PCS 91.BOS
english

One does not refer in the manual explicitly to the devices of the PCS plus/win series, the description applies to all devices. With differentiations between the equipment series the following allocations apply:

PCS topline	**=**	**micro/mini:**	**PCS 009, PCS 090, PCS 095, PCS 095.1, PCS 095.2**
		midi:	**PCS 900, PCS 950, PCS 950c, PCS 950q, PCS 950qc,**
		maxi:	**PCS 9000/9100**
PCS plus	**=**	**micro/mini:**	**PCS 009 plus, PCS 090 plus, PCS 095 plus**
		midi:	**PCS 950 plus, PCS 950c plus, PCS 950q plus, PCS 950qc plus**
PCS win	**=**	**micro/mini:**	**PCS 009 win, PCS 090 win, PCS 095 win**
		midi:	**PCS 950 win, PCS 950c win, PCS 950q win, PCS 950qc win**

Reg 10258/0100

CiS-Nr.: 360.110.0100
Version 2/02.03

Systeme Lauer GmbH & CoKG
Postfach 1465
D-72604 Nürtingen

Operator reference manual: PCS 91.BOS
Version: 04. Februar 2003
Person responsible: Zoch

- We reserve the right make changes to the reference manual without prior notice.
- We can not guarantee the accuracy of the programs and data stored on the diskette and the fault-free state of this information.
- Since diskette represent manipulatable data media, we can only guarantee the physical completeness. The responsibility is limited to a replacement.
- At any time, we welcome suggestions for improvements and remarks on errors.
- The agreement also applies to the special appendices to this reference manual.

Notes for the user

Please read the manual before beginning and keep the manual for later use.

Target group

The manual has been conceived and written for users who are experienced in the use of PCs and automation technology.

Typographical conventions

[KEY]	Keys that are to be pressed by the user are given in square brackets, e.g [CTRL] or [DEL]
`Courier`	On-screen messages are given in the Courier font, e.g. C:\>
`Courier bold`	Keyboard input to be made by the user are given in Courier bold, e.g. C:\>DIR
Italics	Names of buttons to be pressed, menus or other on-screen elements and product names are given in italics.

Pictograms

The manual uses the following pictograms to highlight certain text passages:

Danger!
Possibly dangerous situation. Injury to persons can be the result.

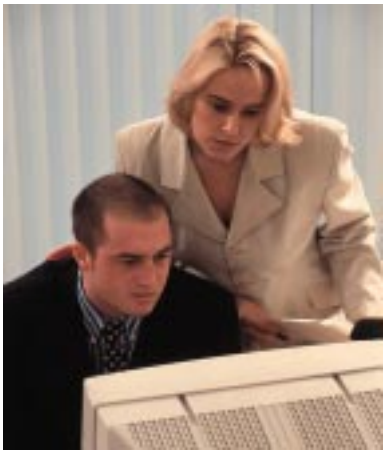
Attention!
Possibly dangerous situation. Property damages can be the result.

Tips and supplementary notes

Inhaltsverzeichnis

Quality and support

In our company, quality comes first. From the electronics component up to the finished device, the quality assurance test competently and comprehensively.
National an internation test standards (ISO, TÜV, Germanischer Lloyd) are the basis.
Within 48 hours, every device passes a 100% check and continuous test under worst case conditions at changing temperatures (0...50°C) and test voltages.
A guarantee for maximum quality.



Our products not only feature a maximum economic efficiency and reliability but also a comprehensive complete service.
You not only receive demo devices but we rather make specialists available who support you in person with your first application.
Qualified user consultation by competent sales engineers is obvious for us.
Our support is for you for the side with advice and deed every day.

We set up training programs and technical training for you in our modern training center or alternatively also in your house.
Request the curent training catalog.

From the consultation up to the user support, from the hotline up to the service, from the reference manual up to the training an all covering and individual service for the entire product line is waiting for you.

Whenever you need us, we are there for you:
dynamically, creatively and enormously efficiently. With the entire experience of a world-wide successful enterprise.

Telephone	07022/9660-132, -231, -230
eMail	support@systeme-lauer .de
Web site	www.lauer-systeme.net
	Systeme Lauer Active Area
	(Download of Software, driver, manuals, Forum...)

Safety regulations

This reference manual contains the most important remarks in order to safely operate the device.

- This operator,s guide, particulary the safety remarks are to be noted byall persons working with the device.
- Furtherrmore, the rules and regulations for the accident prevention applying to the application location are to be observed.
- Use as directed. The device is deigned for the application in the industrial area.
- The device is manufactured to the state of the art and the official safeguarding regulations. Nevertheless, due to the application, dangers or impairments can result to the machine or to material assets.
- The device meets the requirement of the EMC guidelines and harmonized European standards. Any hardware-related modification of the system can influence the EMC behavior.
- The device may not be used without special protective measures in the hazardous area and in plants requiring a special monitoring.
- Do not heat up the buffer batteries. Danger of explosion. Serious burning can be the result.
- The installation and operation may only be performed by trained personnel.
- The operating voltage of the device may only be in the specified ranges.
- You find information on this on the type plate and in the specifications of this reference manual.

Norms

The device is constructed using up-to-date technologies and fulfils the requirements of the following guidelines and norms:

- Compliant with the EMC Directive 89/336/EEC and the German law on electro-magnetic compatibility
- Interference compliant with the generic requirements norm EN 50081-2 and product norm EN 55022:
- Measurement of the conducted interference voltage as per EN 55022
- Measurement of the radiated radio interference field power as per EN 55022 class A
- Interference immunity in compliance with generic requirements norm EN 50082-2 and product norm EN 61000-6-2:
  - Electro-static discharge (ESD) as per with EN 61000-4-2
  - High-frequency electromagnetic fields as per EN 61000-4-3 and ENV 50204
  - Fast transient interference (burst) as per EN 61000-4-4
  - Surge voltages as per EN 61000-4-5
  - High-frequency conducted fields as per EN 61000-4-6
  - Voltage dips and short-term interruptions as per EN 61000-4-11

The assembly and connection instructions contained in this documentation must be followed.

Conformity of this equipment is confirmed by the CE logo.
The EC declaration of conformity can be requested from:

Systeme Lauer GmbH & Co KG
P-O-Box 1465
D-72604 Nürtingen

Information for driverselection

A Lauer driver

Communication with PCS micro/mini, plus/midi und maxi. Requires PLC-Programme.

Operating panel:	PCS 009/090/095/900/920/950 and 9000
PLC-system:	CL300/CL400/CL500
Interface:	PCS 830.1/830.3
Protocol:	Lauer Protocol
Adaptercable:	PCS 736
Kind of driver:	Expander driver

B Buep 19-Expander driver

Communication with PCS micro/mini, plus. Requires PLC-Programme.

Operating panel:	PCS 009/090/095
PLC-system:	CL300
Interface:	PG-connection
Protocol:	Buep 19
Adaptercable:	PCS 706
Kind of driver:	Expander driver

C Buep 19e-Expander driver

Communication with PCS micro/mini, plus/midi. Requires PLC-Programme.

Operating panel:	PCS 009/090/095/900/920/950/950c
PLC-system:	CL200/CL400/CL500
Interface:	PG-connection
	additional CL500: SK500
Protocol:	BUEP 19e
Adaptercable:	PCS 706
Kind of driver:	Expander driver

D Buep 19e-Direct driver

Communication with PCS midi. Requires PLC-Programme.

Operating panel:	PCS 900/920/950/950c
PLC-system:	CL200/CL400/CL500
Interface:	PG-connection
	additional CL500: SK500
Protocol:	BUEP 19e
Adaptercable:	PCS 706
Kind of driver:	Direct driver

A LAUER driver

A1 First commisioning

Adressing of the board

The PCS 830 board is slot-independent. It uses 4EZ/AZ addresses (4 additional input bytes and 4 additional output bytes) and can be freely addressed in steps of 4 via the DIL-switch which is located on the board. Please take care that the set address does not intersect with already used addresses.

Selection of the address

Among others, the areas EZ0 to EZ63 and AZ0 to AZ63 are reserved for analog I/O boards. The PCS 830 board is also addressed in this area. The base address can be set to 0 and to all addresses divisible by 4. Input and output addresses (EZ / AZ) must lie in parallel, i.e. they can not be differently set!

Attention!
Take note, that after re-addressing of the board you must adjust the first three words of the cross-reference data! (Refer to section 1.5 and 1.12)

Example EZ/AZ 4 (EZ4-EZ7, AZ4-AZ7)
6-segment DIL-switch (below the male multipoint connector)

DIL no.:	6	5	4	3	2	1
	A7	A6	A5	A4	A3	A2
Valence:	128	64	32	16	8	4

Baud rate settings

The baud rate for communication between the PCS and the PCS 830 is set via the 8-segment DIL-switch which is located at the upper board edge. Please note, that this switch is only read after a restart and/or after a communication reset.

A LAUER driver

Example: Baud rate setting for 19200 baud.
8-segment DIL-switch (upper board edge):

DIL	8	7	6	5	4	3	2	1	BAUD
	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	1200
	OFF	OFF	OFF	OFF	OFF	OFF	OFF	ON	4800
	OFF	OFF	OFF	OFF	OFF	OFF	ON	OFF	9600
	OFF	OFF	OFF	OFF	OFF	OFF	ON	ON	19200

Status display

The PCS 830 features a 7-segment STATUS display which informs about the state of the board! The following display states are currently possible:

„0“: programmable controller is set to „STOP“, i.e. the PCS 830 waits for the „RUN“ signal.

„1“: The PCS 830 waits for the complete reception of the cross-reference list.

„7“: TIMEOUT fault, i.e. the communication between PCS 830 and PCS was either interrupted (PCS shows „TIMEOUT“ -> cable break) or the PCS has twice received (one after another) wrong data (PCS shows „CONNECT“). In case of the „CONNECT“ fault, noise suppression measures are to be explored and/or to be improved if necessary.

„8“: CONNECT fault, i.e. the PCS 830 has twice received (one after another) wrong data (PCS shows „TIMEOUT“). In this case, noise suppression measures are to be explored and/or to be improved if necessary.

„9“: WATCHDOG fault, i.e. the PCS 830 board is either located in an very noisy environment or the PCS 830 is defective (communications cannot be started anymore).

„-“: The PCS 830 has either sent or received a repeat request. With a reparable fault, the „-“ extinguishes subsequently again. If the fault was not reparable, a „7“(TIMEOUT) or „8“(CONNECT) is subsequently displayed.

„.“: This point lights as so soon as the PCS 830 has received a tested package and is thus ready for the data interchange with the programmable controller. With an increasing data throughput this point will be dimmed. This can be influenced by specifying the ZV, S,... parameter for the serial programming of the PCS.

A LAUER driver

A1.1 Description of the handling software

The enclosed -PCSKOMM PB manages the data transfer between the transfer area (a data block by default) and a PCS 830.
The README.DOC file contains current notes concerning the use of the various projects.

1. The PB uses the MBs 246 to 255 as temporary flags. Writing to these MWs does not influence the PB as long as they are not simultaneously used by interrupt programs. The contents must be written newly on entering the PB and saved on exiting if these MWs are required by other PBs. In no case, interrupt programs are allowed to write to these MWs!

2. The transfer area may not be specified within the data block when using EPROM modules. In this case, the cross-reference DB must be adjusted in such a way that the system area and the used variables are located in the RAM area of the CPU (for instance in the data buffer).

   The implementation is limited (besides the cyclical call of the-PCSKOMM PB) to reading of and writing to the DWs in the transfer DB (only with default assignment!). The presence of the (selectable) transfer DB is not examined. It must be present with the required net length (dynamic) in the programmable controller. Otherwise the programmable controller shows a fault (possibly only at the first call of a present value). A general fault bit (P3) enables the communication status to be evaluated by the ladder diagram. The data transfer has been stopped if this fault bit is set to 1. It can be restarted by setting the (P2) flag. Furthermore, another restart flag is required. This is set after the first successful processing of the handling PB. The restart flag must be reset in OB 7 and 8.

A1.2 Scan time extension

Each task requires a header with one additional word. Since the length of task 5 depends on the number of the external variables in the display (0 to 24 words including headers if the data sources are not consecutive), a maximum of 40 words per cycle will be transferred with the PB. The transfer of the clock happens only every second and costs 3 additional words. All other tasks can be neglected for the average calculation since they only appear isolated.
On average, the processing time of the -PCSKOMM PB totals approximately 10 milliseconds (CL 300).
Refer to section 2.3.1. for the limitation of the package length. The lengths of the individual tasks can be taken from the section 5.3.
The response is optimal if the scan time is selected sufficiently large with the PCSKOMM PB finding a PCS 830 board ready for data interchange in each cycle. The test and the perhaps required repetition of the packages is organized in the PCS 830.
The design of the PCSKOMM PB contains provisions for aborting the communi-cation on every occurring fault and signalling it to the outside. After this, communication must be explecitely restarted using an input (Parameters P2). Of course, this does not affect repetitions caused by reparable faults as these are treated in the PCS 830 internally.

A LAUER driver

Customer-specific presettings are to be inserted at 2 positions:

PB INIT	customer specific presetting for restart
PB COFF	customer specific assignment for an error case

Attention!
In all projects on the floppy disk, the INIT and COFF PBs are to be adjusted specifically for the connected operating console. The contained HLT commands are to be deleted.

A1.3 Parameterization of the PCSKOMM PB

P0: Cross-reference data block (here -QUERWL)
The cross-reference addresses are listed in this block. At the restart, these addresses will be transferred once into the PCS 830. It is ensured, that the download of the cross-reference list is made in the first cycle of the programmable controller. In the CL 300 this first cycle is extended due to the download by approx. 100 ms. During the following cycles, the -PCSKOMM PB receives the corresponding cross-reference address directly from the PCS 830 (time saving).

P1: Communication data block (here -PCSKOMM)
This block is active during processing of the handling block. Cross-reference addresses specified in the cross-reference data block addressing a data word relate to the here indicated data block.

P2: Switch for resetting following a fault (bit)
On a communication fault, communications can be restarted using this bit. The faults indicated in DW3 (default assignment) are then reset automatically! Please note that no edge evaluation is made. I.e. with the P2 bit set, communication is automatically restarted after a fault has appeared! This mechanism can also be realized by connecting the P3 fault output directly to the P2 reset input.

P3: General error message (bit)
The P3 general fault bit is set for any error in the connection between the programmable controller and the PCS 100. DW3 (default assignment) can be evaluated for the exact cause of error. This bit is reset as soon as the communication is running without a fault.

P4: Flag for the first processing of the handling block (bit)
This flag should be reset before the first call (restart) of the handling block. It is automatically set during the first execution of the PB. Among other things, it is used for presetting during a restart.

P5: Customer-specific presetting for communication start
The -INIT PB is selected at communication start. PCS-specific presettings are performed by this PB. The HLT command must be removed.

P6: Customer-specific presetting for communication loss
The -COFF PB is selected at communication loss. PCS-specific presettings are performed by this PB. The HLT command must be removed.

A LAUER driver

A1.4 Set-up of the cross-reference DB

The cross-reference data block contains 256 decimal word addresses. During a restart, these addresses are stored by the download procedure on the PCS 830 board. Values for these addresses must be taken from the CL300 software manual (operation list). The address pointers are required for indirect addressing. The PCS 830 board addresses must be specified in the first 3 words (0-2). In the following words (4-255) the source and/or destination addresses are specified for each word. Therefore, it is possible to assign the system area command word (DW 14) directly to a flag word without copying the flag word to the data word and conversely using a transfer command.
The following table shows a section for detailing the structure of the cross-reference data block.

DB0 name: QUVERWL comment: cross-reference data block RAM/EPROM: R

No.	Symbol	Type	Vz	Datafield	F
D0	RDADR	Word	N	3590;corresponds to EZ6	D
D2	WRADR	Word	N	4102;corresponds to AZ6	D
D4	STADR	Word	N	3589;corresponds to EZ5	D
D6	ERADR	Word	N	2054;corresponds to D3W	D
D8	W4	Word	N	2056;corresponds to D4W	D
D10	W5	Word	N	2058;corresponds to D5W	D
D12	W6	Word	N	2060;corresponds to D6W	D
D14	W7	Word	N	2062;corresponds to D7W	D
D16	W8	Word	N	2064;corresponds to D7W	D
D18	W9	Word	N	2066;corresponds to D7W	D
D20	W10	Word	N	2068;corresponds to D7W	D
D22	W11	Word	N	2070;corresponds to D7W	D
D24	W12	Word	N	2072;corresponds to D7W	D
D26	W13	Word	N	2074;corresponds to D7W	D
D28	W14	Word	N	2076;corresponds to D7W	D
D30	W15	Word	N	2078;corresponds to D7W	D
D32	W16	Word	N	2080;corresponds to D7W	D
D34	W17	Word	N	2082;corresponds to D7W	D
D36	W18	Word	N	2084;corresponds to D7W	D
D38	W19	Word	N	2086;corresponds to D7W	D
D40	W20	Word	N	2088;corresponds to D7W	D
D42	W21	Word	N	2090;corresponds to D7W	D
D44	W22	Word	N	2092;corresponds to D22W	D
D46	W23	Word	N	0256;corresponds to M0W (flag word 0)	D
D48	W24	Word	N	0194;corresponds to A1W (output word 1)	D
D50	W25	Word	N	2098;corresponds to D27W	D
D52	W26	Word	N	2100;corresponds to D26W	D

Attention!
The first 3 addresses in the cross-reference data block are to be adjusted when re-addressing the board (6-segment Dil-switch).

A LAUER driver

Word no.:	Meaning:
D0	RDADR: Boards - base address +2 reading (EZ)
D2	WRADR: Boards - base address +2 writing (AZ)
D4	STAADR: Boards - base address +1 reading (EZ)

A1.5 Operation the PLC with EPROM/EEPROM

Please note the following. When using EPROM and EEPROM memory modules in the programmable controller, the communication area cannot be located within one data block since these data words are fixed in EPROM/EEPROM and thus cannot be manipulated anymore (write operations are not possible).
The only possibility consists in altering the cross-reference data block so that the entire system area (word 3 to word 22 including) and the used variable area (from word 23 to 255 maximum) are located in a modifiable memory area (e.g. flag area, data buffers etc.).

Implementation of the handling PB

1. Power-down the programmable controller
2. Define the PCS 830 address and the baud rate using the DIL-switches
3. Plug-in the board
4. Set the programmable controller to stop and apply power
5. Re-adjust the cross-reference data block addresses if necessary
6. Reset P4 (flag for restart) in the OBs 7 and 8
7. Select and parameterize the -PCSKOMM PB of OB1 (or anywhere else)
8. Adjust INIT and COFF
9. Define the reset pushbutton and set P2 (reset) with the positive edge
10. Assign, link, and download all data blocks into the controller
11. Switch the programmable controller to RUN

The implementation has been successfully completed if the K8000H fault (timeout if no PCS is connected) appears in DW3 (default assignment) and the flags P4 (restart) and P3 (general error message) are both set to logical 1. In addition, the PCS 830 must indicate the „STATUS“ value „7“. An example (OB1) is contained on the floppy disk which indicates a fault on output 0.0, expects a reset pushbutton on E 0.0, uses the flag 20.0, and addresses the PCS 830 board as EZ/AZ 4-7. The baud rate set at the PCS and at the PCS 830 using the DIL-segments is used as communication baud rate.

A LAUER driver

The following transfer operations must be performed for the integration into your program (not required functions can be left out):

- Before calling -PCSKOMM PB: all values read by the PCS must be copied into the corresponding DWs. This affects DW12 to DW22 and all variables but only if you do not directly access the corresponding words by altering the cross-reference data block (example: flag words, data buffers, input words etc.).
- After calling -PCSKOMM PB: all values changed by the PCS must be written back to flags. This affects DW4 to DW14, all message bits with delete behavior 2, and all set values, however only if the cross-reference data block has not been adjusted.

