PCS/VPC 91.AEG
AEG

PCS/VPC 91. AEG

Appendix to the PCS 091/PCS 925/PCS 991/PCS 9091 manuals

PLC systems	(1)	A120 (ALU 200, 201, or 202), A250
PLC interfacing	(2)	with PCS 859 EPROM for KOS 201 or KOS 202 with PCS 776 adapter cable (only for A120)
	(2)	direct driver with the PCS 778 adapter cable
	(2)	expander driver with the PCS 778 adapter cable (only for A250)
PLC systems	(1)	Modicon 984 series with CPU 120, 130, 145, 380, 381, 385, 480, 485, 680, 685, 780, 785, X, A, and B
PLC interfacing	(2)	via MODBUS 1 interface with the PCS 780 adapter cable

Reg 5049/0996ss

Vers. 1/09.96

A

Information for the driver selection

Lauer driver

Fast A120 communication with the PCS micro/mini/midi and maxi. Requires programmable controller program and KOS module

Operating unit:	PCS 009/PCS 090/PCS 095/PCS 900/PCS 920 /PCS 950
Programmable controller system:	AEG A120
Interface:	KOS 201/202 module + PCS 859 (EPROM firmware)
Protocol:	Lauer
Adapter cable:	PCS 776
Driver type:	expander driver

"AEG120DR" driver

'Plug and Play' communication using the PG interface without programmable controller program. Less fast than the expander driver.

Operating unit:	PCS 009/PCS 090/PCS 095/PCS 900/PCS 920 /PCS 950
Programmable controller system:	AEG A120 / A250
Interface:	PG interface
Protocol:	BKOS
Adapter cable:	PCS 778
Driver type:	direct driver

EDITION
SEPTEMBER
1996

Information for the driver selection

A

"MODBUSPS" driver

'Plug and Play' communication between Modicon programmable controllers and PCS

Operating unit:	PCS 009/PCS 090/PCS 095/PCS 900/PCS 920 /PCS 950
Programmable controller systems:	Modicon 984-120, - 380, -381, -385, -480, -485, -680, -685, -780, -785, -X, -A, -B
Interface:	Modbus interface
Protocol:	MODBUS 1 SLAVE
Adapter cable:	PCS 780
Driver type:	direct driver

"AEG250PX" driver

Fast A250 communication with the PCS midi and maxi. Requires programmable controller program. Uses PG interface.

Operating unit:	PCS 009/PCS 090/PCS 095/PCS 900/PCS 920 /PCS 950
Programmable controller system:	AEG A250
Interface:	PG interface
Protocol:	BKOS
Adapter cable:	PCS 778
Driver type:	expander driver

Table of contents

A. Lauer driver

B. AEG120DR driver

Table of contents

Table of contents

Description of the loadable AEG120DR/AEGBKOS driver for the PCS 009/090/095/900 operating consoles. For communication with an AEG 120/250 PLC.
Functionality: Selective direct driver for KOS 201/202 PLCs.

Required devices and accessories:

1. The PCS operating console itself (already parameterized).
2. The PCS 776 connecting cable for the PCS - programmable controller connection via the PU interface.
3. This manual
4. Firmware EPROM PCS 859

Furthermore are necessary (AEG company):

5. An A120 programmable controller with rack, power supply and I/O boards.

.... as well as power supplies for all components.

A1.1 Settings at the board

A1.1.1Addressing of the board

The KOS 201/202 board is only pluggable in the base magazine, i.e. DTA 200 (slots 1, 2 and 3). Slot 1 however, is required by the DNP 205 power supply board, at least when using the ALU 201 or 202. Thus, slot 2 or 3 remains in the DTA 200 rack. The communication with the ALU is performed via a dual port RAM and the PAB1 parallel equipment bus of the system. In addition, 128 IBs and 128 OBs are used. The programmable controller user however, has only contact to the PCS via 128 OBs (64 words altogether).

- Slot selection
  Slot 2: use handling software from the 9002 programming unit directory.
  Slot 3: use handling software from the 9003 programming unit directory.

Basically, slot 1 can also be used since the DNP 205 power supply board can be operated on slot 2 or 3. In addition, all function blocks belonging to the handling software would have to be rewritten however (replace EBx.y and ABx.y with EB1.y and AB1.y).

A1.1.2 Baud rate settings

The baud rate setting for the communication between the PCS and the KOS 201 board is made by the software in OB1. The position (first line; instruction: L KHxx) is mark by a comment! This constant is assigned a flag byte which in turn submits an FB 200 parameter (BAUD). The following baud rates are possible: 1200 baud (L KH00), 4800 baud (L KH01), 9600 baud (L KH02) and 19200 baud (L KH03).

A1.2 Description of the handling software

The enclosed FB 200 manages the data traffic between a transfer area (ABx.1 - ABx.128) and the KOS board.

PB 200 is selected by the FB 200. This is used to copy all Ebx.y (data from the PCS 200) onto ABx.y.

PB 201 and PB 202 are used to offer the user the possibility to install customer specific presettings at restart (PB 201) or fault situations (PB 202) without intervening in FB 200.

Apart from the cyclical call of FB 200 at the beginning of the programmable controller cycle, incl. setting the baud rate with the BAUD parameter, the implementation limits itself to reading and writing ABx.1 - ABx.128. A summary fault bit (ERROR) enables the communication status to be evaluated via the ladder diagram program. If this fault bit is = 1, then the data transfer has been stopped. It can be restarted by presetting a flag (RESET). A further flag (TOGGLM) is required for short/long cycles. This will be toggled and indicates in each programmable controller cycle whether the programmable controller cycle can be terminated after calling FB 200 (short cycle), or whether it is useful to process further statement list commands (long cycle).

A1.3 Parameterization of FB 200

- RESET
  Switch for resetting after faults (input: bit)

If a communication fault appears, then the communication can be restarted with this bit. The faults indicated in ABx.6 and ABx.7 are then automatically reset! It is to be noted that no edge evaluation is made, i.e. when the bit is set RESET starts the communication automatically again after the appearance of a fault! This mechanism can also be realized by applying the fault output ERROR directly to RESET.

- ERROR
  Global error message (output: bit)

If a fault appears anywhere in the connection between PLC and PCS 200, then the ERROR global fault bit is set. ABx.6 and ABx.7 can be procured to determine the exact cause of the fault. This bit is reset as soon as communication runs correctly again.

- BAUD
  Baud rate (input: flag byte)

This flag byte is preset with the desired baud rate before calling FB 200. Possible values are KH00 (1200 baud), KH01 (4800 baud), KH02 (9600 baud) and KH03 (19200 baud).

- TOGGL
  Short / long cycle indication (output: bit)

Since the ALU reads first of all the input bytes at the cycle end, and subsequently writes output bytes and not conversely, is it necessary to insert a short cycle (for each second programmable controller cycle). The short cycle is marked thereby, that this bit is = 0. If this is the case, then the programmable controller cycle can be terminated after calling FB 200 as is realized in the example.

PB201:	customer-specific presetting for the restart
PB202:	customer-specific assignment for the fault case

A1.4 Implementation of the handling FB

1. Turn off the programmable controller
2. Plug the 201/202 KOS board into slot 2 or 3 of the DTA 200
3. Switch the programmable controller to stop and supply voltage
4. Depending on the used slot, add the PB200, PB201, PB202 function blocks from the PCS 9002 or PCS 9003 to each own project. Modify perhaps PB201 and PB202 for your own demands.
5. Expand OB1 corresponding to the example on the floppy disk:
   - load the baud rate in a flag byte
   - call FB 200
   - if TOGGLM = 0 → function block end, otherwise process your own pro gram.
6. Determine the momentary reset pushbutton and set with positive edge (RESET)
7. „Bind" all function blocks and load these into the controller
8. Switch the programmable controller to RUN

If the KH20H fault appears in ABx.6 (time-out, i.e. the PCS is not connected) and the global error message (ERROR) is logical 1, then the implementation is successfully completed. See the trouble-shooting section 1.4 if this is not the case!
An example (OB1) is contained on the floppy disk which indicates faults on output 5.1, and awaits a momentary reset pushbutton at I 4.1. The baud rate submitted in MB1 is set as the baud rate, and must be identical with the baud rate set via the DILs on the PCS 200.

A1.5 Program integration

Regarding the integration into your program, the following transfer operations must be executed (non-required functions can be skipped):

- All values read by the PCS must be copied to the corresponding ABx.y. This affects:
  - word 15 to word 19 (ABx.30 to ABx.39)
  - all used message bits
  - all variables

- All values modified by the PCS must be written back into the corresponding flags/flag bytes/flag words. This affects:
  - word 4 to word 14 (ABx.8 to ABx.29)
  - all message bits with erase behavior 2
  - all message bits for concealed logging with erase behaviors 2 -4
  - all preset values

After restart, the PCS proceeds with the following assumptions:

- Date and time as well as the timer have become invalid and will therefore be transferred immediately.

- All message bits are in the same state as before switching off.

- 0 is selected as idle text.

- All menus have been terminated before switching off, otherwise they must be restarted.

- After restart, the momentary pushbuttons and DIL switches will all be transferred once.

This run-up behavior is to be noted when presetting the transfer area.

Only the listing for slot 2 (9002 programming unit) is printed! The listing for slot 3 is identical with the slot 2 except for the EBx/ABx numbers (EB2.Y and AB2.Y are replaced with EB3.Y and AB3.Y).

