMBER“. Summary:.

Word number:	132
Class:	PRESET
Variable format:	CSTRING
Expression 0 (AP0):	JANUARY
Expression 1 (AP1):	FEBRUARY
up to expression 11 (AP11):	DECEMBER

The variable is inserted into the operating text 20 in the following way:



If the less significant byte of word 132 contains a value of 5, the following is displayed after selecting operating text 20:

NOTES:

1. The bits of the most significant byte of word 132 are ignored during reading and reset to 0 when writing into the programmable controller. This enables detection of changes by the programmable controller program
2. A maximum of 256 expressions (including 0) are admissible.
3. The limit depends on the number of programmed expressions; the minimum value is always 0.
4. A minimum of 3 expressions must be defined. Otherwise, the variable has to be declared as BIT.
5. After editing has been started, exiting the entry field after specifying a value exceeding the limits is not possible.
6. Restoring the previous value with <CLR> is **not** possible.

2.2.1.4 BCD VARIABLE FORMAT

A maximum of 4 digits per word is represented numerically as actual value. One digit represents the numeric value (0-9) of 4 bits.

The BCD variable formats are divided into the following groups:

Variable type	16-bit	32-bit	Number of digits	Leading zero repres.
1. BCD-1	x		1..4	
2. BCD0-1	x		1..4	x
3. BCD-2		x	1..8	
4. BCD0-2		x	1..8	x

Suppose, you assigned a BCD variable (BCD-2) to word 133 as PRESET-P value. You want 8 digits to be displayed. The minimum value which can be entered is set to 90 and the maximum value to 50 000 000. Summary:

Word number: 133
Class: PRESET-P
Variable format: BCD-2
Number of digits: 8
Minimum value: 90
Maximum value: 50000000

Insertion of the variable into the operating text 100 is represented below:

NUMBER OF FINISHED PIECES: ΠΠΠΠΠΠΠΠ CONTINUE: >

If word 133 contains the value $0045 (69) and word 134 the value $5673 (22131), the following is displayed after selecting operating text 100:

NUMBER OF FINISHED PIECES: 455673 CONTINUE: >

The 2 leading zerors are suppressed, since the BCD variable format is used! If a leading zero display is desired, use the BCD0 variable format instead of BCD!

NOTES:

1. Unused most significant bits are ignored and stored as 0's.
2. Scaling and decimal point representation are not possible.

*3. Intermediate values are not stored. Before data can be stored, press <ENTER> or exit the variable field.

* This point is only valid, if the operating pages correspond to the default setting.

4. A balancing entry is also possible <1> <0> <+> would result in the intermediate result of 455683 in the above example. Since this is an intermediate result, it is not stored yet (although the cursor is no longer flashing.)!

*5. After editing has been started, exiting the entry field after specifying a value exceeding the limits is not possible.

6. The sign keys can also be used for incrementing and decrementing values (with Auto Repeat).

7. Restoring the previous value is possible at any time using <CLR>.

* These points are only valid, if the operating page options correspond to the default setting!

2.2.1.5 BIN VARIABLE FORMAT

The binary value of one or two words is converted to determine the actual value. During representation, the decimal point, a sign (if required) and the leading zeros are considered, according to the number of digits specified. The BIN variable formats are divided into the following groups:

Var. type	16-bit	32-bit	Calculator entry	Scaling	Sign	Leading 0's
1. BIN-1	x		x	x		
2. BIN-A	x			x		
3. BIN-2		x	x			
4. BIN-B		x				
5. VBIN-1	x		x	x	x	
6. VBIN-A	x			x	x	
7. VBIN-2		x	x		x	
8. VBIN-B		x			x	
9. BIN0-1	x		x	x		x
10. BIN0-A	x			x		x
11. BIN0-2		x	x			x
12. BIN0-B		x				x
13. VBIN0-1	x		x	x	x	x
14. VBIN0-A	x			x	x	x
15. VBIN0-2		x	x		x	x
16. VBIN0-B		x			x	x

(V)BIN(0)-1 / (V)BIN(0)-2 and (V)BIN(0)-A / (V)BIN(0)-B differ according to the editing mode:

- (V)BIN(0)-1, (V)BIN(0)-2: Calculator entry, i.e. pre-decimal and decimal point places are separately entered (only if decimal point places are used). The <.> key is used for switching.
- (V)BIN(0)-A, (V)BIN(0)-B: Shifting from right to left (beyond the decimal point). The <.> key is not effective.

Suppose, xou assigned a BIN variable (BIN-1) to word 42 as PRESET value. You want to display and enter two pre-decimal point places and one decimal point place. Scaling is also desired. Values between 0 and 100 (0 and 10,0) may be entered into the PCS. However, this value range has to be mapped to the programmable controller range 0...4095 ($0..$0FFF). Leading zeros must be suppressed.

Summary:

Word number:	42
Class:	PRESET
Variable format:	BIN-1
Pre-decimal point places:	2
Decimal point places:	1
Minimum value PCS:	0
Maximum value PCS:	100
Minimum vlaue progr. cont.:	0
Maximum value progr. cont.:	4095

Insertion of the variable into the operating text 120 is described below:

```
ANALOG VOLTAGE IIIII VOLT  CONTINUE:>
```

If word 42 contains the value $0800 (2048), the follwoing is displayed after selecting operating text 120:

```
ANALOG VOLTAGE  5.0 VOLT  CONTINUE:>
```

Handling of a preset value variable on an operating page:

- The numeric keys are used to modify the value.

  (V)BIN(0)-1(2): Pre-decimal and decimal point places are separated. The <.> key is used for switching.

  (V)BIN(0)-A(B): Simple shifting from right to left (the decimal pint is skipped).
- A Balancing entry is possible (not valid for VBIN variables!): e.g. <.> <2> <+>: new representation (example): 5.2!
- <+> / <-> keys: BIN(0)-1,2,A,B: 1 is added/subtracted (even if <.> is pressed).
  VBIN(0)-1,2,A,B: Change of the sign (possible at any time).

Attention!

* ■ Within the limit values, only modified values are stored.

■ If the source value exceeds the limits, inverse arrows are represented.

■ If a value beyond the limit has been entered (only possible when numbers are directly entered), the value is checked when pressing the ENTER key or exiting the field. If an error occurs and if the entered value is below the minimum value, the minimum value is displayed. If the value you entered exceeds the maximum value, the maximum value is displayed. Additionally, the audible signal is generated and no data are written into the programmable controller.

Exiting the inverse field is not possible. If for example the first variable of a text falls beyond the limits, exiting the operating page is not possible either. First of all, the value has to be corrected (if BIN variables are used, +, - or CLR and if VBIN variables are used, only CLR or the corresponding numeric keys are admissible to enter a valid number).

* ■ The indicated value ranges (programmable controller and PCS) may only be negative for VBIN(0) variables. In this case, the corresponding value(s) must be preceded by the minus sign.

* These points are only valid, if the operating page options correspond to the default setting!

2.2.1.6 WORD VARIABLE FORMAT

Word number: 135
Class: PRESET
Variable format: WORD

1. Format when using bit-by-bit representation (corresponds to KB)

The word on the specified address is represented in binary format using 0 and 1 (e.g. a PRESET value has been assigned to word 135): The insertion of the variable into the operating text 99 is represented below:

W 35 BINARY: ┌┬┬┬┬┬┬┬┬┬┬┬┐ CONTINUE:>

If word 135 contains the value $5A5A, the following is displayed after selecting operating text 99:

W 35 BINARY: 01011010 01011010 CONTINUE:>

If the variable is used on an operating page, the <+> and <-> keys are used to position the cursor bit-by-bit. The <0> and <1> keys are used to set the bit at the cursor position to logical 0 or 1.

2. Format when using the dual decimal representation (corresponds to KY)

The word on the specified address is represented using decimal numbers with separation of the high and low byte of the word:

W 35 BINARY: ┌┬┬┬┬┬┬┬┬┬┬┬┐ CONTINUE:>

If word 135 contains the value $5A5A, the following is displayed after selecting operating text 99:

W 35 BINARY: 123.123 CONTINUE:>

0..9: calculator entry of high/low byte; Point: switching between digit high/low byte; +/-: INC/DEC of high/low byte.

3. Format when using the hexadecimal representaion (corresponds to KH)

The word on the specified address is represented word-by-word using the numbers 0...F.

W 35 BINARY: ┌┬┬┬┬┬┬┬┬┬┬┬┐ CONTINUE:>

W 35 BINARY: 5A5A CONTINUE:>

Point: change to the next digit (right direction); 0..9: assigning a number to each digit; +/-: accessing the numbers A...F (pseudo tetrad)

Generally, a modified value is only stored in word 135, if the <ENT> key is pressed or if you exit the variable field.

Attention!

* ■ If the previous value is not changed, no data are stored.

■ Restoring the previous value is possible at any time using the CLR key.

■ The WORD variable format permanently requires 17 characters in the display (the 8 most significant bits are separated by a SPACE from the 8 least significant bits)!

* This point is only valid, if the operating page options correspond to the default setting!

2.2.1.7 TIMER

The variable format TIMER is used to specify a 3-digit time value and to select the time base from 4 possible values.

The TIMER variables reads/writes the content from/into a 16-bit word in the following format:

'00dd cccc bbbb aaaa'

aaaa	=	BCD-coded number D1 (0..9) of the time value
bbbb	=	BCD-coded number D2 (0..9) of the time value
cccc	=	BCD-coded number D3 (0..9) of the time value
dd	=	Time base value (0..3)

Word content '2 1 0 0' - time value 100 corresponds to 100 seconds

|

Time base 2 (corresponds to * 1s)

The texts used to represent the selected time base can be created as desired. To modify a TIMER preset value, the time value and (if required) the time base must be modified. To switch between these two entries, use the '.' key of the PCS.

The time value can be directly modified using the numeric keys. If the time base modification is activated, it can be selected with the <+> / <-> and with the <0> ... <<3> keys.

Timer variable in accordance with the Siemens format with 3 BCD digits and 4 project.AP with a maximum of 37 characters.

Example with an AP comprising 4 characters:

Word format:

Bits 14+15 = These bits are ignored during reading and set to 0 during writing

Bits 12+13 = These bits indicate the corresponding AP

Bit 11...9 = 3-digit BCD number

2.2.1.8 ASCII VARIABLE FORMAT

ASCII variables allow any strings to be entered. The string length must be even (a maximum of 32 digits). All characters (H00 to HFF) can be entered.

Example: An ASCII variable has been assigned to word 136 as PRESET value. A 16-digit serial number is desired to be displayed and entered.

Word number:	136
Class:	PRESET
Variablenformat:	ASCII
Number of characters:	16 (8 words)

Insertion of the variable into the operating text 90 is represented below:

SERIAL NUMBER: ┌┬┬┬┬┬┬┬┬┬┬┬┬┬┬┐ CONTINUE:>

For the word values W136=$4557, W137=$4120, W138=$344E, W139=$4542, W140=$2D38, W141=$3131, W142=$3530 and W143=$3533 (corresponds to the string "EWA-4NEB 8115053), the follwoing is displayed after selecting operating text 90:

SERIAL NUMBER: EWA-4NEB 8115053 CONTINUE:>

If the variable is used on an operting page, use the <.> key to move the cursor (flashing digit) one digit to the right. If the end of the variable (string end) is reached, pressing the <.> key again repositions the cursor at the beginning of the variable. The <+> and <-> keys are used to select any character (including special characters). A modified value is only written in binary format into the transfer area (starting with word 136 (W136..W144), if you press the <ENTER> key or exit the variable field (unless the value has not been modified).

Since ALL characters can be represented, limit values are not checked.

Attention!

* ■ If the previous value is not modified, no data are stored.

■ Restoring the previous value is always possible by pressing CLR.

■ Only even character lengths are admissible!

* This point is only valid, if the operating page options correspond to the default setting!

2.2.2 INTERNAL VARIABLE FORMATS

83 internal variables (already pre-defined) are available which in most cases can only be used efficiently in specific text groups. This is checked by PCSPROWIN before insertion.

If the REPRESENTATION LENGTH column contains an x, the length can be determined by the programmer (e.g. specification of the day of the week according to the language used). Additionally, a default value can be assigned to all preset values. This default value is initialized after transfer of the data record into non-volatile RAM (except for time, date and counter for operating hours). If the preset value is used on an operating page, it can be changed during runtime of the device. The string constants Z075 to Z082 can be modified within PCSPROWIN and are used for printer output formatting.

Firmware	Former desig.	PCSPROWIN design	Type	Class	Repr. length	Default value
[Z001]	ZP	[HINWEISE]	INT_BIN-2	ACTUAL	4	0
[Z002]	ZQ	[WARNUNGEN]	INT_BIN-2	ACTUAL	4	0
[Z003]	ZR	[STOERUNGEN]	INT_BIN-2	ACTUAL	4	0
[Z007]	ZX	[ERR_SCHNITTST]	INT_BIN-2	ACTUAL	2	0
[Z008]	ZA	[TEXTNUMMER]	INT_BIN-2	ACTUAL	4	0
[Z009]	ZC	[ZEIT_MLD_KOMMT]	INT_ZEIT_MLD_KOMMT	ACTUAL	17	0
[Z010]	ZD	[ZEIT_MLD_GEHT]	INT_ZEIT_MLD_GEHT	ACTUAL	17	0
[Z011]	ZE	[ZEIT_MLD_QUITT]	INT_ZEIT_MLD_QUITT	ACTUAL	17	0
[Z012]	ZG	[UHR_STUNDEN]	INT_BIN0-2	PRESET	2	0
[Z013]	ZH	[UHR_MINUTEN]	INT_BIN0-2	PRESET	2	0
[Z014]	ZI	[UHR_SEKUNDEN]	INT_BIN0-2	PRESET	2	0
[Z015]	ZL	[DATUM_JAHR]	INT_BIN0-2	PRESET	2	0
[Z016]	ZK	[DATUM_MONAT]	INT_BIN0-2	PRESET	2	0
[Z017]	ZJ	[DATUM_TAG]	INT_BIN0-2	PRESET	2	0
[Z018]	ZN	[WOCHENTAG_IST]	INT_STRING	ACTUAL	x	0
[Z019]	ZO	[WOCHENTAG_SOLL]	INT_STRING	PRESET	x	0
[Z020]	ZY	[UHRZEIT]	INT_UHRZEIT	ACTUAL	8	0
[Z021]	ZZ	[DATUM]	INT_DATUM	ACTUAL	8	0
[Z022]		[ZEITSCHALTUHR]	INT_STRING	PRESET	16	0
[Z023]		[NOCKEN_NUMMER]	INT_BIN-2	PRESET	1	0
[Z027]		[BAUDRATE]	INT_STRING	PRESET	5	1
[Z028]		[PARITAET]	INT_STRING	PRESET	5	1
[Z029]		[DATENBIT]	INT_STRING	PRESET	1	0
[Z030]		[STOPBIT]	INT_STRING	PRESET	1	1
[Z031]		[RS232/TTY]	INT_STRING	PRESET	5	0
[Z032]		[HISTORYTEXTE]	INT_BIN-2	ACTUAL	4	0
[Z033]	x80	[MLDTXT_ZEILE1]	INT_MLDTXT_ZEILE	ACTUAL	40	0
[Z034]	x81	[MLDTXT_ZEILE2]				
[Z065]		[BETR_STD_IST]	INT_BIN-2	ACTUAL	10	0
[Z066]		[BETR_STD_SOLL]	INT_BIN-2	PRESET	10	0
[Z067]		[HISTORY_EINTR]	INT_BIN-2	ACTUAL	5	0
[Z068]		[DRUCKER_EINTR]	INT_BIN-2	ACTUAL	5	0
[Z069]		[ZSU_EIN_STUNDE]	INT_BIN0-2	PRESET	2	0
[Z070]		[ZSU_EIN_MINUTE]	INT_BIN0-2	PRESET	2	0
[Z071]		[ZSU_EIN_SEK]	INT_BIN0-2	PRESET	2	0
[Z072]		[ZSU_AUS_STUNDE]	INT_BIN0-2	PRESET	2	0
[Z073]		[ZSU_AUS_MINUTE]	INT_BIN0-2	PRESET	2	0
[Z074]		[ZSU_AUS_SEK]	INT_BIN0-2	PRESET	2	0
[Z075]		[DRUCKERTEXTE]	IST_BIN-2	ACTUAL	4	0
[Z076]		<TAB>	INT_DRUCKERSEQUENZ	ACTUAL	0	H09
[Z077]		<ESC>	INT_DRUCKERSEQUENZ	ACTUAL	0	H1B
[Z078]		<LF>	INT_DRUCKERSEQUENZ	ACTUAL	0	H0D H0A
[Z079]		<FF>	INT_DRUCKERSEQUENZ	ACTUAL	0	H0C
[Z080]		<Fe+>	INT_DRUCKERSEQUENZ	ACTUAL	0	H1B H45
[Z081]		<Fe->	INT_DRUCKERSEQUENZ	ACTUAL	0	H1B H46
[Z082]		<Un+>	INT_DRUCKERSEQUENZ	ACTUAL	0	H1B H2D H31
[Z083]		<Un->	INT_DRUCKERSEQUENZ	ACTUAL	0	H1B H2D H30