The PCS assumes the following conditions after restart:

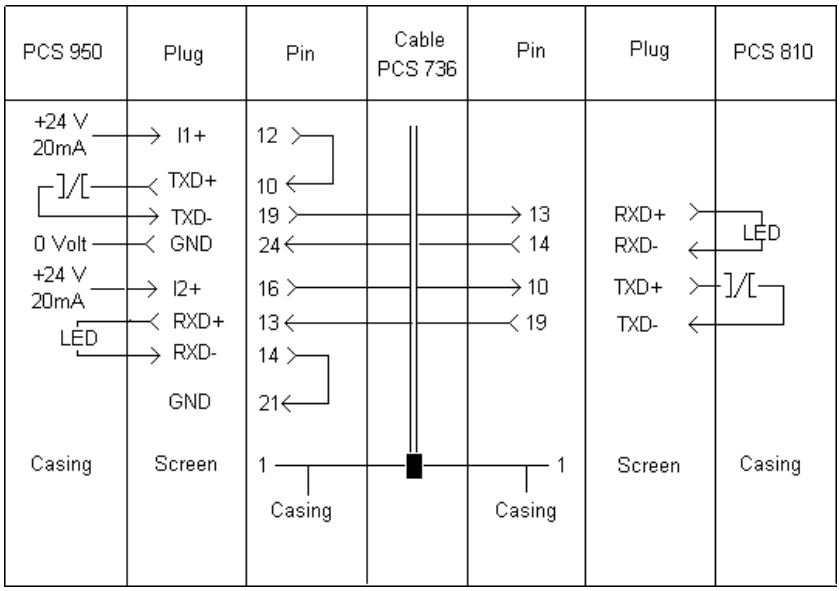
- Date and time have become invalid and thus will be immediately transfer-red.
- All message bits are turned off. If several messages should be activated at the restart they are newly entered in the sequence of their numbers (lowest first) into memory.
- Text number zero is selected as idle text.
- All menus have been terminated before switching off.
- All pushbuttons and DIL-switches are transferred once after restart.

A LAUER driver

A2 Communication

Adaptercable PCS 736

The connection is made via 2 TTY channels. The PCS supplies the line current for both channels. Thus, a strict potential separation exists with regard to the programmable controller.



When using shielded standard cable (4 * 0.14, untwisted) the following recommended maximum length apply:

19200 baud	10 meters
9600 baud	20 meters
4800 baud	40 meters
1200 baud	160 meters

When using low-capacity data cables twisted in pairs the 10-fold lengths can be projected. Interrupting the connection is registered by the PCS 009/090/095/900/9000/PCS plus and in the PCS 830. The fault is indicated in the error word (DW3) for the further evaluation.

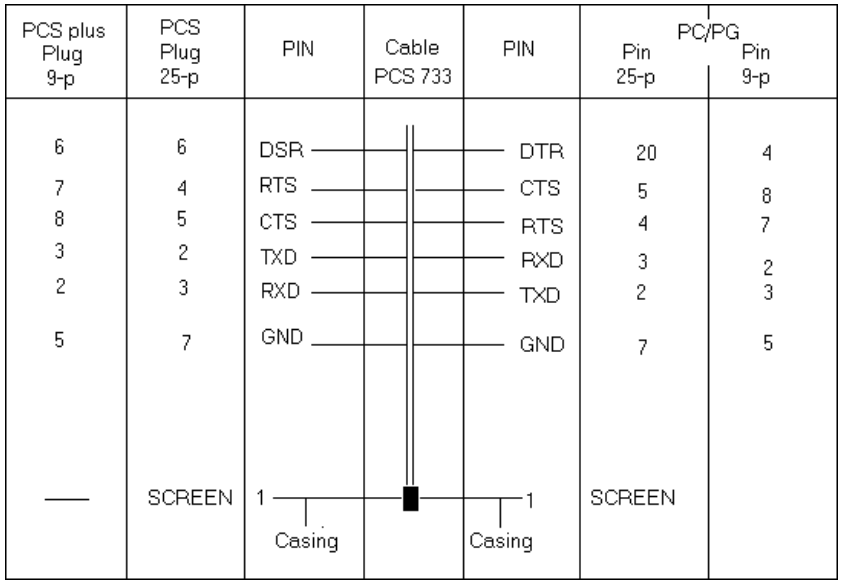
Screening

The shield should be connected on both sides to a metallized connector hood. If non-metallized connector hoods are used, the shield may also be connected to pin 1, but this is not recommended due to interferences, because the data lines should be completely covered by the shield (if possible). However, it should be noticed that grounding on both sides may require an equipotential bonding conductor with a cross section of 10 times that of the shield in order to compensate for earth potential variations and to prevent equalizing currents on the cable shield! This is especially important, if the PCS and the programmable controller are not connected to the same common point (if they are for example installed in different control cabinets).

A LAUER driver

Programming cable PCS 733

Connection PC to PCS
Use this cable for programming (loading of the driver and user program) the PCS.



Data transmission PCS-PCS 830

Data traffic between the PCS and the programmable controller is effected in data packets. Each data packet is assigned a checksum. The package content is checked for possible errors by the PCS 830. Each packet consists of a minimum of one sub-packet which performs a clearly defined task. The tasks to be integrated in a packet are determined by the PCS based on the pro-grammable controller scan-time, the baud rate and the priority of each task. Each of this tasks is assigned a specific start priority. Priority management assures that no task is lost. The indicated packet lengths refer to words, including header.

A LAUER driver

The tasks are listed below in detail:

TASK	PRIOR.	LENGTH	START CRITERION
1. Write keyboard status	8	3	When pressing or releasing a key
2. Reset message bit	7	2	Press CLR for delete behavior 2
3. Write PLC setpoint	6	2..3	Always after changing a setpoint variable and leaving the entry field
4. Read actual values	5	2..24 *)	In priorities 0..2, 6 continuously, otherwise when display is refreshed
5. Read message bits	4	9	Continuously
6. Send PCS status	4	3	After changes
7. Read LEDs, flashing LEDs, memory and display behavior	3	3	Continuously
8. Read command word	3	4**)	Continuously (menu selections only if the temporary buffer is empty)
9. Send time and date	2	3..6	After changes

*) Depends on the number of variables displayed and if the addresses are adjacent. If the addresses are not adjacent, one header is required per non-adjacent variable (1 word).
**) Task with 2 headers and one data word each.

ZV,S parameters, PLC scan-time

The response time can be optimized via the timeout parameter (PCSPRO/ driver parameter). For an optimal response time a data interchange must take place with the PCS 830 in each programmable controller cycle. The PCS adjusts the package length of each package to the PLC scan-time so that the above-mentioned condition is adhered to as much as possible.

Timing sequence of the data transfer:
1111 2222222222222222 33 44444 55555 6666666666666666 (1111)

- **1111**
  The PCS assembles a new package. The required time totals 2.5 milliseconds in the online operation and 10 milliseconds in the off-line operation.
- **2222222222222222**
  The package is transferred. The required time depends on the package length and the baud rate. Since the package length is not constant, this time can on principle not be determined.
- **33**
  The PCS 830 examines the package on plausibility and signals the handling PB the readiness for the data interchange.
- **44444**
  Waiting time, until the handling PB is processed.

A LAUER driver

- **55555**
  The handling PB performs the data interchange in both directions. The required times are to be taken from the description of the handling PB.
- **6666666666666666666**
  The answer package is transferred. Also the length of this package is variable. As soon as the package is completely received by the PCS, a new package is assembled, as described under 1.

If a (minimum) scan-time (without the time required for the handling PB) is specified, the PCS calculates the package length for 2 and 6 during each cycle, to keep time 4 as small as possible.
This logic functions only within certain limit values. So all tasks are guaranteed to be included in the package with a scan time of 60 milliseconds and up and a baud rate of 19200 baud. With a scan time specification of 0 only 1 sub-packet will be included in each data interchange cycle. The average length (without variable transfer) then totals 5.3 words (1.3 headers + 4 words of data).
The specification must be made in multiples of 1/100 seconds. Limit values are 0..50 * 1/100 seconds, the default value totals 400 milliseconds.

Data transfer PCS 830 and I/O BUS

The PCS 830 uses respectively 4 addresses in the additional I/O area (EZ/AZ) on the I/O bus. These addresses have the following meaning (only for diagnosis purposes):

1. Basis address +0 writing (AZ) = Board reset (hardware reset)
   Basis address +0 reading (EZ) = presently without function

2. Basis address +1 writing (AZ) = Reset during communication (software reset)
   Basis address +1 reading (EZ) = Status channel

   Bit 7 = 1 (Frame ready)-board is ready for data inter - change
   Bit 6 = 1 (Fetch list)-request of the cross-reference list
   Bit 5 = 1 (Watchdog)-watchdog has been activated
   Bit 4...3 = presently without function

3. Basis address +2,3 writing (AZ) + reading (EZ) = Data channel
   Here, the data will be submitted.

The data transfer is managed by the enclosed -PCSKOMM PB, so no evaluation is required by the user. The indicated meanings are only relevant for diagnosis purposes. However, attention must be paid that the user program never accesses the PCS 830 board. Otherwise correct communication between the PCS 009/090/095/900/9000/9100/PCS plus and PCS 830 is not guaranteed anymore.
This start behavior must be observed when presetting the transfer data block.

A LAUER driver

A2.1 Description PCS 830.3/.1 handling

Settings of the board
adressing (6-segment DIL switch)

The PCS board 830.3 is an interface module with three built-in TTY (20 mA) interfaces (active or passive). The PCS 830.1 is identical but it is only equipped with 1 TTY interface (2 current sources).
It is used for the communication between a BOSCH programmable controllers CL300, CL400, CL500 or PC600 and one (PCS 830.1) or three (PCS 830.3) operating consoles. The PCS 830.3 and PCS 830.1 boards are slot-independant. The PCS 830.3 uses altogether 12 additional input bytes and additional output bytes (EZ/AZ). 4 EZ/AZ addresses are required for each interface (channel). The PCS 830.1 requires exactly 4 EZ/AZ addresses.
Each channel can be freely addressed in steps of 4 via one 6-segment DIL-switch. Please take care that the set address does not intersect with already used addresses or channels! Refer to section 2.1 (Specifications of the PCS 830.3/1) for the position of the DIL-switch segments of the corresponding channel.

Selection of the address
Among others, the areas EZ0 to EZ63 and AZ0 to AZ63 are reserved for analog I/O boards. The PCS 830.3/31 board is also addressed in this area. The base address can be set to 0 and to all addresses divisible by 4. Input and output addresses (EZ/AZ) must lie in parallel, i.e. they can not be differently set.

Attention!
Take note, that after re-addressing of the board you must adjust the first three words of the cross-reference data.

Example Setting of the base address to EZ/AZ 4 (EZ4-EZ7, AZ4-AZ7)

6-segment DIL-switch:

DIL no.:	6	5	4	3	2	1
	A7	A6	A5	A4	A3	A2
Valence:	128	64	32	16	8	4

A LAUER driver

Address assignment table

The following table contains the possible address settings for the 3 channels of the PCS 830.3 (or 1 channel for the PCS 830.1). Take note, that DIL 5 and DIL 6 must be set to OFF (logical 0) for single processor systems (CL300, PC 600). The assignment is set to the corresponding ZE central unit for a CL500 multi-processor system.

EZ/AZ	D0W	D1W	D2W	DIL6	DIL5	DIL4	DIL3	DIL2	DIL1	CL500,	ZE:
0-3	0E02	1002	0E01	0	0	0	0	0	0	0	
4-7	0E06	1006	0E05	0	0	0	0	0	1	0	D4
8-11	0E0A	100A	0E09	0	0	0	0	1	0	0	D5
12-15	0E0E	100E	0E0D	0	0	0	0	1	1	0	D6
16-19	0E12	1012	0E11	0	0	0	1	0	0	0	
20-23	0E16	1016	0E15	0	0	0	1	0	1	0	
24-27	0E1A	101A	0E19	0	0	0	1	1	0	0	
28-31	0E1E	101E	0E1D	0	0	0	1	1	1	0	
32-35	0E22	1022	0E21	0	0	1	0	0	0	0	
36-39	0E26	1026	0E25	0	0	1	0	0	1	0	
40-43	0E2A	102A	0E29	0	0	1	0	1	0	0	
44-47	0E2E	102E	0E2D	0	0	1	0	1	1	0	
48-51	0E32	1032	0E31	0	0	1	1	0	0	0	
52-55	0E36	1036	0E35	0	0	1	1	0	1	0	
56-59	0E3A	103A	0E39	0	0	1	1	1	0	0	
60-63	0E3E	103E	0E3D	0	0	1	1	1	1	0	
0-60	0EXX	10XX	0EXX	0	1	x	x	x	x	1	*)
0-60	0EXX	10XX	0EXX	1	0	x	x	x	x	2	*)
0-60	0EXX	10XX	0EXX	1	1	x	x	x	x	3	*)

0: logical 0 (OFF)
1: logical 1 (ON)
x: Arbitrary setting
*): only allowed for CL500 (DIL 1..4 settings depend on the EZ/AZ address -> refer to ZE central unit settings)
D4..D6: Factory set state and address setting as an example (channel 1..3 D4..D6)
D0..2W: RDADR/WRADR/STADR (CL300 and CL500 entries in the QUVERWX data block)

A LAUER driver

Channel parameters
(8-segment DIL-switch)

Using one 8-segment DIL-switch, parameter settings for the communication between the PCS-operating console and the PCS 830.3 can be separately made for each channel (at the PCS 830.1 likewise only one channel). The position of the DIL-switch segments for the corresponding channel can be take from section 1.16 and 1.17 (Specifications). The following settings can be made via these DIL-switch segments:

- Maximum number of allowed repeat requests WDHA; DIL 5-8
- Diagnosis function DIAG (test of the received cross-reference list); DIL 4
- Baud rate BAUD for the communication; DIL 1,2

Please take note that these switch segments are only read after a restart or after a communications reset. DIL 3 is reserved for later expansions and should be set for compatibility reasons to OFF.

Example and factory set state (channel 1..3 D1..D3):
Number of repeat requests: 1 max.
Diagnosis function: OFF
Baud rate: 19200 baud

8-segment DIL-switch:

BAUD	WDHA	DIAG	DIL8	DIL7	DIL6	DIL5	DIL4	DIL3	DIL2	DIL1
1200	X	OFF	X	X	X	X	0	0	0	0
4800	X	OFF	X	X	X	X	0	0	0	1
9600	X	OFF	X	X	X	X	0	0	1	0
19200	X	OFF	X	X	X	X	0	0	1	1
X	1	OFF	0	0	0	0	0	0	X	X
X	2	OFF	0	0	0	1	0	0	X	X
X	3	OFF	0	0	1	0	0	0	X	X
X	4	OFF	0	0	1	1	0	0	X	X
X	5	OFF	0	1	0	0	0	0	X	X
X	6	OFF	0	1	0	1	0	0	X	X
X	7	OFF	0	1	1	0	0	0	X	X
X	8	OFF	0	1	1	1	0	0	X	X
X	9	OFF	1	0	0	0	0	0	X	X
X	10	OFF	1	0	0	1	0	0	X	X
X	11	OFF	1	0	1	0	0	0	X	X
X	12	OFF	1	0	1	1	0	0	X	X
X	13	OFF	1	1	0	0	0	0	X	X
X	14	OFF	1	1	0	1	0	0	X	X
X	15	OFF	1	1	1	0	0	0	X	X
X	16	OFF	1	1	1	1	0	0	X	X
X	X	ON	X	X	X	X	1	0	X	X

0: logical 0 (OFF)
1: logical 1 (ON)
X: arbitrary

A LAUER driver

Status indicators

The PCS 830.3/31 features a 7-segment „ST.x“ display which informs about the state of the channel.

The following display states are currently possible:

„0“ programmable controller is set to „STOP“, i.e. the PCS 830.3/31 waits for the „RUN“ signal and/or for the outputs to be enabled. This display is simultaneously shown for all three channels.

„1“ The corresponding channel waits for the complete reception of the cross-reference list. This channel was not parameterized correctly, or the address (6-segment DIL) was incorrectly set, or the first three words of the cross-reference data block have been incorrectly set if this display does not extinguish.

„2“ (flashing). The corresponding channel is set to the DIAG diagnosis function. The cross-reference list was correctly received. This function is only possible along with the CL300 and CL500 and the examples contained on the floppy disk.

„7“ TIMEOUT fault - time infringement of the corresponding channel. Interruption of the communication between the PCS 830.3/31 and the PCS operating console. Possible fault conditions:

PCS operating console: „TIMEOUT“

Cause: Cable break or the handling block has not been processed for sometimes.

Remedy: Check the cabling and select the handling block cyclically. Extend the timeout time of the operating console if possible. PCS operating console: „CONNECT(ION)“ or „TOO MANY REPETITIONS“.

Cause: The operating console has received multiple times wrong data (following another).

Remedy: Checking the noise suppression measures and increase the WDHA if possible.

„8“ CONNECT fault, i.e. the corresponding channel of the PCS 830.3 has received multiple times wrong data (following another) (the PCS operating console shows „TIMEOUT“). In this case, noise suppression measures are to be explored and/or to be improved if necessary!

„9“ WATCHDOG fault, i.e. either the PCS 830.3/31 board is located in a very noisy environment, or it is defective (communication cannot be restarted). If at all, this display appears for all channels simultaneously.

„-“ The corresponding channel has either sent or received a repeat request. With a reparabel fault, the „-“ extinguishes subsequently again. If the fault was not reparabel, a „7“ (TIMEOUT) or „8“ (CONNECT) is subse-quently displayed. Constant flickering of this segment informs about a very noisy environment.

A LAUER driver

„." This point lights as so soon as the corresponding channel has received a tested package and is thus ready for the data interchange between the PCS 830.3/31 and the programmable controller. With an increasing data throughput this point will be dimmed (scan time depended). This can be influenced by specifying the ZV, S,... timeout parameter and the „AA" driver parameter. This determines both the time monitoring for the communication as well as the maximum packet length for the serial transfer. For some PCS operating consoles, this time can also be influenced by deactivating the transfers (LEDs, message blocks).

„F" (flashing). The corresponding channel is set to the DIAG diagnosis function. The cross-reference list was incorrectly received. This function is only possible along with the CL300 and CL500 and the examples con-tained on the floppy disk.

A2.2 Description of the handling software

The following directories are located on the floppy disk:

CL300:	3P83110.300	for CL300
CL400:	4P83110.C00	for CL400
CL500:	5P83110.500	for CL500
PC600:	6P83110.600	for PC600

The enclosed -PCSKOMM PB (for the PC600 it is an FB) determines the data transfer between the transfer area (a data block by default) and the PCS 830.3. The data block can be parameterized and can be used for all three channels of the PCS 830.3. It can also be used for the PCS830.1. Call examples along with the handling blocks are available on the floppy disk. The README.DOC file contains current notes concerning the use of the various projects. The calls of the handling block for the interface two and three must be removed if you use the PCS 830.1.

1. The handling block uses the MBs 246 to 255 (PC600: MBs 502 to 512) as temporary flags. Writing to these MWs does not influence the handling block as long as they are not simultaneously used by interrupt programs. The contents must be written newly on entring the PB and saved on exit if these MWs are required by other PBs or FBs. In no case interrupt programs are allowed to write to these MWs. This also applies to the parameterizable INIT PB or FB which is called at communication start. The use of these flag bytes in this block is not recommended.
2. The transfer area may not be specified within the data block when using EPROM modules (for the programmable controller program). In this case, the cross-reference DBs must be adjusted in such a way that the system area and the used variables are located in the RAM area of the CPU (for instance in the data buffer).

A LAUER driver

The implementation is limited (besides the cyclical call of the PCSKOMM PB) to reading of and writing to the DWs in the transfer data block (only with default assignment!). The presence of the (selectable) transfer data block is not examined. It must be present in the programmable controller with the required net length (dynamic). Otherwise the programmable controller shows a fault (possibly only at the first call of a present value). A general fault bit ERORX (P3) enables the communication status to be evaluated by the ladder diagram. The data transfer has been stopped if this fault bit is set to 1. It can be restarted by setting the RSETX (P2) flag. Furthermore, another flag FIRSTRUX (P4) for the restart is required. This is set after the first successful processing of the handling block. The restart flag must be reset in start OBs (as in the example on the floppy disk).

Customer-specific blocks (PBs) are selected at 2 positions in the PCSKOMM block:

1: (INIT) customer-specific presetting for the communication start

2: (COFF) customer-specific assignment for the communication loss

They are to be created and/or to be extended (perhaps separately per channel).

Attention!
These blocks are of course available on the floppy disk (common for all channels). However, they contain only one „HLT“ command with subsequent examples for the corresponding PCS operating consoles.

You should delete the „HLT“ command and extend the block corresponding to the used operating console before you execute the example. If this is not considered, the CPU enters the STOP state with the status „1“ shown on the display.

Scan time extension

The processing time of the handling block depends on several elements. This is for one the specification of the programmable controller scan-time (ZV, S,.. and/or „AA“ driver parameter) in the operating console. If the specified time is sufficiently large compared to the selected baud rate (e.g. exceeding 500 ms at 19200 baud), all cyclical tasks of the operating console will be contained in each data packet i.e. all tasks are cyclically replaced. On the other hand the number of the words to be replaced depends on the number of variables (present values) in the display and the message words to be transferred. This number is adjustable for some PCS operating consoles.

Control	**Process time/ch.**
CL300	approx. 10 ms
CL400/500	approx. 650 µs
PC600	approx. 3 ms

The response is optimal if the scan time is specified sufficiently large (in the PCS) with the -PCSKOMM PB finding a PCS 830.3/1 board ready for data interchange in each cycle.