9002 PCS9002\FB200

NETWORK: 0001

NAME: HANT_PCS

```
BEZ: RESET (E/Ex/A/Ax/M/Mx/SM/SMx/T/Z/TN/B2/B8/B16/ANZ)   (I/O)      M
I
BEZ: ERROR (E/Ex/A/Ax/M/Mx/SM/SMx/T/Z/TN/B2/B8/B16/ANZ)   (I/O) M    O
BEZ: BAUD (E/Ex/A/Ax/M/Mx/SM/SMx/T/Z/TN/B2/B8/B16/ANZ)    (I/O) MB   I
BEZ: TOGGLM (E/Ex/A/Ax/M/Mx/SM/SMx/T/Z/TN/B2/B8/B16/ANZ)  (I/O) M    O
      :***

NETWORK: 0002

      :L    =BAUD
      :U    KH 3
      :=    AB 2.1      BAUD RATE -> COMMAND BYTE
      :U    SM 2  ^     activating flag
      :SPZ  =NORM       FIRST CYCLE?
      :L    AB 2.1
      :O    KH C0
      :=    AB 2.1      THEN RESET COMMAND  -> COMMAND B.
      :BA   PB 201      AND MACHINE SPEC. PRESETTINGS
      :SP   =END
NORM  :UN   =TOGGLM
      :=     =TOGGLM
      :SPZ  =END        SHORT CYCLE?,  IF YES -> END
      :L    AB 2.1
      :O    KH 40
      :=    AB 2.1      LONG CYCLE -> COMMAND B.
      :BA   PB 200      COPIES ALL EB2.X -> AB2.X
      :L    AB 2.6
      :O    AB 2.7
      :<>    KH 0       FAULT FOUND?
      :=     =ERROR     THEN SET FAULT BIT AND
      :BAB  PB 202      CALL EMERGENCY SETTINGS!
      :U    =ERROR
      :U    =RESET
      :SPZ  =END        IF FAULT AND RESET, THEN
      :BA   PB 201      MACH. SPEC. PRESETTING AND COMM.START
END:***
```

```
NETWORK: 0003

      :BE

PCS 9002\PB200

NETWORK: 0001

      :L    EB 2.6
      :=    AB 2.6
      :L    EB 2.7
      :=    AB 2.7
      :L    EB 2.8
      :=    AB 2.8
      :L    EB 2.9
      :=    AB 2.9
      :L    EB 2.10
      :=    AB 2.10
      :L    EB 2.11
      :=    AB 2.11
      :L    EB 2.12
      :=    AB 2.12
      :L    EB 2.13
      :=    AB 2.13
      :L    EB 2.14
      :=    AB 2.14
      :L    EB 2.15
      :=    AB 2.15
      :L    EB 2.16
      :=    AB 2.16
      :L    EB 2.17
      :=    AB 2.17
      :L    EB 2.18
      :=    AB 2.18
      :L    EB 2.19
      :=    AB 2.19
      :L    EB 2.20
      :=    AB 2.20
      :L    EB 2.21
      :=    AB 2.21
      :L    EB 2.22
      :=    AB 2.22
      :L    EB 2.23
      :=    AB 2.23
      :L    EB 2.24
      :=    AB 2.24
      :L    EB 2.25
      :=    AB 2.25
      :L    EB 2.26
      :=    AB 2.26
      :L    EB 2.27
      :=    AB 2.27
      :L    EB 2.28
      :=    AB 2.28
      :L    EB 2.29
      :=    AB 2.29
      :L    EB 2.30
      :=    AB 2.30
      :L    EB 2.31
      :=    AB 2.31
      :L    EB 2.32
      :=    AB 2.32
      :L    EB 2.33
      :=    AB 2.33
```

```
:L    EB 2.34
:=    AB 2.34
:L    EB 2.35
:=    AB 2.35
:L    EB 2.36
:=    AB 2.36
:L    EB 2.37
:=    AB 2.37
:L    EB 2.38
:=    AB 2.38
:L    EB 2.39
:=    AB 2.39
:L    EB 2.40
:=    AB 2.40
:L    EB 2.41
:=    AB 2.41
:L    EB 2.42
:=    AB 2.42
:L    EB 2.43
:=    AB 2.43
:L    EB 2.44
:=    AB 2.44
:L    EB 2.45
:=    AB 2.45
:L    EB 2.46
:=    AB 2.46
:L    EB 2.47
:=    AB 2.47
:L    EB 2.48
:=    AB 2.48
:L    EB 2.49
:=    AB 2.49
:L    EB 2.50
:=    AB 2.50
:L    EB 2.51
:=    AB 2.51
:L    EB 2.52
:=    AB 2.52
:L    EB 2.53
:=    AB 2.53
:L    EB 2.54
:=    AB 2.54
:L    EB 2.55
:=    AB 2.55
:L    EB 2.56
:=    AB 2.56
:L    EB 2.57
:=    AB 2.57
:L    EB 2.58
:=    AB 2.58
:L    EB 2.59
:=    AB 2.59
:L    EB 2.60
:=    AB 2.60
:L    EB 2.61
:=    AB 2.61
:L    EB 2.62
:=    AB 2.62
:L    EB 2.63
:=    AB 2.63
:L    EB 2.64
:=    AB 2.64
:L    EB 2.65
:=    AB 2.65
:L    EB 2.66
```

```
:=    AB 2.66
:L    EB 2.67
:=    AB 2.67
:L    EB 2.68
:=    AB 2.68
:L    EB 2.69
:=    AB 2.69
:L    EB 2.70
:=    AB 2.70
:L    EB 2.71
:=    AB 2.71
:L    EB 2.72
:=    AB 2.72
:L    EB 2.73
:=    AB 2.73
:L    EB 2.74
:=    AB 2.74
:L    EB 2.75
:=    AB 2.75
:L    EB 2.76
:=    AB 2.76
:L    EB 2.77
:=    AB 2.77
:L    EB 2.78
:=    AB 2.78
:L    EB 2.79
:=    AB 2.79
:L    EB 2.80
:=    AB 2.80
:L    EB 2.81
:=    AB 2.81
:L    EB 2.82
:=    AB 2.82
:L    EB 2.83
:=    AB 2.83
:L    EB 2.84
:=    AB 2.84
:L    EB 2.85
:=    AB 2.85
:L    EB 2.86
:=    AB 2.86
:L    EB 2.87
:=    AB 2.87
:L    EB 2.88
:=    AB 2.88
:L    EB 2.89
:=    AB 2.89
:L    EB 2.90
:=    AB 2.90
:L    EB 2.91
:=    AB 2.91
:L    EB 2.92
:=    AB 2.92
:L    EB 2.93
:=    AB 2.93
:L    EB 2.94
:=    AB 2.94
:L    EB 2.95
:=    AB 2.95
:L    EB 2.96
:=    AB 2.96
:L    EB 2.97
:=    AB 2.97
:L    EB 2.98
```

```
      :=    AB 2.98
      :L    EB 2.99
      :=    AB 2.99
      :L    EB 2.100
      :=    AB 2.100
      :L    EB 2.101
      :=    AB 2.101
      :L    EB 2.102
      :=    AB 2.102
      :L    EB 2.103
      :=    AB 2.103
      :L    EB 2.104
      :=    AB 2.104
      :L    EB 2.105
      :=    AB 2.105
      :L    EB 2.106
      :=    AB 2.106
      :L    EB 2.107
      :=    AB 2.107
      :L    EB 2.108
      :=    AB 2.108
      :L    EB 2.109
      :=    AB 2.109
      :L    EB 2.110
      :=    AB 2.110
      :L    EB 2.111
      :=    AB 2.111
      :L    EB 2.112
      :=    AB 2.112
      :L    EB 2.113
      :=    AB 2.113
      :L    EB 2.114
      :=    AB 2.114
      :L    EB 2.115
      :=    AB 2.115
      :L    EB 2.116
      :=    AB 2.116
      :L    EB 2.117
      :=    AB 2.117
      :L    EB 2.118
      :=    AB 2.118
      :L    EB 2.119
      :=    AB 2.119
      :L    EB 2.120
      :=    AB 2.120
      :L    EB 2.121
      :=    AB 2.121
      :L    EB 2.122
      :=    AB 2.122
      :L    EB 2.123
      :=    AB 2.123
      :L    EB 2.124
      :=    AB 2.124
      :L    EB 2.125
      :=    AB 2.125
      :L    EB 2.126
      :=    AB 2.126
      :L    EB 2.127
      :=    AB 2.127
      :***
NETWORK: 0002

      :BE
```

- RESTART-PB

```
9002 PROGRAMMING UNIT  \ PB201

NETWORK: 0001

      :L    KH 0
      :=    AB 2.6           FAULT WORD RESET
      :=    AB 2.7
      :=    AB 2.8           SET KEY BITS TO 0
      :=    AB 2.9
      :=    AB 2.10
      :=    AB 2.11
      :=    AB 2.12
      :=    AB 2.13
      :=    AB 2.14          DATE/TIME INVALID 1
      :=    AB 2.15
      :=    AB 2.35
      :=    AB 2.36          COMMAND WORD A (KH 0F00)
      :=    AB 2.38          COMMAND WORD B (KH 0800)
      :=    AB 2.39          COMMAND WORD C (KH 0000)
      :L    KH F
      :=    AB 2.34
      :L    KH 80
      :=    AB 2.37
      :***

NETWORK: 0002

      :BE
```

- PB FOR COMMUNICATION LOSS

```
PCS 9002\PB202

NETWORK: 0001

      :L    KH 0
      :=    AB 2.8           ZERO KEY BITS!!
      :=    AB 2.9
      :=    AB 2.10
      :=    AB 2.11
      :=    AB 2.12
      :=    AB 2.13
      :=    AB 2.14          DATE/TIME INVALID!!
      :=    AB 2.15
      :***

NETWORK: 0002

      :BE
```