2.2.2.1 IMPLEMENTATION OF THE INTERNAL VARIABLES

	IDLE TEXT	OPERATING TEXT/STATUS TEXT	MESSAGE TEXT	HISTORY TEXT	HLP TEXT FOR IDLE PRIORITY	HLP TEXT FOR OPERATING PRIORITY	HLP INF./HELP TEXT FOR INFORMATION PRIORITY	HELP TEXT FOR WARNING PRIORITY	HELP TEXT FOR FAILURE PRIORITY	HELP TEXT FOR HISTORY PRIORITY	OPERATING PRINTER TEXT	MESSAGE PRINTER TEXT
[Z001] [HINWEISE]	X	X	X	-	X	X	X	X	X	X	X	-
[Z002] [WARNUNGEN]	X	X	X	-	X	X	X	X	X	X	X	-
[Z003] [STOERUNGEN]	X	X	X	-	X	X	X	X	X	X	X	-
[Z007] [ERR_SCHNITTST]	X	X	X	-	X	X	X	X	X	X	X	-
[Z008] [TEXTNUMMER]	X	X	X	X	X	X	X	X	X	X	X	X
[Z009] [ZEIT_MLD_KOMMT]	-	-	X	X	-	-	X	X	X	X	-	X
[Z010] [ZEIT_MLD_GEHT]	-	-	X	X	-	-	X	X	X	X	-	X
[Z011] [ZEIT_MLD_QUIT]	-	-	X	X	-	-	X	X	X	X	-	X
[Z012] [UHR_STUNDEN]	X	X	-	-	-	-	-	-	-	-	-	-
[Z013] [UHR_MINUTEN]	X	X	-	-	-	-	-	-	-	-	-	-
[Z014] [UHR_SEKUNDEN]	X	X	-	-	-	-	-	-	-	-	-	-
[Z015] [DATUM_JAHR]	X	X	-	-	-	-	-	-	-	-	-	-
[Z016] [DATUM_MONAT]	X	X	-	-	-	-	-	-	-	-	-	-
[Z017] [DATUM_TAG]	X	X	-	-	-	-	-	-	-	-	-	-
[Z018] [WOCHENTAG_IST]	X	X	X	X	X	X	X	X	X	X	X	X
[Z019] [WOCHENTAG_SOLL]	X	X	-	-	-	-	-	-	-	-	-	-
[Z020] [UHRZEIT]	X	X	X	X	X	X	X	X	X	X	X	X
[Z021] [DATUM]	X	X	X	X	X	X	X	X	X	X	X	X
[Z022] [ZEITSCHALTUHR]	X	X	-	-	-	-	-	-	-	-	-	-
[Z023] [NOCKEN_NUMMER]	X	X	-	-	-	-	-	-	-	-	-	-
[Z027] [BAUDRATE]	X	X	-	-	X	X	X	X	X	X	-	-
[Z028] [PARITAET]	X	X	-	-	X	X	X	X	X	X	-	-

	IDLE TEXT	OPERATING TEXT/STATUS TEXT	MESSAGE TEXT	HISTORY TEXT	HLP TEXT FOR IDLE PRIORITY	HLP TEXT FOR OPERATING PRIORITY	HLP INF./HELP TEXT FOR INFORMATION PRIORITY	HELP TEXT FOR WARNING PRIORITY	HELP TEXT FOR FAILURE PRIORITY	HELP TEXT FOR HISTORY PRIORITY	OPERATING PRINTER TEXT	MESSAGE PRINTER TEXT
[Z029] [DATENBIT]	X	X	-	-	X	X	X	X	X	X	-	-
[Z030] [STOPBIT]	X	X	-	-	X	X	X	X	X	X	-	-
[Z031] [RS232/TTY]	X	X	-	-	X	X	X	X	X	X	-	-
[Z032] [HISTORYTEXTE]	X	X	X	X	X	X	X	X	X	X	X	-
[Z033..34] [MLD_TXT_ZEILE1...2]	-	-	-	X	-	-	-	-	-	X	-	X
[Z065] [BETR_STD_IST]	X	X	X	X	X	X	X	X	X	X	X	X
[Z066] [BETR_STD_SOLL]	X	X	-	-	-	-	-	-	-	-	-	-
[Z067] [HISTORY_EINTR]	-	-	-	X	-	-	-	-	-	X	-	-
[Z068] [DRUCKER_EINTR]	-	-	-	-	-	-	-	-	-	-	-	X
[Z069] [ZSU_EIN_STUNDE]	X	X	-	-	-	-	-	-	-	-	-	-
[Z070] [ZSU_EIN_MINUTE]	X	X	-	-	-	-	-	-	-	-	-	-
[Z071] [ZSU_EIN_SEK]	X	X	-	-	-	-	-	-	-	-	-	-
[Z072] [ZSU_AUS_STUNDE]	X	X	-	-	-	-	-	-	-	-	-	-
[Z073] [ZSU_AUS_MINUTE]	X	X	-	-	-	-	-	-	-	-	-	-
[Z074] [ZSU_AUS_SEK]	X	X	-	-	-	-	-	-	-	-	-	-
[Z075] [DRUCKERTEXTE]	X	X	X	-	X	X	X	X	X	X	X	-
[Z076] <TAB>	-	-	-	-	-	-	-	-	-	-	X	X
[Z077] <ESC>	-	-	-	-	-	-	-	-	-	-	X	X
[Z078] <LF>	-	-	-	-	-	-	-	-	-	-	X	X
[Z079] <FF>	-	-	-	-	-	-	-	-	-	-	X	X
[Z080] <Fe+>	-	-	-	-	-	-	-	-	-	-	X	X
[Z081] <Fe->	-	-	-	-	-	-	-	-	-	-	X	X
[Z082] <Un+>	-	-	-	-	-	-	-	-	-	-	X	X
[Z083] <Un->	-	-	-	-	-	-	-	-	-	-	X	X
External variable	X	X	X	X	X	X	X	X	X	X	X	X

The available internal variables are briefly described below:

- HINWEISE / WARNUNGEN / STÖRUNGEN (information/warnings/failures)

  The number of messages currently enabled for the corresponding priority is displayed.

- ERR_SCHNITTSTELLE (ERR interface)

  The maximum number of failed (repeated) packages after RESET is displayed. It always refers to 100 packages and characterizes the reliability of the data transmission which, on the other hand, depends on the cable length, the cable type and the intensity of the electric and magnetic interference fields. An error rate up to 1% is acceptable. This information is valid for all drivers supporting the internal variable ZX.

- TEXTNUMMER (text number)

  This variable indicates the current text number.

- ZEIT_MELD_GEHT / KOMMT / QUIT (time when a message was displayed, cleared and acknowledged)

  These variables are only efficient within the message priorities, the log buffer and the printer message buffer. If the value is not yet specified, zeros are displayed.

- UHR_STUNDEN/MINUTEN/SEKUNDEN + DATUM_MONAT/JAHR/TAG + WOCHENTAG (clock: hours/minutes/seconds + date: month/day/year + day of the week)

  These variables are used to set the clock. Since consistent clock values must be entered, all the 7 variables must be used within one operating page. The current clock values are stored temporarily when loading an operating page and - when exiting the operating page - are entirely written to the clock, if at least one value has been modified.

- ZEITSCHALTUHR (timer)

  The texts for 8 timers can be specified here. This value is used as index for the corresponding timers. The length is fixed to 16 characters !

- NOCKEN_NUMMER (cam number)

  8 ON/OFF times can be programmed within each timer. The 8 time ranges are ORed to a bit and written into the programmable controller as result of each timer. This cam number is used as an index to one of these 8 pairs of time instances. If only one cam number is used, this variable is not required. In this case, the ZSU_EIN/AUS (timer ON/OFF) variable refers to the first pair of time instances.

- BAUDRATE, PARITAET, DATENBIT, STOPBIT, RS232/TTY (baud rate, parity, data bit, stop bit, RS232/TTY)

  These parameters determine the PRN interface configuration. They can be modified online as preset values and are stored in non-volatile RAM. During data record transfer, these parameters are initialized to the default values specified here.

- HISTORYTEXTE/DRUCKERTEXTE (history texts/printer texts)

  This variable represents the number of messages in the history buffer (for display messages) or in the printer message buffer.

- MLDTXT_ZEILE1 + 2 (message text, lines 1 + 2)

  This variable inserts the n.th line of the corresponding message text into the log or message printer text. So all messages can be covered by creating a general range text (see section 2.6.5, HISTORY DISPLAY).

- BETR_STD_IST / SOLL (counter for operating hours: actual / preset values)

  Only preset values can be used to set the counter for operating hours to a specific value. The counter is not initialized during data record switching. However, if a firmware of a later version is loaded, the counter for operating hours is set to zero. (A corresponding message is displayed in this case.) The counter can then be corrected using this variable.

- HISTORY_EINTR, DRUCK_EINTR (history entry, printer entry)

  A counter is incremented for each log message to obtain uninterrupted logging. Since buffers overwrite each other, interruptions may occur. This number is only sensible in history texts and message printer texts. It is set to 0 when buffers are cleared and increments up to 9999. If the value exceeds 10000, inverse fields are displayed and if it exceeds 65535+1, zeros are displayed.

- ZSU_EIN/AUS_STUNDE/MINUTE/SEKUNDE (timer ON/OFF, hours/minutes/seconds)

  These times can be modified ONLINE and are managed internally by means of the TIMER and CAM NUMBER indices. After data record switching, the specified values are initialized. If the preset values are changed ONLINE, they are stored in the non-volatile memory area. Modification of individual parts is immediately effective. The following conditions are valid:

  - If the ON switching time corresponds to the OFF switching time, this cam is disabled.
  - If the ON switching time is below the OFF switching time, this cam is enabled every day.
  - If the ON switching time exceeds the OFF switching time, this cam is enabled across day boundaries.

- TAB, ESC, LF, FF, Fe+, Fe-, Un+, Un-

  These constants represent strings of any type with a maximum length of 8 characters (all characters except H00). The significance does not necessarily correspond to the designations, since these strings can be defined in any way in PCSPROWIN. They can only be used in message printer texts and operating printer texts.

2.2.3 HANDLING OF VARIABLES

All variables are automatically read out by the PCS or read starting with the indicated word number. The word number (DW, MW, DM, Counter..) used in the specific programmable controller or the designation is indicated in the PCS 091.x driver manual. This is also valid for preset values (the value read out is represented as default value - see also the section „Variables on operating pages“).

The following rules are valid for refreshing of variables (ACTUAL values or disabled PRESET(-P) values):

- All priority classes are constantly refreshed. The refreshing rate depends on different factors (number of variables displayed, driver type, transmission rate (baud rate), number of tasks specified for a transmission package as well as the programmable controller response time which also depends on the scan time of the programmable controller). The optimal value is probably a refreshing rate of 8 for each second).
- Since not all variables required for display or for the operating printer may be fetched during one transmission cycle, the values used in the PCS 950 may be from different programmable controller cycles. However, the values are only displayed, if all variable values are available.
- There is no difference between internal and external variables. If the variable values are not yet transmitted, empty fields (spaces) are displayed. If the value which has been read in exceeds the limit values stored in the PCS, inverse arrows (depending on the state - above or below the limit) are displayed in the variable field.
- PRESET-P variables are evaluated in the same way as ACTUAL values, as long as bit 7 of word 38 is set to logical 0.
- PRESET(-P) variables are first of all read, underlined and „frozen“ on the display. After „freezing“ this variable, modifications of the value through the programmable controller is not visualized. As soon as a key for preset value editing is pressed, a flashing cursor is displayed and the rest of the variable is represented statically (underlined). This is, however, not valid for balancing entries and for BIT and CSTRING variables, since they are immediately stored.

- There is no difference between internal and external variables. If the variable values are not yet transmitted, empty fields (spaces) are displayed. If the value which has been read in exceeds the limit values stored in the PCS, inverse arrows (depending on the state - above or below the limit) are displayed in the variable field.
- PRESET-P variables are evaluated in the same way as ACTUAL values, as long as bit 7 of word 38 is set to logical 0.
- PRESET(-P) variables are first of all read, underlined and „frozen" on the display. After „freezing" this variable, modifications of the value through the programmable controller is not visualized. As soon as a key for preset value editing is pressed, a flashing cursor is displayed and the rest of the variable is represented statically (underlined). This is, however, not valid for balancing entries and for BIT and CSTRING variables, since they are immediately stored.

Storing of the preset values can be specified individually for each operating page in PCS*PRO*^{WIN}. For this purpose, select an operating page in the appropriate window and enter the character <@> via the keyboard (on German keyboards <Alt-Gr+Q>). For normal applications, the default settings should be sufficient. This default setting is described below:

- After modification, a preset value (PRESET or PRESET-P) is stored, if you press the <ENTER> key or exit the variable field (admissible arrow keys). This is, however, not valid, if you close an operating page. In this case, the last value displayed is always stored.
- If an enabled PRESET-P value is displayed and if bit 7 of word 14 is set to 0, the first PRESET value which can be edited on this display page is searched for and underlined (not yet edited).
- After a PRESET(-P) value has been stored by the PCS, it is read twice (different programmable controller cycles) an then compared to the value edited before. If these values are different, the audible warning signal sounds and the instantaneous value of the programmable controller is underlined. This enables the programmable controller to check the limit values dynamically. To exit the variable field or the operating page, you have to confirm the value proposed by the programmable controller by pressing <ENTER> or an admissible <arrow key> (if transmission was initiated in this way).
- Make sure that the correct value („level") is specified by the programmable controller, if the dynamic limit check is applied to scaled binary variables and if variables are used for which the programmable controller range exceeds the PCS range. Example: The value range of the PCS is between 0 and 1000 and that of the programmable controller between 0 and 65535. A value of 10 on the PCS display corresponds to 655 in the programmable controller and a value of 11 on the PCS display to a value of 721 in the programmable controller. If the programmable controller writes a value of 670, exiting of the operating page will be constantly inhibited, since the value written by the PCS (655) always differs from 670.

2.3 TEXT GROUPS

There are 10 groups of texts which can be created as desired. The following overview indicates the maximum text and line numbers. This does not mean that all maximum values can be used. The maximum data record capacity is 128 kByte and the texts are dynamically allocated, i.e. texts which have not been created do not require memory capacity.