A LAUER driver

The test and the repetition of the packages is organized in the PCS 830.3/1.
The design of the PCSKOMM handling block contains provisions for aborting the communication on every coccuring fault and signalling it to the outside. After this, communication must be explecitely restarted using the RSETx input (P2 parameter). Of course, this does not affect repetitions caused by reparable faults as these are treated in the PCS 830.3/1 internally.

A2.3 Parameters of the PCSKOMM PB'S (FB'S)

Attention!
Check the action/reaction of the PLC!
Check the desired action/reaction of the programmable controller after a restart following a communication interruption to avoid unwanted malfunctionings.

P0: Cross-reference data block (here: -QUVERW1..QUVERW3)
The cross-reference addresses are listed in this block. At the restart, these addresses will be transferred once into the PCS 830.3/31. It is ensured, that the download of the cross-reference list of is made in the first cycle of the programmable controller. In the CL 300 this first cycle is extended due to the download by approx. 100 ms. During the following cycles, the PCSKOMM PB receives the corresponding cross-reference address directly from the PCS 830.3/1 (time saving).

When using several channels and/or PCS 830.3/1 boards, scan time problems can be created along with the user program. Therefore, the handling block contains the additional P5 bit parameter. This bit signals the user program that the cross-reference list was transferred in the just finished programmable controller cycle. This way it can be prevented that several cross-reference lists are transferred in the same programmable controller cycle. This has been realized in the example.

CL300, CL500: block parameters (example - QUVERW1, DB0)

PC600: constant K00XXH (example - QUVERW1, K0H); xx = DBnr.

P1: Communication data block (here: -PCSDB1..PCSDB3)
This block is active during processing of the handling block. Cross-reference addresses specified in the cross-reference data block and addressing a data word relate to the here indicated data block.

CL300, CL500: block parameters (example - PCSDB1, DB1)

PC600: constant K00XXH (example - PCSDB1, K1H); xx = DBnr.

P2: Switch and/or pushbutton for resetting following a fault (here: RSET1.. RSET3)
On a communication fault, communications of the corresponding channel can be restarted using this bit. The faults indicated in DW3 (default assignment) are then reset automatically. Please note that no edge evaluation is made. I.e. with the P2 bit set communication is auto-matically restarted after appearing of a fault! This mechanism can also be realized by connecting the P3 fault output directly to the P2 reset.

CL300, CL500, PC600: bit parameters (example - RSET1, E0.0)

A LAUER driver

P3: Genral error message (here: -EROR1..EROR3)
The P3 general fault bit is set for any error in the connection between the programmable controller and the PCS operating console. DW3 (default assignment) can be evaluated for the exact cause of error. This bit is reset as soon as the first packet has been correctly received.

CL300, CL500, PC600: bit parameters (example - EROR1, A0.0)

P4: Flag for the first processing of the handling block (here: FIRSTRU1.. FIRSTRU3)
This flag should be reset before the first call (restart) of the handling block. It is automatically set during the first execution of the PB. Among other things, it is used for presetting during a restart.

CL300, CL500, PC600: bit parameters (example - FIRSTRU1, M20.0)

P5: Flag for cross-reference data block has been transferred in the current cycle (here: -QVL_OUT)
Using this common (global) bit it can be prevented that the entire cross-reference lists are transferred in the first cycle if several channels and/or PCS 830.3/31 interface modules are inserted. This bit is set in the handling block as soon as the cross-reference list of a channel is transferred.

Note!
The transfer of the cross-reference list happens only once at the start. To prevent a scan time infringement, the transfer of the cross-reference lists can be distributed across the first programmable controller cycles if several channels are used.

Description of the execution sequence (as in the example realized):

This bit is reset before any of the PCS handling blocks are called. This bit must be scanned starting with the call of the second handling block. The following handling blocks may not be called if this bit is set to logical 1.

CL300, CL500, PC600: bit parameters (example - QVL _ OUT, M20.3)

P6: Presetting PB (here: -INIT)
Here, the PB must be specified which should be called at the communication start of the corresponding channel. This block must be generated for customer-specific presettings. This block is of course available on the floppy disk (once for all three channels) but it contains only one „HLT“ instruction which will switch the CPU to STOP (status „1“) with the remark to program the block.

CL300, CL500: block parameters (example -INIT, PB1)

PC600: constant K80XXH (example -INIT, K8000H); xx = block no.

A LAUER driver

P7: Presetting PB (here: -COFF)

Here, the PB must be specified which should be called at the communication start of the corresponding channel. This block must be generated for customer-specific presettings. This block is of course available on the floppy disk (once for all three channels) but it contains only one „HLT“ instruction which will switch the CPU to STOP (status „1“) with the remark to program the block.

CL300, CL500: block parameters (example - COFF, PB1)

PC600: constant K80XXH (example - COFF, K8001H); xx = block no.

A2.4 Set-up of the cross-reference data block

A cross-reference data block contains 256 decimal word addresses and must be specified for each interface (channel). During a restart, these addresses are stored by the download procedure on the PCS 830.3/1 board. Values for these addresses must be taken from the CL300, CL500, or PC600 software manual (operation list). The address pointers are required for indirect addressing. The PCS 830.3 board addresses must be specified in the first 3 words (0-2). In the following words (4-255) the source and/or destination addresses are specified for each word. Therefore, it is possible to assign the system area command word (DW 14) directly to a flag word without copying the flag word to the data word and conversely using a transfer command.

The following table shows a section for detailing the structure of the cross-reference data block:

- **CL300, CL400, CL500**

```
          No.          Symbol      Type  Sg          Data field  F
D         0      RDADR Word  N     0E06        H
D         2      WRADR Word  N     1006        H
D         4      STADR Word  N     0E05        H
D         6      ERADR Word  N     0806        H
D         8      W4    Word  N     0808        H

DB 2 name: QUVERW2 comment: Cross-reference db PCS2     RAM/
EPROM:   R
   No.          Symbol      Type  Sg          Data field  F
D         0            Word  N     0E0A        H
D         2            Word  N     100A        H
D         4            Word  N     0E09        H
D         6            Word  N     0806        H
D         8      W4    Word  N     0808        H

DB 4 name: QUVERW3 comment: Cross-reference db PCS3     RAM/
EPROM:   R
    No.          Symbol      Type  Sg          Data field  F
D         0            Word  N     0E0E        H
D         2            Word  N     100E        H
D         4            Word  N     0E0D        H
D         6            Word  N     0806        H
D         8            Word  N     0808        H
```

A LAUER driver

Example (CL300):

```
D        44    W22   Word  N     2092  ;corresponds to D22W    D
D        46    W23   Word  N     0256  ;corresponds to M0W (flag
word 0)  D
D        48    W24   Word  N     0194  ;corresponds to A1W
(output word 1)      D
D        50    W25   Word  N     2098  ;corresponds to D27W    D
D        52    W26   Word  N     2100  ;corresponds to D26W    D
```

Attention!
Take note, that after re-adressing of the board (6-seg. DIL-switch) you must adjust the first three words of the cross-reference data.

Word no.:Meaning:
D0 RDADR: board base address +2 reading (EZ)
D2 WRADR: board base address +2 writing (AZ)
D4 STAADR: board base address +1 reading (EZ)

- **PC600**

```
   DB  0 name: QUVERW1 comment: Cross-reference db PCS1 RAM/
EPROM:   R
   No.          Symbol      Type  Sg          Data field  F
D        0      RESADR      Word  N     0104        H
D        2      STATADR     Word  N     0105        H
D        4      WR/RDADR    Word  N     0106        H
D        6      ERADR Word  N     3843        D
D        8      W4    Word  N     3844        D

DB 2 name: QUVERW2 comment: Cross-reference db PCS2     RAM/
EPROM:   R
   No.          Symbol      Type  Sg          Data field  F
D        0           Word  N     0108        H
D        2           Word  N     0109        H
D        4           Word  N     010A        H
D        6           Word  N     3843        D
D        8           Word  N     3844        D

DB 4 name: QUVERW3 comment: Cross-reference db PCS3     RAM/
EPROM:   R
    No.          Symbol      Type  Sg          Data field  F
D        0           Word  N     010C        H
D        2           Word  N     010D        H
D        4           Word  N     010E        H
D        6           Word  N     3843        D
D        8           Word  N     3844        D
```

Attention!
Take note, that after re-adressing of the board (6-seg. DIL-switch) you must adjust the first three words of the cross-reference data.

Word no.:Meaning:
D0 RESADR: board base address +0 (EZ/AZ)
D2 STATADR: board base address +2 (EZ/AZ)
D4 STAADR: board base address +1 (EZ/AZ)

A LAUER driver

- **CL500**

Attention!
The entire system (all processors) must be stopped and restarted after changing the cross-reference list. Otherwise the cross-reference list will not be newly transferred.

No outputs (independantly of the processor) may be disabled when starting the entire system. This would result in a faulty reception of the cross-reference list.

A2.5 Operation the PLC with EPROM/EEPROM

When using EPROM and EEPROM memory modules in the programmable controller, the communication area cannot be located within one data block since these data words are fixed in EPROM/EEPROM and thus cannot be manipulated anymore (write operations are not possible).
The only possibility consists in altering the cross-reference data block so that the entire system area (refer to the PCS operating console manual) and the used variable area are located in a modifiable memory area (e.g. flag area, data buffers etc.).

A2.6 Implementation of the handling PB/FB

1. Power-down the programmable controller
2. Define the PCS 830.3/1 address, the baud rate and the number of repitions using the DIL-switches (6-segment and 8-segment). DIL 4 must be set to OFF (diagnosis function).
3. plug-in the board
4. Set the programmable controller to stop and apply power
5. Re-adjust the cross-reference data block addresses if nece
6. Reset P4 (flag for restart) in the starting OBs (for the CL 300 this is OB 7 and OB 8)
7. Select and parameterize the -PCSKOMM handling PB (FB) of OB1 (or anywhere else). This is perhaps required by QVL_OUT (P5) when using several channels.
8. Determine reset pushbutton and „wire „with P2 (reset)
9. Program -INIT and -COFF presetting blocks (perhaps for each channel separately)

Attention!
These blocks are of course available on the floppy disk (common for all channels). However, they contain only one „HLT“ command with subsequent examples for the corresponding PCS operating consoles.

10. Assign, link, and download all data blocks into the controller
11. Switch the programmable controller to RUN

A LAUER driver

The implementation has been successfully completed if the K8000H fault (time-out - no PCS connected) appears in DW3 (default assignment) and the flags P4 (restart) and P3 (general error message) are both set to logical 1. In addition, the PCS 830.3/1 must indicate the „STATUS“ value „7“ for all accessed channels.
An example (OB1) is contained on the floppy disk which indicates a fault on output A0.0..0.2, expects a reset pushbutton on E0.0..E0.2, uses the flag M20.0..M20.3 and addresses the PCS 830.3/1 board as EZ/AZ 4-15 (channel 1: 4-7, channel 2: 8-11, channel 3: 12-15). The baud rate is set to correspond to the settings of the 8-segment DIL.

A2.6 Program integration

The following transfer operations must be performed for the integration into your program (not required functions can be left out):

- Before calling the -PCSKOMM handling block all values read by the PCS must be copied into the corresponding DWs. This affects:

  All words read by the PCS (refer to the PCS operating console manual)

  But only if you do not directly access the corresponding words by altering the cross-reference data block (example: flag words, data buffers, input words etc.).

- After calling the -PCSKOMM handling block all values changed by the PCS must be written back to flags. This affects:

  All words written to by the PCS (refer to the PCS operating console manual)

  Dut only if the cross-reference data block has not been adjusted.

A LAUER driver

A3 Program listing

Tip!
Inspect the function of the handling software to avoid failures of the PCS or PLC.

Statement listing CL300

Prg. contr. documentation Bosch CL300 Version 3.30z Date: Apr. 4, 1995
Projekt: 3P83110/ZS0 File: OB1.P3O (OB1) Page: 2

```
      PZ:   1
1     UN    B     -QVL_OUT
2     U     B     -QVL_OUT
3     =     B     -QVL_OUT

      ;*****HANDLING BLOCK PCS1*****
4     BA          -PCSKOMM,8
      P0          -QUVERW1    ;CROSS-REFERENCE DATA BLOCK
      P1          -PCSDB1     ;COMMUNICATIONS DATA BLOCK
      P2    B     -RSET1      ;SWITCH FOR RESTART AFTER A COMMUNICATION ERROR
      P3    B     -EROR1      ;GENERAL ERROR MESSAGE
      P4    B     -FIRSTRU1   ;FLAG FOR FIRST SCAN (-PCSKOMM)
      P5    B     -QVL_OUT    ;FLAG FOR TRANSFER CROSS-REFERENCE DATA BLOCK
      P6          -INIT ;INIT (RESTART PRESETTING;COMMUNICATIONS START)
      P7          -COFF ;COFF (FAULT PRESETTING;COMMUNICATIONS LOSS)

      ;*****HANDLING BLOCK PCS2*****
      PZ:   2
5     UN    B     -QVL_OUT    ;ONE XREF DATA BLOCK PER SCAN CYCLE ->PCS830.3
6     BAB         -PCSKOMM,8  ;*****HANDLING BLOCK PCS2*****
      P0          -QUVERW2    ;CROSS-REFERENCE DATA BLOCK
      P1          -PCSDB2     ;COMMUNICATIONS DATA BLOCK
      P2    B     -RSET2      ;SWITCH FOR RESTART AFTER A COMMUNICATION ERROR
      P3    B     -EROR2      ;GENERAL ERROR MESSAGE
      P4    B     -FIRSTRU2   ;FLAG FOR FIRST SCAN (-PCSKOMM)
      P5    B     -QVL_OUT    ;FLAG FOR TRANSFER CROSS-REFERENCE DATA BLOCK
      P6          -INIT ;INIT (RESTART PRESETTING;COMMUNICATIONS START)
      P7          -COFF ;COFF (FAULT PRESETTING;COMMUNICATIONS LOSS)

      ;*****HANDLING BLOCK PCS3*****
      PZ:   3
7     UN    B     -QVL_OUT    ;ONE XREF DATA BLOCK PER SCAN CYCLE ->PCS830.3
8     BAB         -PCSKOMM,8  ;*****HANDLING BLOCK PCS3*****
      P0          -QUVERW3    ;CROSS-REFERENCE DATA BLOCK
      P1          -PCSDB3     ;COMMUNICATIONS DATA BLOCK
      P2    B     -RSET3      ;SWITCH FOR RESTART AFTER A COMMUNICATION ERROR
      P3    B     -EROR3      ;GENERAL ERROR MESSAGE
      P4    B     -FIRSTRU3   ;FLAG FOR FIRST SCAN (-PCSKOMM)
      P5    B     -QVL_OUT    ;FLAG FOR TRANSFER CROSS-REFERENCE DATA BLOCK
      P6          -INIT ;INIT (RESTART PRESETTING;COMMUNICATIONS START)
      P7          -COFF ;COFF (FAULT PRESETTING;COMMUNICATIONS LOSS)
```

Prg. contr. documentation Bosch CL300 Version 3.30z Date: Apr. 4, 1995
Projekt: 3P83110/ZS0 File: OB7.P3O (OB7) Page: 4

```
      PZ:   1
1     U     B     -FIRSTRU1
2     UN    B     -FIRSTRU1
3     =     B     -FIRSTRU1   ;FLAG FOR FIRST SCAN PCS1 SET TO ZERO
      PZ:   2
4     U     B     -FIRSTRU2
```

A LAUER driver

```
5     UN    B     -FIRSTRU2
6     =     B     -FIRSTRU2   ;FLAG FOR FIRST SCAN PCS2 SET TO ZERO
      PZ:   3
7     U     B     -FIRSTRU3
8     UN    B     -FIRSTRU3
9     =     B     -FIRSTRU3   ;FLAG FOR FIRST SCAN PCS3 SET TO ZERO
10    BE
```

Prg. contr. documentation Bosch CL300 Version 3.30z Date: Apr. 4, 1995
Projekt: 3P83110/ZS0 File: OB8.P3O (OB8) Page: 5

```
      PZ:   1
1     U     B     -FIRSTRU1
2     UN    B     -FIRSTRU1
3     =     B     -FIRSTRU1   ;FLAG FOR FIRST SCAN PCS1 SET TO ZERO
      PZ:   2
4     U     B     -FIRSTRU2
5     UN    B     -FIRSTRU2
6     =     B     -FIRSTRU2   ;FLAG FOR FIRST SCAN PCS2 SET TO ZERO
      PZ:   3
7     U     B     -FIRSTRU3
8     UN    B     -FIRSTRU3
9     =     B     -FIRSTRU3   ;FLAG FOR FIRST SCAN PCS3 SET TO ZERO
10    BE
```

Prg. contr. documentation Bosch CL300 Version 3.30z Date: Apr. 4, 1995
Projekt: 3P83110/ZS0 File: INIT.P30 (PB1) Page: 10

```
;*************************************************************************************************
      ;** INIT (APPLICATION-SPECIFIC PRE-ASSIGNMENTS; PCS COMMUNICATIONS START)

;*************************************************************************************************

1     HLT
      ;*********************************PRE-ASSIGNMENTS FOR COMMUNICATIONS START
      ;*********************************INSERT OR ADD !!!!!

      ;*****EXAMPLE PCS 090/095*****
2     L     W     K0FC8H,A
3     T     W     A,D13W      ;COMMAND WORD A
4     L     W     K0080H,A
5     T     W     A,D14W      ;COMMAND WORD B
6     BE
      ;*****EXAMPLE PCS 100*****
7     L     W     K0D,A
8     T     W     A,D6W ;DATA
9     L     W     K0F00H,A
10    T     W     A,D14W      ;COMMAND WORD
11    BE
      ;*****EXAMPLE PCS 200 AND PCS 300*****
12    L     W     K0D,A
13    T     W     A,D7W ;DATA
14    T     W     A,D19W      ;COMMAND WORD C
15    L     W     K0080H,A
16    T     W     A,D18W      ;COMMAND WORD B
17    L     W     K0F00H,A
18    T     W     A,D17W      ;COMMAND WORD A
19    BE
```

A LAUER driver

Statement listing CL400/CL500 — Since the listing of the CL400 is absolutely identical to the CL500 it is not printed here.