- CALL EXAMPLE

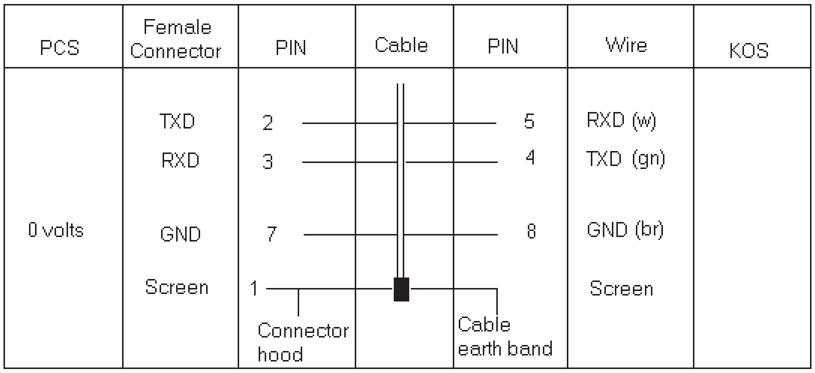
This example shows how the parameterization on the enclosed floppy disk is realized.

```
 PS9002 \ OB1

NETWORK: 0001

      :L    KH 3       * * ENTER BAUD RATE HERE
      :=    MB 1       * * 0: 1200 BAUD, 1:  4800 BAUD
      :BA   FB 200     * * 2: 9600 BAUD, 3: 19200 BAUD
NAME  :HANT_PROGRAMMING UNIT
RESET :     M 1.1
ERROR :     M 1.2
BAUD  :     MB 1
TOGGLM:     M 1.3
      :UN   M 1.3       IF  =  0  =>  SHORT CYCLE!!
      :BEB
      :NOP              ********************************
      :NOP              ** !!!!  WARNING  !!!!        **
      :NOP              ** ONLY OWN PLC PROGRAM       **
      :NOP              ** FROM HERE ONWARDS!!!!      **
      :NOP              ********************************
      :U    M 1.2
      :=    A 5.1       GLOBAL ERROR MESSAGE
      :U    E 4.1
      :FLP  M 1.4
      :=    M 1.1       RESET ONLY ON POSITIVE EDGE
      :***

NETWORK: 0002

      :BE
```

```
  1  Rev. 0  31.07.91  SYSTEME  Date  31.07.1991
                       LAUER    Off.  Ran.        PCS200 (KOS201 STECKPL.2)
                       Check                      PCS300 (KOS201 STECKPL.2)
  No.Change  Date      Name     Stand                                    AEG
```

A3.1 Communication cable programming UNIT/KOS

The connection is made with one 3-wire cable (TXD, RXD, GND)



PCS	Female Connector	PIN	Cable	PIN	Wire	KOS
	TXD	2		5	RXD (w)	
	RXD	3		4	TXD (gn)	
0 volts	GND	7		8	GND (br)	
	Screen	1			Screen	
		Connector hood		Cable earth band		

When using shielded standard cable (3 * 0.14, not twisted in pairs) a maximum length of 15 meters is recommended. A connection interruption is registered in both the PCS 200 and 850 and noted in the fault word (ABx.6/ABx.7) for further evaluation.

A3.1.1 Screening of the communication cable

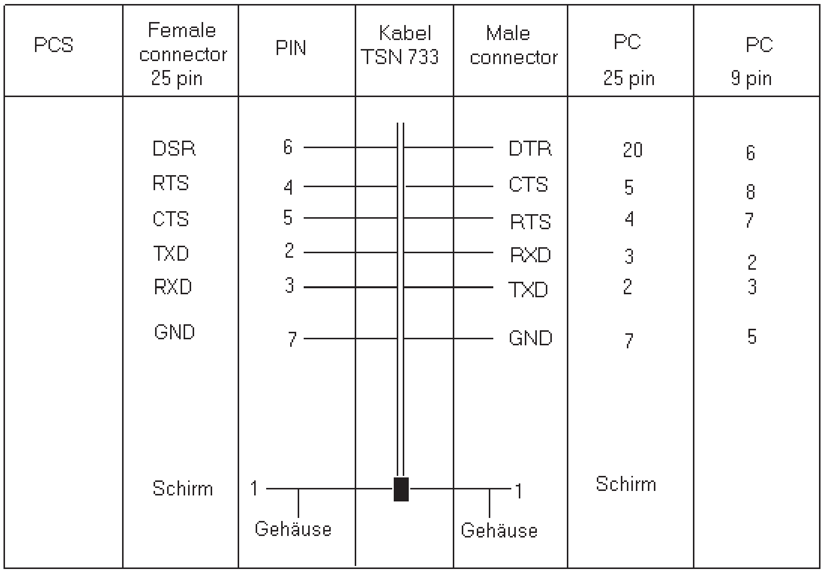
The screen should be connected on both sides to a metal coated connector case. The screen can also be connected to pin 1 when using non-metal coated connector cases. However, this is not recommended for error-technical reasons. There, the data lines should be covered as completely as possible by the screen! With grounding at both sides is to be noted however that possibly (because of ground potential shifts) a potential compensation wire is required having at least 10 times the screen cross section (reason: compensation currents should not flow through the cable screen if possible!). This applies particularly if PCS and programmable controller are not connected to the same earthing point. This is for example valid where the PCS and programmable controller are not housed in one switching cabinet! For double sided earthing, use a cable earthing bar, type CER 001, AEGS E-NR. 424 244 739 on the PCS side. Install according to AEG instructions!

Sketch KOS boards (top view)

1	°	gn				•	1
2						•	2
3	°	yel				•	3
4	°	yel	←	TXD	→	•	4
5	°	yel	←	RXD	→	•	5
6						•	6
7						•	7
8				GND	→	•	8
9						•	9
10						•	10
11						•	11
12	°	yel	←	Ready		•	12
13						•	13
14						•	14
15						•	15
16						•	16
17						•	17
18						•	18
19						•	19
20						•	20
21	°	red				•	21
22	°	red				•	22

°- LED
•- screw-type terminal

A3.2Programming cable PCS 733



PCS	Female connector 25 pin	PIN	Kabel TSN 733	Male connector	PC 25 pin	PC 9 pin
	DSR	6		DTR	20	6
	RTS	4		CTS	5	8
	CTS	5		RTS	4	7
	TXD	2		RXD	3	2
	RXD	3		TXD	2	3
	GND	7		GND	7	5
	Schirm	1 Gehäuse		1 Gehäuse	Schirm	

Description of the loadable AEG 120DR/AEGBK05 driver for the PCS 009/090/095/900 operating consoles. For the communication with an AEG 120/250 PLC.
Functionality: Selective direct driver for AEG 120/AEG 250 PLCs.

Required devices and accessories:
1. The PCS operating console itself (already parameterized).
2. The PCS 778 connecting cable for the PCS - programmable controller connection via the PU interface.
3. This manual with AEG master floppy disk (PCSKOMM project for A120), (AEG250DR project for A250).
4. Floppy disk and PCSPRO/PCS9092 manual with AEG120DR driver.

Furthermore are necessary (AEG company):
5. An A120/A250 programmable controller with rack, power supply and I/O boards.
6. AEG Modicon Dolog AKF programming software with programming cable.

.... as well as power supplies for all components.

B1.1 Delimitation

The successful parameterization of the PCS as described in the PCSPRO and/or PCS9092 manual is assumed. This appendix relates exclusively to the use of a PCS together with an AEG A120/A250 controller. This controller is defined in the following as programmable controller. The AEG-specific terms and the programming of the programmable controller with the Dolog AKF software are assumed as known. The used communication protocol is defined as BKOS protocol. The enclosed handling software was created for an ALU 202 (A120) and ALU 151 (A250).

B1.2 Loading the AEG 120 DR driver into the PCS

During configuration of the PCS, the application program with data and a selected driver are both transferred. The driver is called „AEG120DR.DRV" (AEGBKOPS.DRV for PCS900). The procedure is the same as described in the PCSPRO/PCS9092 manual.

The following variables for the AEG120DR driver can be set:

- The AA variable and/or COM_TIMEOUT: PCS time-out time
  The time-out time determines the maximum admissible time for the processing of the synchronization word in the programmable controller. As default, this time is 400 for AA. This corresponds to 4000 ms = 4 seconds. Admissible values for the AA variable are from 200 to 999 (= 2 to 9.9 sec.). The AA time-out time is only activated with the „SYNC AND TIME-OUT" setting.
- The AC, AD, AE, AF variables and/or COM_MODE: Synchronization
  DIL switches 5 and 6 and/or the rotary switch on the rearside of the PCS can be set to 2 values:
  - „NO SYNC, NO TIME-OUT „; 2)" SYNC AND TIME-OUT „.

  With „SYNC", the usage of a synchronization word is mandatory. „NO SYNC" uses no synchronization word. „TIME-OUT" requires the usage of a time-out timer (with respect to the time specified in the AA variable). „NO TIME-OUT" prevents a time monitoring of the communication in the PCS.
- The AL variable and/or COM_DAT_STR: START ADDRESS WORD.
  Determines the first used flag word in the programmable controller. Thereby, the variable has an offset function. The possible values for flag words are 1 to 1956. Default setting is 100. Therefore, the flag word is 100. Thus, the PCS word 0 is assigned to flag word 100.
- The AT variable and/or CPM_DAT_END: END ADDRESS WORD.
  Determines the last used flag word in the programmable controller. The possible values are from 30 to 1985. Default value is 355. This variable influences the word area to be used in PCSPRO.
  Thus, the used area is AM-AL+1 = x words in size. E.g. for AM = 355 and AL = 100 the flag area is 256 words.
  - The AJ variable and/or COM_MAXLEN: Maximum number of packages per communication cycle.

  In a communication cycle, the PCS forms sub-packages for each task. The number of these sub-packages can be limited. This results in reducing the time for a communication cycle. This can be advantageous for fast jog operations (as far as possible without variable display). On the other hand, the refreshing of variables thereby takes longer. Limit this variable only if all other measures have not resulted in sufficient time savings. The possible values are between 1 and 50. Default value is 50.

B1.3 PCS connection to the programmable controller

1. Set the DIL-switches 8 and 9 at the rearside of the 090/095/900 PCS to „OFF". Nothing needs to be changed for the 9000 PCS.
2. Load the corresponding software into the programmable controller and set DIL switches 5 and 6 or the PCS rotary switch correspondingly.
3. Switch off the PCS and then on again (power reset). At least, the ERR LED must light now.
4. Connect the programming interface of the programmable controller to the PCS using the PCS 778 cable. If you have activated the „SYNC" operation, then you should set the programmable controller restart input to „1".
5. Now, the ERR LED is deactivated at the PCS.

B1.4 Trouble-shooting

Here, the most frequent faults are listed which occur during the first commissioning and permanent operation:

1 DIL switch no. 8 is set to ON or the rotary switch is to a position >3. If this is the case, then the PCS enters a diagnosis routine after powering up. This is required only for test purposes. Remedy: set DIL switch 8 to OFF or the rotary switch to 0 and restart the PCS (by cycling power or pressing the momentary reset push-button above the DIL switch).