1. **128 IDLE TEXTS:**
2. **127 OPERATING TEXTS:**
3. **1024 MESSAGE TEXTS:** Text pages with a length up to two lines. These texts are permanently assigned to the message bits and displayed as INFORMATION TEXTS, WARNING TEXTS and FAILURE TEXTS.
4. **1024 HISTORY TEXTS:** These text pages may contain up to two lines and are displayed during history buffer representation.
5. **1024 MESSAGE PRINTER TEXTS:** These text pages are used to print the content of the message printer buffer. The content of these pages depends on the specific printer being used.
6. **255 OPERATING PRINTER TEXTS:** These texts which depend on the specific printer are selected numerically by the programmable controller and are immediately printed (maximum line length: 132, maximum line number: 126).
7. **HELP TEXTS FOR IDLE PRIORITY:** Each help text is displayed on a page with a maximum of 32 lines and can be activated at any time during ONLINE mode by pressing the <HLP> key.

   128 HELP TEXTS FOR IDLE PRIORITY: These texts are assigned to idle text numbers.
8. **127 HELP TEXTS FOR OPERATING PAGE PRIORITY:** These texts are assigned to the operating text numbers (max. 32 lines).
9. **1024 HELP TEXTS FOR INFORMATION/WARNING/FAILURE PRIORITY** with a maximum of 32 lines. They are assigned to the message text numbers.
10. **1024 HELP TEXTS FOR HISTORY PRIORITY:** These texts (max. 32 lines) are visualized during history buffer display, if the Btt<HELP> key is pressed. They are assigned to history texts.

A DEFAULT TEXT can be created for each text group. This DEFAULT TEXT is displayed within the indicated number range, if no main text has been assigned to the specified number.

If the number of display text lines exceeds the number of lines available in the working range, use the <ARROW DOWN> key to display the other pages (also called additional pages) and the <ARROW UP> key to switch back to the first page (also called main page). If paging is possible, the corresponding arrow key LEDs are ON. If a text consists of an odd number of lines, the last display line is empty.

On these pages, texts or values which are changed are represented by VARIABLES.

The only task which has to be accomplished by the programmable controller is to supply (almost continuously) the corresponding variable value. Any other additional programmable controller program is not required.

2.4 OPERATING PAGES/ORGANIZATION OF OPERATING PAGES

By checking the status of bit 7 of command word C (W38), the user can determine at any time, if PRESET-P variables can be changed. If bit 7 = 0, only preset values can be changed. If bit 7 = 1, PRESET and PRESET-P variables can be modified.

If a PRESET-P value is selected (underlined), setting bit 7 of word 38 to logical 0 selects the first PRESET value of this display page which can be edited. All preset and actual values which are not selected are continuously updated, just as actual values.

Defining access levels (password protection) enables variable modifications to be suppressed. Variables can only be changed, if the access level (0..9) defined for an operating page is below or equal to the currently enabled access level.

2.4.1 STARTING OPERATING PAGES

The programmable controller program writes an operating page number (1..127) to the less significant byte of command word B (W38), bit 0..6.

Bit 7 of command word B (W38) determines at any time, if a PRESET-P value may be changed. If bit 7 is set to logical 0, the variable is exited, if the preset value currently edited is a PRESET-P value. If the bit is already set to 0 during selection, a PRESET-P variable is evaluated just as an actual value (skipped).

2.4.2 TERMINATION OF OPERATING PAGES

Operating pages are terminated by the programmable controller by setting the operating page number of command word C (W38) (bits 0..6) to logical 0.

Terminating an operating page corresponds to closing a page, i.e. a preset value modified by pressing a key is written into the programmable controller (corresponds to pressing the <ENTER> key).

Exiting the operating page is, however, only possible, if a modified preset value has been read twice from the programmable controller data area and if it corresponds to the previously written value *. This enables the programmable controller to detect and to refuse interlocks or min./max. value exceedings (dynamical limit value check). If the preset value is not accepted by the programmable controller and for this reason immediately overwritten, the variable field containing the variable value currently proposed by the programmable controller remains active (underlined). Terminating an operating page is only possible, if the written preset value corresponds to the read out preset value. To inform the user that a certain preset value input is inadmissible, an INFORMATION text can be displayed e.g. which has to be acknowledged (by pressing the <CLR> key for example). This acknowledgement does not affect the preset value, but has the same effect as an interrupt.

However, this is not true for preset values which have been assigned a default value exceeding the limits (represented by inverse fields). In this case, the operating page can be terminated, if editing has not been started yet.

The actual termination of the operating page is detected on the negative edge of bit 0 of word 16 (PCS state).

* Is only valid, if the operating page options are set to the default values using PCS*PRO*^{WIN}.

2.4.3 STUCTURE OF THE OPERATING PAGES

Any structure can be assigned to an operating page (1..127).

When creating operating texts, an optimal operator guidance should be kept in mind. By using the (programmable) special characters ARROW UP, ARROW DOWN, ARROW RIGHT and ARROW LEFT, an unequivocal assignment should be defined (for example character $18 = Arrow Up, $19 = Arrow Down, $1A = Arrow right, $1B = Arrow left).

Additionally, activation of the arrow key LEDs on operating pages (bit 8 of command word B (W37) is set to logical 0) allows an optical guidance of the operator for a certain page. For example, if an arrow key LED is constantly ON, another variable may be selected on the operating page.

2.4.4 VARIABLES ON OPERATING PAGES

When loading a new operating page, the first preset value read out of the programmable controller is first of all underlined („frozen"). If this default value exceeds the specified limits, inverse fields are displayed instead of the preset value. If an editing key (<CLR>, but <+> or <-> are also possible) is pressed once, an admissible value is displayed. If the value stored in the programmable controller is below the minimum value, this is the minimum value and, vice versa, the maximum value.

If the default value has been modified once (except for balancing entry), the input position is marked by a flashing cursor.

If the underline is continuous, the variable is in the so-called CLEAR mode. To enable the EDITING mode, an editing key must be pressed. If the continuous variable types BIT and CSTRING are used, no EDITING mode is available, since data are stored immediately after modification. The balancing entry represents another exception: After calculating the new intermediate result, the variable is completely underlined, although the editing mode is active. After pressing the CLR key, the system switches back to the CLEAR mode and the initial value read out of the programmable controller is displayed again.

Attention!
The flashing cursor only indicates an intermediate result. This means that the currently displayed value does not correspond to the value in the controller!

Numeric values can also be modified during addition or subtraction mode (also called balancing entry): <digit>, <digit>, .. <plus>, possible with BCD(0) and BIN(0)-1,2,A,B. Subsequently, the editor is in the basic state (variable underlined). This also indicates an intermediate result which has not been stored yet!

Storing a preset value and defining the word address (in word 18, bits 8-15) take place during the same cycle. The number of bytes previously written is specified in the least significant byte. If a bit variable is used, 0 is stored and word 19 contains the bit mask of the preset value. After modification of a bit, the bit at the corresponding position is set to 1. Checking word 18 in the programmable controller program enables a preset value change to be detected. After processing of the preset value, word 18 should be set to zero by the programmable controller program.

2.4.5 ARROW KEYS ON OPERATING PAGES

The arrow keys can be used on an operating page to access another variable. If an inadmissible arrow key is pressed and bit 11 of word 37 is set to 0, the audible error signal sounds. Additionally, the admissible arrow keys are displayed, if bit 8 of word 37 is set to logical 0. If an LED is continuously ON, another variable on the same display page may be selected.

If several preset value variables are used on an operating page, access is possible via the arrow keys.

- <ARROW LEFT><-RIGHT>:

  If several preset value variables are used in the text, all operating page lines are supposed to be side by side and the next variable is searched for. If the arrow key LEDs are enabled and another preset variable which can be edited is located in the direction of the arrow, the corresponding LED is constantly ON.

- <ARROW DOWN><-UP>

  If variables are located on several display lines, the first variable (left) of the line which corresponds to the arrow direction is selected. If the arrow key LEDs are enabled, the corresponding LED is ON.

Termination of an operating page can be evaluated using word 16, bit 0. If this bit is set to logical 0, the operating page is disabled. The exact moment of termination is detected by evaluating the negative edge.

2.4.6 ADMISSIBLE KEYS FOR OPERATING PAGES

Operation of the integrated editor		
Variable type	Key	Function
BIT	PLUS	Sets a bit which was set to logical 0 to logical 1 (immediately written into the programmable controller).
	MINUS	Resets a bit which was set to logical 1 to logical 0 (immediately written into the programmable controller).
	* ARROWS	Exits this variable, if allowed. The next variable (in the direction of the arrow) is searched for.
STRING	* PLUS	Increments the value of a variable, if the value is still within the limits.
	* MINUS	Decrements the value of a variable, if the value is still within the limits.
	CLR	Restores the previous display value (previously read from the programmable controller).
	ENTER	Writes the selected value into the programmable controller, if it has been changed and not sent yet.
	* ARROWS	Write the selected value, if it has been changed and not sent yet and search for the next variable in the direction of the arrow.
CSTRING	* PLUS	Increments the value of a variable, if the value is still within the limits (in contrast to STRING it is immediately written into the programmable controller).
	* MINUS	Decrements the value of a variable, if the value is still within the limits (in contrast to STRING it is immediately written into the programmable controller).
	CLR	Restores the previous display value (previously read from the programmable controller)
	* ARROWS	Exits this variable, if allowed. The next variable (in the direction of the arrow) is searched for.

* = Auto repeat

Variable type	Key	Function
BCD-1 BCD-2 BCD0-1 BCD0-2	* PLUS/ MINUS	Addition/subtraction of n within the limits BCD-2 (balancing entry) where: * n = 1, if no number has been entered yet or * n = entered value, if any number has already been entered
	CLR	Restores the previous display value (previously read from the programmable controller).
	ENTER	Writes the selected value into the programmable controller, if it has been modified, but not sent yet.
	* ARROWS	Write the selected value, if it has been modified, but not sent yet and search for the next variable in the direction of the arrow.
	* NUMBERS	
BIN-A BIN-B BIN0-A BIN0-B	* PLUS/ MINUS	Addition/subtraction of n within the limits (balancing entry) where: * n = 1, if no number has been entered yet or * n = entered value, if any number has already been entered
	CLR	Restores the previous display value (previously read from the programmable controller).
	ENTER	Writes the selected value into the programmable controller, if it has been modified, but not sent yet.
	* ARROWS	Write the selected value, if it has been modified, but not sent yet and search for the next variable in the direction of the arrow.
	* NUMBERS	Used for direct entry: numbers are shifted from right to left (even beyond an existing decimal point).
BIN-1 BIN-2 BIN0-1 BIN0-2	* PLUS/ MINUS	Addition/subtraction of n within the limits (balancing entry) where: * n = 1, if no number has been entered yet or * n = entered value, if any number has already been entered
	CLR	Restores the previous display value (previously read from the programmable controller).
	ENTER	Writes the selected value into the programmable controller, if it has been modified, but not sent yet.
	* ARROWS	Write the selected value, if it has been modified, but not sent yet and search for the next variable in the direction of the arrow.
	* NUMBERS	Used for direct entry: numbers are entered as on a calculator.
	(*)POINT	Switches to decimal point places (if defined).

* = Auto repeat; (*) = Auto repeat, abut without any sensible meaning

Variable type	Key	Function
VBIN-A VBIN-B VBIN0-A VBIN0-B	* PLUS	Specifies the »+« sign.
	* MINUS	Specifies the »-« sign.
	CLR	Restores the previous display value (previously read from the programmable controller).
	ENTER	Writes the selected value into the programmable controller, if it has been modified, but not sent yet.
	* ARROWS	Write the selected value, if it has been modified, but not sent yet and search for the next variable in the direction of the arrow.
	* NUMBERS	Used for direct entry: numbers are shifted from right to left (even beyond an existing decimal point).
VBIN-1 VBIN-2 VBIN0-1 VBIN0-2	* PLUS	Specifies the »+« sign.
	* MINUS	Specifies the »-« sign.
	CLR	Restores the previous display value (previously read from the programmable controller).
	ENTER	Writes the selected value into the programmable controller, if it has been modified, but not sent yet.
	* ARROWS	Write the selected value, if it has been modified, but not sent yet and search for the next variable in the direction of the arrow.
	* NUMBERS	Used for direct entry: numbers are entered as on a calculator.
	(*)POINT	Switches to decimal point places (if defined).
WORD	* PLUS	Positions the cursor one bit to the right towards the least significant bit (LSB).
	* MINUS	Positions the cursor one bit to the left towards the most significant bit (MSB).
	CLR	Restores the previous display value (previously read from the programmable controller).
	ENTER	Writes the selected value into the programmable controller, if it has been modified, but not sent yet.
	* ARROWS	Write the selected value, if it has been modified, but not sent yet and search for the next variable in the direction of the arrow.
	* NUMBERS	Only the keys <0> and <1> can be used: <0> Sets a bit to 0 and positions the cursor one digit to the right, if possible. If the cursor reaches the end of the variable, it is positioned on the most significant bit (MSB). <1> Sets a bit to 1 and positions the cursor one digit to the right, if possible. If the cursor reaches the end of the variable, it is positioned on the most significant bit (MSB).

* = Auto repeat; (*) = Auto repeat, abut without any sensible meaning

Variable type	Key	Function
ASCII	* PLUS	Displays the character with the next higher character code. If the end of the character table is reached, the first character of the table is displayed.
	* MINUS	Displays the character with the next lower character code. If the beginning of the character table is reached, the last character of the table is displayed.
	CLR	Restores the previous display value (previously read from the programmable controller).
	ENTER	Writes the selected value into the programmable controller, if it has been modified, but not sent yet.
	* ARROWS	Write the selected value, if it has been modified, but not sent yet and search for the next variable in the direction of the arrow.
	* POINT	Positions the cursor one digit to the right. If the end of the variable is reached, the cursor is repositioned on the first character of the variable.
	* NUMBERS	Used to directly enter numbers and advance to the next input position.
WORD-KH	* PLUS	Increments the digit by 1 the cursor is positioned on
	* MINUS	Decrements the digit by 1 the cursor is positioned on
	CLR	Restores the previous display value.
	ENTER	Writes the selected value into the PCS, if it has been modified, but not sent yet.
	* ARROWS	Write the selected value, if it has been modified, but not sent yet and search for the next variable in the direction of the arrow.
	* NUMBERS	Used to directly enter numbers (0..9) on the corresponding digit.
	* POINT	Moves the cursor from left to right. If the rightmost cursor position is reached, the cursor is repositioned on the left digit.
WORD-KY	* PLUS	Increments the digit by 1 the cursor is positioned on
	* MINUS	Decrements the digit by 1 the cursor is positioned on
	CLR	Restores the previous display value.
	ENTER	Writes the selected value into the PCS, if it has been modified, but not sent yet.
	* ARROWS	Write the selected value, if it has been modified, but not sent yet and search for the next variable in the direction of the arrow.
	* NUMBERS	Used for direct entry: Numbers are shifted from right to left (calculator entry).
	* POINT	Switches between high byte and low byte of the data word (decimal format).

* = Auto repeat

All functions mentioned here are only valid, if the operating pages options are set to the default parameters.

2.5 ACCESS LEVELS/PASSWORD PROTECTION

10 access levels (0..9) which are protected by a password (4-digit number) can be configured in the PCS 950. Level 0 is always active, if no password has been entered or configured. Each operating page can be assigned to an access level. After loading the corresponding operating text, the PRESET and PRESET-P variables contained herein can only be modified if at least the access level to which this operating text is assigned is currently active.

The password dialogue used to activate a new access level is called via the internal softkey function 256 (LOGIN). The configured password text is then displayed in the working area and used for a hidden password entry. The internal variable [PASSWORD] must be configured for this purpose. If a wrong password is successively entered three times, the dialogue is automatically disabled. If the correct password is entered, the access level corresponding to the password is activated. The internal variable [PASSWORD_LEVEL] contains the active access level (0..9) which can be deactivated using the internal softkey function 257 (LOGOUT). After calling this function, access level 0 is active.