Prg. contr. documentation — Bosch CL500 Version 3.30z — Date: Apr. 4, 1995
Projekt: 5P83110/ZS0 — File: OB1.P5O (OB1) — Page: 2

```
      DEF         SM30.3.-LOGO
      PZ:   1
1     U     B     -LOGO
2     =     B     -QVL_OUT

      ;*****HANDLING BLOCK PCS1*****
3     BA          -PCSKOMM,8
      P0          -QUVERW1    ;CROSS-REFERENCE DATA BLOCK
      P1          -PCSDB1     ;COMMUNICATIONS DATA BLOCK
      P2    B     -RSET1      ;SWITCH FOR RESTART AFTER A COMMUNICATION ERROR
      P3    B     -EROR1      ;GENERAL ERROR MESSAGE
      P4    B     -FIRSTRU1   ;FLAG FOR FIRST SCAN (-PCSKOMM)
      P5    B     -QVL_OUT    ;FLAG FOR TRANSFER CROSS-REFERENCE DATA BLOCK (=1)
      P6          -INIT ;INIT (RESTART PRESETTING;COMMUNICATIONS START)
      P7          -COFF ;COFF (FAULT PRESETTING;COMMUNICATIONS LOSS)

      ;*****HANDLING BLOCK PCS2*****
      PZ:   2
4     UN    B     -QVL_OUT    ;ONE XREF DATA BLOCK PER SCAN CYCLE ->PCS830.3
5     BAB         -PCSKOMM,8
      P0          -QUVERW2    ;CROSS-REFERENCE DATA BLOCK
      P1          -PCSDB2     ;COMMUNICATIONS DATA BLOCK
      P2    B     -RSET2      ;SWITCH FOR RESTART AFTER A COMMUNICATION ERROR
      P3    B     -EROR2      ;GENERAL ERROR MESSAGE
      P4    B     -FIRSTRU2   ;FLAG FOR FIRST SCAN (-PCSKOMM)
      P5    B     -QVL_OUT    ;FLAG FOR TRANSFER CROSS-REFERENCE DATA BLOCK (=1)
      P6          -INIT ;INIT (RESTART PRESETTING;COMMUNICATIONS START)
      P7          -COFF ;COFF (FAULT PRESETTING;COMMUNICATIONS LOSS)

      ;*****HANDLING BLOCK PCS3*****
      PZ:   3
6     UN    B     -QVL_OUT    ;ONE XREF DATA BLOCK PER SCAN CYCLE ->PCS830.3
7     BAB         -PCSKOMM,8
      P0          -QUVERW3    ;CROSS-REFERENCE DATA BLOCK
      P1          -PCSDB3     ;COMMUNICATIONS DATA BLOCK
      P2    B     -RSET3      ;SWITCH FOR RESTART AFTER A COMMUNICATION ERROR
      P3    B     -EROR3      ;GENERAL ERROR MESSAGE
      P4    B     -FIRSTRU3   ;FLAG FOR FIRST SCAN (-PCSKOMM)
      P5    B     -QVL_OUT    ;FLAG FOR TRANSFER CROSS-REFERENCE DATA BLOCK (=1)
      P6          -INIT ;INIT (RESTART PRESETTING;COMMUNICATIONS START)
      P7          -COFF ;COFF (FAULT PRESETTING;COMMUNICATIONS LOSS)

8     PE
```

Prg. contr. documentation — Bosch CL500 Version 3.30z — Date: Apr. 4, 1995
Projekt: 5P83110/ZS0 — File: OB1.P3O (OB1) — Page: 4

```
      DEF         SM30.3.-LOGO
      PZ:   1
1     U     B     -LOGO
2     =     B     -FIRSTRU1   ;FLAG FOR FIRST SCAN PCS1 SET TO ZERO
3     =     B     -FIRSTRU2   ;FLAG FOR FIRST SCAN PCS2 SET TO ZERO
4     =     B     -FIRSTRU3   ;FLAG FOR FIRST SCAN PCS3 SET TO ZERO
5     BE
```

A LAUER driver

Prg. contr. documentation Bosch CL500 Version 3.30z Date: Apr. 4, 1995
Projekt: 5P83110/ZS0 File: OB6.P5O (OB6) Page: 5

```
      DEF         SM30.3.-LOGO
      PZ:   1
1     U     B     -LOGO
2     =     B     -FIRSTRU1   ;FLAG FOR FIRST SCAN PCS1 SET TO ZERO
3     =     B     -FIRSTRU2   ;FLAG FOR FIRST SCAN PCS2 SET TO ZERO
4     =     B     -FIRSTRU3   ;FLAG FOR FIRST SCAN PCS3 SET TO ZERO
5     BE
```

Prg. contr. documentation Bosch CL500 Version 3.30z Date: Apr. 4, 1995
Projekt: 5P83110/ZS0 File: OB7.P5O (OB7) Page: 6

```
      DEF         SM30.3.-LOGO
      PZ:   1
1     U     B     -LOGO
2     =     B     -FIRSTRU1   ;FLAG FOR FIRST SCAN PCS1 SET TO ZERO
3     =     B     -FIRSTRU2   ;FLAG FOR FIRST SCAN PCS2 SET TO ZERO
4     =     B     -FIRSTRU3   ;FLAG FOR FIRST SCAN PCS3 SET TO ZERO
5     BE
```

Prg. contr. documentation Bosch CL500 Version 3.30z Date: Apr. 4, 1995
Projekt: 3P83110/ZS0 File: OB8.P5O (OB8) Page: 7

```
      DEF         SM30.3.-LOGO
      PZ:   1
1     U     B     -LOGO
2     =     B     -FIRSTRU1   ;FLAG FOR FIRST SCAN PCS1 SET TO ZERO
3     =     B     -FIRSTRU2   ;FLAG FOR FIRST SCAN PCS2 SET TO ZERO
4     =     B     -FIRSTRU3   ;FLAG FOR FIRST SCAN PCS3 SET TO ZERO
5     BE
```

Prg. contr. documentation Bosch CL500 Version 3.30z Date: Apr. 4, 1995
Projekt: 5P83110/ZS0 File: INIT.P5O (PB1) Page: 12

```
;*************************************************************************************************
      ;** INIT (APPLICATION-SPECIFIC PRE-ASSIGNMENTS; PCS COMMUNICATIONS START)

;*************************************************************************************************

1     HLT                       ;!!!!! INSERT OR ADD PCS AND EQUIPMENT-
      ;*********************************PRE-ASSIGNMENTS FOR COMMUNICATIONS START
      ;*********************************START HERE !!!!!

      ;*****EXAMPLE PCS 090/095*****
2     L     W     K0FC8H,A
3     T     W     A,D13W      ;COMMAND WORD A
4     L     W     K0080H,A
5     T     W     A,D14W      ;COMMAND WORD B
6     BE
      ;*****EXAMPLE PCS 100*****
7     L     W     K0D,A
8     T     W     A,D6W ;DATA
9     L     W     K0F00H,A
10    T     W     A,D14W      ;COMMAND WORD
11    BE
      ;*****EXAMPLE PCS 200 AND PCS 300*****
12    L     W     K0D,A
13    T     W     A,D7W ;DATA
14    T     W     A,D19W      ;COMMAND WORD C
15    L     W     K0080H,A
16    T     W     A,D18W      ;COMMAND WORD B
17    L     W     K0F00H,A
18    T     W     A,D17W      ;COMMAND WORD A
19    BE
```

A LAUER driver

Prg. contr. documentation Bosch CL500 Version 3.30z Date: Apr. 4, 1995
Projekt: 5P83110/ZS0 File: COFF.P5O (PB2) Page: 13

```
;*************************************************************************************************
      ;** COFF (APPLICATION-SPECIFIC FAULT PRE-ASSIGNMENTS; PCS COMMUNICATIONS LOSS)

;*************************************************************************************************

1     HLT                    ;!!!!! INSERT PCS AND EQUIPMENT-SPECIFIC
     ;********************************* FAULT PRESSTINGS FOR COMMUNICATIONS
     ;********************************* LOSS HERE !!!!!
     ;********************************* !!!!! ATTENTION: !!!!!
     ;********************************* SET AT LEAST HERE KEYWORDS TO ZERO
     ;********************************* (PCS-SPECIFIC) !
      ;*****EXAMPLE PCS 090/095*****
2     L     W     K0D,A
3     T     W     A,D4W ;KEYS
4     T     W     A,D5W
5     T     W     A,D23W
6     BE

     ;*****EXAMPLE PCS 100*****
7     L     W     K0D,A
8     T     W     A,D4W ;KEYS
9     T     W     A,D5W
10    T     W     A,D6W
11    BE

     ;*****EXAMPLE PCS 200 AND PCS 300*****
12    L     W     K0D,A
13    T     W     A,D4W ;KEYS
14    T     W     A,D5W
15    T     W     A,D6W
16    T     W     A,D7W
17    BE
```

Statement listing PC600

Prg. contr. documentation Bosch PC600 Version 3.30z Date: Apr. 4, 1995
Projekt: 6P83110/ZS0 File: OB1.P6O (OB1) Page: 2

```
      DEF         K0H, -QUVERW1
      DEF         K1H, -PCSDB1
      DEF         K2H, -QUVERW2
      DEF         K3H, -PCSDB2
      DEF         K4H, -QUVERW3
      DEF         K5H, -PCSDB3
      DEF         K8000H, -INIT
      DEF         K8001H, -COFF

      PZ:   1
1     UN    B     -QVL_OUT
2     U     B     -QVL_OUT
3     =     B     -QVL_OUT

      ;*****HANDLING BLOCK PCS1*****
4     BA          -PCSKOMM,8
      P0          -QUVERW1    ;CROSS-REFERENCE DATA BLOCK
      P1          -PCSDB1     ;COMMUNICATIONS DATA BLOCK
      P2    B     -RSET1      ;SWITCH FOR RESTART AFTER A COMMUNICATION ERROR
      P3    B     -EROR1      ;GENERAL ERROR MESSAGE
      P4    B     -FIRSTRU1   ;FLAG FOR FIRST SCAN (-PCSKOMM)
      P5    B     -QVL_OUT    ;FLAG FOR TRANSFER CROSS-REFERENCE DATA BLOCK (=1)
      P6          -INIT ;INIT (RESTART PRESETTING;COMMUNICATIONS START)
      P7          -COFF ;COFF (FAULT PRESETTING;COMMUNICATIONS LOSS)
```

A LAUER driver

```
      ;*****HANDLING BLOCK PCS2*****
      PZ:   2
5     UN    B     -QVL_OUT    ;ONE XREF DATA BLOCK PER SCAN CYCLE ->PCS830.3
6     BAB         -PCSKOMM,8
      P0          -QUVERW2    ;CROSS-REFERENCE DATA BLOCK
      P1          -PCSDB2     ;COMMUNICATIONS DATA BLOCK
      P2    B     -RSET2      ;SWITCH FOR RESTART AFTER A COMMUNICATION ERROR
      P3    B     -EROR2      ;GENERAL ERROR MESSAGE
      P4    B     -FIRSTRU2   ;FLAG FOR FIRST SCAN (-PCSKOMM)
      P5    B     -QVL_OUT    ;FLAG FOR TRANSFER CROSS-REFERENCE DATA BLOCK (=1)
      P6          -INIT ;INIT (RESTART PRESETTING;COMMUNICATIONS START)
      P7          -COFF ;COFF (FAULT PRESETTING;COMMUNICATIONS LOSS)

      ;*****HANDLING BLOCK PCS3*****
      PZ:   3
7     UN    B     -QVL_OUT    ;ONE XREF DATA BLOCK PER SCAN CYCLE ->PCS830.3
8     BAB         -PCSKOMM,8
      P0          -QUVERW3    ;CROSS-REFERENCE DATA BLOCK
      P1          -PCSDB3     ;COMMUNICATIONS DATA BLOCK
      P2    B     -RSET3      ;SWITCH FOR RESTART AFTER A COMMUNICATION ERROR
      P3    B     -EROR3      ;GENERAL ERROR MESSAGE
      P4    B     -FIRSTRU3   ;FLAG FOR FIRST SCAN (-PCSKOMM)
      P5    B     -QVL_OUT    ;FLAG FOR TRANSFER CROSS-REFERENCE DATA BLOCK (=1)
      P6          -INIT ;INIT (RESTART PRESETTING;COMMUNICATIONS START)
      P7          -COFF ;COFF (FAULT PRESETTING;COMMUNICATIONS LOSS)

9     PE
```



```
      PZ:   1
1     U     B     -FIRSTRU1
2     UN    B     -FIRSTRU1
3     =     B     -FIRSTRU1   ;FLAG FOR FIRST SCAN PCS1 SET TO ZERO
      PZ:   2
4     U     B     -FIRSTRU2
5     UN    B     -FIRSTRU2
6     =     B     -FIRSTRU2   ;FLAG FOR FIRST SCAN PCS1 SET TO ZERO
      PZ:   3
7     U     B     -FIRSTRU3
8     UN    B     -FIRSTRU3
9     =     B     -FIRSTRU3   ;FLAG FOR FIRST SCAN PCS1 SET TO ZERO

10    BE
```



```
      PZ:   1
1     U     B     -FIRSTRU1
2     UN    B     -FIRSTRU1
3     =     B     -FIRSTRU1   ;FLAG FOR FIRST SCAN PCS2 SET TO ZERO
      PZ:   2
4     U     B     -FIRSTRU2
5     UN    B     -FIRSTRU2
6     =     B     -FIRSTRU2   ;FLAG FOR FIRST SCAN PCS2 SET TO ZERO
      PZ:   3
7     U     B     -FIRSTRU3
8     UN    B     -FIRSTRU3
9     =     B     -FIRSTRU3   ;FLAG FOR FIRST SCAN PCS2 SET TO ZERO

10    BE
```

A LAUER driver


```
      PZ:   1
1     U     B     -FIRSTRU1
2     UN    B     -FIRSTRU1
3     =     B     -FIRSTRU1   ;FLAG FOR FIRST SCAN PCS3 SET TO ZERO
      PZ:   2
4     U     B     -FIRSTRU2
5     UN    B     -FIRSTRU2
6     =     B     -FIRSTRU2   ;FLAG FOR FIRST SCAN PCS3 SET TO ZERO
                                PZ:   3
7     U     B     -FIRSTRU3
8     UN    B     -FIRSTRU3
9     =     B     -FIRSTRU3   ;FLAG FOR FIRST SCAN PCS3 SET TO ZERO

10    BE
```



```
;*************************************************************************************************
      ;** INIT (APPLICATION-SPECIFIC PRE-ASSIGNMENTS; PCS COMMUNICATIONS START)

;*************************************************************************************************

1     HLT                       ;!!!!! INSERT OR ADD PCS AND EQUIPMENT-
      ;*********************************SPECIFIC PRE-ASSIGNMENTS FOR COMMUNICATIONS
      ;*********************************START HERE !!!!!

      ;*****EXAMPLE PCS 090/095*****
2     L     W     K0FC8H,A
3     T     W     A,D13W      ;COMMAND WORD A
4     L     W     K0080H,A
5     T     W     A,D14W      ;COMMAND WORD B
6     BE
      ;*****EXAMPLE PCS 100*****
7     L     W     K0D,A
8     T     W     A,D6W ;DATA
9     L     W     K0F00H,A
10    T     W     A,D14W      ;COMMAND WORD
11    BE
      ;*****EXAMPLE PCS 200 AND PCS 300*****
12    L     W     K0D,A
13    T     W     A,D7W ;DATA
14    T     W     A,D19W      ;COMMAND WORD C
15    L     W     K0080H,A
16    T     W     A,D18W      ;COMMAND WORD B
17    L     W     K0F00H,A
18    T     W     A,D17W      ;COMMAND WORD A
19    BE
```

A LAUER driver


```
;*************************************************************************************************
      ;** COFF (APPLICATION-SPECIFIC FAULT PRE-ASSIGNMENTS; PCS COMMUNICATIONS LOSS)

;*************************************************************************************************

1     HLT                    ;!!!!! INSERT PCS AND EQUIPMENT-SPECIFIC
     ;********************************* FAULT PRESSTINGS FOR COMMUNICATIONS
     ;********************************* LOSS HERE !!!!!
     ;********************************* !!!!! ATTENTION: !!!!!
     ;********************************* SET AT LEAST HERE KEYWORDS TO ZERO
     ;********************************* (PCS-SPECIFIC) !

      ;*****EXAMPLE PCS 090/095*****
2     L     W     K0D,A
3     T     W     A,D4W ;KEYS
4     T     W     A,D5W
5     T     W     A,D23W
6     BE
     ;*****EXAMPLE PCS 100*****
7     L     W     K0D,A
8     T     W     A,D4W ;KEYS
9     T     W     A,D5W
10    T     W     A,D6W
11    BE
     ;*****EXAMPLE PCS 200 AND PCS 300*****
12    L     W     K0D,A
13    T     W     A,D4W ;KEYS
14    T     W     A,D5W
15    T     W     A,D6W
16    T     W     A,D7W
17    BE
```

A LAUER driver

A4 Technical apendix

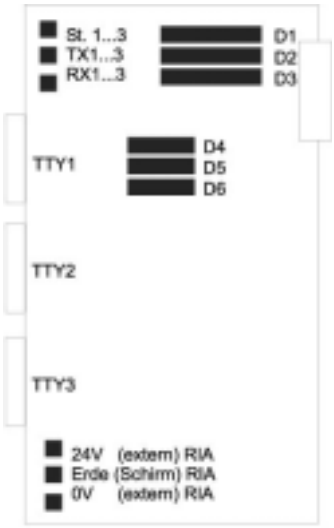
A4.1 PCS 830.3

This part of the manual contains special informations which are reqiered for the setup: cable assignment, transfer addresses and a short explanation of the data transfer using the communication port and the I/O bus.

Mechanical design	Board with 64-pole DIN 41612 „C“ connector
Power supply	Via motherboard: 12 V
Current requirement	300 mA max. (typically 250 mA)
Address allocation	12 addresses in the EZ/AZ area; 4 successive addresses (selectable via DIL-switches) in the additional I/O area (EZ/AZ) with EZ/AZ-addresses lying in parallel
Interfaces	3 x TTY (20 mA)
TTY current sources	Internal/external selectable
Potential separation	Yes, optocouplers hp 4100/4200
Dimensions	233.4 * 160 * 30 mm
Front elements	3 7-segment status displays „St.x“ 3 LED „TXx“ = transmit current 3 LED „RXx“ = receive current 3 JD female connector „TTY1“ 3-pole inline connector (RIA, 24V, earth, 0V)
Current source supply	24V DC (current requirement 50 mA max.) stabilized
Connections	24V, earth, 0V (potentially isolated)

A LAUER driver

Circuit board diagram



X1 (DIN 41612 „C“ connector)

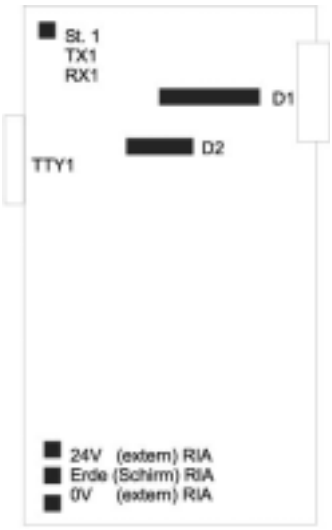
D1-3: 8-segment DILs, channel parameters (1 per cannel)
D4-6: 6-segment DILs, addressing in the EZ/AZ-field (1 per cannel)

A4.2 PCS 830.1

Mechanical design	Board with 64-pole DIN 41612 „C“ connector
Power supply	Via motherboard: 12 V
Current requirement	250 mA max. (typically 200 mA)
Address allocation	4 addresses in the EZ/AZ area; 4 successive addresses (selectable via DIL switches) in the additional I/O area (EZ/AZ) with EZ/AZ-addresses lying in parallel
Interfaces	1 x TTY (20 mA)
TTY current sources	Internal/external selectable
Potential separation	Yes, opto-couplers hp 4100/4200
Dimensions	233.4 * 160 * 30 mm
Front elements	1 7-segment status displays „St.x“ 1 LED „TXx“ = transmit current 1 LED „RXx“ = receive current 1 JD female connector „TTY1“ 3-pole inline connector (RIA, 24V, earth, 0V)
Current source supply	24V DC (current requirement 50 mA max.) stabilized
Connections	24V, earth, 0V (potentially isolated)

A LAUER driver

Circuit board diagram



X1 (DIN 41612 „C“ connector)

D1: 8-segment DIL, channel parameters
D2: 6-segment DIL, addressing
RIA (24V, earth, 0V): 3-pole inline connector for the active operation of the current sources

A4.3 Interfaces TTY1..3

Pin assignment of the TTY interfaces:

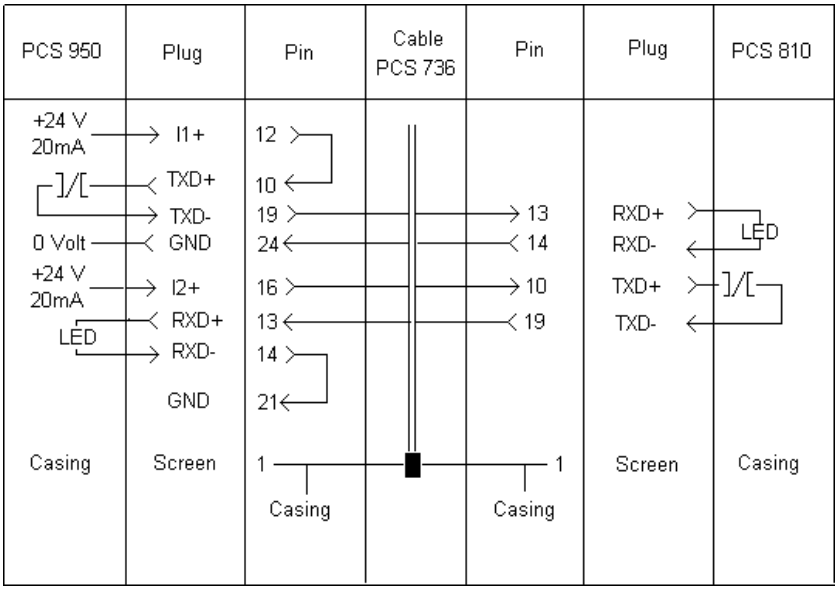
Pin no.	Meaning
Case	screen (connec. with frontplate and RIA shield. termin.)
1	„ „ „ „ „ „ „
10	TX+
12	current source 1 (20mA) active with stab. 24 V supplied
13	RX+
14	RX -
16	current source 2 (20mA) active with stab. 24 V supplied
19	TX -
21,24	0 V (external)

2 additional current sources per interface can be activated by connecting stabilized 24 VDC to the 3-pole RIA terminal if a printer without it´s own current source is connected to the PCS operating console (e.g. PCS 900). Then, these current sources supply the line current for the communication between the PCS operating console and the PCS 830.3/1 TTYx interface while the current sources built into the PCS operating console supply the line current for the printer port.
The PCS 736 cable can also be used when using these current sources. Only the cable ends are to be reversed („Prog. contr. interface“ cable end is plugged into the PCS operating console and the „PCS“ cable end is plugged into the PCS 830.3/1).

A LAUER driver

Communications cable PCS 736

For each interface, the connection is made via 2 TTY channels. The line currents for both channels are supplied by the PCS operating console. Thus, the programmable controller is galvanically isolated.