2 A time-out is reported in the programmable controller. In this case, the fault is to be searched at the programmable controller - PCS connection. Maybe, the cable is defective or plugged in incorrectly. Maybe, a wrong driver is also loaded into the PCS. An incorrectly defined flag word area is also possible.

3 The communication is active (PCS ERR LED is deactivated) but after a certain time the following message appears on the upper display line of the PCS:

COMMUNICATION ERROR

An explanatory help text is shown on the lower display line:

TIMEOUT COMMUNICATION:

Transfer fault in the BKOS protocol. In this case, the PCS - programmable controller connection is routed in a noisy environment, the cable length is too long or the grounding is insufficient. Also, this message can result from a too small or incorrectly defined flag word area in the programmable controller.

TIMEOUT PLC:

The determined time-out time (AA driver variable) was exceeded. This can occur only in the „SYNC" operating mode and means that the synchronization word (data word 3) was not processed in the programmable controller in time. Has the handling software been loaded into the programmable controller? Is the data area selected correctly? Is the restart input on „1" level?

A helpful diagnosis for the 090/095/900 PCS is the output of the PCS status on the display. This display can be accessed in PCS versions later than 201.6 by the following combination: press the Help key after a PCS start (e.g. after a reset). The ERR LED must light constantly. This option is available no more after starting the communication (ERR LED is deactivated or flashes). By pressing the Help plus arrow keys you can display the PCS version, the data block version, and the driver version as well as the selected driver variables.

References for the connection of the PCS to a programmable controller:

❒ Connect the cable screening to the central earthing point of the switching cabinet!

❒ Ensure appropriate chassis groundings with regard to the PCS housing on the one hand and the programmable controller 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 interference 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. Use a separate power supply for the PCS (24 V, 10 VA if the supply voltage is influenced by high interference. 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 equi-potential 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!

The used AEG120DR driver is a direct driver, i.e. it replaces directly the data words between the programmable controller and the PCS. Either a small or no programmable controller program is required for that resulting only in a small scan time burden for the programmable controller. Using the BKOS protocol, the programmable controller and the PCS communicate via RS-232C with 9600 bauds, 8 bits, ODD parity and 1 stop bit. It is possible to run a synchronous or asynchronous communication operation since only the exact data words which are absolutely necessary are replaced.

SYNC or NOSYNC?

Since the data interchange occurs asynchronously by writing and reading via several programmable controller cycles, the data written by the programmable controller can be overwritten by the PCS and conversely. Thus, there is no data consistency. Possible solutions are either a strict separation of read and write data words (→ NO SYNC) or the usage of a synchronization word (→ SYNC). The usage of a synchronization word enables the application of a time-out timer on the PCS side.

DETERMINE THE DATA AREA

A common data area must be defined to enable access of the programmable controller and the PCS. This must be physically available and defined in the programmable controller.
In the PCS, this area is defined via the AL and AM driver variables. If this area in the programmable controller is selected as too small then a time-out can appear during the operation!

EFFECTIVE RESPONSE TIMES: PCS - PROGRAMMABLE CONTROLLER

The response time of the protocol depends extensively on the tasks executed in the PCS. If variables are displayed or even processed, then the communication cycle time is essentially higher than with text without variables. Also, the transfer of the message bit area and LED words influences this time. Limit these transfers.
The number of sub-packages in the AJ driver variables can be limited to enable a frequent transfer of the keys. The disadvantage thereby is, that the reading and writing of variables require more time.
Summing up, the key → LED time can be between 0.5 ms and 3 seconds.

B2.1 Asynchronous Communication

- Actual and preset value data words must be strictly separated (writing accesses can interfere). Even then, e.g. the reading of a variable which extends across several data words can result in a fault and of course then, if the variable is read although only a part of the variable has been written.
- Bit variables should be used only once per word since the access of the PCS occurs only word-by-word. A word that has been retrieved, changed, and written back by the PCS can overwrite another bit variable using this word! This is also valid for the message bit area with erase behavior 2 (deleting of the bit in the programmable controller).
- Time-out monitoring is possible only in the programmable controller. Therefore, the PCS sends one incremented count word in word 3 in each communication cycle. The usage of a time-out timer is described in the handling software.

The advantages compared to the SYNC operation are:

- Faster data interchange. The programmable controller scan time is not taken into account when calculating the response.
- There is no need for a communication program in the programmable controller. Only the flag word area must be available having the correct size. Access to the flag word area is possible at any time in the programmable controller program.

B2.2 Synchronous communication

If you want to use the entire functional extent of the PCS, then the access to the data in the programmable controller must be synchronized, i.e. programmable controller and PCS access alternately. In addition, a synchronization word is transferred to the programmable controller. PB2 examines this word and enables the programmable controller user program access. If the user program is finished with the processing of the data words, then the synchronization word is changed and the PCS accesses the data area. While the PCS processes the data, the user program may not access to the data. This Ping-Pong game offers the possibility to realize a time-out monitoring also in the PCS. Whenever the PCS reads the inverted synchronization word, the timer is restarted. If the timer expires then a time-out exists.

Using this alternating access, actual and preset values can be mixed, bit variables used, erase behavior 2 realized, etc. Therefore, the entire intelligence of the PCS is available. The disadvantage for you is that the reaction speed between PCS and programmable controller is lowered. Furthermore, before accessing the data, the programmable controller program must always scan whether access is allowed or not.

The time-out time, i.e. the time that passes since the last writing of DW3 up to the time-out message in the programmable controller, should be set to a minimum of 2 seconds. In the PCS, the time-out time is set via the „AA" driver variable or COM_TIMEOUT.

B2.2.1 Procedure

To realize synchronous communication between the PCS and programmable controller you must:

- select the „SYNC" setting on the PCS,
- load the handling software into the programmable controller.

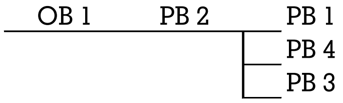
This handling software is described in the following. Of course you can also solve these tasks differently in your software. It is only important that you keep to the following procedure:

1. The used data area assigned to the PCS and programmable controller must be determined in the programmable controller (e.g. 355 flag words) and communicated to the PCS via the AL and AM driver variables. The programmable controller flag word area should be initialized.
2. Thereupon, the PCS sends an order number (byte-by-byte, every time incremented by one) in PCS word 3, e.g.: „01". This is the signal for the programmable controller, that the data area can be processed. Furthermore, a time-out timer can be restarted (if this is used).
3. If the processing of the data area is finished, then the inverted sync word of PCS word 3 is copied to the PCS word 2, e.g. with „FE". This is the signal for the PCS to access to the data area. Nothing more may be changed now in the data area by the programmable controller program!

From now on, step 2 and 3 will be cyclically executed. If a time-out appears, then processing is resumed with step 2 with order number „01" again.

B2.2.2 Description of the handling software

The existing handling software is required only for synchronous operation. It can however, also be used unchanged for asynchronous operation (caution, a loss of the programmable controller communication is not noticed in the PCS!). The data word area is envisaged for flag word 100..355. If you want to displace the flag word area, then you must rewrite the handling software completely.
Overview:



OB1:
Selects PB2 (PCS communication).

PB2:
Manages the communication processing. PB1 (initialization) is selected in the 1st. cycle. If the communication fails, then PB3 is selected once. After a communication loss the synchronizing word is examined only, if a restart input 0 → 1 transition exists. Then, PB3 (data area processing) is selected if it is ascertained in the synchronizing word that the PCS has sent an order (MW 102 <> MW103). After that, the sync word processes and the time-out timer is restarted.

PB1:
Initializes the PCS command words and the time-out timer. Attach your initializations here.

PB3:
The real data access occurs here. Evaluate here the keys, set menus, LEDs, etc..

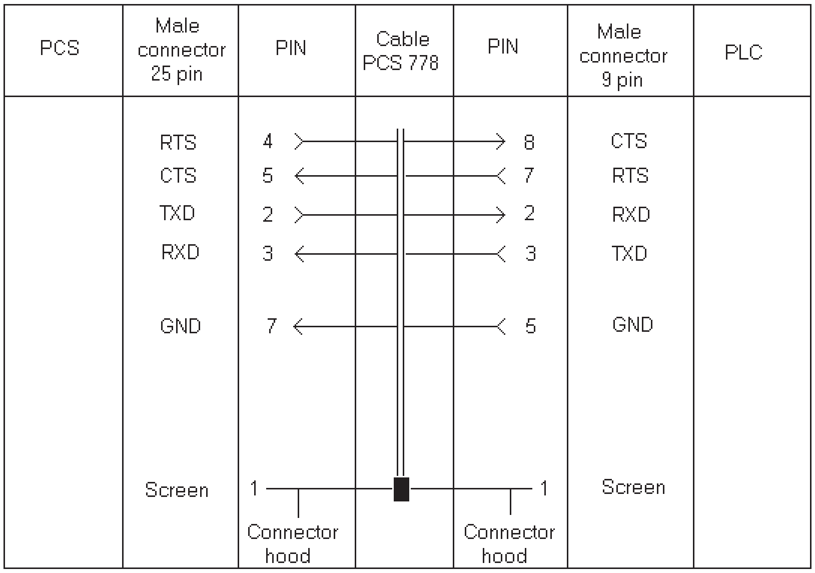
PB4:
Resets the keys and synchronous words at communication loss. Attach your measures here.

Used programmable controller areas:

```
MW 100       FLAG WORD AREA PCS-WORD     0..255
...
MW 355

E  2.1      RESTART INPUT
A3.1        FAULT OUTPUT
M  1.1      TIMER TIME-OUT PROCESSING
M  1.2      RESET TIMER TIME-OUT
M  1.3      EDGE COMMUNICATION LOSS
M  1.4      EDGE RESTART
M  1.5        "
M  1.6      ACCESS AFTER COMM. LOSS
SM 1        AWP is active
SM 2        ON flag
TSW1        TIME-OUT TIMER PRESET VALUE
T 1         TIME-OUT TIMER
PB1         INITIALIZATION
PB2         DATA ACCESS MANAGEMENT
PB3         DATA ACCESS
PB4         COMMUNICATION LOSS
```

B3.1 Adapter cable PCS 778

The connection is made via the RS232C - programmable controller interface.