2.6 PRIORITY MANAGEMENT

In the PCS 950, 8 priority levels are available and several priorities can be simultaneously activated. The highest activated and enabled priority is authorized to display texts and to process key codes. If a priority is deactivated or disabled, authorization is reassigned. Disabled priorities are only used in the background (e.g. message bits are nevertheless evaluated and events are stored in the history and printer message buffer).

The behavior of the PCS can be derived from the PCS status which is available in the transfer area of the programmable controller (words 16 and 17) and from the interlocking bits in command word A (W36):

- Word 16 (bits 0..7) contains all active priorities, even if they are currently disabled and thus not displayed.
- Word 17 (bits 12..15) contains the currently displayed, i.e. the most significant activated and enabled priority.
- In command word A (W37, bits 8..11) all priorities (except for the idle priority) can be individually suppressed at any time. For example, this feature can be used to inhibit that an operating page is interrupted by an information, a warning or a failure text, as long as this page is active.

Pressing the <HELP> key does not change priorities. As long as this key is pressed, assignment of authorization to a certain priority is not modified.

Priorities can be interlocked at any time in the programmable controller. It must, however, be noticed that this may cause unwanted actions during operation (preset value input, advancing of messages or history). For this reason, a blocking time of 0.5 seconds is Priorities can be interlocked at any time in the programmable controller. It must, however, be noticed that this may cause unwanted actions during operation (preset value input, advancing of messages or history). For this reason, a blocking time of 0.5 seconds is initiated, when priorities are changed. During this time, control or numeric key entries are ignored and the audible signal sounds when a key is pressed (prerequisite: bit 11 of word 37 = 0).

2.6.1 ON/OFF CONDITIONS

The conditions for switching ON and OFF priorities are described below:

lowest		0	=	IDLE TEXT	{idle texts 0..127}
:				Displayed, if no higher priority is active.	
:	*)	1	=	OPERATING PRIORITY	{operating texts 0..255}
:				(De)activated by the pr. controller (word 38 bits 0-7).	
:	*)	2	=	INFORMATION TEXTS	{message texts 0.1023}
:				Activated by a 0 -> 1 transition of at least one message	
:				bit to which a text with INFORMATION priority is assigned.	
:				Deactivated according to the deletion procedure selected	
:				for the corresponding INFORMATION message text.	
:	*)	3	=	WARNING TEXTS	{message texts 0.1023}
:				Activated by a 0 -> 1 transition of at least one message	
:				bit to which a text with WARNING priority is assigned.	
:				Deactivated according to the deletion procedure selected	
:				for the corresponding WARNING message text.	
:	*)	4	=	FAILURE TEXTS	{message texts 0.1023}
:				Activated by a 0 -> 1 transition of at least one message	
:				bit to which a text with FAILURE priority is assigned.	
:				Deactivated according to the deletion procedure selected	
:				for the corresponding FAILURE message text.	
:		5	=	HISTORY DISPLAY	{History texts 0.1023}
:				Activated by setting bit 1 of word 36 and deactivated by	
:				resetting bit 1 of word 36.	
:		6	=	Currently not used.	
:	**)	7	=	ERROR PRIORITY	{fixed text}
:				Activated by interface or start test errors and deactivated	
:				in most cases by a programmable controller RESET command	
:				STOP/RUN transition or restart of the progr. controller	
:	*)	8	=	OFFLINE MENU activated by HELP + CLR	{fixed text}
highest				(Can be interlocked during communication by bit 15 of word 36)	

**) With most drivers, this error is indicated in error word W3 of the programmable controller. The error word type depends on the specific driver being used and must be looked up in the corresponding PCS 091.x manual.

*) Always available, if the programmable controller communication is not established!

2.6.2 IDLE TEXT PRIORITY

This priority class (0) includes the idle texts from 0 to 127. The programmable controller determines which of these idle texts (bits 8..14 of word 38) is displayed and if the idle text is flashing (bit 15 of word 38). The numeric and control keys are ineffective in this case. If they are nevertheless pressed, the audible error message is suppressed and the control keys can be used for control purposes. The <HLP> key enables display of a defined help text for the idle priority. PRESET-, PRESET-P- and ACTUAL value variables can be used, but PRESET values cannot be entered. All variables are cyclically refreshed. IDLE TEXT 0 is assigned a special function: it is displayed immediately after the PCS is switched on, even if communication with the programmable controller has not been started yet. If the idle text 0 contains a variable, it is replaced by spaces, until the variable can be read out of the programmable controller. This is a comfortable method used to detect communication start. If a non-defined idle text is selected, the idle text previously displayed remains active.

2.6.3 OPERATING PRIORITY

127 operating pages are available within this priority class (1).

Loading an operating page requires use of command word C (W38), bits 0..6.

Before an operating priority can be started, an operating page must be specified.

Bit 7 of word 14 determines, if PRESET-P variables may be changed. If bit 7 is set to logical 0, only PRESET value variables can be modified. If bit 7 is set to logical 1, PRESET and PRESET-P variables can be changed. This bit can be modified at any time by the programmable controller.

2.6.4 MESSAGE PRIORITIES

In these priority classes (2, 3 and 4) texts are called by setting a bit in the message area (words 41 to 104 (maximum)). A MESSAGE TEXT with a maximum of two lines is assigned to each of the 1024 bits. For each of the 128 texts, an individual MESSAGE PRIORITY (specified during programming) can be specified. These priorities are described below:

- INFORMATION TEXT PRIORITY (priority 2)
- WARNING TEXT PRIORITY (priority 3)
- FAILURE TEXT PRIORITY (priority 4)

These priority classes only differ with regard to the priority level (functions are identical). Setting a message bit to which no message text, log text or message printer text is assigned does not show any effect.

The system normally attempts to record the time in the correct order when transitions are detected. To retain the timing for machine operation, messages are fetched in blocks, each containing 128 bits. If several bits are set during one cycle, the lower text numbers within the blocks is assigned a higher priority.

2.6.4.1 DELETION PROCEDURE

The deletion procedure can be specified for each individual message bit in PCSP*RO*^{WIN}. 5 deletion procedures are used:

Deletion procedure 1 (deletion by the programmable controller):
The text remains active as long as the corresponding bit is set to 1. If the programmable controller resets this bit, the message text is deleted. The bit is only read by the PCS. The operator prompt LED (?) is OFF.

Deletion procedure 2 (manual deletion with message bit reset):
The text is displayed by a 0 -> 1 transition and can be acknowledged with <CLR>. This resets the message bit in the programmable controller and therefore removes the text from the display.
Resetting the message bit by the programmable controller corresponds to using the <CLR> key function.
If this deletion procedure is used, the message bit may only be set once in the programmable controller program (no continuous assignment!), otherwise the message is repeatedly displayed after pressing <CLR>. The operator prompt LED (?) is immediately switched OFF, after <CLR> has been pressed.

Deletion procedure 3 (manual deletion without message bit reset):
The text is displayed by a 0 -> 1 transition and can be removed at any time. This must, however, be acknowledged with <CLR>, independently of the message bit status.
The message bit (in the programmable controller) must be reset by the programmable controller program.
The operator prompt LED (?) is immediately switched OFF, after <CLR> has been pressed.

Deletion procedure 4 (manual deletion if the message bit is set to 0):
The text is displayed by each 0 -> 1 transition and can only be removed with <CLR>, if the message bit is set to 0 by the programmable controller.
The message bit status is indicated by the operator prompt LED (?):
Flashing: The bit is set lo logical 1, deletion is impossible.
ON: The bit is set to logical 0, deletion is possible.

Deletion procedure 5 with acknowledgement:
In case of a rising edge, the text flashes and is underlined.
Flashing can be removed by pressing <CLR>.
Deletion is only performed, if the bit is reset to 0 and <CLR> has been pressed.

2.6.5 HISTORY DISPLAY

Up to 128 messages can be stored in the PCS 950. For display and printing of texts, two different buffers are used. Data are stored in the display buffer, as soon as COME, GONE and ACKNOWLEDGED times are determined (only deletion procedure 3, 4 and 5). Data in the two buffers can be overwritten at any time, i.e. entries are always possible. An uninterrupted representation of stored messages can, however, not be guaranteed. The variable [HISTORY_EINTR] (history entry) which assigns a consecutive number from 0 to 65535 to each entry is used to facilitate the analysis.

HISTORY TEXTS are entered into the buffer. If specific history texts are not desired, the variables [MLDTXT_ZEILE1] and [MLDTXT_ZEILE2] (message text lines 1 and 2) are used to refer to the corresponding message text line. Creating a HISTORY RANGE TEXT thus enables generation of a general form for all message texts.

2.6.5.1 ACTIVATION OF THE HISTORY DISPLAY

Activation of the history display requires several conditions to be met:

- Bit 12 of word 36 (interlocking bit) must be set to 1.
- Bit 1 of word 36 (activation bit) must be set to 1.
- At least one message must have been entered into the buffer. This is indicated by bit 11 of word 13: if this bit is set to logical 1, messages are stored in the log buffer.

  History texts are displayed in one (large characters) or two lines (small characters). If the working area contains more than two lines, several history texts can be displayed within the activated priority. Alternatively, the history texts can be listed from top to bottom according to the „FIFO“ or „LIFO“ representation (determined by the PCSPROWIN software):

LIFO: The last history text which has been activated (recent text) is immediately displayed in the history list box as topmost entry. It is displayed with an inverse background, i.e. it is active. All previous history texts are listed below, if not inhibited by the working area capacity. If the listing is complete and the programmable controller does not activate another history text, the ARROW UP and ARROW DOWN keys can be used to page through the history list box.

FIFO: The first history text which has been activated, is displayed in the history list box as topmost entry. It is displayed with an inverse background, i.e. it is active. All more recent history texts are listed below, if not inhibited by the working area capacity. If the listing is complete, the ARROW UP and ARROW DOWN keys can be used to page through the list. New history texts are appended to the last entry and are, in contrast to LIFO, only visible after being selected.

On history displays, the following functions are assigned to the arrow keys:

- Arrow Left: Activates the topmost list box entry. This enables the oldest (FIFO) or recent (LIFO) history text to be selected. Afterwards, the selecting bar is positioned on the beginning of the list box.
- Arrow Right: Activates the bottommost list box entry. This enables the recent (FIFO) or oldest (LIFO) history text to be selected. Afterwards, the selecting bar is positioned on the end of the list box.
- Arrow Up: Activates the next list box entry positioned above the current entry. This enables the more previous (FIFO) or more recent (LIFO) history text to be selected. If the selecting bar is positioned on the beginning of the list box, the history texts are scrolled in the list box one position towards the end of the list. If available, a history text is moved to the position of the selecting bar (before scrolling, the text was positioned above the visible list box).
- Arrow Down: Activates the next list box entry positioned below the current entry. This enables the more recent (FIFO) or more previous (LIFO) history text to be selected. If the selecting bar is positioned at the end of the list box, the history texts are scrolled in the list box one position towards the beginning of the list. If available, a history text is moved to the position of the selecting bar (before scrolling, the text was positioned below the visible list box).

  The arrow key LEDs are ON (function which can be deactivated), if the corresponding key is effective.

  For each history text within the message priority, a help text with up to 32 lines and referring to the currently selected entry can be created. The arrow key LEDs indicate (function which can be deactivated), if paging within the help text is possible.

2.6.5.2 VARIABLES IN THE HISTORY BUFFER

The following variables are stored:

- All external variables (from the programmable controller)
- Text number [TEXTNUMMER]
- Identification number [HISTORY_EINTR]
- Date and time came [ZEIT_MLD_KOMMT]
- Date and time acknowledged [ZEIT_MLD_QUITT]
- Date and time gone [ZEIT_MLD_GEHT]
- [MLDTEXT_ZEILE1] and [MLDTEXT_ZEILE2] (message text, lines 1 and 2)

Spaces indicate variables that cannot be displayed.

2.6.5.3 ERASING THE HISTORY BUFFER

The rising edge of bit 12 of word 37 (command word B) enables the history buffer to be erased. The identification numbers [HISTORY_EINTR] recommence with 0. Bit 12 of word 13 indicates when the erase job was detected by the PCS.

2.6.6 COMMUNICATION ERROR PRIORITY

The priority level (7) described in this section is controlled by the programmable controller driver. The associated texts cannot be modified. The header is identical for all drivers. The second line is explained in the corresponding driver manual.