When using shielded standard cable (2 * 2 * 0.14, untwisted) the following recommended maximum length apply:

19200 baud	10 meters
9600 baud	20 meters
4800 baud	40 meters
1200 baud	160 meters

When using low-capacity data cables twisted in pairs the 10-fold lengths can be projected (example: Belden cable 8723; alternative: 2-fold foil shielded 4-wire 2*2*0,2 mm cables twisted in pairs).
Interrupting the connection is registered by the PCS operating console and shown on the PCS 830.3 display (status „St.x“). The fault is indicated in the error word (DW3) for the further evaluation.

Screening
Connector hoods are used, the shield may also be connected to pin 1, but this is not recommended due to interferences, because the data lines should be completely covered by the shield (if possible). It should be noticed, however, that grounding on both sides may require an equipotential bonding conductor with a cross section of 10 times that of the shield (in order to compensate for ground potential variations and to prevent equalizing currents on the cable shield)! This is especially important, if the PCS and the programmable controller are not connected to the same common point (if they are for example installed in different control cabinets).

A LAUER driver

A4.4 Data transfer PCS 830.x I/O BUS

The PCS 830.3/31 uses for each channel (separately settable) 4 addresses respectively in the EZ/AZ area on the I/O bus. These addresses have the following meaning (only for diagnosis purposes):

ADDRESS: base address +0
DIRECTION: writing (AZ)
-- presently not used
DIRECTION: reading (EZ)
-- presently not used

STATUS CHANNEL
ADDRESS: base address +1
DIRECTION: reading (EZ)
bit 7 = 1 (frame ready) - data interchange readiness!
bit 6 = 1 (fetch list) - cross-reference list transfer!
bit 5 = 1 (watchdog) - watchdog has been activated!
bit 4..0 presently not used
DIRECTION: writing (AZ)
-- reset during communication (software reset)
for exiting the menus (PCS 100)

DATA CHANNEL
ADDRESS: base address +2 and bass address +3 (word-by-word)
DIRECTION: reading (EZ) and writing (AZ)
COMMANDS PCS -> prog. contr.
DATA PCS <-> prog. contr.

The data transfer is managed by the enclosed PCSKOMM PB, so no evaluation is required by the user. The indicated meanings are only relevant for diagnosis purposes. However, attention must be paid that the user program never accesses the base addresses +2, +3 of the corresponding channel. Otherwise, correct communication between the PCS operating console and the PCS 830.3/31 is not guaranteed anymore. Consequences are e.g. stop of the programmable controller, corruption of the programmable controller program and data.

B Buep 19-Expander driver

B1 First commissioning

Preface

The successful parameterization of the PCS 009/090/095/PCS plus as described in the PCS manual is presupposed. This appendix relates exclusively to the use of the PCS together with the BOSCH CL300 programmable controller via the programming interface by means of the BUEP19 protocol. In the following, this controller is referred to as programmable controller and the driver to be loaded into the PCS as 3P90BUEP. The BOSCH specific terminology and the programming of the programmable controller with the BOSCH software are presupposed as known.

Required devices and accessories

The following products are required for the operation of a programmable controller with an already parameterized PCS 009/090/095/PCS plus (Systeme Lauer company):

1. The PCS PCS 009/090/095/PCS plus operating console itself (already parameterized).
2. The PCS 706 connecting cable for the connection programmable controller via the TTY interface or LCA 035/235 for the connection via the RS-232C interface.
3. PCS 091 manual plus this appendix (PCS 91.BOS)
4. Floppy disk with PCS 91.BOS handling block.

Furthermore are required (BOSCH company):

5. Programming card, programming cable, software, and the copy protection device for programming the programmable controller
6. CL300 rack (BGT 300, BGT 300-K, BGT 301, or BGT 301-K)
7. Power supply NT 300 or NT 301
8. ZE 300 or ZE 301 CPU board (with 16k or 32k RAM module)
9. Optionally a digital I/O board

Loading of the 3P90BUEP driver

Both, the application program with data and a selected driver are transferred during the configuration. The procedure described in the PCS 091 manual applies also here.

Variable settings

The following settings must be made in the OPTIONS/DRIVER PARAMETERS menu of PCSPRO software for the correct operation of the driver. These are: baud rate and interface, (DB number) and (Timeout time).

B Buep 19-Expander driver

Interfaces

Interface and baud rate can be selected from combinations set with the DIL switches 5 and 6 on the rearside of the PCS 090. The choice consists of: RS-232C or TTY, 19200, 9600, 4800, or 1200 baud. The speed of the communication depends decisively on the baud rate (e.g. a changed variable must be read 2 times!). Select only on a slower baud rate if you have longer distances between the participants. We recommend to communicate with 19200 baud via TTY.
The relation baud rate to maximum distance is as follows:

Baudrate	max length RS-232C	max. length TTY
19200	15m	150m*)
9600	15m	300m*)
4800	15m	350m*)
1200	15m	350m*)

*) When using a shielded and twisted cable with 14 x 0,14mm, a line resistance of less than 138 /km, and a capacity of less than 120 pF/m.

Communication data block

The number of the communication data block containing the send ad receive buffers must be passed on to the 3P90BUEP drivers. The value 0 (default value) must be entered if you do not change the communications software in the programmable controller. The communication data block consists of 2 times 32 data words (= 128 bytes).

Timeout time

The timeout time to be set is the maximum time, the programmable controller needs for the cyclical call of the PCSKOMM communication program plus the communication time. The time is settable between 0.5 and 2.55 seconds. Time monitoring is activated only if the order read by the PCS is not yet the current one.
Thus, the resulting timeout parameter (element x 10 ms) and the timeout parameter P6 (Kx.1 = x 100ms) for the programmable controller without user program is as follows:

Baudrate	Timeout PCS	P6 (timeout progr. contr.)
19200	50	K2.1
9600	50	K3.1
4800	50	K5.1
1200	170	K17.1

Attention!
You must add the maximum scan time to the above indicated times.
Example: 19200 baud, 800 ms maximum scan time -> timeout PCS = 130, P6 = K10.1.

B Buep 19-Expander driver

Connection to the PLC

1. Set DIL switches 8 and 9 on the rear side of the PCS to OFF. The DIL switches 5 and 6 are set according to the selected baud rate and interface. Initiate a PCS reset.
2. Supply operating voltage (19..33V) to the PCS. At least the ERR LED must light now.
3. Connect the programming interface of the programmable controller to the PCS using a suited cable.
4. Now, the ERR LED at the PCS must be activated.

Trouble-shooting

Errors which might occur during first commissioning are listed below:

1. DIL switch 8 is set to ON. In this case, the PCS starts a diagnosis routine after being switched on. This routine is only required for test purposes. Remedy: Set the DIL switch to OFF and restart the PCS (by switching it off for a short time or by shortly pressing the RESET push-button located above the DIL switch).
2. The DIL switches 5 and 6 (interface and baud rate selection) are not correctly set. Check the settings at the PCS using the HLP key and arrowdown key.
3. Is the correct cable being used? This is PCS 706 for TTY (observe the polarity) and LCA 035/235 for RS-232 (polarity not important) or a cable as described in chapter 2.5.
4. Has the correct program being loaded into the programmable controller? Has the programmable controller being switched off following a „Program load“? Does the P2 parameter indicate a „Rest-art after fault“?
5. DW 3 of the PCS data block shows a value <> 0KH in case of an error. If this the case the error is to be searched in the programmable controller to PCS connection. The cable is possibly defective.
6. Communication is established but after a certain time the PCS displays the following message:

   ======== COMMUNICATION ERROR =========

   In this case, please read the following section.

Communication ERROR

When communicating with each other the PCS operates as master (AST) and the programmable controller as slave (PST). Thus, it is the PCS´s task to set up the communication and to perform monitoring. Thereby, 3 criteria must be monitored: is communication active, is communication processing in the programmable controller active, and does the programmable controller indicate faults. Data are transferred to/read from the programmable controller via the BUEP 19 protocol. The timeout times for the protocol are fixed (3.2s max.) which allows no statement about data processing. Therefore, the processing timeout is set using the PROJECT/DRIVER PARAMETER menu in PCSPRO.

B Buep 19-Expander driver

- TIMEOUT
  Time monitoring for the serial data interchange is active in the PCS as well as in the programmable controller. In case of an error, an error message is displayed on the PCS and the ERROR LED flashes. The PCS tries to set up communication again in the background. The error message is erased if this has been successful. In the PCSPRO, the timeout time for order processing is adjustable in the range of 500ms to 2.55 s.

  The following message is displayed in case of a complete communication loss with the programmable controller:

  ========== COMMUNICATION ERROR ==========

  TIMEOUT COMMUNICATION!

  The following display appears in case the tasks are not processed by the programmable controller. This is the case if the PCSKOMM program will not be executed or the restart input is not set after a programmable controller timeout. The cable is possibly defective.

  ========== COMMUNICATION ERROR ==========

  TIMEOUT PLC

- PLC ERROR RECEIVED
  Ba means of the protocol, the programmable controller indicates if a fault has appeared. It will receive a fault byte which informs about the appeared fault. Please refer to the BOSCH „CL300 Computer linking module R301“ manual for the possible fault description. In the shown example, the fault value „2C „ is indicated which corresponds to addressing an inadmissible DB (is variable AG correct?).

  ========== COMMUNICATION ERROR ==========

  PLC ERROR RECEIVED: 2C!

- TOO MANY REPETITIONS
  The faulty position is repeated if a fault appears in the communication. The communication is interrupted and the following error message is output after 3 unsuccessful repetitions. In this case the PCS/ programmable controller connection has been routed through a noisy environment, the cable is too long for the selected baud rate, or earthing is insufficient.

  ========== COMMUNICATION ERROR ==========

  TOO MANY REPETITIONS!

B Buep 19-Expander driver

Notes concerning the connection of the PCS to a PLC:

- Connect the cable screening to the central common of the control cabinet.
- Ensure appropriate chassis groundings with regard to the PCS housing on the one hand and the PLC bus board on the other. Remember that a copper grounding strip, due to its large surface, ensures a considerably higher RF conductivity than normal stranded interconnecting wire.
- Avoid, as far as possible, high frequency interferences, because damping is very difficult in this case. The progr. contr. and the PCS are electrically isolated by optocouplers, but this isolation is not effective in case of high-speed transients, because optocouplers feature a coupling capacitance (although it is very low).
- Ensure clear supply voltage reference points. To facilitate this, the power supply is potential-free.
- If the supply voltage is influenced by high interferences, use a separate power supply for the PCS (24 V, 10 VA). It should be equipped with appropriate noise filters. In this case, 0 Volt can directly be connected to protective earth at the PCS.
- Ensure a minimum distance of 200 mm between noise sources and the PCS/the communication cable. This especially concerns inductors and frequency converters.
- Please take care that the serial data lines are covered completely (if possible) by the shield. Use a metallized connector hood at PCS as well as at the progr. contr. side and ensure a highly conductive connection between the connector hood and the shield. Please notice that grounding on both sides may require an equipotential bonding conductor with a cross section of 10 times that of the shield. This is especially important, if the PCS and the progr. contr. are not connected to the same common point (if they are for example installed in different control cabinets). This is necessary to prevent equalizing currents on the cable shield.

B Buep 19-Expander driver

B1.2 Description of the handling software

The 3P90BUEP communication block is located on the PCS 91.BOS floppy disk. The scan time without user program (only PCSKOMM and the I/O cycle) totals about 40 ms. A maximum of 80 ms are possible. Since the maximum scan time of the ZE300 totals 1600ms the user program should not require more than approx. 1500 ms.
The effective response time speed between programmable controller and PCS depends on the baud rate and the cyclical call time of PCSKOMM. The response time is approx. 0.2 seconds without user program and 19200 baud communication rate. Please match the P6 timeout time and the PCS Timeout time parameters correspondingly (refer to 2.1). However, the actual response time can be increased for variables since a changed set value variable is read 2 times before it is assumed accepted.

Description of the programs

- OB 1

Call of the programs using parameters. Change the settings for the PCSKOMM program here. Thus, no changes are required in PCSKOMM. Link your application program in after calling the PCSKOMM program. Thereby, activated keys are immediately processed.

- OB7, OB8
  Set the flag for the programmable controller start.
- PCSKOMM communication program.
  Reads reception buffer of TRANSDB and processes a received order packet. Enters corresponding data into PCSDB and reads from it. Writes data to be read into the transmitting buffer of TRANSDB. The communication is monitored using the timer P7. P3 and P5 are set if the P7 timeout time has expired without a new order being entered. IF P2 is = 1, the communication starts automatically again after communications error if an order packet has been received. Please specify the presettings for the PCS at the -PCSVORB label: LEDs, display and memory behavior, enable priorities, ect. Please specify the reactions to communication loss at the -PCSNOTF label: reset the PCS status (keys,..), reset of variables (the communication can fail during a menu), and start conditions of the PCS for the restart.

Parameterization of the PCSKOMM handling block:

P0: Communication data block (here: DB0, 64 words)
The send and receive buffers are specified by this block. Orders of the PCS and processed data (for reading orders) are stored in these buffers. The block must consist of 64 data words.

P1: PCS control block (here: DB1, 255 words)
The programmable controller and the PCS exchange information via this data block. It is activated during the processing of the handling block.

P2: Switch for restart after faults (bit, here E0.0)
Using this bit, the communication can be restarted if a communications error appears (faults are then reset automatically!). It is to be considered that no edge evaluation is made, i.e. with bit P2 set, the communication is automatically restarted after a fault has appeared.

B Buep 19-Expander driver

P3: General error message (bit, here A0.0)
The P3 general fault bit is set for any error in the connection between the programmable controller and the PCS. This bit is automatically reset as soon as the communication is running without a fault.

P4: Flag for the first processing of the handling block (bit, here M20.0)
This flag should be reset before the first call of the handling block. At the first execution, it is automatically set to 1.

P5: Flag bit for appeared timeout faults (here: M20.2)

P6: Time value for timeout monitoring (refer also to 1.4.3). Structure:
e.g. K2.2 = 2 seconds timeout
multiplication element 1 = 100 ms, 2 = 1 s
value

P7: Name of the used timer for timeout (here: T15)

P8: Flag bit for waiting for the first order (here: M20.1)
TRANSDB is the data block containing the send and receive buffers. Must be 64 words in length. PCSDB is the information interchange data block between the PCS and the programmable controller. Must be 255 words in length.

B1.3 Implementation of the software

The README.DOC file on the PCS 91.BOS floppy disk contains current notes concerning the use of the various projects.
The 200.0-213.7 flags are used as temporary flags by the communication program. They may also be used by other programs as temporary flags. The P7 timeout timer (here: T15) and the flags P4, P5 and P8 (here: M20.0-M20.2) must not be used by other programs.

Implementation of the PLC software

- Equip the rack
- Set the digital I/O boards to address 0 (for this set all DILs to OFF)
- Copy the corresponding project stored on the PCS 91.BOS floppy disk to the hard disk
- Supply voltage to the programmable controller
- Transfer the programmable controller program into the controller using the loader (if necessary, link in your application program, correct the configuration and link newly)
- Turn off power to the programmable controller (important!), remove the programming cable, set the programmable controller interface parameters (refer to 2.4), turn on power to the programmable controller and set the mode switch to RUN
- Connect the PCS and the programmable controller using an communications cable described in sections 3 (are DIL-switch segments 5 and 6 at the PCS correspondingly set).
- Apply 24 volts for a short-term to input E0.0 (Precondition: 24 V input module).

B Buep 19-Expander driver

Now, the ERR LED (PCS) and the A0.0 output LED (programmable controller) must have been deactivated. Also the 7-segment display at the CPU must remain dark. Thus, the communication is active.

Attention!
In all projects on the floppy disk, the INIT and COFF PBs are to be adjusted specifically for the connected operating console. The contained HLT commands are to be deleted.

B1.4 Settings of the PLC

Please set the segments of the switch of the ZE 300/301 board as follows:

S1 switch

DIL	8	7	6	5	4	3	2	1
	X	X	X	X	X	OFF	ON	ON 1200 BAUD
	X	X	X	X	X	ON	OFF	ON 4800 BAUD
	X	X	X	X	X	ON	ON	OFF 9600 BAUD
	X	X	X	X	X	ON	ON	ON 19200 BAUD

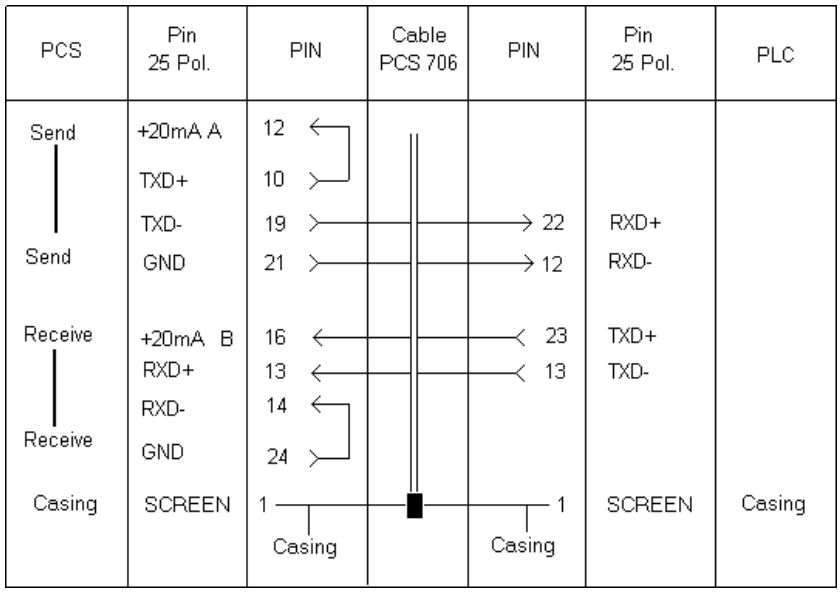
The switch is located on the upper edge of the ZE 300/301 board. The exact location can be taken from the „BOSCH CL300 devices manual“. Turn shortly off the programmable controller after changing switch positions.

B Buep 19-Expander driver

B2 Communication

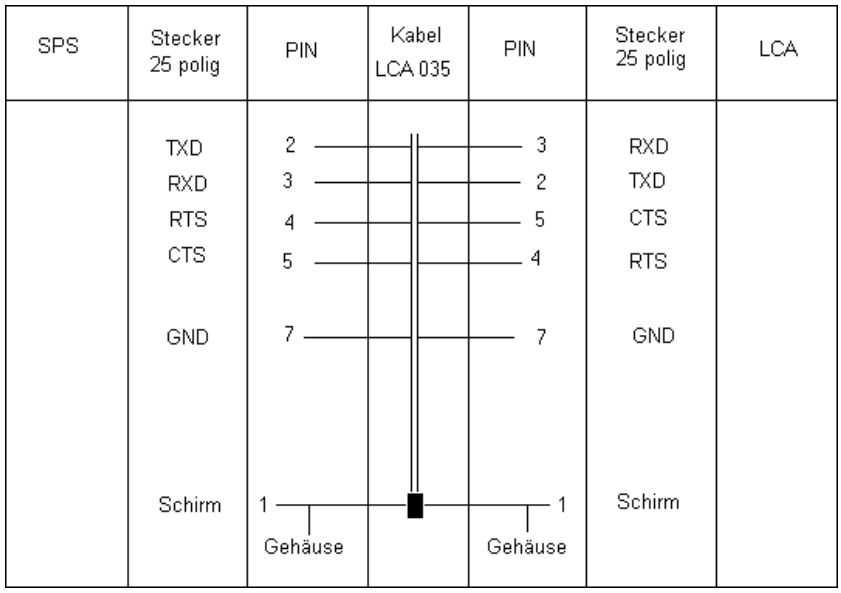
Adapter cable PCS 706

Connection of the PCS - PLC via TTY interface



LCA 035/235

PCS to PLC via RS-232 interface
Use a zero-modem cable for the connection of the PLC to PCS via RS-232C. Systeme Lauer offers the LCA 035/235 cable for this purpose. The necessary connections are as follows:

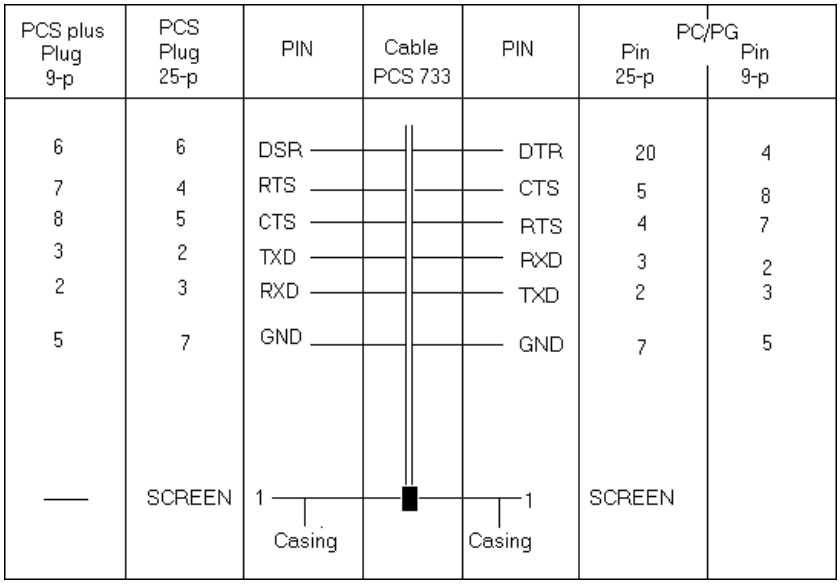


B Buep 19-Expander driver

Screening
The shield should be connected on both sides to a metallized connector hood. If non-metallized connector hoods are used, the shield may also be connected to pin 1, but this is not recommended due to interferences, because the data lines should be completely covered by the shield (if possible). It should be noticed, however, that earthing on both sides may require an equipotential bonding conductor with a cross section of 10 times that of the shield in order to compensate for earth potential variations and to prevent equalizing currents on the cable shield! This is especially important, if the PCS and the programmable controller are not connected to the same common point (if they are for example installed in different control cabinets).