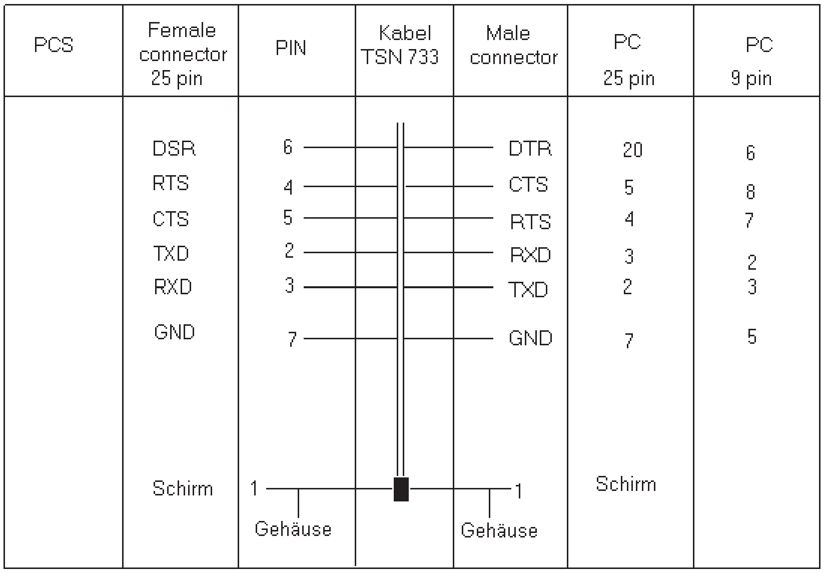
PCS	Male connector 25 pin	PIN	Cable PCS 778	PIN	Male connector 9 pin	PLC
	RTS	4 →		→ 8	CTS	
	CTS	5 ←		← 7	RTS	
	TXD	2 →		→ 2	RXD	
	RXD	3 ←		← 3	TXD	
	GND	7 ←		← 5	GND	
	Screen	1 Connector hood		1 Connector hood	Screen	

B3.1.1 Screening

The screen should be connected on both sides to a metal coated connector case. The screen can also be connected to pin 1 when using non-metal coated connector cases. However, this is not recommended for error-technical reasons. There, the data lines should be covered as completely as possible by the screen! With grounding at both sides is to be noted however that possibly (because of ground potential shifts) a potential compensation wire is required having at least 10 times the screen cross section (reason: compensation currents should not flow through the cable screen if possible!). This applies particularly if PCS and programmable controller are not connected to the same earthing point. This is for example valid where the PCS and programmable controller are not housed in one switching cabinet!

B3.2Programming cable PCS 733

PC -PCS connection



PCS	Female connector 25 pin	PIN	Kabel TSN 733	Male connector	PC 25 pin	PC 9 pin
	DSR	6		DTR	20	6
	RTS	4		CTS	5	8
	CTS	5		RTS	4	7
	TXD	2		RXD	3	2
	RXD	3		TXD	2	3
	GND	7		GND	7	5
	Schirm	1		1	Schirm	
		Gehäuse		Gehäuse		

```
PB1;
/#Initialization#/;
L  KH 0/* ZERO KEYS */;
=  MW 104   /* PCS 090 = MW104,105 */;
=  MW 105   /* PCS 095 = MW104,105,123 */;
=  MW 106   /* PCS 900 = MW104,105,106 */;
=  MW 107   /* PCS9000 = MW104,105,106,107 */;
=  MW 123;
NOP         /* insert enables start. here */;
NOP         /* PCS 090, 095 */;
NOP         /* KH0FC8 after MW 113 */;
NOP         /* KH0080 after MW 114 */;
NOP         /* PCS 900 */;
NOP         /* KH1F00 after MW 136 */;
NOP         /* KH00FF after MW 137 */;
NOP         /* KH0080 after MW 138 */;
NOP         /* PCS 9000 */;
NOP         /* KH0000 after MW 114 */;
NOP         /* KH00FF after MW 115 */;
NOP;
=  MW 102   /* zero sync. words */;
=  MW 103;
UN SM 1           /* initialize time-out timer */;
=  M 1.1;
=  M 1.2;
L  K 4;
=  TSW 1;
U  M 1.1;
SE T 1;
DZB   1000MS;
L  TSW 1;
U  M 1.2;
R  T 1;
=  A 2.1;
*** ;
BE;
PB2;
   /#Synchronization block#/;
   U  SM 119                  /*     1st cycle*/;
   BAB      PB 1              /* initialization */;
   U  A 2.1            /* communication loss?*/;
   FLP      M 1.3;
   BAB      PB 4       /* COFF program */;
   U  E 2.2       /* edge for restart */;
   FLP      M 1.4;
   =  M 1.5;
   O  M 1.6;
   O  E 2.2;
   O  M 1.5;
   =  M 1.6;
   UN M 1.6       /* restart after fault?*/;
   U  A 2.1;
   SPB      =END       /* no end */;
   L  MW 103           /* access to data field OK?*/;
   == MW 102;
   SPB =END       /* no: end */;
   BA PB 3        /* yes Pb3: processing MW */;
   UN SM 1;
   =  M 1.5;
   =  M 1.6;
   L  MW 103           /* process sync. word */;
   X  KH FFFF;
   =  MW 103;
   =  MW 102;
   UN SM 1        /* timer reset */;
   =  M 1.1;
   SP =END2;
END:  O     SM1   /* start Timer */;
   =  M 1.1;
END2: U     M 1.1       /* timer processing */;
   SE T 1;
   DZB     1000MS;
   L  TSW 1;
   U  M 1.2;
   R  T 1;
   =  A 2.1       /* fault output */;
   *** ;
   BE;
PB3;
   /#User program (example)#/;
   NOP                  /* example for one */;
   NOP                  /* PCS 090 */;
   NOP;
   L  MW 104;
   <> KH 8000;
   SPB     =TEST1;
   L  KH 81;
   =  MW 114;
   L  KH FC8;
   =  MW 113;
TEST1:      L     MW 104;
   <> KH 4000;
   SPB     =TEST2;
   L  KH 0;
   =  MW 114;
TEST2:      L     MW 104;
   <> KH 2000;
   SPB     =TEST3;
   L  KH FFFF;
   =  MW 115;
TEST3:      L     MW 104;
   <> KH 1000;
   SPB     =TEST4;
   L  KH 0;
   =  MW 115;
TEST4:      L     MW 104;
   U  KH FF00;
   =  MW 110;
   *** ;
BE;
PB4;
   /#COFF program #/;
   L KH 0;
   =  MW 102      /* zero sync. words */;
   =  MW 103;
   =  MW 104      /* ZERO KEYS */;
   =  MW 105      /* depending on PCS */;
   =  MW 106;
   =  MW 107;
   =  MW 123;
NOP         /* further instructions here ..*/;
*** ;
BE;
```

Description of the loadable MODBUSPS driver for the PCS operating console. For the communication with an AEG MODICON 984 PLC.
Functionality: Selective MODBUS 1 direct driver for AEG MODICON PLC.

Required devices and accessories:
The following products are required for the operation of a programmable controller with an already parameterized PCS (Systeme Lauer company):

1. The PCS operating console itself (already parameterized).
2. The PCS 780 connecting cable for the PCS - programmable controller connection via the MODBUS interface.
3. This (PCS 91.AEG) manual with AEG master floppy disk (MODBUSPS\PCSKOMM project),
4. Floppy disk and PCSPRO and/or PCS9092 manual with MODBUSPS driver.

Furthermore are necessary (AEG company):
5. A MODICON 984 programmable controller with power supply, rack and I/O boards.
6. AEG LMODSOFT programming software with programming cable.

.... as well as power supplies for all components.

C First commissioning

C.1.1 Delimitation

The successful parameterization of the PCS, as described in the PCSPRO and/or PCS9092 manual, is assumed. This appendix relates exclusively to the use of a PCS together with a Modicon controller of the AEG company. This controller is defined in the following as programmable controller. The AEG-specific terms and the programming of the programmable controller with the LMODSOFT software are assumed as known. The used communication protocol is defined as MODBUS 1 protocol. The enclosed handling software was created on a Modicon 984-131 CPU with 2 Modbus interfaces.

This driver is capable of running with all CPU types that feature a MODBUS 1 slave interface. Up to now the following 984 CPU board series are known:

-120, -130, -145, -380, -381, -385, -480, -485, -680, -685, -780, -785, - X, - A and B.

Warning!
Use only the PCSPRO and/or PCS9092 software for the configuration. Other software packages can initiate malfunctions in the PCS and programmable controller.

C.1.2 Loading of the MODBUSPS driver to the PCS

During configuration of the PCS, the application program with data and a selected driver are both transferred. The driver is called „MODBUSPS.DRV „. The procedure is the same as described in the PCSPRO/PCS9092 manual.

The following variables for the MODBUSPS driver can be set:

- The AA variable and/or COM_TIMEOUT: PCS time-out time.
  The Time-out time determines the maximum admissible time for the processing of the synchronization word in the programmable controller. As default, this time is 300 for AA. This corresponds to 3000 ms = 3 seconds. Admissible values for the AA variable are from 200 to 999 (= 2 to 9.9 sec.). The AA time-out time is only activated with the „SYNC AND TIME-OUT” setting.
- The FROM variable and/or COM_ERR_REPT: Maximum number of repetitions
  As MODBUS master, the PCS repeats the order if a fault appears in the communication. After x faults following another (set by the AB variable) a loss of the communication is displayed on the PCS and the communication is restarted. Values are possible between 1 and 50, default value is 5.
- The AC, AD, AE or AF variable: Baud rate and transfer type DIL switches 5 and 6 or the rotary switch on the rearside of the PCS can be used with 4 settings.

Displayed is the default setting:

Rotary switch	Dil5	Dil6	Variable	Default setting
0	off	off	AC/COM_MODE0	19200 RS-232 + SYNC
1	on	off	AD/COM_MODE1	19200 RS232 +NOSYNC
2	off	on	AE/COM_MODE2	9600 RS-232 +SYNC
3	on	on	AF/COM_MODE3	9600 RS232 +NOSYNC

The first number defines the used MODBUS baud rate. The LMODSOFT programming default setting is 9600 baud, use 19200 baud (if possible). The communication is thereby almost twice as fast.
For other MODBUS linkages, the interface can be selected also as TTY or RS-422. These two are not dealt with here.
With „SYNC", the usage of a synchronization word is mandatory. „NO SYNC" uses no synchronization word. The „SYNC" setting implies also the activation of a time-out timer in the PCS with the values specified by the „AA" variable.