This priority is enabled, if an already established communication with the programmable controller is interrupted.

```
===== COMMUNICATION ERROR =====
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

Deactivation of this priority is only possible by restarting communication at the programmable controller.

Warning!
The PCS actions which have not been performed yet due to communication interruption are transmitted into the programmable controller after restart. Check the correct action/reaction of the PCS and the progr. contr. after restart!

2.6.7 OFFLINE MENU

This priority level (8) can be accessed with the key combination <HELP> + <CLR>, if communication has been established and the priority has been enabled by setting bit 15 of word 36 (command word A) logical 1). This priority prepares access to the OFFLINE MENU which can only be used if communication is NOT ESTABLISHED. A security inquiry (fixed text) which has to be confirmed with <ENTER> is then displayed. Pressing any other key deactivates this priority.

Warning!
Confirmation with ENTER aborts communication immediately. Check action/ reaction of the programmable controller. Malfunctions in the programmable controller are possible. After exiting the OFFLINE MENU, a restart is performed. Additionally, restarting communication at the programmable controller is required!

2.7 PRINTER

The PCS 950 printer function is used for the following tasks:

- OPERATING PRINTER

  255 different printer texts with any variables can be numerically selected with this function via word 40. The text is output as soon as possible. This selection is not stored.

- MESSAGE PRINTER

  Messages are stored (in a similar way as for the history buffer) which can be printed later (but also continuously). Currently, 1024 messages can be stored in the buffer which is non-volatile. These messages can be printed in the forward or backward direction or multiple times via a programmable controller command. The relevant control addresses are word 37 (command word B), word 39 (command word D) and the status in words 13, 14 and 15.

The priority assigned to the OPERATING PRINTER is higher than the priority of the MESSAGE PRINTER. This means that an OPERATING PRINTER job is performed even between individual MESSAGE PRINTER pages.

2.7.1 PRINTER TEXTS

The OPERATING and MESSAGE PRINTER texts may contain all characters except H00. Within PCS*PRO*^{WIN}, an entry line may contain up to 132 characters and number of lines is limited to 126. The actual printer line length is determined by the printer control characters which have to be inserted explicitly.

The pseudo variables <TAB, <LF>, <FF>, Fe+>, Fe->, <Un+> and <Un-> can be used to simplify operation. They represent 8 strings (hex format) which can be individually edited and contain a maximum of 8 characters.

- VARIABLES IN MESSAGE PRINTER TEXTS

  The following variables are stored:

  - [TEXTNUMMER]: Message number
  - [DRUCKER_EINTR]: Identification number between 0 and 9999
  - [ZEIT_MLD_KOMMT]: Time when message was displayed.
  - [ZEIT_MLD_GEHT]: Time when message was removed.
  - [ZEIT_MLD_QUITT]: Time when message was acknowledged.
  - [MLDTXT_ZEILE1] and [MLD_TXT_ZEILE2] (message text, lines 1 and 2)

  The other printable variables are read during printout. Non-printable variables (e.g. variables in message text lines) are represented as spaces with a specific length.

- VARIABLES IN OPERATING PRINTER TEXTS

  All programmable controller variables and internal variables that do not refer to messages may be used here.

2.7.2 MESSAGE PRINTER

2.7.2.1 MESSAGE ENTRY

An entry is always assigned to messages and can be suppressed via bit 14 of word 37 (command word B). If this bit is set to 1, the entry is ignored. Feedback of this bit status is possible via bit 14 of word 13.

Several attributes enabling customizing of the printer output are assigned to each message printer text.

- CAME

  The text is entered, as soon as a rising edge of the message bit is detected. In these texts, only the CAME time is efficient.

- GONE

  The text is entered, as soon as a falling edge of the message bit is detected (independently of the deletion procedure). The variable [ZEIT_MLD_QUITT] (time when message was acknowledged) is not effective or valid for all deletion procedures.

- ACKNOWLEDGED

  The text is entered, when the <CLR> key is pressed (only with deletion procedure 2..5). The variable [ZEIT_MLD_GEHT] (time when message was removed) is possibly not valid yet.

An attribute must not necessarily be selected. In this case, message entry is suppressed. If several attributes are selected, the same text is entered several times.

Background logging requires the following conditions to be met:

- No displayable message text may be configured (this number is also inadmissible for the range text).
- The attribute CAME or GONE must be selected in the message printer text and only the variable [ZEIT_MLD_KOMMT] or [ZEIT_MLD_GEHT] is valid.

If only the CAME attribute has been selected, the corresponding message bit is reset in the programmable controller by the PCS itself if background logging is enabled.

Orientation within the log buffer is possible by storing the number of the previous messages in W14 and the number of recent messages in W 15 of the programmable controller. The number of messages currently stored in the message log buffer is calculated by adding these two values. These values are independent of the print direction. If bit 13 of command word B is set to logical 0 (FIFO principle), word 15 contains the number of words which still have to be printed. If bit 13 of command word B is set to logical 1 (LIFO principle), word 14 contains the number of messages which still have to be printed.

- CIRCULAR BUFFER

  If the message log buffer is full (1024 stored messages), each new entry overwrites the oldest entry. If the overwritten entry corresponds to the entry which currently has to be printed, the current printout is completed and the system continues with the next possible entry - according to the print direction - or terminates the print job. Uninterrupted logging can only be detected via the variable [DRUCKER_EINTR].

* Within a message printer text or history text, variables can also be used. The message text and the currently valid programmable controller variables are entered together into the corresponding circular buffer.

The maximum number of messages which have been entered now depends on the number of variables per message text. A total of 4 KByte of history buffer and 20KB of message printer buffer is generally available. The buffer is dynamically allocated during runtime. A message which has to be entered always requires a data header of 22 bytes plus the total number of bytes for all variable types inserted into the message text (2 bytes per word; 4 bytes per double word; ASCII: number of bytes depends on the length). These are used for the data body. If a history/message printer texts contains for example 8 text lines with 3 double word variables in each of them and one text line with an ASCII variable comprising 18 characters, this requires in the corresponding circular buffer 3*4*8 bytes plus 2*9 bytes for the ASCII variable plus 22 bytes for the header, a total of 136 bytes = 68 words. So the following equation is valid: Data header (= 22 bytes) plus total of data bytes in the data body of a message text is equal to the required space in the history or message printer buffer. A maximum of 128 messages can be stored in the history circular buffer and a maximum of 416 messages in the message printer buffer.

2.7.3 OPERATING PRINTER

If an operating printer is selected data are directly transferred to the printer if the operating printer text number exceeds 0 in command word E (word 40). If the job is completed the PCS 950 resets this word to 0. The operating printer is activated at the end of a message if an operating printer text has to be output while the message printer is active. Afterwards, message printer operation continues.

2.7.4 PRINTER OUTPUT

There is a difference between CHARACTER and PAGE logs. Since a printer is not always ready to accept characters, a character oriented handshake procedure must be used for the interface. Page logging is performed by the programmable controller. In this case, log entries to be printed are selected according to the messages.

IMPORTANT : EACH PRINTER OUTPUT MUST BE TRIGGERED BY THE PROGRAMMABLE CONTROLLER.

2.7.4.1 LOW LEVEL PROTOCOLS

For low level protocols, two procedures are used:

- XON/XOFF (when using RS 232 and TTY)

  The printer sends XOFF, if it is not ready and XON, if is ready. After a cold restart, the PCS supposes that the printer is ready. Since removing the cable during XOFF phase may permanently prevent the PCS from receiving any other XON signal, difficulties may occur when using printers which are temporarily connected. Some printers offer (as a remedy) a so-called PERMANENT XON, i.e. XON is repeated every two seconds if the printer is ready.

  XON/XOFF offers the advantage that no additional handshake lines are required. This enables a TTY operation. Sine electrical isolation is possible in this case, this procedure features a higher noise immunity than RS 232.

- RTS/CTS (only when using RS 232)

  The printer output RTS is connected to the CTS input of the PCS 950 printer interface. (For some printers, this signal is also referred to as BUSY signal.) Printing is enabled, if the signal is HIGH. Printer output is stopped if the printer cable is removed.

The additional control line requires the RS232 configuration (without electrical isolation) to be selected. The following signals are required in the PCS:

TXD	Transmitted data
CTS	Handshake
GND	0 Volt (identical to a supply voltage of 0 Volt)
SCREEN	identical to enclosure (not 0 Volt !)

2.7.4.2 PAGE LOG (MESSAGE PRINTER)

In this case, you must send the print job, specify the print direction, enable the print job and, if necessary, adjust the printer pointer. The entire printing process can thus be customized via a few programmable controller program lines. Word 37 (command word B), word 39 (command word D), and word 40 (command word E) are available as tools:

■ ABORT OF THE PRINT JOB (W 37 BIT 15)

If set to logical 1, this bit aborts each current print job as well as the reference point adjustment. The command word D (word 39) is set to zero by the PCS and any further action is inhibited.

■ PRINT DIRECTION (W 37 BIT 13)

This bit determines the print direction and the reference point adjustment:

- log. 1 (LIFO) The printer outputs messages in the direction of the previous messages. Printer pointer commands (negative value in command word D = word 37) are directed towards recent messages.
- log. 0 (FIFO) The printer outputs messages according to the entry order (direction of the recent messages). Printer pointer commands (negative value in command word D = word 37) are directed towards older messages.