Programming cable PCS 733

Connection PLC to PCS
Use this cable for the programming (loading of the driver and user program) of the PCS.



Data transmission PCS - PLC

Data transfer between the PCS and the programmable controller is effected in data packets. Each data packet is assigned a checksum (CRC). The package content is checked for possible errors by the programmable controller operating system and the PCS. Each packet consists of a minimum of one sub-packet which performs a clearly defined task. The tasks to be integrated in a packet are determined by the PCS based on numerous single criteria. A criterion represents the importance of a task. Each of these tasks is assigned a specific start priority. Priority management assures that no task is lost. The highest priorities are processed first. These priorities have nothing in common with the priority management (PCS manual)! The indicated packet lengths refer to words, including header. Data word 13 is used to prevent transmission of specific data words (PCS manual).

B Buep 19-Expander driver

The tasks are listed below in detail:

TASK	PRIOR.	LENGTH	START CRITERION
1. Write keyboard status	8	3	When pressing or releasing a key
2. Reset message bit	7	2	Press CLR for delete behavior 2
3. Write PLC setpiont	6	2..3	Always after changing a setpoint variable and leaving the entry field
4. Read actual values	5	2..24*)	In priorities 0..6, 12 continuously, otherwise when display is refreshed
5. Read message bits	4	9 max.	Continuously
6. Send PCS status	4	5	After changes
7. Read LEDs, flashing LEDs, memory and display behavior	3	4	Continuously
8. Read command word	3	3	Continuously (menu selections only if the temporary buffer is empty)

*) Depends on the number of variables displayed and if the addresses are adjacent. If the addresses are not adjacent, one header is required per non-adjacent variable (1 word).

Data transfer procedure

When communicating with each other the PCS operates as master (AST). It is the master´s task to set up the communication and to send orders to the PLC. PCS and PLC communicate with the following settings: TTY or RS-232C, 19200/9600/4800/1200 baud, 8 bit, EVEN parity, 1 stop bit. Only the interchange of the packets is described in this manual. Refer to the BOSCH „BUEP 19 transfer protocol“ manual for the structuring of the packets.

B Buep 19-Expander driver

Structure of the reading cycle:

PCS	——	**PLC**
ENQ	→	
	←	ACK
READ DATA DB AREA	→	
	←	ACK
EOT	→	
	←	ENQ
ACK	→	
	←	DATA THE DB AREA
ACK	→	
	←	EOT

Structure of the writing cycle:

PCS	——	**SPS**
ENQ	→	
	←	ACK
WRITE DATA DB AREA	→	
	←	ACK
EOT	→	
	←	ENQ
ACK	→	
	←	ORDER SUCCESSFULLY PERFORMED
ACK	→	
	←	EOT

The packets designated above as „data“ specify the real content of the communication. These data are written into the receive buffer of the communication data block of the programmable controller (write cycle) and are processed by the communication program of the programmable controller. Information is written into the send buffer in the communication data block of the programmable controller if the processing of the data has been completed. This send buffer is read by the reading cycle. The validity of an order is verified via an order number in the data. The send and receive buffers use 128 bytes in a data block and are not adjustable within this data block.

C Buep 19e-Expander driver

C1 First commissioning

Preface

The successful parameterization of the PCS as described in the PCS 091/991/925/995 manual is presupposed. This appendix relates exclusively to the use of the PCS micro/mini/midi and plus together with the BOSCH CL200/CL400/CL500 PLC via the programming interface by means of the BUEP19E protocol. In the following, this controller is referred to as programmable controller and the driver to be loaded into the PCS as 5P90BUEP. The BOSCH specific terminology and the programming of the programmable controller with the BOSCH software are presupposed as known.

Attention!
Use only the software PCSPRO or PCSPRO^{WIN} for programming. Other software packages can release failures of the PCS and PLC.

The following products are required for the operation of a programmable controller with an PCS (Systeme Lauer company):

1. PCS 009/090/095/900/920/950/950c/PCS plus operating console
2. The PCS 706 connecting cable for the PCS to programmable controller connection via the TTY interface or a zero-modem cable for the connection via the RS-232C interface
3. PCSPRO floppy disk with 5P90BUEP driver, version 1.3 or later
4. PCS 091/925/995 manual
5. PCS 91.BOS appendix including BOSCH handling software floppy disk

Furthermore are required (BOSCH company):

1. Programming card, programming cable, software (version 3.12 or later) and the copy protection device for programming the PLC
2. CL500/CL400/CL200 racks
3. Power supply NT 3
4. Coordinator board SK 500 (not for CL400/CL200)
5. One or several ZS 500 CPU boards (with 32k or 64k RAM module)
6. One or several I/O boards
7. Power supplies for the PCS console and Bosch I/O boards

C Buep 19e-Expander driver

Loading of the 5P90BUEP Driver

After selecting the 5P90BUEP driver in PCSPRO you have to set various variables which are used by the drivers for the correct addressing of the programmable controller. These are: baud rate, interface, data block number, board address, number of sub-packets and timeout time.

Interfaces

Interface and baud rate can be selected from combinations set with the DIL switches 5 and 6 on the rearside of the PCS 090. The choice consists of: RS-232C or TTY, 19200, 9600, 4800, or 1200 baud. The speed of the communication depends decisively on the baud rate. Select only on a slower baud rate if you have longer distances between the participants. We recommend to communicate with 19200 baud via TTY.

The relation baud rate to maximum distance is as follows:

Baudrate	**max length RS-232C**	**max. length TTY**
19200	15m	150m*)
9600	15m	300m*)
4800	15m	350m*)
1200	15m	350m*)

*) When using a shielded and twisted cable with 14 x 0,14mm, a line resistance of less than 138 /km, and a capacity of less than 120 pF/m.

Communication data block

The number of the communication data block containing the send and receive buffers must be passed on to the 5P90BUEP drivers (default value=2). The communication data block consists of 69 data words (= 138 bytes).

Slot address

The slot address (= block address) of the CPU to be addressed must be passed on to the 5P90BUEP driver. Possible values are 0..3 for the CL 500, 30 for the CL 400, and 0 for the CL200 (default settings is 0). Once you have set the address you can plug the communications cable into an arbitrary programming unit interface in the CL 500 rack (including the SK 500). Always the CPU on this slot is addressed.

Number of sub-packets

The number of sub-packets per sent packet can be set. If you select a small value, the scan time load of the Expander module in the PLC is decreased. However, simultaneously the response time of the communication is increased. Possible values are between 1 and 50 (default value).

Timeout time

The timeout time to be set is the time in which a complete communication cycle must have been started including packet processing in the programmable controller. Thus, the timeout time is composed out of 1 programmable controller scan time plus the data transfer time. The timeout time is settable between 2.5 and 6 seconds. 2,5 seconds is default value.

C Buep 19e-Expander driver

Connection of the PCS to the PLC

1. Set DIL-switch 8 to „OFF“ and DIL 9 to „ON“ (on the rearside the PCS). Set the DIL-switches 5 and 6 according to the selected baud rate and interface. Initiate a reset of the PCS.
2. Apply operating voltage (19..33V) to the PCS. At least the ERR LED must light now.
3. Connect the programming interface of the programmable controller to the PCS using a suited cable.
4. Set the P2 restart input to „1“.
5. Now the ERR LED at the PCS must be deactivated. Idle text 0 is shown on the display of the PCS.

After setting and loading the PLC corresponding to section 2, you can connect the PCS to the PLC corresponding to the following instructions.

Attention!
Check the function of the PCS and PLC after programming or driver installation. All parameterized functions have to be checked. Otherwise failures of the PCS or PLC are possible.

Trouble-shooting

Errors which might occur during first commissioning are listed below:

1. DIL switch 8 is set to ON. In this case, the PCS starts a diagnosis routine after being switched on. This routine is only required for test purposes. Remedy: Set the DIL switch to OFF and restart the PCS (by switching it off for a short time or by shortly pressing the RESET push-button located above the DIL switch).
2. The DIL switches 5 and 6 (interface and baud rate selection) are not correctly set. Check the settings at the PCS using the HLP key and arrow-down key.
3. Is the correct cable being used? This is PCS 706 for TTY (observe the polarity) and LCA 035/235 for RS-232 (polarity not important) or a cable as described in chapter 3.
4. Has the correct program being loaded into the programmable controller? Does the P2 parameter indicate a „Restart after fault“?
5. In dataword 3 of the PCS datablock will be reportet an error wiith a value different to zero. In this case the error should be searched in the PLC-PCS connection. Possibly the cable is defect.
6. Communication is established but after a certain time the PCS displays the following message:

   ======== COMMUNICATION ERROR ========

   In this case, please read the following section.

C Buep 19e-Expander driver

Communication ERROR

When communicating with each other the PCS operates as master (AST) and the PLC as slave (PST). Thus, it is the PCS´s task to set up the communication and to perform monitoring. Thereby, 3 criteria must be monitored: is communication active, is communication processing in the PLC active, and does the PLC indicate faults. Data are transferred to/ read from the PLC via the BUEP19E protocol. The timeout times for the protocol are fixed (2.4s max.) which allows no statement about data processing. Therefore, the processing timeout is set using the PROJECT/ DRIVER PARAMETER menu in PCSPRO.
In case of an error, an error message is displayed on the PCS and the ERROR LED flashes. The PCS tries to set up communication again in the background. The error message is erased if this has been successful. Error message (whereby the fault text on the bottom line is described below):

========== COMMUNICATION ERROR ==========
FAULT TEXT

- **TIMEOUT COMMUNICATION**
  This error message is output if the protocol timeout time of 2.4 expires. TIMEOUT PLC. This display is shown in case the task package is not processed by the programmable controller. This is the case if the program will not execute PCS_KOMM, or the restart input is not set following a communication loss. The Timeout time for the order processing is settable via the timeout parameter in the range of 500ms to 2.55 s.
- **PLC ERROR RECEIVED**
  The programmable controller indicates via the BUEP19E protocol if a fault has appeared. This can happen if the addressed data block is too small or not present or an inadmissible access occurs. Therefore, examine the settings of the driver variables and the definitions in the programmable controller (e.g. symbol file).
- **TOO MANY REPETITIONS**
  The faulty position is repeated if a fault appears in the communication. The communication is interrupted and the following error message is output after 3 unsuccessful repetitions. In this case the PCS/PLC connection has been routed through a noisy environment, the cable is too long for the selected baud rate or earthing is insufficient.
- **PLC IN STOP**
  This error message is output if the status of the PLC has been check and the CPU was not in „RUN“ mode.
- **WRONG BLOCK NUMBER**
  The block address of the CPU (set via CPU slot) to be addressed is checked by the protocol. This error message is output if the corresponding CPU is not present.

C Buep 19e-Expander driver

Notes concerning the connection of the PCS to a PLC:

- Connect the cable screening to the central common of the control cabinet.
- Ensure appropriate chassis groundings with regard to the PCS housing on the one hand and the PLC bus board on the other. Remember that a copper grounding strip, due to its large surface, ensures a considerably higher RF conductivity than normal stranded interconnecting wire.
- Avoid, as far as possible, high frequency interferences, because damping is very difficult in this case. The PLC and the PCS are electrically isolated by optocouplers, but this isolation is not effective in case of high-speed transients, because optocouplers feature a coupling capacitance (although it is very low).
- Ensure clear supply voltage reference points. To facilitate this, the power supply is potential-free.
- If the supply voltage is influenced by high interferences, use a separate power supply for the PCS (24 V, 10 VA). It should be equipped with appropriate noise filters. In this case, 0 Volt can directly be connected to protective earth at the PCS.
- Ensure a minimum distance of 200 mm between noise sources and the PCS/the communication cable. This especially concerns inductors and frequency converters.
- Please take care that the serial data lines are covered completely (if possible) by the shield. Use a metallized connector hood at PCS as well as at the PLC side and ensure a highly conductive connection between the connector hood and the shield. Please notice that grounding on both sides may require an equipotential bonding conductor with a cross section of 10 times that of the shield. This is especially important, if the PCS and the PLC are not connected to the same common point (if they are for example installed in different control cabinets). This is necessary to prevent equalizing currents on the cable shield.

C Buep 19e-Expander driver

C1.2 Description of the handling software

The 3P90BUEP communication block is located on the BOSCH floppy disk. The scan time load is less than 1 ms. The effective response time speed between PLC and PCS depends on the baud rate and the cyclical call time of PCS_KOMM. The response time is approx. 0.2 seconds without user program and 19200 baud communication rate.

Description of the programs

- **OB1**
  Call of the programs using parameters. Change the settings for the PCS_KOMM program here. Thus, no changes are required in PCS_

  KOMM. Link your application program in after calling the PCS_KOMM program. Thereby, activated keys are immediately processed.

  TEST Example for a data access:

  F1 activates the menu 1, F2 disabled all menus, F3 sets the message bits M0..15, F4 disables these message bits again. The keys are copied to the LEDs.

- **PCS_KOMM communication program**
  Reads reception buffer in P1 and processes a received order packet. Enters corresponding data into P0 and reads from it. Writes data to be read into the transmitting buffer of P1. The communication is monitored using the timer P3. P5 is set if the P4 timeout time has expired without a new order being entered. IF P2 is = 1, the communication starts automatically again after communications error if an order packet has been received. Please specify the presettings for the PCS at the -START_KO label: LEDs, display and memory behavior, enable priorities, ect. Please specify the reactions to communication loss at the -KOMM_FEL label: reset the PCS status (keys,..), reset of variables (the communication can fail during a menu), and start conditions of the PCS for the restart.

- **INIT**
  Typical presets for a restart (e.g. release of massage texts)

- **COFF**
  Emergency case presets (e.g. keys set zero)

Parameterization of the PCSKOMM handling block:

P0: PCS communication data block (here: DB1, 256 words)
The programmable controller and the PCS exchange informations via this data block.

P1: Send and receive data block (here: DB2, 138 bytes)
The send and receive buffers are specified by this block. Orders of the PCS and processed data (for reading orders) are stored in these buffers. The block must consist of 69 data words.

P2: Switch for restart after faults (bit, here E0.0)
Using this bit, the communication can be restarted if a communications error appears (faults are then reset automatically!). It is to be considered that no edge evaluation is made, i.e. with bit P2 set, the communication is automatically restarted after a fault has appeared.

C Buep 19e-Expander driver

P3: Name of the used timer for timeout (here: T0)

P4: Time value for timeout monitoring. Structure:
e.g. K4.2 = 4 seconds timeout
multiplication element 1 = 100 ms, 2 = 1 s
value

P5: General error message (bit, here M100.0)
The P5 general fault bit is set for any error in the connection between the programmable controller and the PCS. This bit is automatically reset as soon as the communication is running without a fault.

C1.3 Implementation of the software

The README.DOC/PROJECT.DOC files on the PCS 91.BOS floppy disk contain current notes concerning the use of the various projects.

Implementation of the PLC software

- Equip the rack
- Set the block address on CPU and I/O boards
- Set the interface on the CPU and the SK (if possible set all to the same parameters)
- Copy the corresponding project stored on the PCS 91.BOS floppy disk to the hard disk
- Supply voltage to the PLC and the I/O boards
- Run the PLC programming software, select the project for the CPU, enter the configuration in the SK table and transfer it into SK (under LOADER).
- Transfer the PLC program into the controller using the loader (if necessary, link in your application program and link newly)
- Connect PCS and PLC using an adapter cable described in section 3.6 (are the DIL-switch segments 5 and 6 at the PCS correspondingly set?). The location for attaching the adapter cable is arbitrary. How-ever, the block address must be correct and the communication baud rate must correspond with the PCS.
- Set the E0.0 input to „1“

Now, the ERR LED (PCS) and the A0.0 output LED (PLC) must have been deactivated. Also the 7-segment display at the SK must remain dark. Thus, the communication is active.
Using this protocol, the addressed CPU can be accessed from an arbitrary programming unit connector. This way, it is possible to manage several PCSs by one CPU if each PCS is assigned it´s own communication block and data block. However, it is possibly dangerous to assign several PCSs to the same data block (the data transmitted by the PCS are then no longer clearly alloca-table). A parallel mode of the programming unit and the PCS is easily possible.

C Buep 19e-Expander driver

C1.4 Board settings

Please set the segments of the switch of the ZS500 and SK500 boards for setting the interface parameters as follows:

S6 switch

DIL	1	2	3	4	5	6	7	8	
	ON	OFF	ON	ON	ON	ON	ON	OFF	19200 BAUD
	ON	OFF	ON	ON	OFF	ON	ON	OFF	9600 BAUD
	ON	OFF	ON	ON	ON	OFF	ON	OFF	4800 BAUD
	ON	OFF	ON	ON	ON	ON	OFF	OFF	1200 BAUD

Please set the segments of switch 5 of the ZS500 and SK500 boards (I/O groups in parenthesis) for setting the board address as follows:

DIL	1(7)	2(8)	board
	OFF	OFF	0
	ON	OFF	1
	OFF	ON	2
	ON	ON	3

For adjustment of the transfer rate adjust the DIP-switches for the CL200 as following:

DIL	1	2	Baud
	OFF	OFF	9600
	ON	OFF	19200
	OFF	ON	38400

Tip!
Inspect the function of the handling software to avoid failures of the PCS or PLC.

R200 coupling card settings

Pay attention that in the CPU of the CL200 is loadrd at least the Firmware version 1.5 of 06.03.1996 eingespielt ist.
On the settings of the card pay attention to the DIP-switches S1 and S2.

DIP-Switches

DS1	DS2	Protokoll
ON	OFF	BUEP19e-Protocol

DS3	DS4	
ON	ON	Parity

DS5	DS6	DS7	Transfer rate
OFF	OFF	OFF	38 400 Baud
ON	ON	ON	19 200 Baud
OFF	ON	ON	9 600 Baud
ON	OFF	ON	4 800 Baud

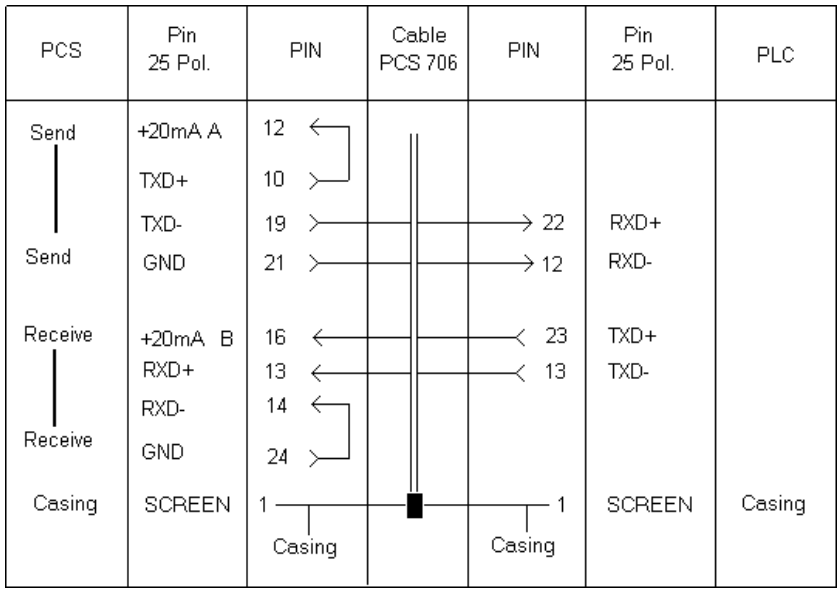
DS8	
OFF	Control signals not evaluate

C Buep 19e-Expander driver

C2 Communication

Adapter cable PCS 706

Connection PCS to PLC via TTY interface



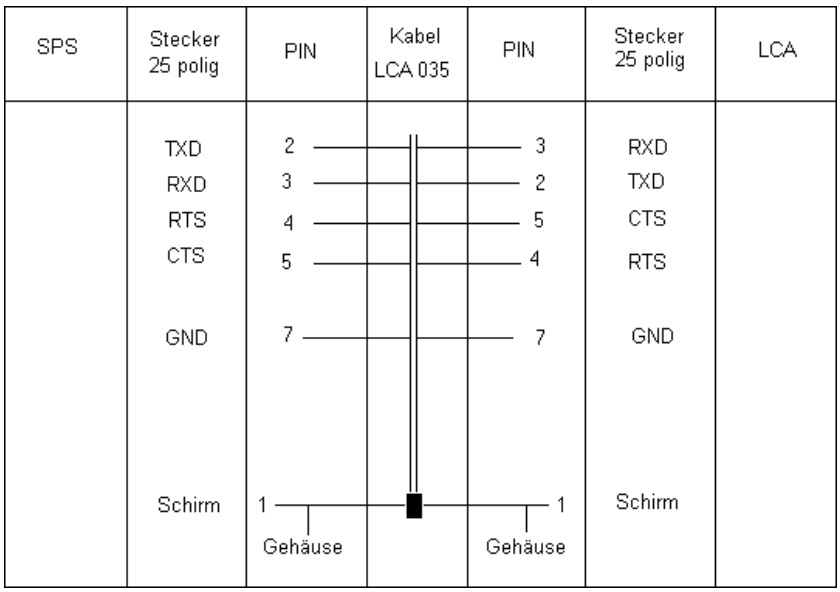
Screening

The shield should be connected on both sides to a metallized connector hood. If non-metallized connector hoods are used, the shield may also be connected to pin 1, but this is not recommended due to interferences, because the data lines should be completely covered by the shield (if possible). It should be noticed, however, that earthing on both sides may require an equipotential bonding conductor with a cross section of 10 times that of the shield in order to compensate for earth potential variations and to prevent equalizing currents on the cable shield! This is especially important, if the PCS and the programmable controller are not connected to the same common point (if they are for example installed in different control cabinets).