- The AL variable and/or COM_DATA_STR: START ADDRESS WORD.
  Determines the first used register word in the programmable controller. Thereby, the variable has an offset function. The possible values for register words are 1 to 32737. Default setting is register word 400. Thus, the PCS word 0 is assigned to flag word 400.
- The AM variable and/or COM_DATA_END: END ADDRESS WORD.
  Determines the last used register word in the programmable controller. Possible values are 30 to 32767. Default value is 655. This variable influences the word area to be used in PCSPRO.
  Thus, the used area is AM-AL+1 = x words in size. E.g. for AM = 655 and AL = 400 the flag area is 256 words.
- The AO variable and/or COM_SL_NUM: Address of the accessed programmable controller.
  Although the RS-232C interface is a point-to-point connection, a programmable controller address between 1 and 32 can be entered in the Modbus 1 protocol. Default value is 1.
- The AS variable and/or COM_PAR+STP: Parity and stop bits of the interface
  The default setting of the interface is even parity with 1 stop bit. In case you use an other setting, then you can select between the following settings: even/odd/non-parity, each with 1 or 2 stop bits.

C.1.3 Connection of the PCS to the programmable controller

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

1. Set the DIL-switches 8 and 9 at the rearside of the 090/095/900 PCS to „OFF". Nothing needs to be changed for the PCS 9000.
2. Load the corresponding software into the programmable controller and set DIL switches 5 and 6 or the PCS rotary switch correspondingly.
3. Switch off the PCS and then on again (power reset). At least, the ERR LED must light now.
4. Connect the programming interface of the programmable controller to the PCS using the PCS 780 cable. If you have activated the „SYNC" operation, then you should set the programmable controller restart input to „1".
5. Now, the ERR LED is deactivated at the PCS.

C.1.4 Trouble-shooting

Here, the most frequent faults are listed which occur during the first commissioning and permanent operation:

- DIL switch no. 8 is set to ON or the rotary switch is to a position >3. If this is the case, then the PCS enters a diagnosis routine after powering up. This is only required for test purposes. Remedy: set DIL switch 8 to OFF or the rotary switch to 0 and restart the PCS (by cycling power or pressing the momentary reset push-button above the DIL switch).
- A time-out is reported in the programmable controller. In this case, the fault is to be searched at the programmable controller - PCS connection. Maybe, the cable is defective or plugged in incorrectly. Maybe, a wrong driver is also loaded in the PCS. An incorrectly defined flag word area is also possible.

Warning!
Check the action/reaction of the programmable controller!
After restarting the programmable controller following a communication loss, the desired programmable controller reaction/action is to be examined to avoid malfunctions.

- The communication is active (PCS ERR LED is deactivated) but after a certain time the following message appears on the upper display line of the PCS:
  COMMUNICATION ERROR
  An explanatory help text is shown on the lower display line:

  TIMEOUT COMMUNICATION:
  Transfer fault in the MODBUS 1 protocol. In this case, the PCS - programmable controller connection is routed in a noisy environment, the cable length is too long or the grounding is insufficient. Also, this message can result from a too small or incorrectly defined flag word area in the programmable controller.
  TIMEOUT PLC:
  The determined time-out time (AA driver variable) was exceeded. This can occur only in the „SYNC" operating mode and means that the synchronization word (data word 3) was not processed in the programmable controller in time. Has the handling software been loaded into the programmable controller? Is the data area selected correctly? Is the restart input on „1" level?

A helpful diagnosis for the PCS 090/095/900 is the output of the PCS status on the display. This display can be accessed by the following combination: press the Help key after a PCS start (e.g. after a reset). The ERR LED must light constantly. This option is available no more after starting the communication (ERR LED is deactivated or flashes). By pressing the Help plus arrow keys you can display the PCS version, the data block version, and the driver version as well as the selected driver variables.

References for the connection of the PCS to a programmable controller:

❒ Connect the cable screening to the central earthing point of the switching cabinet!

❒ Ensure appropriate chassis groundings with regard to the PCS housing on the one hand and the programmable controller 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 interference 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.

❒ Use a separate power supply for the PCS if the supply voltage is influenced by high interference. 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 equi-potential bonding conductor with a cross section of 10 times that of the 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)! This is necessary to prevent equalizing currents on the cable shield!

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

The used MODBUSPS driver is a direct driver, i.e. it replaces directly the data words between programmable controller and PCS. Either a small or no programmable controller program is required for that resulting only in a small scan time burden for the programmable controller. Using the MODBUS 1 protocol, the programmable controller and the PCS communicate via RS-232C with 19200 or 9600 bauds and 8 data bits. It is possible to run a synchronous or asynchronous communication operation since only the exact data words which are absolutely necessary are replaced. During communication, the PCS is the master and the programmable controller the slave. Thus, only one point-to-point connection is possible for the interconnection.

SYNC or NOSYNC?

Since the data interchange occurs asynchronously by writing and reading via several programmable controller cycles, the data written by the programmable controller can be overwritten by the PCS and conversely. Thus, there is no data consistency. Possible solutions are either a strict separation of read and write data words (→ NO SYNC) or the usage of a synchronization word (→ SYNC). The usage of a synchronization word enables the application of a time-out timer on the PCS side.
At a CPU „Stop“, the communication is not interrupted. This CPU state can only be detected in the sync. operation via the time-out timer.

DETERMINE THE DATA AREA

A common data area must be defined to enable access of the programmable controller and the PCS. This area must be physically available and defined in the programmable controller.
In the PCS, this area is defined via the AL and AM driver variables. If this area is incorrectly selected in the programmable controller then a time-out can appear during the operation!

EFFECTIVE RESPONSE TIMES: PCS - PROGRAMMABLE CONTROLLER

The response time of the protocol depends extensively on the tasks executed in the PCS. If variables are displayed or even processed, then the communication cycle time is essentially higher than with text without variables. Also, the transfer of the message bit area and LED words influences this time. Limit these transfers. You can do this also in the running operation, e.g. to realize a key-driven jog operation.
The following table presents a reference for the speed of the communication. The so-called „key → LED time“ is measured. That means the time in which a key is transferred to the programmable controller and an LED is set in the PCS. This time is made up of 2 communication cycles and a programmable controller scan time. The response time, i.e. the time a key is signaled in the programmable controller is only half the indicated time!

The measurements were made with a PCS 095 in synchronous operation while the following data are displayed:

8 consecutive double word variables
8 non-consecutive DW variables.
1 word variable
no variable

Key → LED [seconds]
(the programmable controller scan time must be counted)

Baud rate	Port	8 DW flush	8 DW non flush	1 word	0 word
19200	1	0.26	0.42	0.24	0.21
19200	2	0.50	0.80	0.35	0.28
9600	1	0.46	0.72	0.40	0.37
9600	2	1.05	0.67	0.60	0.53

C.2.1 Asynchronous communication

- Actual and preset value data words must be strictly separated (writing accesses can interfere). Even then, e.g. the reading of a variable which extends across several data words can result in a fault and of course then, if the variable is read although only a part of the variable has been written.

- Bit variables should be used only once per word since the access of the PCS occurs only word-by-word. A word that has been retrieved, changed, and written back by the PCS can overwrite another bit variable using this word! This is also valid for the message bit area with erase behavior 2 (deleting of the bit in the programmable controller).

- Time-out monitoring is possible only in the programmable controller. Therefore, the PCS sends one incremented count word in word 3 in each communication cycle. The usage of a time-out timer is described in the handling software.
The advantages compared to the SYNC operation are:

- Faster data interchange. The programmable controller scan time is not taken into account when calculating the response.

- There is no need for a communication program in the programmable controller. Only the flag word area must be available having the correct size. Access to the flag word area is possible at any time in the programmable controller program.

C.2.2 Synchromous communication

If you want to use the entire functional extent of the PCS, then the access to the data in the programmable controller must be synchronized, i.e. programmable controller and PCS access alternately. Therefore, a synchronization word is transferred to the programmable controller. The handling software examines this word and enables the programmable controller user program access. If the user program is finished with the processing of the data words, then the synchronization word is changed and the PCS accesses the data area. While the PCS processes the data, the user program may not access to the data. This Ping-Pong game offers the possibility to realize a time-out monitoring also in the PCS. Whenever the PCS reads the inverted synchronization word, the timer is restarted. If the timer expires then a time-out exists.

Using this alternating access, actual and preset values can be mixed, bit variables used, erase behavior 2 realized, etc. Therefore, the whole intelligence of the PCS is available. The disadvantage for you is that the reaction speed between PCS and programmable controller is lowered. Furthermore, before accessing the data the programmable controller program must always scan whether access is allowed or not.

The time-out time, i.e. the time that passes since the last writing of word 3 up to the time-out message in the programmable controller, should be set to a minimum of 2 seconds. In the PCS, the time-out time is set via the „AA" driver variable or COM_TIMEOUT.

C2.2.1 Procedure

To realize synchronous communication between the PCS and programmable controller you must:

- select the „SYNC" setting on the PCS,
- load the handling software into the programmable controller.

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

1. The used data area assigned to the PCS and programmable controller must be determined in the programmable controller (e.g. 255 flag words) and communicated to the PCS via the AL and AM driver variables. The programmable controller flag word area should be initialized.
2. Thereupon, the PCS sends an order number (byte-by-byte, every time incremented by one) in PCS word 3, e.g.: „01". This is the signal for the programmable controller, that the data area can be processed. Furthermore, a time-out timer can be restarted (if this is used).
3. If the processing of the data area is finished, then the inverted sync word of PCS word 3 is copied to PCS word 2, e.g. with „FE". This is the signal for the PCS to access the data area. Nothing more may be changed now in the data area by the programmable controller program!
   From now on, step 2 and 3 will be cyclically executed. If a time-out appears, then processing is resumed with step 2 with order number „01" again.