The PCS only reads this bit in connection with a new print job/printer pointer command. Intermediate modifications are ignored.

■ PRINT JOB / PRINTER POINTER COMMAND (word 39)

A signed value must be written into this word to trigger a corresponding action. If the job is completed or the log buffer end reached, the PCS resets this value to zero. This is, however, only effective, if bit 15 of word 37 (command word B) is set to logical 0.

■ POSITIVE VALUE

A positive value triggers a print job. The value range is from +1 to +32767 (KH0001 to KH 7FFF).

■ NEGATIVE VALUE

A negative value is used to adjust the printer pointer. The direction is determined by BIT 13. The value range is from -32767 to -1 (KH8001 to KHFFFF).

- -32768

erases the entire log buffer (KH8000). After erasing, 0 new and 0 old messages are specified by W14 by W15.

Important!
Each print job and each printer pointer adjustment are IMMEDIATELY terminated, if the PCS is switched OFF!

2.8 TIMER

8 timers are available in the PCS 950 (word 13, bits 0..7). Each of these timers is equipped with up to 8 cams and each cam is assigned a switching ON and a switching OFF instant. These instants are repeated every day. Hours, minutes and seconds are specified. The 8 cams are ORed in a bit and transmitted as a complete byte. Transmission only takes place after modifications and during communication restart. A certain presetting is required in the programmable controller which becomes valid during communication failures. The switching ON/OFF instants can be preset via PCS*PRO*^{WIN}. If the corresponding instants are used as preset values on operating pages (variables [NOCKEN_NUMMER] (cam number), [ZSU_EIN/AUS_STUNDE/MIN./SEK.] (timer ON/OFF: hour/minute/second)), they can also be modified ONLINE. The modified values remain in non-volatile RAM. The following applies to the values:

- SWITCHING ON INSTANT = SWITCHING OFF INSTANT
  This cam is ignored.
- SWITCHING ON INSTANT < SWITCHING OFF INSTANT
  This cam is set to logical 1 during the day. The latest possible instant for switching OFF is 23:59:59 h.
- SWITCHING ON INSTANT > SWITCHING OFF INSTANT
  This cam is active from the switching ON instant beyond midnight until the switching OFF instant of the following day. Since the conditions are continuously calculated, setting the clock becomes immediately effective, when terminating the operating page.

Attention :

This is also valid when the switching ON and OFF instants are changed, i.e. modifying the switching ON and OFF instants may cause the cams to be immediately set/reset. To define an unequivocal switching ON and OFF instant, the cams should not be evaluated when the configuration operating page is active.

2.9 ALARM OUTPUT/ALARM CONTACT

A floating relay contact is available between connections 6 and 7.

Only low voltage (24V) and 0.5A may be applied to this contact.

It is closed, if all of the following conditions are met:

- The PCS 950 is switched ON and supply voltage is applied.
- Communication with the programmable controller is enabled.
- At least one message with the attribute ALARM CONTACT is enabled.
- The enable bit (bit 3 of word 36 (command word A)) is set to logical 1.

If one of these conditions is not met, the contact will not be closed.

2.10 REAL TIME CLOCK

Normally, time and date are available in words 9 to 12 of the programmable controller. If the programmable controller is equipped with its own real time clock, synchronization with the PCS clock is possible. For this purpose, bits 6 and 5 of word 36 must first be set to 1 (inhibit transmission). If a 0->1 transition of bit 0 is detected, the PCS reads time, date and day of week one scan later from the programmable controller, sets the internal clock appropriately and resets bit 0 of word 36.

2.11 COUNTER FOR OPERATING HOURS

The counter for operating hours to which the internal variable [BETR_STD_IST] is assigned, outputs data which are stored in non-volatile RAM. When switching on the device, the counter is checked for plausibility. If this check fails, a failure message is displayed after switching on. If necessary, the operating hours can be adjusted via the interval variable [BETR_STD_SOLL]. The counter for operating hours is active only during communication with the programmable controller and if bit 2 of word 36 (command word A) is set.

3 DRIVING THE PCS 950

Driving of the PCS 950 is effected by a transfer area comprising a maximum of 256 words which are located in the programmable controller.

The programmable controller controls all PCS functions and accesses these data by „write“ and „read“ operations via these words.

Communication between the programmable controller and the PCS is controlled by the handling software supplied with the corresponding PCS 91.xxx manual. The communication processor (PCS 810-1 for example) and the operating system of the programmable controller and the PCS are responsible for data storage and the communication log.

Since the parameters of the transfer area (its length, capacity and functionality for example) depend on the driver being used, it is recommended to use the corresponding „driver manual“ PCS 91.x.

Basically, two data exchange procedures are available.

The expander driver (e.g. L1) represents the first procedure for data exchange. Data are exchanged via a sending and receiving area. The PCS writes „jobs“ into this area and reads response data. To execute these „jobs“ stored in the transfer area, an expander program is required in the programmable controller. This program may considerably affect the scan time. Reduced data volumes are exchanged with a high communication reaction speed. Access to the transfer area is possible at any time (except for interrupt programs). This procedure can be used for programmable controller systems with indirect addressing and if scan time of these controllers is negligible.

The direct driver (e.g. AS511) represents the second principle. This type of driver writes and reads data directly into/from the transfer area of the programmable controller. For this reason, an expander program is not required in the programmable controller. In this case, large data volumes are exchanged with a low communication reaction speed. Since data exchange is asynchronous with regard to the programmable controller scan time, only a synchronized transfer area access is admissible in the programmable controller program. Synchronization takes place using a transfer area word and the programmable controller scan time is only affected to a small degree. This procedure is used for systems without indirect addressing and in which the scan time may not be adversely affected.

Warning!
Use only the appropriate driver for the programmable controller being used, otherwise malfunctions may occur in the PCS and in the programmable controller!

System area: W0..3

W0..2	Internally used, disabled for the user.
W3	Communication error word. For details, see the corresponding „driver manual" PCS 091.x.

2. Status area: PCS status (written into the programmable controller)

KEYS:

W4	Key bits <F1..F8>, <F9..10>, <ARR_DOWN>, <ARR_UP>, <ARR_RIGHT>, <ARR_LEFT>, <->, <+>
W5	<F11-F20>, <CLR>, <ENTER>, DIL 4-1 <HELP>, <.>, <9..0>, Reserved
W6-W7	Rest

TIME AND DATE

W9-12	Year, month, day, day of week, hour, minute, second

PCS status

W13-17	Acknowledge bits, timer, (W14) number of old printer messages, (W15) number of new printer messages, (W16), priority state, (W17) numerical priority, text number on the display.

PRESET VALUE STATUS

W18-19	Data word number, length, (W19) bit mask

3. Command area: (read out of the programmable controller)

LED STATUS, DISPLAY and MEMORY MODE

W20	LED driving, F1..F10, green + display modes messages,
W21	LED driving, F1..F10, flashing green + display modes messages
W22,23	LED driving, F1..F10, yellow, flashing yellow
W24,25	LED driving, F11..F20, green, flashing green
W26,27	LED driving F11..F20, yellow, flashing yellow, the remaining are reserved
W28..33	Reserved
W34	Bit map number for working area
W35	Status page and softkey row

COMMAND WORDS

W36	Enable priority + inhibit transmission + activation bits
W37	Printer driving, inhibit specific LEDs + message block transfer
W38	Idle text number + operating text number
W39	Print job
W40	Operating printer text number

4. Message area: W41...110

MESSAGE BLOCKS

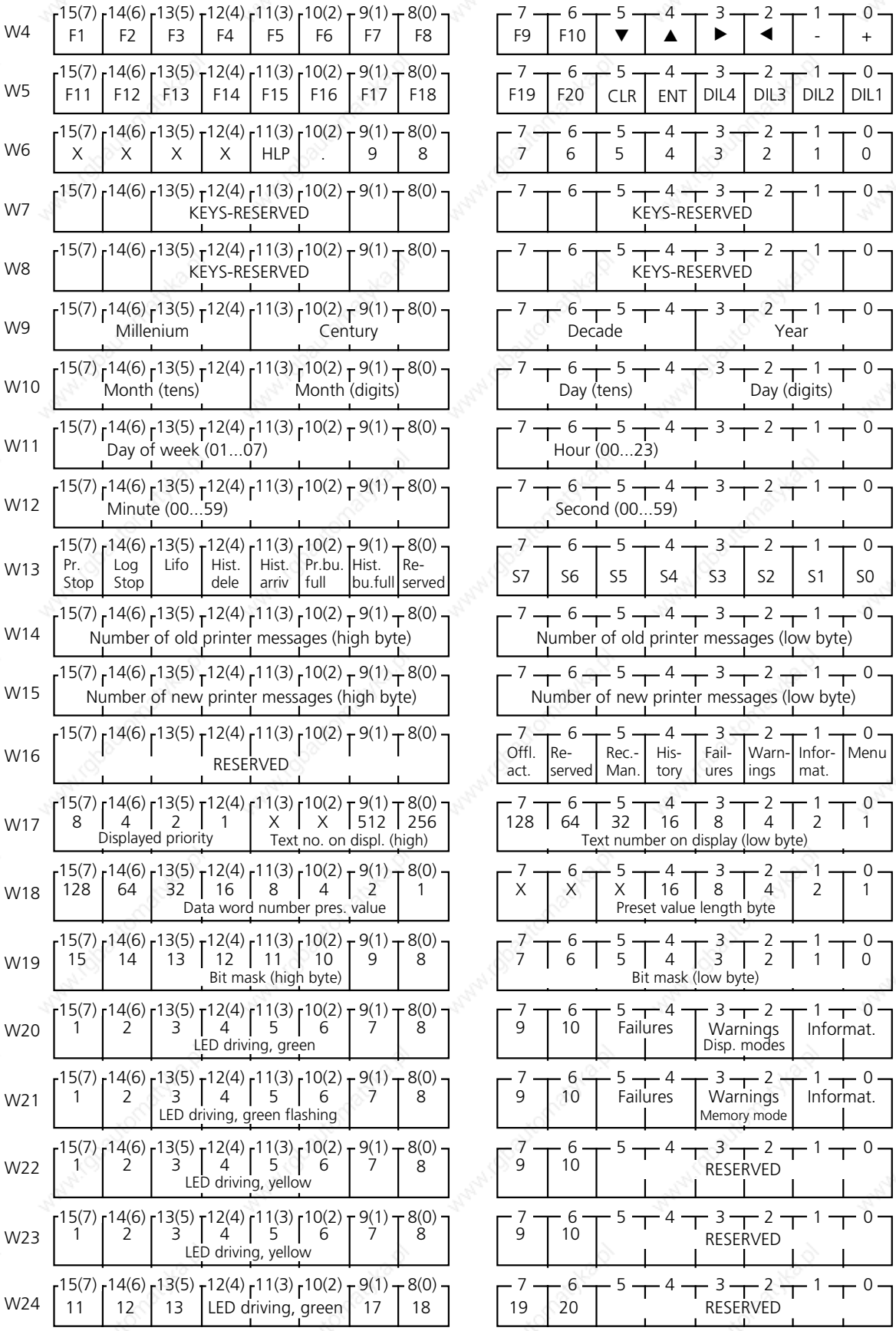
W41-48	Block 1
W49-56	Block 2
W57-64	Block 3
W65-72	Block 4
W73-80	Block 5
W81-88	Block 6
W89-96	Block 7
W97-104	Block 8

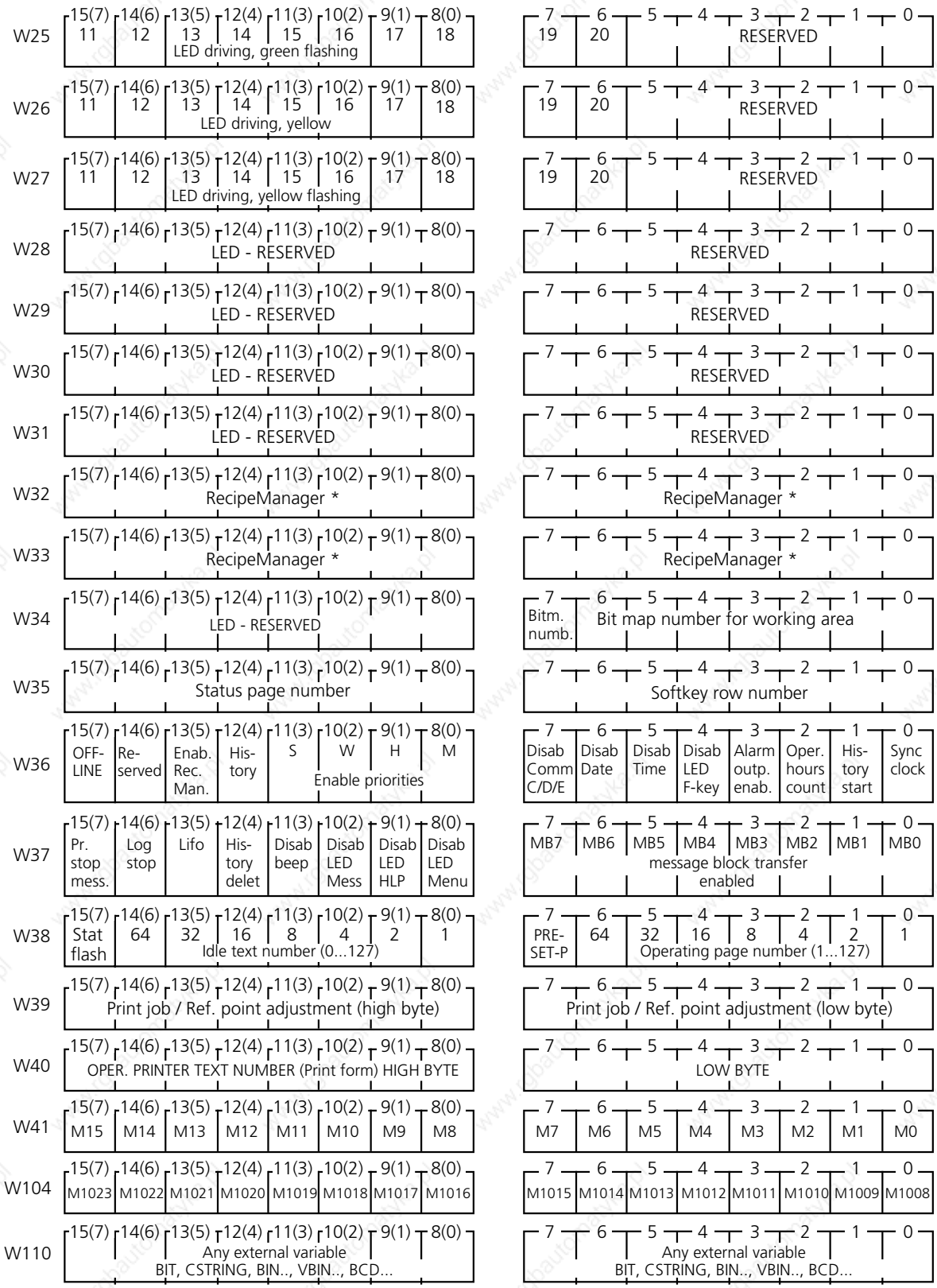
5. Expansion area: W105..W109
This area is reserved for possible enhancements.

6. Variable area: W 110..255

W110..255	Reserved for variables

3.1 BRIEF OVERVIEW OF THE TRANSFER AREA

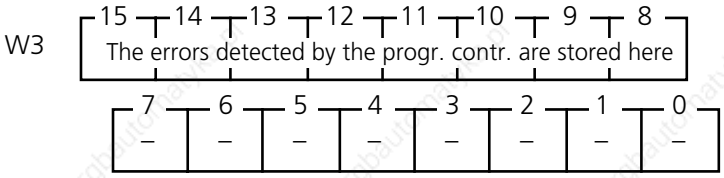




* in preparation

3.2 SYSTEM AREA

Words W0..2 are reserved for functions which depend on the driver.



The errors detected by the programmable controller are stored in the HIGH byte of word W3. Since these errors depend on the driver, use of the PCS 091.x manual is recommended. If a communication error occurs, this word indicates the error cause. These data have to be at hand when asking for support information at the telephone.

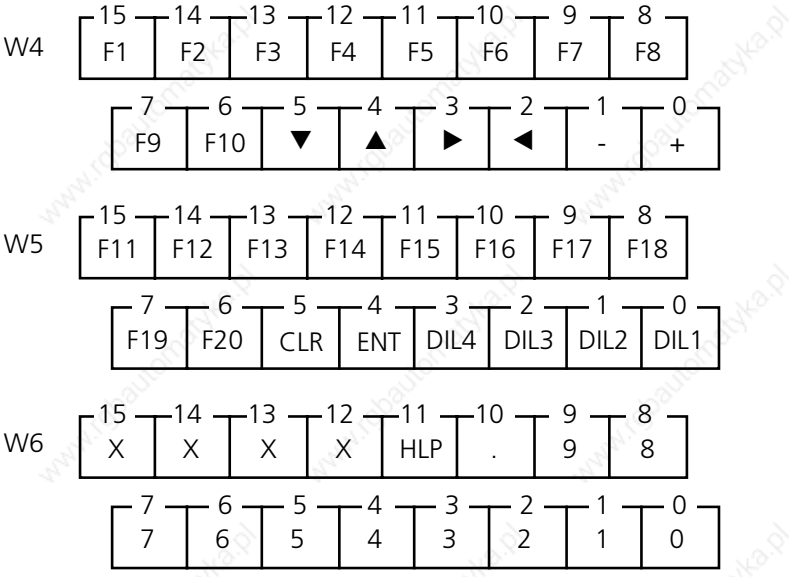
This error aborts communication (by the programmable controller). The system only tries to reestablish communication after a COMMUNICATION RESET initiated by the programmable controller. Operation continues in the PCS where it was interrupted. Information may be lost due to execution of an (application specific) preset and emergency program stored in the programmable controller. If desired, this can be inhibited by appropriate programming.

Warning!
Check the action/reaction in the PCS and in the programmable controller to avoid malfunctions after a programmable controller restart.

3.3 STATUS AREA

The key bits, time, date and the PCS status are written into this area by the PCS.

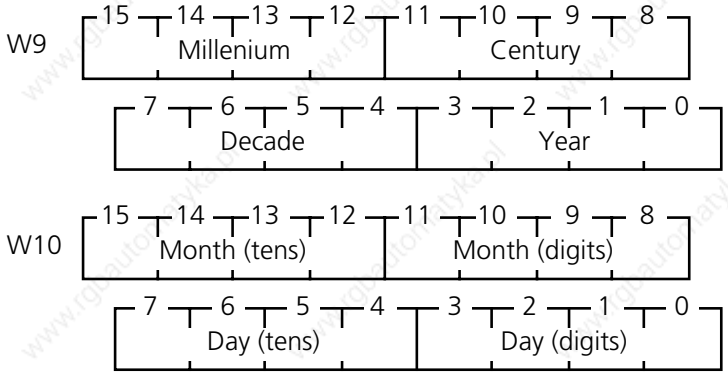
3.3.1 FUNCTION KEYS, CONTROL KEYS AND NUMERIC KEYS



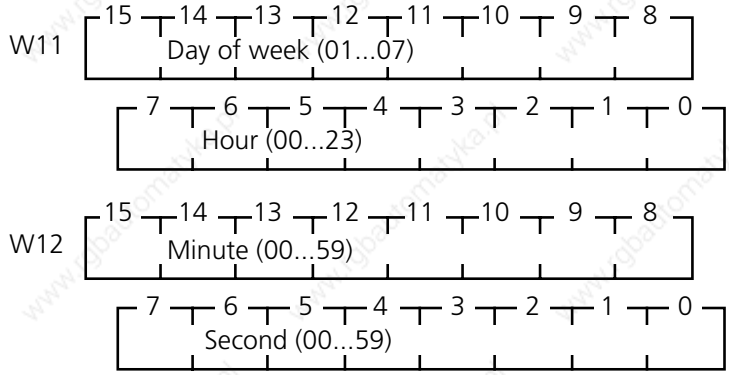
These key bits are set to logical 1 as long as the corresponding key is pressed and communication takes place without errors. Care should be taken when using the control keys of the LOW byte of W5, since they are also required in different priorities (operating page priority for example) to edit preset values.

Words 7 and 8 are reserved for the keyboard extension (PCS 891).

3.3.2 DATE AND TIME



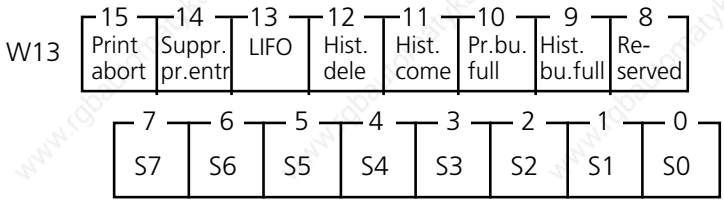
The date is represented in the BCD format. It is, however, only valid, if the millennium is larger than 0. For this purpose, the millennium should be preset using KH0000 in the restart program. If the year is below 93, 20xx is presupposed.



These contents also are only valid, if W9 <> 0. Although a seconds clock pulse can be derived from the seconds, it must be noticed that signalling of a new time may be delayed up to one second (especially with long programmable controller scan times). Apart from that, the contents are always valid, since the PCS always refreshes words W9..W12 during one cycle.