C Buep 19e-Expander driver

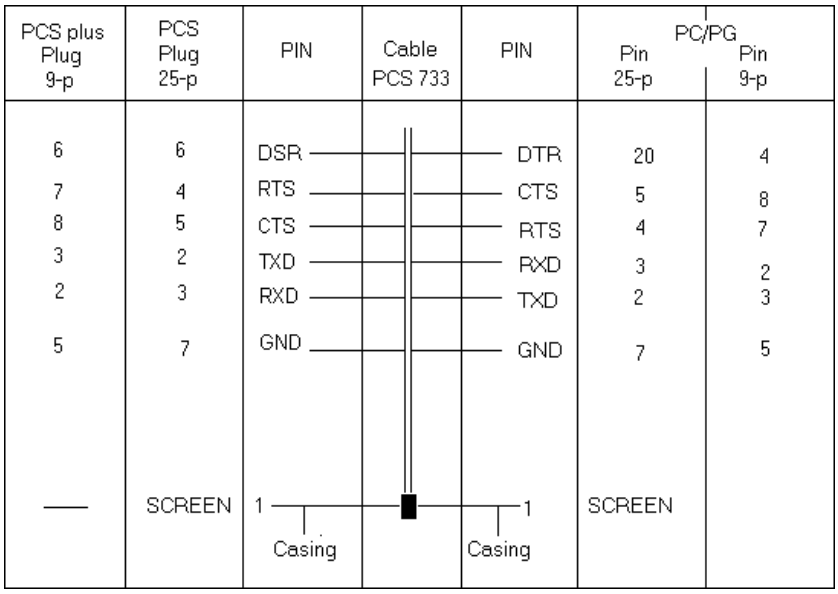
LCA 035/235

PCS to PLC via RS-232 interface
Use a zero-modem cable for the connection of the PLC to PCS via RS-232C. Systeme Lauer offers the LCA 035/235 cable for this purpose. The necessary connections are as follows:



Programming cable PCS 733

Connection PLC to PCS
Use this cable for the programming (loading of the driver and user program) of the PCS.



C Buep 19e-Expander driver

Data transmission PCS-PLC

Data transfer between the PCS and the PLC is effected in data packets. Each data packet is assigned a checksum (CRC). The package content is checked for possible errors by the PLC operating system and the PCS. Each packet consists of a minimum of one sub-packet which performs a clearly defined task. The tasks to be integrated in a packet are determined by the PCS based on numerous single criteria. A criterion represents the importance of a task. Each of these tasks is assigned a specific start priority. Priority management assures that no task is lost. The highest priorities are processed first. These priorities have nothing in common with the priority management (PCS manual). The indicated packet lengths refer to words, including header. Data word 13 is used to prevent transmission of specific data words (PCS manual).

The tasks are listed below in detail:

TASK	PRIOR.	LENGTH	START CRITERION
1. Write keyboard status	8	3	When pressing or releasing a key
2. Reset message bit	7	2	Press CLR for delete behavior 2
3. Write PLC setpoint	6	2..3	Always after changing a setpoint variable and leaving the entry field
4. Read actual values	5	2..24*)	In priorities 0..6, 12 continuously, otherwise when display is refreshed
5. Read message bits	4	9 max.	Continuously
6. Send PCS status	4	5	After changes
7. Read LEDs, flashing LEDs, memory and display behavior	3	4	Continuously (menu selections only if the temporary buffer is empty)
8. Read command word	3	3	Continuously

*) Depends on the number of variables displayed and if the addresses are adjacent. If the addresses are not adjacent, one header is required per non-adjacent variable (1 word).

Data transfer procedure

When communicating with each other the PCS operates as master (AST).

C Buep 19e-Expander driver

It is the master´s task to set up the communication and to send orders to the PLC. PCS and PLC communicate with the following settings: TTY or RS-232C, 19200/9600/4800/1200 baud, 8 bit, EVEN parity, 1 stop bit. Only the interchange of the packets is described in this manual. Refer to the BOSCH „BUEP 19 transfer protocol“ manual for the structuring of the packets.

Structure of the reading cycle:

PCS	——	**PLC**
ENQ	→	
	←	ACK
READ DATA DB AREA	→	
	←	ACK
EOT	→	
	←	ENQ
ACK	→	
	←	DATA THE DB AREA
ACK	→	
	←	EOT

Structure of the writing cycle:

PCS	——	**PLC**
ENQ	→	
	←	ACK
WRITE DATA DB AREA	→	
	←	ACK
EOT	→	
	←	ENQ
ACK	→	
	←	ORDER SUCCESSFULLY PERFORMED
ACK	→	
	←	EOT

The packets designated above as „data“ specific the real content of the communication. These data are written into the receive buffer of the communication data block of the programmable controller (write cycle) and are processed by the communication program of the programmable controller. Information is written into the send buffer in the communication data block of the program-mable controller if the processing of the data has been completed. This send buffer is read by the reading cycle. The validity of an order is verified via an order number in the data. The send and receive buffers use 138 bytes in a data block and are not adjustable within this data block.

D Buep 19e-Direct driver

D1 First commissioning

Preface

The successful parameterization of the PCS as described in the PCSPRO manual is presupposed. This appendix relates exclusively to the use of the PCS 900/950 together with the BOSCH PLC by means of the BUEP19E protocol. In the following, this controller is referred to as PLC and the driver to be loaded into the PCS as B19EDIR. The BOSCH specific terminology and the programming of the PLC with the BOSCH software are presupposed as known.

Attention!
Use only the software PCSPRO or PCSPRO^{WIN} for programming. Other software packages can release failures of the PCS and PLC.

Required devices and accessories

The following products are required for the operation of a PLC with an PCS (Systeme Lauer company):

1. PCS 900/920/950 operating console
2. The PCS 706 connecting cable for the PCS to PLC connection via the TTY interface or a zero-modem cable for the connection via the RS-232C interface
3. PCSPRO floppy disk with 5P90BUEP driver, version 2.4 or later
4. PCS 991/925/995 manual
5. PCS 91.BOS appendix including BOSCH handling software
6. Optionally: PG-MUX PCS 809.BOS for the simultaneous operation of the programming unit and the PCS via one programming interface.

Furthermore are required (BOSCH company):

1. „PROFI“ software (version 3.3 or later), programming cable, and the copy protection device for programming the PLC
2. CL500 rack (or other PLC supporting the BUEP19E protocol)
3. Power supply NT 3
4. Coordinator board SK 500
5. One or several ZS 500/501 CPU boards (with 32k or 64k RAM module)
6. One or several I/O boards
7. Power supplies for the PCS console and Bosch I/O boards

D Buep 19e-Direct driver

Loading of the 5P90BUEP driver

After selecting the 5P90BUEP driver in PCSPRO you have to set various variables which are used by the drivers for the correct addressing of the PLC. These are: baud rate, interface, cross-reference list, board address, and timeout time.

Interfaces

Interface and baud rate can be selected from combinations set with the DIL switches 5 and 6 on the rearside of the PCS. The choice consists of: RS-232C or TTY, 19200, 9600, or 4800 baud. The speed of the communication depends decisively on the baud rate. Select only on a slower baud rate if you have longer distances between the participants. We recommend to communicate with 19200 baud via TTY. With Sync you decide for a synchronized data interchange between the PCS and the programmable controller, Nosync disables the synchronization (see also Chapter 4.2 and 4.3).

The relation baud rate to maximum distance is as follows:

Baud rate	**max. length RS232**	**max. length TTY**
19200	15m	150m*)
9600	15m	300m*)
4800	15m	350m*)

*) When using a shielded and twisted cable with 14 x 0,14mm, a line resistance of less than 138 /km, and a capacity of less than 120 pF/m.

Slot address

The slot address (= block address) of the CPU to be addressed must be passed on to the B19EDIR driver. Possible values are 0..3 for the CL 500, 30 for the CL 400, and 0 for the CL200 (default settings is 0). Once you have set the address you can plug the communications cable into an arbitrary BUEP19E interface in the rack (including the SK 500). Always the CPU on this slot is addressed.

Timeout time

The timeout time to be set is the time in which a complete communication cycle must have been started in the PLC. Thus, the timeout time is composed of 1 PLC scan time plus the data transfer time. The timeout time is settable between 2 and 9.9 seconds. 4 seconds is default value.

Data block number

The number of the transfer data block must be passed on to the B19EDIR drivers (default value=2). Possible values are in the range of 0 and 255. The default setting is 1. The transfer data block consists of 256 data words (= 512 bytes). The PLC and the PCS exchange information via this block.

D Buep 19e-Direct driver

Connection to the PLC

After setting and loading the programmable controller corresponding to section 4.5 and 4.6, you can connect the PCS to the PLC corresponding to the following instructions.

Attention!
Check the function of the PCS after parameterization and/or driver installation. All parameterized functions must be examined. Otherwise, malfunctions of the PCS and/or the programmable controller are possible.

1. Set DIL-switch 8 to „OFF“ and DIL 9 to „ON“ (on the rearside the PCS). Set the DIL-switches 5 and 6 according to the selected baud rate and interface. Initiate a reset of the PCS.
2. Apply operating voltage (19..33V) to the PCS. At least the ERR LED must light now.
3. Connect the programming interface of the PLC to the PCS using a suited cable.
4. Set the P2 restart input to „1“.
5. Now the ERR LED at the PCS must be deactivated. Idle text 0 is shown on the display of the PCS.

Trouble-shooting

Errors which might occur during first commissioning are listed below:

1. DIL switch 8 is set to ON. In this case, the PCS starts a diagnosis routine after being switched on. This routine is only required for test purposes. Remedy: Set the DIL switch to OFF and restart the PCS (by switching it off for a short time or by shortly pressing the RESET push-button located above the DIL switch).
2. The DIL switches 5 and 6 (interface and baud rate selection) are not correctly set. Check the settings at the PCS using the HLP key and arrow-down key. There is possibly also an incorrect DIL-switch setting of a corresponding BOSCH board.
3. Is the correct cable being used? This is PCS 706 for TTY (observe the polarity) and LCA 035/235 for RS-232 (polarity not important) or a cable as described in chapter 4.7.
4. Has the correct program being loaded into the PLC? Does the P2 parameter indicate a „Restart after fault“?
5. Communication is established but after a certain time the PCS displays the following message:

   ======== COMMUNICATION ERROR ========

   In this case, please read the following section.

D Buep 19e-Direct driver

Communication ERROR

When communicating with each other the PCS operates as master (AST) and the PLC as slave (PST). Thus, it is the PCS´s task to set up the communication and to perform monitoring. Thereby, 3 criteria must be monitored: is communication active, is communication processing in the PLC active, and does the PLC in dicate faults. Data are transferred to/ read from the PLC via the BUEP19E protocol. The timeout times for the protocol are fixed (2.4s max.) which allows no statement about data processing. Therefore, the processing timeout is set using the PROJEKT/ DRIVER PARAMETER menu in PCSPRO.
In case of an error, an error message is displayed on the PCS and the ERROR LED flashes. The PCS tries to set up communication again in the background. The error message is erased if this has been successful.

Error message (whereby the fault text on the bottom line is described below):

========== COMMUNICATION ERROR ==========
FAULT TEXT

- **TIMEOUT COMMUNICATION**
  This error message is output if the protocol timeout time of 2.4 seconds expires.
- **TIMEOUT PLC**
  This display is shown in case the task packet is not processed by the PLC. This is the case if the program will not execute PCS_ KOMM, or the restart input is not set following a communication loss. The Timeout time for the order processing is settable via the timeout parameter in the range of 2 s to 9.9 s. This fault appears only during synchronous operation.
- **PLC ERROR RECEIVED**
  The PLC indicates via the BUEP19E protocol if a fault has appeared. This can happen if the addressed data block is too small or not present or an inadmissible access occurs. Therefore, examine the settings of the driver variables and the definitions in the PLC (e.g. symbol file).
- **TOO MANY REPETITIONS**
  The faulty position is repeated if a fault appears in the communication. The communication is interrupted and the following error message is output after 3 unsuccessful repetitions. In this case the PCS/PLC connection has been routed through a noisy environment, the cable is too long for the selected baud rate, or earthing is insufficient.
- **PLC IN STOP**
  This error message is output if the status of the PLC has been check and the CPU was not in „RUN“ mode.
- **WRONG BLOCK NUMBER**
  The block address of the CPU (set via CPU slot) to be addressed is checked by the protocol. This error message is output if the corresponding CPU is not present.

D Buep 19e-Direct driver

Notes concerning the connection of the PCS to a PLC:

- Connect the cable screening to the central common of the control cabinet.
- Ensure appropriate chassis groundings with regard to the PCS housing on the one hand and the PLC bus board on the other. Remember that a copper grounding strip, due to its large surface, ensures a considerably higher RF conductivity than normal stranded interconnecting wire.
- Avoid, as far as possible, high frequency interferences, because damping is very difficult in this case. The progr. contr. and the PCS are electrically isolated by optocouplers, but this isolation is not effective in case of high-speed transients, because optocouplers feature a coupling capacitance (although it is very low).
- Ensure clear supply voltage reference points. To facilitate this, the power supply is potential-free.
- If the supply voltage is influenced by high interferences, use a separate power supply for the PCS (24 V, 10 VA). It should be equipped with appropriate noise filters. In this case, 0 Volt can directly be connected to protective earth at the PCS.
- Ensure a minimum distance of 200 mm between noise sources and the PCS/the communication cable. This especially concerns inductors and frequency converters.
- Please take care that the serial data lines are covered completely (if possible) by the shield. Use a metallized connector hood at PCS as well as at the progr. contr. side and ensure a highly conductive connection between the connector hood and the shield. Please notice that grounding on both sides may require an equipotential bonding conductor with a cross section of 10 times that of the shield. This is especially important, if the PCS and the PLC are not connected to the same common point (if they are for example installed in different control cabinets). This is necessary to prevent equalizing currents on the cable shield.

Attention!
Check the function of the PCS after parameterization and/or driver installation. All parameterized functions must be examined. Otherwise malfunctions of the PCS and/or the programmable controller are possible.

D Buep 19e-Direct driver

The BOSMULPS communication block is located on the BOSCH floppy disk. The effective response time speed between PLC and PCS depends on the baud rate, the used interface, and the data volume. E.g. the SK interface is therefore approx. 50% slower than the direct CPU interface.

- Access types
  The used B19EDIR driver is a direct driver i.e. it exchanges data words between the PLC and the PCS directly. No or just a small PLC program is required. Therefore, the scan time in the PLC is only slightly loaded.
- SYNC or NOSYNC
  Since the data interchange occurs asynchronous via several PLC writing and reading cycles, data written by the PLC can be overwritten by the PCS and conversely. Thus, a data consistency is not given. Possible solutions are either a strict separation of read and write data words (-> NO SYNC) or the use of a synchronization word (-> SYNCS). The use of a synchronization word enables the use of a timeout timer on the PCS side.

D1.1 Asynchronous communication

- Preset value and actual value data words must be strictly separated (write accesses can disturb each other). Even then, for example the reading of a variable which extends across several data words can result in an error at the time when the variable is read although only a part of it has been written.
- Bit variables should be used only 1 time per word since the access of the PCS occurs only word-by-word. A word fetched by the PCS which is changed and written back can overwrite another bit variable in this word. This applies also for the message bit area in delete behavior 2 (resetting of the bit in the PLC).
- Timeout monitoring is possible only in the PLC. To implement this function, the PCS sends a count word which is incremented by 1 in each communication cycle in the 3rd word. The use of the timeout timer is described in the handling software.

Advantages in comparison to the SYNC operation are:

- A faster data interchange. The scan time of the PLC does not influence the calculation of the response time.
- No program is needed in the PLC for communicating. Only the word area with the correct size must be available. Access to the word area in the PLC program must be possible at any time.

D Buep 19e-Direct driver

D1.2 Synchronous communication

The access to the data in the PLC must be synchronized if you want to make use of the entire functional extent of the PCS. I.e. the PLC and the PCS access data alternately. Therefore, a synchronization word is transmitted to the PLC. OB1 examines this word and releases the access of the PLC application program in the „test“ program. The synchronization word in PCS_KOMM is changed and the PCS accesses the data area if the application program has terminated the processing of the data words. While the PCS processes the data, the application program may not access to the data. This Ping-Pong game implements the possibility to perform a timeout monitoring both in PCS_KOMM as well as in the PCS. The timer is newly started whenever the PCS reads the inverted synchronization word. A timeout is present if the time of the timer expires. Implementing the alternating access allows for mixing of actual and preset values, using of bit variables, and realizing delete behavior 2 etc. Therefore, the entire intelligence of the PCS is at your disposal. The disadvantage is the decreasing reaction speed between PCS and programmable controller. Furthermore, access admission must be determined everywhere in the PLC program before accessing data.
The timeout time, i.e. the time which passes from the last writing to DW3 up to the timeout message in the PLC should be set to a minimum of 4 seconds. The timeout time in the PCS is set via the „AA“ or COM_TIMEOUT driver variable.

GENERAL PROCEDURE
To implement a synchronous communication between the PCS and the programmable controller:

- select the“SYNCHRONIZATION“ setting on the PCS
- load the PCS_SYNCS handling software in the PLC

This handling software is described in the following. Of course you can also implement these tasks differently in your software. It is only important to adhere to the following procedure:

1. The data area used by the PCS and the programmable controller must be defined in the programmable controller. The parameters are to be passed on to the PCS via the AL and AM driver variables. The programmable controller word words have initialization values and should be correctly pre-assigned.
2. The PCS sends the order number (byte-by-byte incremented by 1 every time) in word 3, e.g: „01“. Thus, word 2 is different from word 3. This signals the programmable controller that the data area can be processed.
3. After terminating the processing of the data area, the inverted sync word is copied from word 3 to word 2 and word 3, e.g. using the „FE“ data. This is the signal for the PCS to access to the data area. From now on, the programmable controller program is not allowed to perform any changes in the data area anymore! Furthermore, a timeout timer can be newly started (if one is used).

   From now on, steps 2 and 3 are cyclically performed. On a timeout, the communication loss sequence (in the example COFF) is executed once. The procedure is resumed at step 2 with order number“ 01".

D Buep 19e-Direct driver

D1.3 Description of the handling software

- **OB1**
  Call of the programs using parameters. Change the settings for the PCS_KOMM program here. Thus, no changes are required in PCS_KOMM. The „TEST“ data access program is selected in PCS_KOMM.
- **Test**
  Example for a data access:
  F1 activates the menu 1, F2 disabled all menus, F3 sets the message bits M0..15, F4 disables these message bits again. The keys are copied to the LEDS.
- **PCS_KOMM Communication program**
  The program evaluates whether an access to the data area is allowed in the programmable controller program. The „TEST“ program is called if an access is enabled. The sync word is processed after a data access. Also, a timeout timer will be using „TIMER“ to monitor the communica-tion. „COFF“ is selected and the M100.0. flag is set if the timer times out. With the E0.0 restart input set to „1“, communication will be restarted automatically. Otherwise it remains disabled.