C2.3 Description of the handling software

Warning!
Check the functioning of the handling software to avoid PCS and/or programmable controller malfunctions.

The existing handling software is required only for synchronous operation. It can however, also be used unchanged for asynchronous operation (caution, a loss of the programmable controller processing is not noticed in the PCS!). The data word area is envisaged for flag word 400..655. If you want to displace the flag word area, then you must rewrite the handling software completely. The presettings and accesses were written for a PCS 090 / 095. For the PCS 900/ 9000, the presettings and the accesses must be adjusted to the data word area.

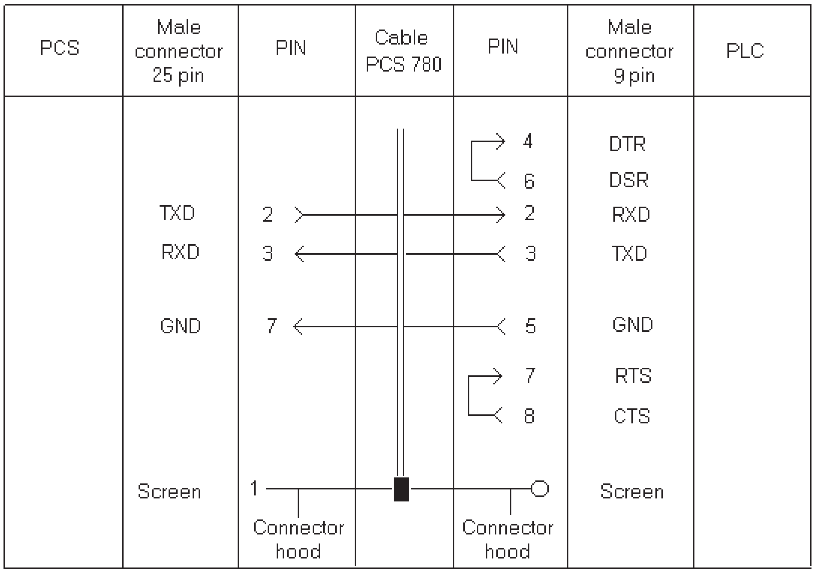
Network 1:
When starting the programmable controller, the 41750 table area > 40400 data area is loaded (with a length of 25 words). You can define the presettings for the PCS in this table area. This depends on the PCS type!

Network 2:
The communication time-out timer is located here. With a positive fault edge at output 1 it initiates the transfer of table area 41800 → data area 40400. That means particular values for the communication loss are to be entered in the table area 41800 (length 25 words). Please also take note of the PCS-type dependent values here.
Furthermore, here is examined whether an access to the data area is allowed. This is true when data word 40402 differs from 40403. Then, the flag 101 (ACCESS) is set.

Network 3+4:
An example program is executed here if access is allowed (ACCESS = 1): the F1 key sets menu 1, the F2 key deletes menus, the F3 key sets all messages in DW 15, and the F4 key deletes these again. Key word 4 is copied to LED word 10.
As last(!) access action, the sync. word 40403 is inverted and copied to 40402.

C3.1 PCS 780 adapter cable

The connection is made via a programmable controller MODBUS 1 interface (RS-232).



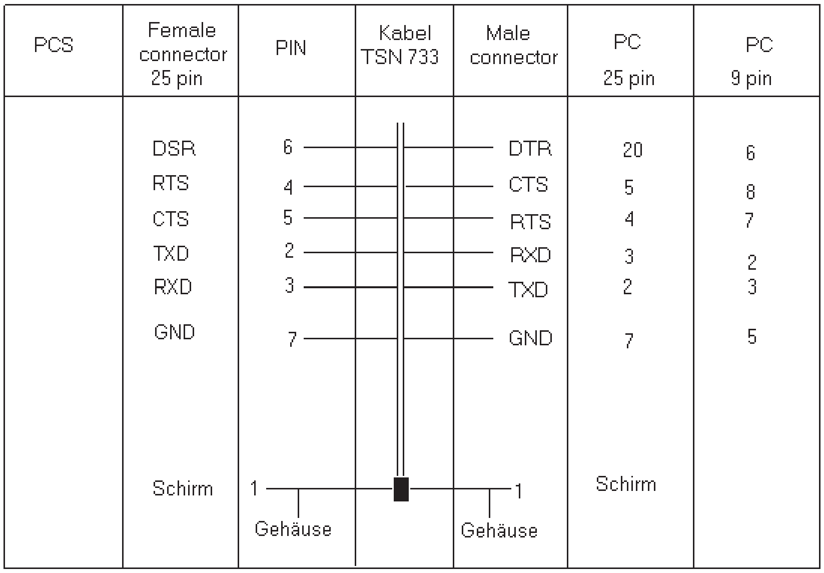
PCS	Male connector 25 pin	PIN	Cable PCS 780	PIN	Male connector 9 pin	PLC
				4	DTR	
				6	DSR	
	TXD	2		2	RXD	
	RXD	3		3	TXD	
	GND	7		5	GND	
				7	RTS	
				8	CTS	
	Screen	1 Connector hood		Connector hood	Screen	

C3.2 Screening

The screen should be connected on both sides to a metal coated connector case. The screen can also be connected to pin 1 when using non-metal coated connector cases. However, this is not recommended for error-technical reasons. There, the data lines should be covered as completely as possible by the screen! With grounding at both sides is to be noted however that possibly (because of ground potential shifts) a potential compensation wire is required having at least 10 times the screen cross section (reason: compensation currents should not flow through the cable screen if possible!). This applies particularly if PCS and programmable controller are not connected to the same earthing point. This is for example valid where the PCS and programmable controller are not housed in one switching cabinet!

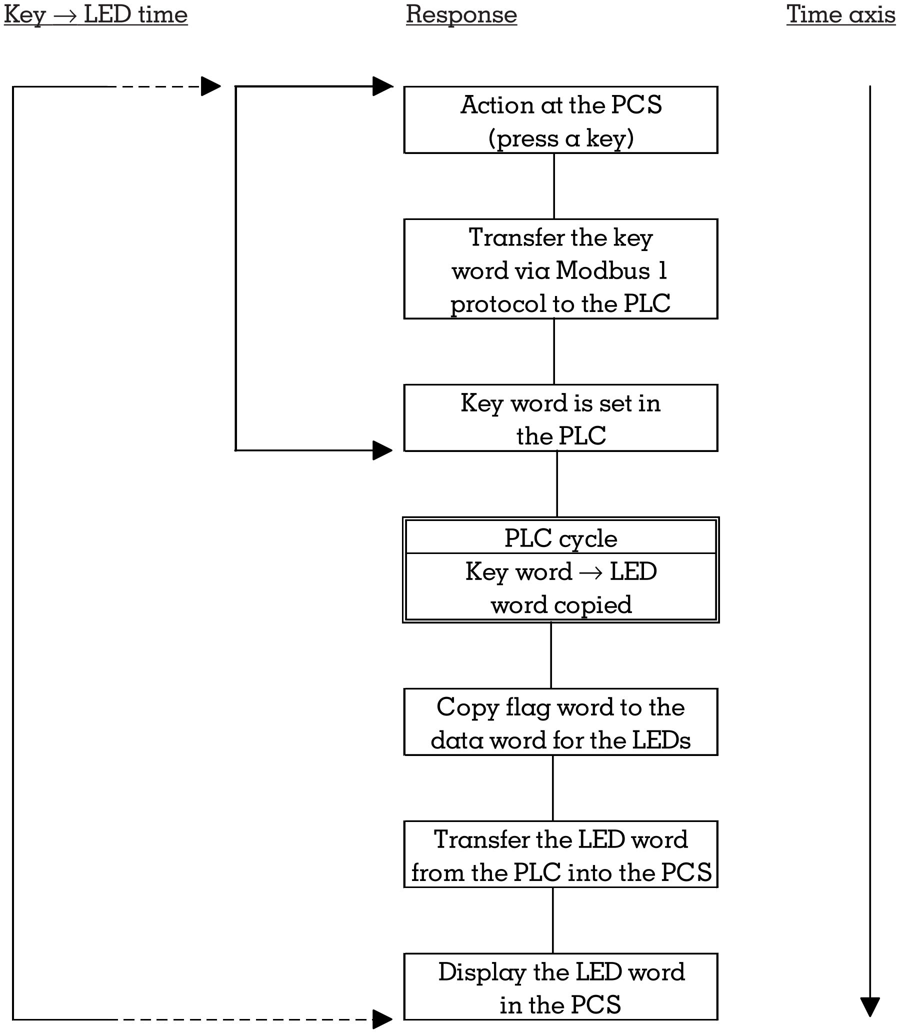
C3.2 Programming cable PCS 733

PC - PCS Connection



PCS	Female connector 25 pin	PIN	Kabel TSN 733	Male connector	PC 25 pin	PC 9 pin
	DSR	6		DTR	20	6
	RTS	4		CTS	5	8
	CTS	5		RTS	4	7
	TXD	2		RXD	3	2
	RXD	3		TXD	2	3
	GND	7		GND	7	5
	Schirm	1 Gehäuse		1 Gehäuse	Schirm	

C3.3 Data transfer sequences


Key → LED time
Response
Time axis
Action at the PCS (press a key)
Transfer the key word via Modbus 1 protocol to the PLC
Key word is set in the PLC
PLC cycle
Key word → LED word copied
Copy flag word to the data word for the LEDs
Transfer the LED word from the PLC into the PCS
Display the LED word in the PCS

The response time and the key → LED time can be influenced by your programming on the one hand and by the programmable controller scan time on the other hand by the amount of transferred data. Since during a transfer cycle collected data are sent and/or retrieved, a small amount of data is imperatively necessary for small response times. The amount of data dependents on the current number of variables in the display. The fewer variables are shown in the display, the quicker is the reaction time!
For jog operation we recommend to select a menu via the menu command word. This menu should not contain any variables for display. Furthermore, the transfer of message words is to be limited to the actually necessary number.