3.3.3 PCS STATUS

3.3.3.1 CONTROL BITS



The function of the individual bits is explained below:

0-7: Timers

9: The history buffer for history variables is full, i.e. all messages in the circular history buffer are replaced by new ones.

10: The printer buffer for message printer variables is full, i.e. all messages in the circular message printer buffer are replaced by new ones.

11: At least one message is stored in the history buffer.

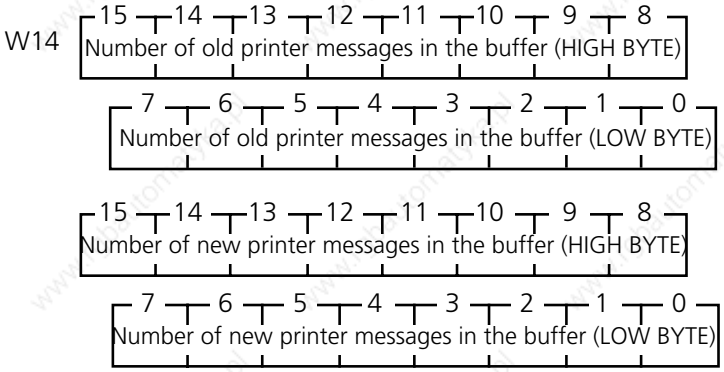
12: Response bit for history buffer erasure

13: Print direction or printer pointer adjustment, 1 = most recent entry first, then in the direction of previous messages

14: Printer messages NOT entered, if logical 1

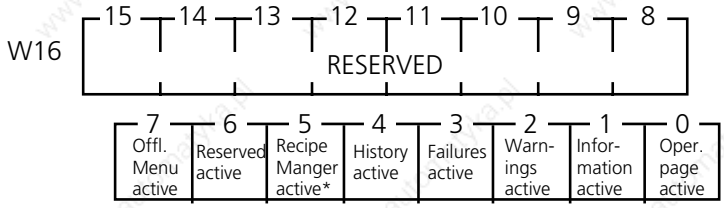
15: Abort print job or printer pointer adjustment

3.3.3.2 PRINTER STATUS



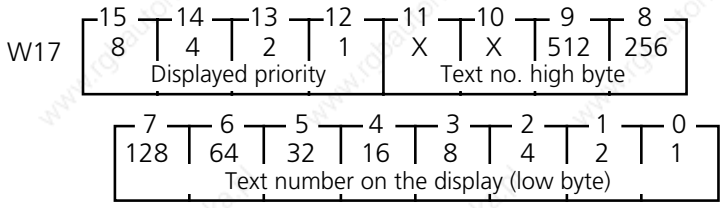
These numbers always refer to the next message to be printed. If the most recent message has been printed in FIFO mode, W15 contains a value of 0 and W14 the total number of stored messages (max. 128). In LIFO mode, messages are printed in the direction of the previous messages. If all messages are printed, W14 contains a value of 0 and W15 the total number of messages.

3.3.3.3 PRIORITY STATUS



* in preparation

The low byte is set to logical 1 for each activated priority (independently of disabling or authorization assignments).



Word 17 contains numerical data of the priority to which authorization is assigned. If, for example, a failure message bit is set and if the failure priority is enabled and the history buffer is deactivated, bits 12 to 15 contain a number of 4 and bits 0 to 9 the number of the message being displayed. The priority indicates that this is a message text. The following values are possible:

0: Idle priority

1: Operating priority

2: Information priority

3: Warning priority

4: Failure priority

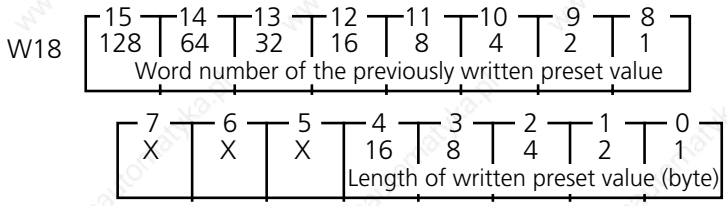
5: History priority

6: RecipeManager *) *) in preparation

7: Communication error (not contained here, see W3)

8: Offline priority (or initial stage of the actual Offline menu)

3.3.3.4 PRESET VALUE STATUS



HIGH byte, bit

0..7: Word number of the previously written preset value (binary)

The word number of the preset value previously edited is stored here in the binary format. If a programmable controller program expects entry of a certain preset value, W18 (or the HIGH byte only) can previously be set to zero. As soon as the value of this byte is below or exceeds 0, the preset value input can be evaluated by the programmable controller program. If the value does not correspond to the expected preset value, you must set W8 once again to zero and wait for the result etc.

LOW byte, bit

5..7: Currently not used.

0..4: Length of previously written preset value (binary, number of bytes)

If the PCS has written a preset value, evaluation of bits 0..4 enables the number of written bytes to be determined. Evaluating the bit mask described below allows evaluation of the variable type.

Number of bytes:

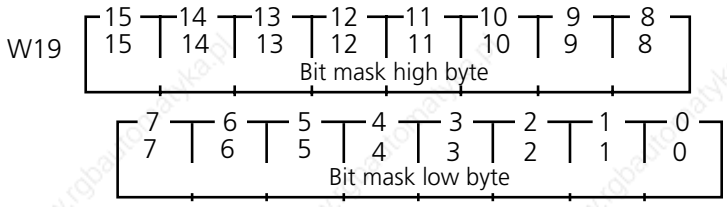
0: BIT variable

2: 16-bit variable, just as (C)STRING; BCD(0)-1; (V)BIN(0)-1,A; WORD; ASCII

4: 32-bit variable, just as (C)STRING; BCD(0)-2; (V)BIN(0)-2,B; ASCII

>4: ASCII variable

If a bit variable has been written (number of bytes = 0), the modified bit number can be determined by using the bit mask stored in W19.



A bit number can be determined by using the bit mask stored in W19. The modified bit is set to logical 1 in the bit mask while all other bits are set to logical 0. The new state of the corresponding bit can be determined by ANDing the bit mask and the word number stored in W18.

Example:

A bit variable which was programmed in word 41 (bit 11) as PRESET value is changed on an operating page (previously logical 0, but logical 1 after pressing the <+> key). After modification, W18 and W19 contain the following values:

W18: 00101001 xxx00000

W19: 00001000 00000000

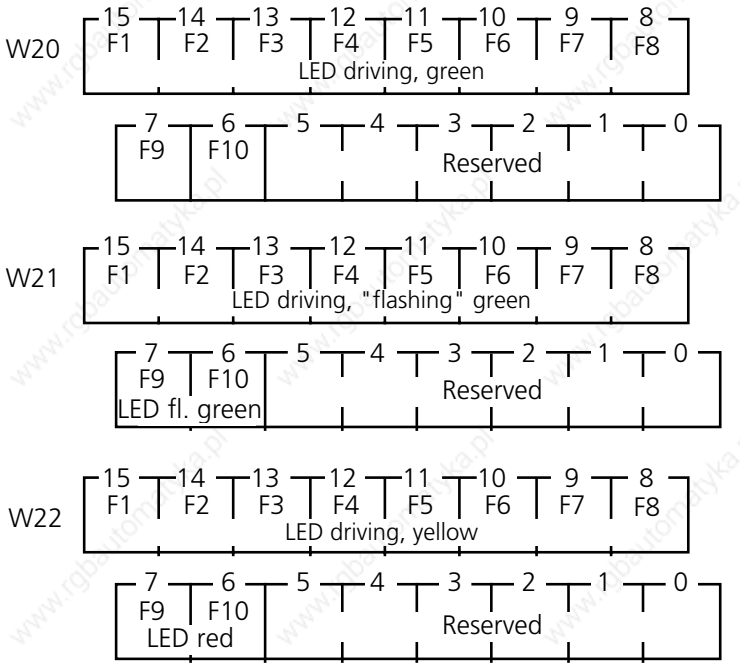
By logically ANDing W19 and W41, the value 00001000 00000000 is obtained (thus <> 0). So bit 11 was set to logical 1.

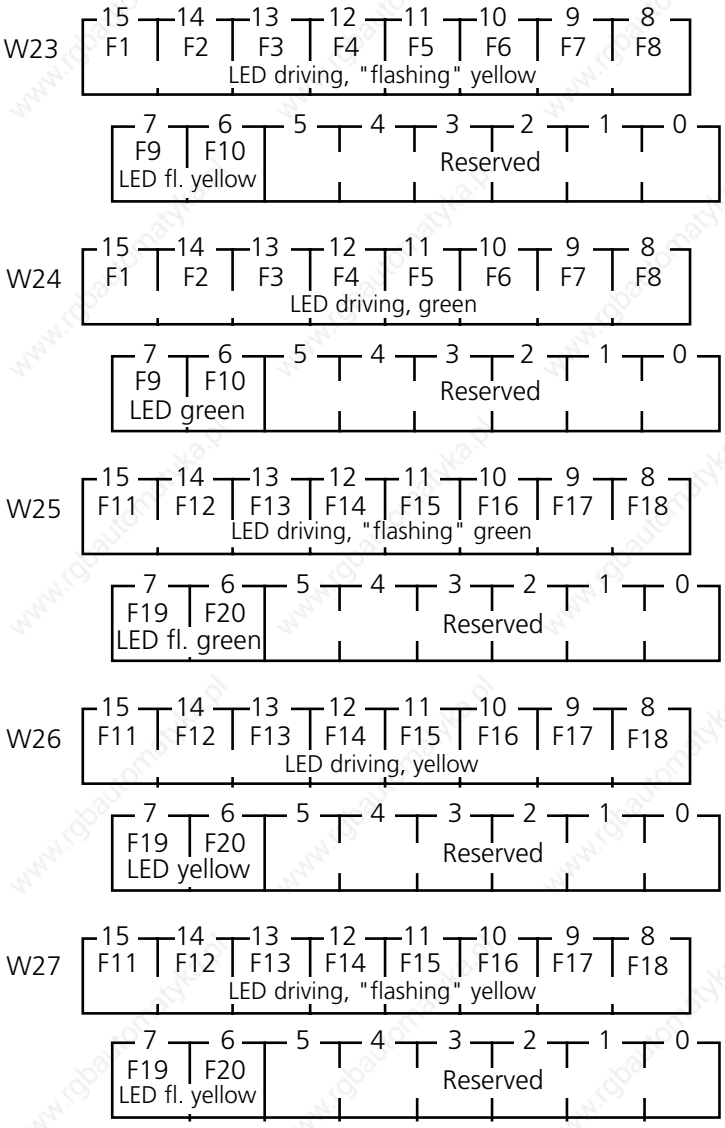
3.4 COMMAND AREA

The PCS 950 can be controlled by writing individual words to this area.

The LEDs as well as command words A-E with interlocking bits, transmission disabling, idle text number, operating page number, message printer jobs and operating printer jobs are described below:

3.4.1 LED STATUS





One green and one yellow LED are freely available per function key <F1..F8>. Since two bits are assigned to each LED, 4 different LED states are possible.

- OFF
- ON
- Flashing (75% light phase, 25% dark phase)
- Inverse flashing (25% light phase, 75 % dark phase)

If one LED is flashing and another is inversely flashing, both are lit alternately. If one LED is light, the other is dark and vice versa.

The states of an LED are determined by the two associated bits of word Wx and Wx+1.(W20 and W21, W22 and W23, W24 and W25, W26 and W27).

LED status assignment:

Wx, bit no. y	Wx+1,bit no. y	State
0	0	OFF
0	1	INVERSE FLASHING
1	0	ON
1	1	FLASHING

Example:

W20: 00001111 00xxxxxx,

W21: 00000000 11xxxxxx

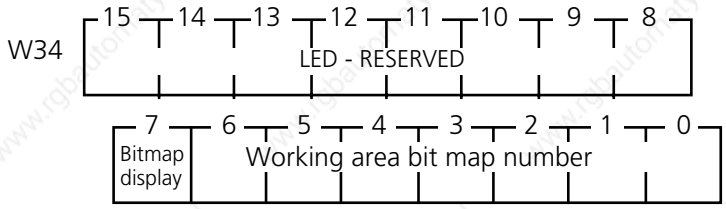
-> green LEDs above F5..F8: NORMALLY ON

-> green LEDs above F9..F10:INVERSE FLASHING

3.4.2 COMMAND WORDS

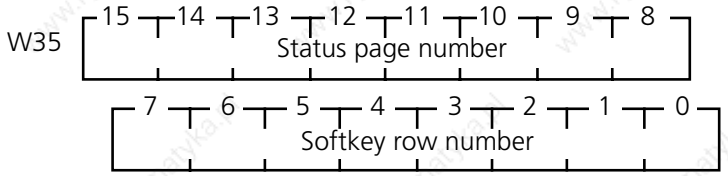
These words are used to control the functions of the PCS 950. They contain enabling bits, transmission disabling, clock control, idle text number, operating page number, message printer and operating printer control.

3.4.2.1 WORKING AREA BITMAP



A background bitmap which is automatically displayed after selecting the operating page can be assigned to each text page (except for message and history texts). If bit 7 of W34 is set to logical 1, the bitmap number assigned in the data record is suppressed and the bit map number retrieved from bits 0..6 of W34 is displayed. If this number is not configured, no bitmap is displayed.

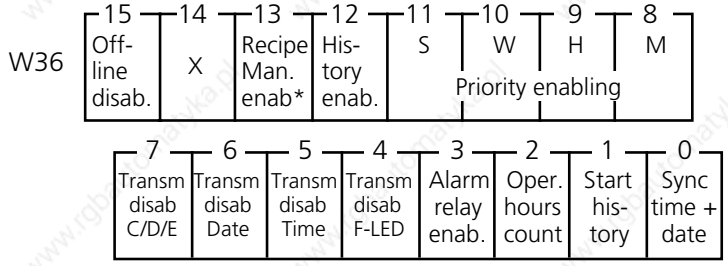
3.4.2.2 STATUS PAGE AND SOFTKEY ROW



The number of the status page which has to be displayed is assigned to the high byte of this data word. The low byte contains the softkey row number.

3.4.2.3 PRIORITY ENABLING/PRIORITY ACTIVATION

Command word A:



The function of the individual bits is described below:

15: If set to logical 1, <HELP> + <CLR> will display the initial stage of the actual Offline menu. This is, however, only possible, if communication is established.

14: Not used

13: If set to logical 1, the RecipeManager in the PCS can be selected. If set to logical 0, selection is not possible* (in preparation)

12: If set to logical 1, the history buffer can be displayed. If set to logical 0, display is inhibited. After a temporary suppression, the same message is displayed as before.

11: If set to logical 1, the failure priority is displayed. If set to logical 0, display is suppressed.

10: If set to logical 1, the warning priority is displayed. If set to logical 0, display is suppressed.

9: If set to logical 1, the information priority is displayed. If set to logical 0, display is suppressed.

8: If set to logical 1, the operating page priority is displayed. If set to logical 0, display is suppressed. (If a preset value input has been started, it is also aborted.)

7: If set to logical 1, transmission of the command words C, D and E is suppressed (scan time reduction). If set to logical 0, transmission is enabled (normal operation).

6: If set to logical 1, transmission of the date is suppressed (scan time reduction). If set to logical 0, transmission is enabled. To transmit the programmable controller time into the PCS, set bits 5 and 6 to logical 1.

5: If set to logical 1, transmission of the time is suppressed. If set to logical 0, transmission is enabled. To transmit the programmable controller time into the PCS, set bits 5 and 6 to logical 1.

4: If set to logical 1, transmission of the function key LEDs is entirely suppressed (scan time reduction, 8 words !)

3: If set to logical 1, the alarm relay is enabled. Additionally, other conditions must be met (see description). If set to logical 0, the alarm relay is always deactivated (open contact).

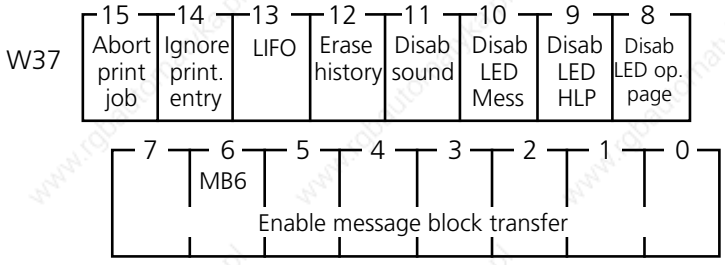
2: If set to logical 1, the counter for operating hours is enabled. The counter is only active, if communication with the programmable controller is established.

1: If set to logical 1, the history buffer is displayed. If set to logical 0, display is disabled. With each transition (0->1) the most recent message is started from (if a message has been entered into the log buffer).

0: If set to logical 1, the time is once transferred from the programmable controller into the PCS. This bit may only be set. It is reset to 0 by the PCS during the same programmable controller scan after transmission. So this bit may not be permanently set to logical 1 !

3.4.2.4 CONFIGURATION BITS

Command word B



15: If set to logical 1, a current print job (only message printer) can be aborted. As long as the bit is set to 1, each print job is immediately acknowledged but not executed. This bit is read before a message is printed. However, if the printer is not ready and is currently printing a message, an abort is not possible. To ascertain that the abort has been detected by the PCS, this bit is duplicated at the same bit position in word 13.

15: If set to logical 1, a current print job (only message printer) can be aborted. As long as the bit is set to 1, each print job is immediately acknowledged but not executed. This bit is read before a message is printed. However, if the printer is not ready and is currently printing a message, an abort is not possible. To ascertain that the abort has been detected by the PCS, this bit is duplicated at the same bit position in word 13.

14: If set to logical 1, data entry into the message printer buffer is suppressed. Before a message is entered, this bit is checked by the PCS. To ascertain that the PCS has read this bit, it is duplicated at the same bit position in word 13.

13: If set to logical 0, messages are printed in the direction of recent messages. If set to logical 1, printing takes place in the direction of the older messages. Before a message is printed, this bit is checked. To ascertain that the PCS has read this bit, it is duplicated at the same bit position in word 13.

12: A rising edge (0->1 transition) erases the history buffer (for the display). To ascertain that the PCS has read this bit, it is duplicated at the same bit position in word 15.

11: If set to logical 1, the audible PCS warning signal which sounds after a wrong operation is disabled.

10: If set to logical 1, the arrow key LEDs are deactivated (forced to OFF) within the message priority and the history priority.

9: If set to logical 1, the arrow key LEDs are deactivated (forced to OFF) within the HELP texts.