Attention!
Check the functioning of the handling software to avoid malfunc-tioning of the PCS and/or programmable controller.

Parameterization of the PCS_ KOMM handling block:
P0: PCS transfer data block (here: DB1, 256 words)
The programmable controller and the PCS exchange informations via this data block.

P1: Switch for restart after faults (bit, here E0.0)
Using this bit, the communication can be restarted if a communi-cations error appears (faults are then reset automatically!). It is to be considered that no edge evaluation is made, i.e. with bit P2 set, the communication is automatically restarted after a fault has appeared.

P2: Time value for timeout monitoring
Structure:
e.g. K2.2 = 2 seconds timeout
multiplication element 1 = 100 ms, 2 = 1 s
value

P3: General error message (bit, here M100.0)
The P5 general fault bit is set for any error in the connection between the programmable controller and the PCS. This bit is automatically reset as soon as the communication is running without a fault.

P4: Initialization program
The data words for the programmable controller start are pre-assigned in this PB.

P5: Communication loss program
The data words for the communication loss are pre-assigned using this PB. The block will be executed once in case of an error.

D Buep 19e-Direct driver

Other:

Timer T1 „TIMER“ is used for timeout monitoring in the PCS_KOMM program.

INIT

Determines the initialization values of the PCS data area. Here, you can pre-assign the interlocks for the data transfer. Allow only so much data exchange as needed. With the exchange of fewer data, the communica-tion speed is increased.

COFF

Determines the measures at a communication loss. The block is executed once after a communication loss.

Attention!
Check the functioning of the handling software. The INIT and COFF blocks must be pre-assigned for the respective PCS. The program listing shows some examples.

D1.4 Implementation of the software

The README.DOC/PROJECT.DOC files on the PCS 91.BOS floppy disk contain current notes concerning the use of the various projects.

Procedure:

- Equip the rack
- Set the block address on CPU and I/O boards
- Set the interface on the CPU and the SK (if possible set all to the same parameters)
- Copy the corresponding project (e.g. BOSMULPS.500) stored on the PCS 91.BOS floppy disk to the hard disk
- Supply voltage to the programmable controller and the I/O boards
- Run the programmable controller programming software, select the project for the CPU, enter the configuration in the SK table and transfer it into SK (under LOADER).
- Transfer the programmable controller program into the controller using the loader (if necessary, link in your application program and link newly)
- Connect PCS and programmable controller using an adapter cable described in section 4 (are the DIL-switch segments 5 and 6 at the PCS correspon-dingly set?). The location for attaching the adapter cable is arbitrary. How-ever, the block address must be correct and the communication baud rate must correspond with the PCS. Plugging the communications cable into the addressed CPU will increase the communication speed.

D Buep 19e-Direct driver

- Set the E0.0 input to „1“

Now, the ERR LED (PCS) and the A0.0 output LED (PLC) must have been deactivated. Also the 7-segment display at the SK must remain dark. Thus, the communication is active.
Using this protocol, the addressed CPU can be accessed from an arbitrary programming unit connector. This way, it is possible to manage several PCSs by one CPU if each PCS is assigned it´s own communication block and data block. However, it is possibly dangerous to assign several PCSs to the same data block (the data transmitted by the PCS are then no longer clearly allocatable). A parallel mode of the programming unit and the PCS is easily possible.

D1.5 Boarding settings

Please set the segments of the switch of the ZS500 and SK500 boards for setting the interface parameters as follows:

S6 switch

DIL	1	2	3	4	5	6	7	8	
	ON	OFF	ON	ON	ON	ON	ON	OFF	19200 BAUD
	ON	OFF	ON	ON	OFF	ON	ON	OFF	9600 BAUD
	ON	OFF	ON	ON	OFF	ON	ON	OFF	4800 BAUD
	ON	OFF	ON	ON	ON	OFF	ON	OFF	1200 BAUD

Please set the segments of switch 5 of the ZS500 and SK500 boards (I/O groups in parenthesis) for setting the board address as follows:

DIL	1(7)	2(8)	Board
	OFF	OFF	0
	ON	OFF	1
	OFF	ON	2
	ON	ON	3

For adjustment of the transfer rate adjust the DIP-switches for the CL200 as following:

DIL	1	2	Baud
	OFF	OFF	9600
	ON	OFF	19200
	OFF	ON	38400

Tip!
Inspect the function of the handling software to avoid failures of the PCS or PLC.

D Buep 19e-Direct driver

R200 coupling card settings

Pay attention that in the CPU of the CL200 is loadrd at least the Firmware version 1.5 of 06.03.1996 eingespielt ist.
On the settings of the card pay attention to the DIP-switches S1 and S2.

DIP-Switches

DS1	**DS2**	**Protokoll**
ON	OFF	BUEP19e-Protocol

DS3	**DS4**	
ON	ON	Parity

DS5	**DS6**	**DS7**	**Transfer rate**
OFF	OFF	OFF	38 400 Baud
ON	ON	ON	19 200 Baud
OFF	ON	ON	9 600 Baud
ON	OFF	ON	4 800 Baud

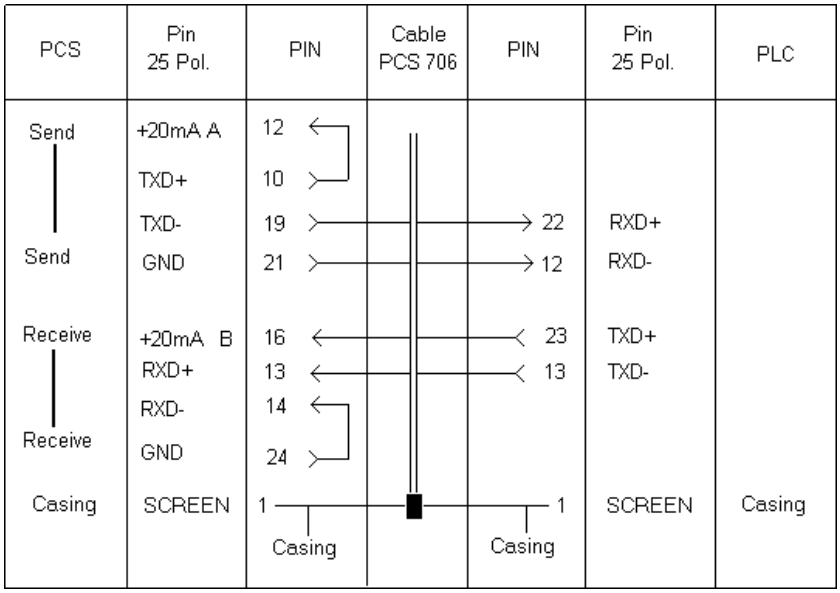
DS8	
OFF	Control signals not evaluate

D Buep 19e-Direct driver

D2 Communication

Adapter cable PCS 706

Connection PCS-PLC via TTY interface



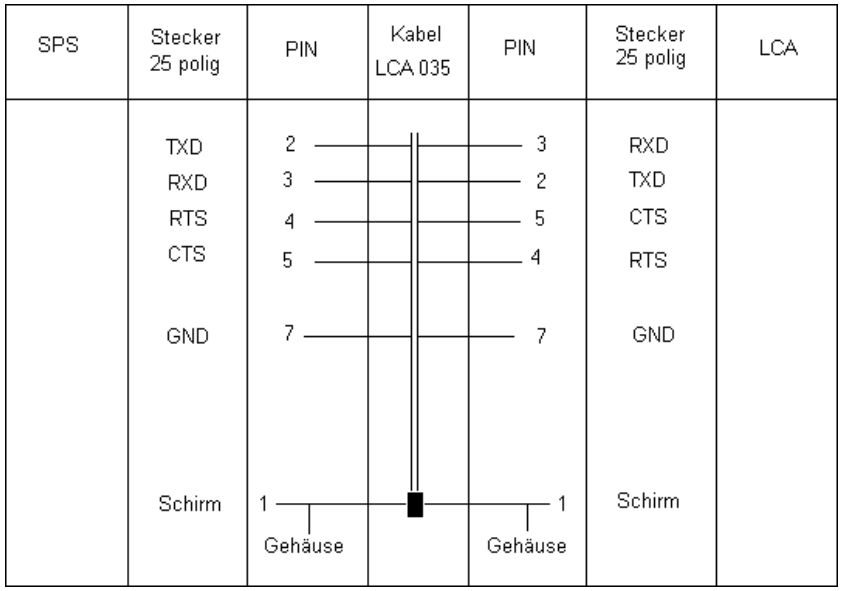
Screening

The shield should be connected on both sides to a metallized connector hood. If non-metallized connector hoods are used, the shield may also be connected to pin 1, but this is not recommended due to interferences, because the data lines should be completely covered by the shield (if possible). It should be noticed, however, that earthing on both sides may require an equipotential bonding conductor with a cross section of 10 times that of the shield in order to compensate for earth potential variations and to prevent equalizing currents on the cable shield! This is especially important, if the PCS and the programmable controller are not connected to the same common point (if they are for example installed in different control cabinets).

D Buep 19e-Direct driver

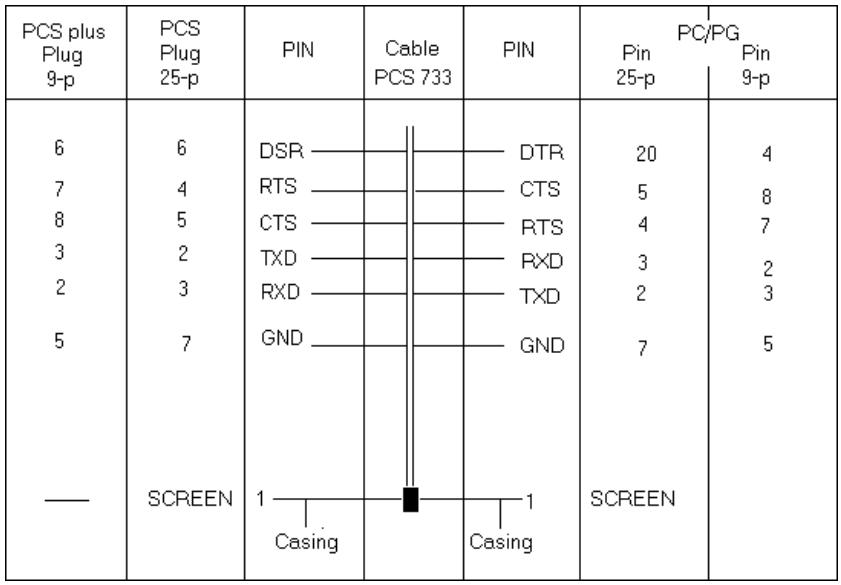
LCA 035/235

PCS to PLC via RS-232 interface
Use a zero-modem cable for the connection of the PLC to PCS via RS-232C. The necessary connections are as follows:



Programming cable PCS 733

Connection PLC to PCS
Use this cable for the programming (loading of the driver and user program) of the PCS.



D Buep 19e-Direct driver

D3 Program listing

```
      ;***************************************************************************
      ;     OB1
      ;***************************************************************************
      ; File name:      OB1
      ; Function:       Organizational block

      ;***************************************************************************
      ;      P r o g r a m
      ;***************************************************************************
1  BA       -PCS_KOMM,6
   P0 W     -DB1
   P1 B     -RESTART
   P2 W     K4.2
   P3 B     M100.0
   P4 W     -INIT
   P5 W     -COFF

   PZ:  1
2  U  B     M100.0      ; display communication loss
3  =  B     A0.0

4  PE             ; ------------------------------------------->
   ;Program end

-COFF       PB3   PCS communication loss
-DB1              DB1   PCS data field
-INIT             PB2   PCS initialization values
-PCS_KOMM   PB0   SYNC block for the CL500
-RESTART          E0.0  Restart input for communication loss


   Para.    Res.  Ind   Symbol      < >         Comment     | Version: 1.0
   P0 W           P0          <     DB    Application db, keys, LED, variable, ...
   P1 B           P1          <     E     restart input
   P2 W           P2          <     K     „watch dog“ time value
   P3 B           P3          >     M     communication error output
   P4 W           P4          <     PB    communication start pre-assignments
   P5 W           P5          <     PB    communication loss pre-assignments


      ;***************************************************************************
      ;     PCSKOMM
      ;     Synchronization block for the communication with a PCS 090/095
      ;     via 5P90BUEP.drv drivers
      ;***************************************************************************
      ; File name: PCS_KOMM
      ; Function:       Sync block for the CL00
      ;                       Transfer format: BUEP19E
      ;

      ;***************************************************************************
      ;      D e c l a r a t i o n
      ;***************************************************************************

      ; Block parameters:
      ;~~~~~~~~~~~~~~~~~~~~~~~~~
      ;BA    -PCS_KOMM,7
      ;P0    W     DB1   ;     <=    DB    Application db, keys, LED, variable ..
      ;P1    B     m     ;     <=    M     autom. restart after „watch dog“
      ;P2    W     k4.2  ;     <=    K     „watch dog“ time value
      ;P3    B     M100.0      ;     =>    M     communication error „watch dog“ active
      ;P4    W     PB2   ;     <=    PB    pre-assignment block start communication
      ;P5    W     PB3   ;     <=    PB    error block communication loss
```

D Buep 19e-Direct driver

```
     ;*************************************************************************
     ;      P r o g r a m
     ;*************************************************************************
1     BX          -P0   ; open PCS DB
     ;*************************************************************************
     ;      Processor reset => INIT
     ;———————————————————-
      PZ:   1
2     U     B     -P_RI ; start after Halt
3     O     B     -NEU_RI     ; restart
4     SPB         -START_KO ; ------>
      ; no processing in case of timeout and restart input = 0
      ;———————————————————————————
      PZ:   2
5     UN    B     -P1   ; timeout ?
6     U     B     -P3   ; restart ?
7     BEB
      ; Sync word 2 = Sync word 3 ? yes = end
      ;———————————————————————————-
8     L     W     DX4,A
9     L     W     DX6,B
-NEU_RI           SM20.7      RI    on power-up or prog. restart
-P0               P0    DB    Application db, keys, LED, variable, ...
-P1               P1    E     restart input
-P3               P3    M     comm. error output
-P_RI             SM20.0            Processor RI after every processor Halt
10    VGL   W     B,A
11    SPZ         -NO_AUFTR
12    BA    -TEST       ; Data word access
      ;=======================================
      ;                                       Order processed
            -AUFT_END         ; <———
      ; negate sync word and => D2, D3
      ; ——————————————————-
13    L     W     D6,D        ; read order no. D3
14    XO    W     KFFFH,D           ; negate order no.
15    T     W     D,DX6       ; write order no. D2,D3
16    T     W     D,DX4
      ; watch dog
      ; ———————
      ; time routine 0 edge for new triggering
      ; ————————————————————
17    U     W     -P2,D       ; time value
      PZ:   3
18    U     B     -LOGO
19    SE          D,-TIMER    ; time routine = 0
      PZ:   4
20    U     B     -LOG1       ; reset watchdog flag
21    R     B     -P3
22    BE                ; ————————————————————>
      ;                                  End order processed
      ; =======================================
      ;                                  End order processing
      ; ***************************************************************
      ;                                       no order
      ;     -NO_AUFTR         ; <———————
23    L     W     -P2,D       ; time value
      PZ:   5
24    U     B     -LOG1
25    SE          D,-TIMER
      PZ:   6
26    U     B     -TIMER            ; time routine
27    BEI               ; ————————————————————>
-LOG0 SM30.3      logical 0
-LOG1 SM31.1      logical 1
-P2         P2          K     time value for watch dog
-P3         P3          M     comm. error output
```

D Buep 19e-Direct driver

```
-TEST PB1   PCS data access example
-TIMER      T0          PCS communication watch dog timer

   ; set fault  watch dog has triggered
   ;————————————————————————————
      -KOM_FEL
28 S  B     -P3   ; watch dog is active
29 X0 W     D,D
   ; initialize application data block
   ;————————————————————————-
30 BA       -P0   ; open user data block
31 BA       -P5   ; COFF
32 BE             ; ————————————————————>
   ;                              End watch dog
   ; ******************************************************************


   ; ******************************************************************
   ;                              Start communication
      -START_K0         ; <————
   ; time routine 0 edge
   ; ——————————-
33 L  W     -P2,D       ; time value for watch dog
   PZ:      7
34 U  B     -LOG0
35 SE       D,-TIMER    ; time routine = 0

   ; reset fault
   ; ——————-
   PZ:      8
36 U  B     -LOG1
37 R  B     -P3         ; watch dog is active = 0

   ; initialize application data block
   ; ————————————————————
38 BA       -P0         ; open user data block 1.DB
39 BA       -P4         ; INIT
40 BE             ; ————————————————————>
   ;                              End start communication
   ; ******************************************************************

   ; Program end
-LOG0 SM30.3      logical 0
-LOG1 SM31.1      logical 1
-P0         P0          DB    application data block, keys, variable, ...
-P2         P2          K     time value for watch dog
-P3         P3          M     comm. error output
-P4         P4          PB    communication start presetting
-P5         P5          PB    communication loss presetting
-TIMER      T0          PCS communication watch dog timer
```

D Buep 19e-Direct driver

```
     ;*************************************************************************
     ;      TEST: Example program for PCS 090/095 data access
     ;*************************************************************************
1    BA          -DB1  ; open data block

2    L     W     D8,A
3    VGL   W     K8000H,A
4    SPN         -TEST1

5    L     W     K0081H,A    ; F1 -> preset value menu 1
6    T     W     A,D28

           -TEST1

7    L     W     D8,A
8    VGL   W     K4000H,A
9    SPN         -TEST2

10   L     W     K0000H,A    ; F2 -> preset value menu 1 off
11   T     W     A,D28

           -TEST2

12   L     W     D8,A
13   VGL   W     K2000H,A
14   SPN         -TEST3

15   L     W     KFFFFH,A    ; F3 -> message bits M0..15 on
16   T     W     A,D30

           -TEST3

17   L     W     D8,A
18   VGL   W     K1000H,A
19   SPN         -TEST4

20   L     W     K0000H,A    ; F4 -> message bits M0..15 off
21   T     W     A,D30

           -TEST4

22   L     W     D8,A
23   T     W     B,D20 ; keys -> LEDs

24   BE
```

D Buep 19e-Direct driver

```
;*************************************************************************************************
      ;** COFF (APPLICATION-SPECIFIC ERROR PRESETTING; PCS COMMUNICATIONS LOSS)

;*************************************************************************************************

                              ;!!!!! ADD PCS AND EQUIPMENT-SPECIFIC
     ;********************************* ERROR PRESSTINGS FOR COMMUNICATIONS
     ;********************************* LOSS HERE !!!!!
     ;********************************* !!!!! ATTENTION: !!!!!
     ;********************************* SET AT LEAST HERE KEYWORDS TO ZERO
     ;********************************* (PCS-SPECIFIC) !

      ;*****EXAMPLE PCS 090/095*****
1     L     W     K0D,A
2     T     W     A,D4W ;KEYS
3     T     W     A,D5W
4     T     W     A,D23W

     ;*****EXAMPLE PCS 900*****
      ;L    W     K0D,A
      ;T    W     A,D4W ;KEYS
      ;T    W     A,D5W
      ;T    W     A,D6W
                        ;set possible additional keys also to 0
      ;T    W     A,D9W ;data

5     BE



;*************************************************************************************************
      ;** INIT (APPLICATION-SPECIFIC PRESETTINGS; PCS COMMUNICATIONS START)

;*************************************************************************************************

                               ;!!!!! ADD OR INSERT PCS AND EQUIPMENT-SPECIFIC
      ;*********************************ERROR PRESETTINGS FOR COMMUNICATION
      ;*********************************START HERE !!!!!

      ;*****EXAMPLE PCS 090/095*****
1     L     W     KOD,A
2     T     W     A,D4W ;KEYS
3     L     W     A,D5W
4     T     W     A,D23W
5     L     W     K0FC8H,A
6     T     W     A,D13W      ;COMMAND WORD A
7     L     W     K0080H,A
8     T     W     A,D14W      ;COMMAND WORD B
      ;*****EXAMPLE PCS 900*****
      ;L    W     K0D,A
      ;T    W     A,D4W ;KEYS
      ;T    W     A,D5W
      ;T    W     A,D6W
      ;T    W     A,D9W ;data
      ;T    W     A,D39W      ;COMMAND WORD D
      ;L    W     K0080H,A
      ;T    W     A,D38W      ;COMMAND WORD C
      ;L    W     K00FFH,A
      ;T    W     A,D37W      ;COMMAND WORD B
      ;L    W     K1F00H,A
      ;T    W     A,D36W      ;COMMAND WORD A

5     BE
```

Index

Index