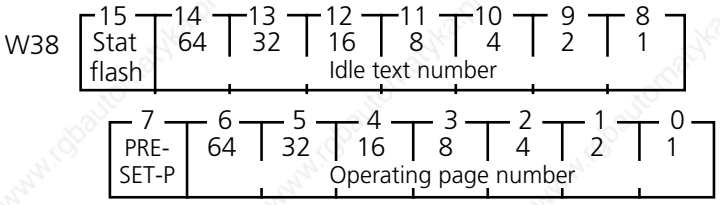
8: If set to logical 1, the arrow key LEDs are deactivated (forced to OFF) within operating pages.

0-7: If set to logical 1, message blocks can be transmitted selectively. To reduce scan time, all blocks can be disabled for example (with logical 0) and be enabled when an error occurs (with logical 1).

Warning!
Disabling transmissions may cause undesirable actions, if disabling takes place at the wrong moment. For example, a set message cannot be deleted at first after disabling the transmission (suppose deletion procedure 4 has been selected), although the message bit is set to logical 0 in the programmable controller! For this reason, disable transmission only if you are sure that no unwanted effects will occur!

3.4.2.5 IDLE TEXT / OPERATING PAGE NUMBER

Command word C



15: If set to logical 1, a displayed idle text will flash.

8-14: Represents the idle text number (0..127) in binary coded format.

Idle text number:

This is the displayed operating text number if the idle text priority is active. The idle text number (identical to the operating text number) can be modified by the programmable controller at any time. The variables of the text (preset values) cannot be modified (edited). Idle text (operating text) 0 is displayed each time the PCS 950 is switched on (even without communication).

If bit 15 is set to logical 1, the entire idle text flashes. If bit 15 is set to logical 0, the idle text does not flash (static display).

Example: Idle text 23 should flash. For this reason, a value of 10010111 x0000000 has to be written into word W38.

7: If set to logical 1, PRESET-P variables may be edited in the operating page priority.

0..6: Represents (in binary coded format) the number of the idle text which has to be activated (1..127, since 0 = end of operating page).

Operating page number:

By writing a value into bits 0..6, an operating page is activated or terminated and by writing a binary coded value > 0 (1..127) into bits 0..6, it is started. If an operating page contains one or several PRESET value variables, the first one is activated (at first underlined). Preset values can be entered now. For cursor positioning, use the <arrow keys>. The arrow key LEDs can be used at any time as visual help (if bit 8 of word 37 = command word A is set to logical 0).

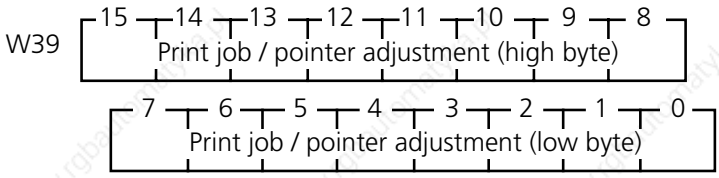
To terminate an operating page, simply set bits 0..6 to 0. PCSPROWIN allows you to determine for each operating page, if it is ABORTED or if the preset values are ACCEPTED (default setting).

If bit 7 is set to logical 1, PRESET-P variables can be modified at any time on the currently displayed operating page. As soon as the bit is set to logical 0, PRESET-P variables may no longer be modified. In this way, operating pages or variables which depend on a key switch position may be defined. In this case, PRESET-P variables are evaluated just as actual values.

If bit 7 changes from 0 to 1 during editing a PRESET-P value, the operating page is aborted. No data are stored and the cursor is positioned on the first PRESET value of the operating page.

3.4.2.6 MESSAGE PRINTER COMMAND

Command word D



This word is used to control the message printer. By writing a 16-bit number <> 0, one of the following three commands is executed:

- Number exceeding 0, below/equal to H7FFF <32767)

  This is a print job. The printer tries to output a number of messages corresponding to this number or to the number of stored messages. The first message is the message which has not been printed yet. The direction is determined by the LIFO bit.

- Number below 0 (-1 to - 32767)

  This is a printer pointer adjustment. The message printer buffer is reduced by this number of messages (up to the upper limit). The direction depends on the LIFO bit. If this bit is set to 0, adjustment takes place in the direction of the older messages. If this bit is set to 1, adjustment takes place in the direction of the recent messages.

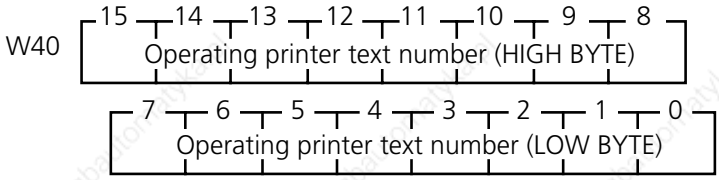
- H8000 or -32768

  This number causes the message printer buffer to be erased.

After execution, the PCS sets this word to zero. Afterwards, a new job can be detected.

3.4.2.7 OPERATING PAGE COMMAND

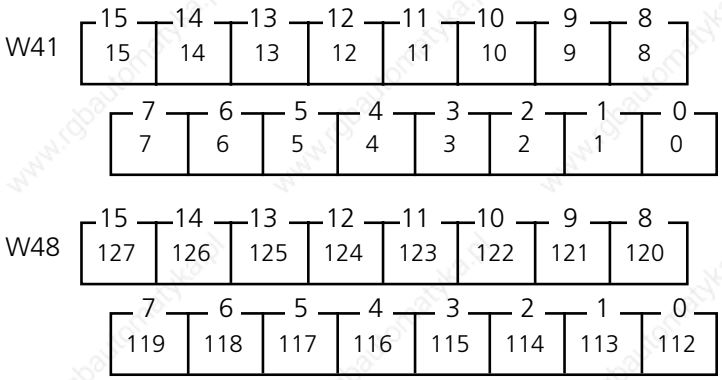
Command word E



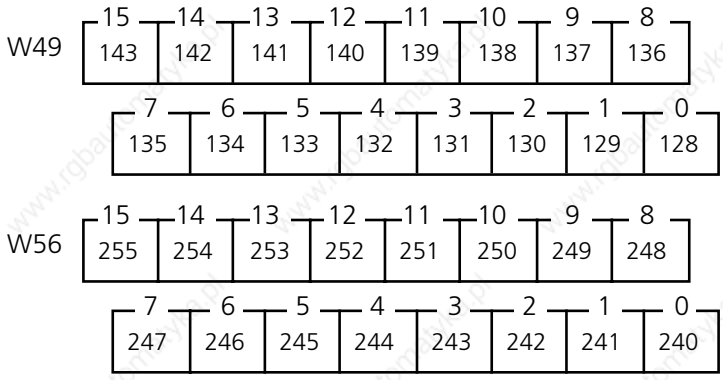
This word controls the operating printer. By writing a 16-bit number <> 0, an operating printer form is printed. If the print job is completed or if this operating printer text does not exist, this word is set to 0 by the PCS. Afterwards, a new job can be detected.

3.5 MESSAGE AREA

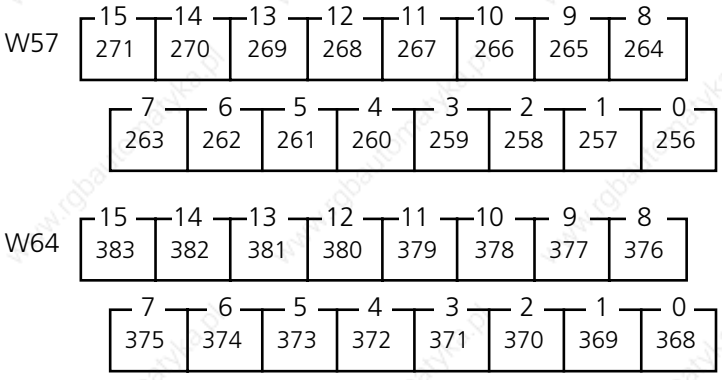
W41-W48: MESSAGE BLOCK 0



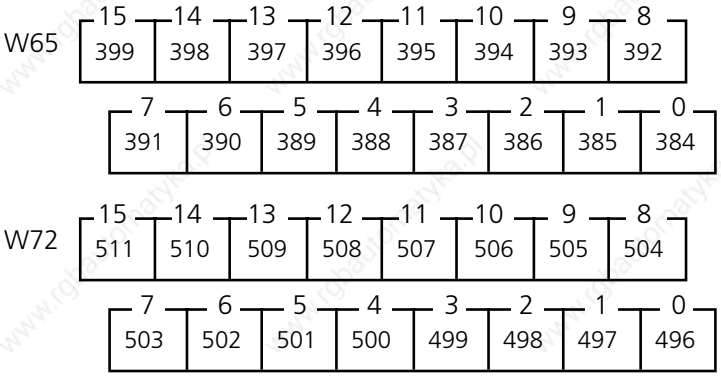
W49-W56: MESSAGE BLOCK 1



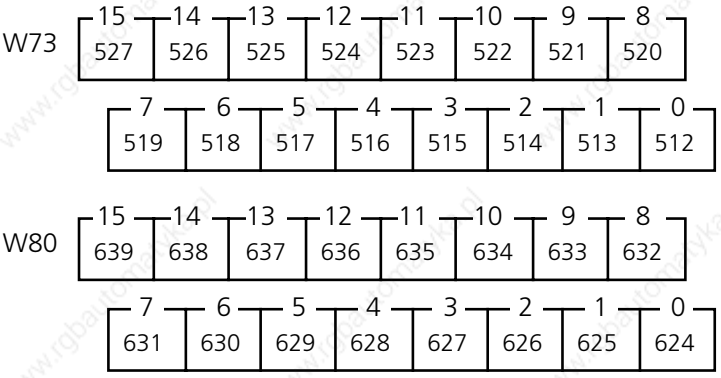
W57-W64: MESSAGE BLOCK 2



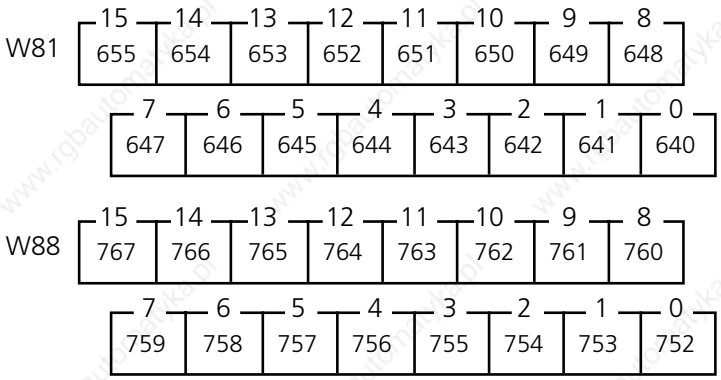
W65-W72: MESSAGE BLOCK 3



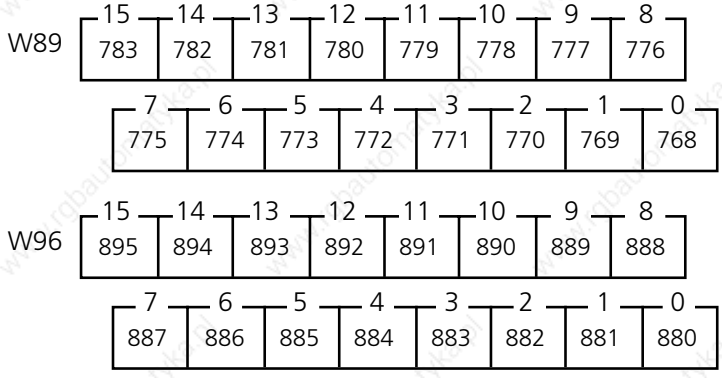
W73-W80: MESSAGE BLOCK 4



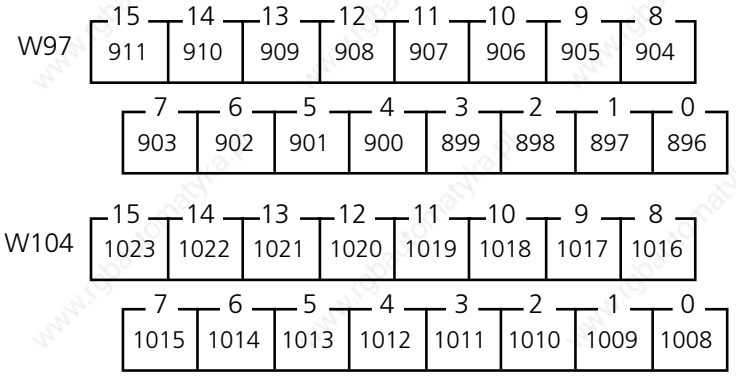
W81-W88: MESSAGE BLOCK 5



W89-W96: MESSAGE BLOCK 6



W97-W104: MESSAGE BLOCK 7



A MESSAGE TEXT (0..1023, max. 32 lines per text) is assigned to each bit. A specific priority (S,W,H) and a specific deletion procedure (1-5) may be defined for each text.

■ CONVERSION OF MESSAGE TEXT NUMBER INTO DW NUMBER

W = pre-decimal point places of (message text/16) + 41

Bit = decimal point places of (message text/16) * 16

Example:

Calculation of the bit position of message text 165:

165 / 16 = 10.3125

10 + 41 51

0.3125 * 16 = 5

Message text number 165 corresponds to W51 Bit5.

■ EVALUATION

Only 1 message block is read within one programmable controller scan. If an evaluation is required during each cycle, only 1 message block may be used and an appropriate SCAN TIME for the programmable controller must be selected (see TECHNICAL APPENDIX).

■ COLD RESTART

Since the PCS message buffer is non-volatile, the programmable controller memory should be of the same type. Otherwise, log buffer messages may be entered several times during a cold restart.

■ TRANSMISSION

For each message block, command word B (W 37) contains a bit which enables transmission to be suppressed. If these bits are modified during operation, all bits of this block must be set to 0. Normally, these bits are only initialized during a cold restart and are not modified afterwards.

3.6 VARIABLE AREA

The variable area is located between word W110 and word W255 (maximum). If this area is insufficient, the expansion area W105-W109 can also be used. However, this area is reserved for possible extensions. So care must be taken when using it (programmable controller program changes !).

If no messages or only a part of the available messages is required, words 41..104 may also be used for variables.

For detailed information about the different variable types, see the sections describing variables, texts and operating pages.

Before data are exchanged, all variables (preset and actual values) must be written into the corresponding words. After data exchange, a maximum of one preset value (corresponding to preset value states in W18 and 19) must be read out of the corresponding words.

An adjacent addressing of variables of the same display page is recommended. In this way, programmable controller scan time may partially be reduced.

Assigning addresses to the individual variables is effected in PCSPROWIN.

The variable formats STRING, CSTRING, BCD, BIN, WORD and ASCII are right-justified and ascending, e.g. BIN-2 in W50..W51 (W50 is the HIGH and W51 the LOW word). The BIT format can be used for each individual bit (to address all 16 bits individually, 16 BIT variables must be defined). One and the same word can be used as source and destination for several variables (even different formats). If used as preset value, only one variable format should be assigned to a word. However, one or several actual value variables (even of different formats) can be simultaneously assigned to this word.

Leading (unused) bits in the preset values of the STRING, CSTRING, BCD(0)-1 (lengths 1..3) and BCD(0)-2 (lengths 1..7) type are ignored during reading and reset to 0 when writing data into the programmable controller. The BIT type only modifies the addressed bit.

Preset values should be defined according to the admissible MIN/MAX values before executing a cold restart, since they are required as default values for editing. If the MIN/MAX range is exceeded, inverse fields are represented which are only removed after correcting the preset values.

If 32-bit variables are used, the word with the lower number is the more significant word and the word with the higher number the less significant word.

Variables of the BIT and CSTRING type are written into the programmable controller immediately after modification. All the other types are only stored after pressing the <ENTER> key or when exiting the variable field (configurable).

4 Index

5 IMPORTANT USER INFORMATION

5.1 IDEOGRAMS AND SYMBOLS

The following symbols and ideograms are used in this manual:

Danger!
Directly imminent danger which causes death and most serious injuries.

Warning!
Possibly dangerous situation which can cause death and most serious injuries.

Caution!
Possibly dangerous situation which can cause light and less serious injuries.

Attention!
Possibly harmful situation which can cause damage to the product or its environment.

Mechanical pressure causes damage to the product.

Information concerning safety when using the devices in an Ex area.

Information and notes which must be additionally observed.

5.2 SAFETY RELATED INFORMATION

- The device may only be connected to the systems specified by Systeme Lauer.
- The device meets the current technical state of the art.
- Only trained and qualified persons who have familiarized themselves with the product are allowed to install and operate the device.
- The responsibility of persons operating the device must be clearly determined in order to avoid undefined competencies.
- The relevant safety regulations and standards must be observed.
- Opening the device is not allowed. Systeme Lauer is not responsible for resulting damages.
- Before commissioning the device, this instruction manual must be read thoroughly.
- Modifications of or changes to the design of the device are not allowed. Systeme Lauer is not responsible for resulting damages.
- The supply voltage of the device must be within the range specified in the section „Specifications". Systeme Lauer is not responsible for damages resulting from non-compliance to this requirement.
- The latest manuals and documentations are valid.

The specifications published by Systeme Lauer were determined with our methods and facilities; characteristics are only guaranteed in this respect. The user is responsible for testing and determining the suitability for the specific application or for use under actual conditions. Systeme Lauer does not assume any warranty for this.

Modifications reserved