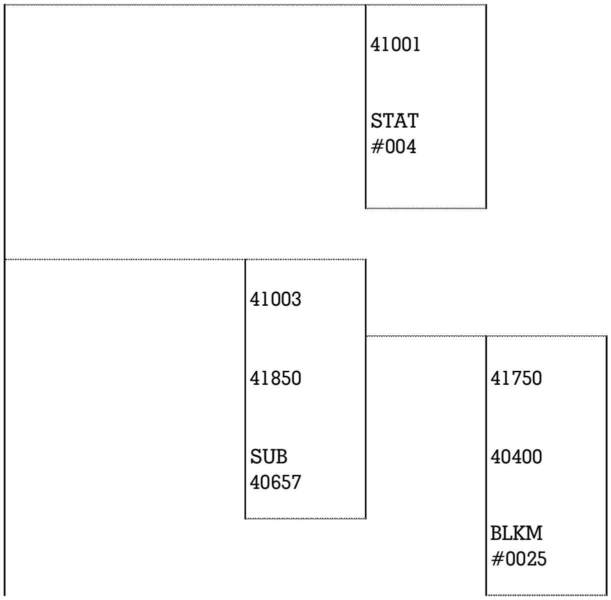
C

Listing

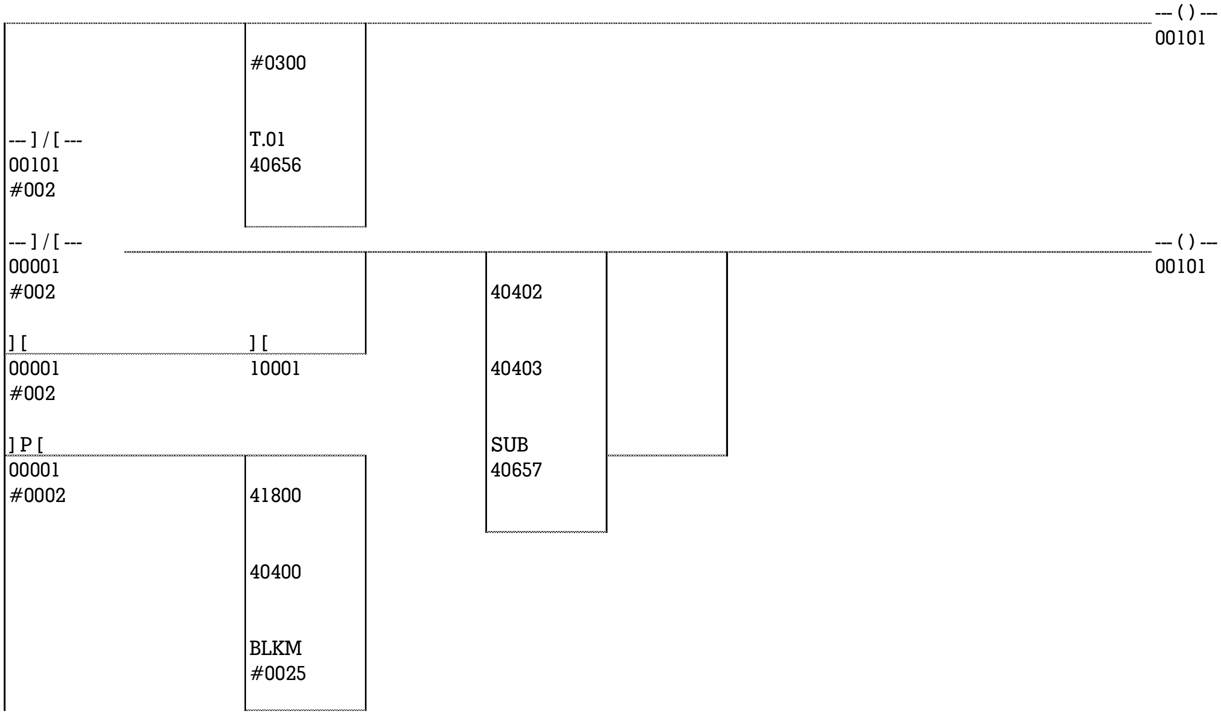
Configuration overview

```
                                  | Size of prgr.memory   02970
     Programmable controller :    | No. words I/O spec.   00153
     Pr. contrl. type 984 COMPACT +—————————————————
     Model                   A13X |  I/O  :
     Memory                  4.0K |  number of segments       2
     Extended memory            K |  I/O drops                1
                                  |  I/O modules              18
                                  +—————————————————
   —————————————+  Special :
                                  |  Battery coil         00081
      Areas  :                    |  Timer register       41862
      0xxxx    00001  - 01536     |  Date/time            41863-41870
      1xxxx    10001  - 10512     |
      3xxxx    30001  - 30048     +—————————————————
      4xxxx    40001  - 41872     |
                                  |
                                  |
                                  |
                                  |
                                  |


                       CONFIGURATION EXPANSION BLOCKS

                        No configuration expansions found


                                      PORTS

Number Modus  Data bits Parity   Stop bits  Baud rate     Key addr.  Delay

MODBUS
01     RTU         8       EVEN       1       19200            1     10 ms
02     RTU         8       EVEN       1       19200            1     10 ms


                     SEGMENT SCHEDULER & CONSTANT  CYCLE
       Number of drops:  1                      Number of segments:      2

   ————————OFF—————————   ———————     —     ——
   CONST. CYCLE:        MIN. CYCLE:       ms     Ref time:          -
   ———————————————————————————————————
    Seq.         Ref.                              Program    Drop       Drop
    number        type       Number   State        segment    read       output
      1         CONTINUOUS                           01         01         01
      2         EOL

                                 USED I/O
                            984-120/130/145 CONTROLLERS

    Progr. contrl.:      COMPACT A13X   Rack           :             1
    No. of inputs :      32             No. of outputs:              16

   ———————————————————————————————————
    Slot     Module      Reference numbers          Data     Module
              type      Input          Output        type     description

     101      984                                          PLC COMPACT
     102      984                                          PLC COMPACT
     103      DEP216 10001 -10016                          Help Alt-H
     104      DEP216 10017 -10032                          Help Alt-H
     105      DAP216                00001 -00016           16-O 24V
```

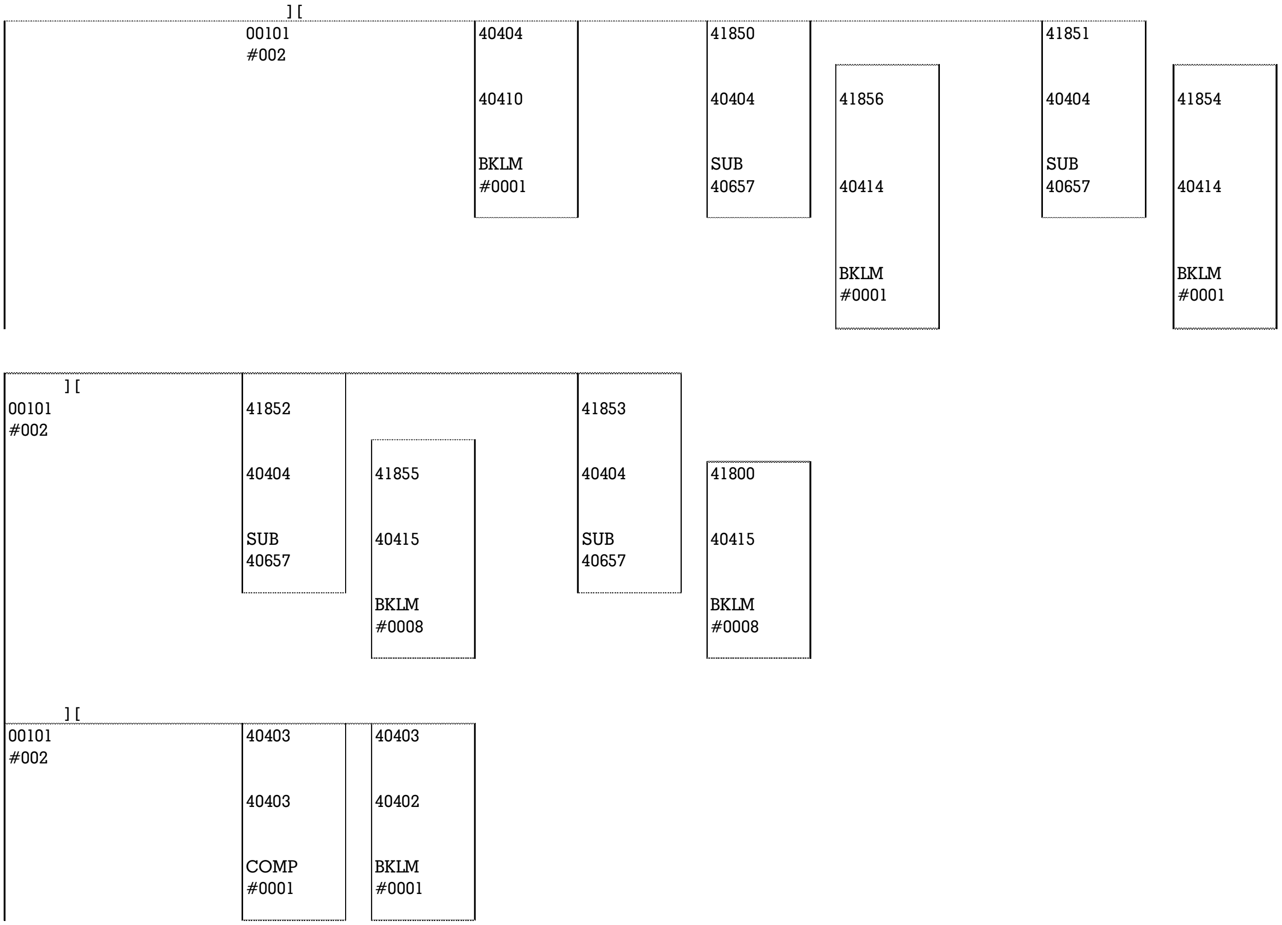
Segment: 01 Netzwerk: #00001



Segment: 01 Netzwerk: #00002



Segment: 01 Netzwerk: #00003



```
ASCII
Numerical operand list


ADDR  DEC  HEX      ASC  SYMBOL  LABEL     DESCRIPTION                  USED


00001  = 0 E             COM_ERROR                                 Net #002 1,* -( )
00100  = 1 E             INIT                                      Net #001 3,* -( )
00101  = 0 E              ACCESS       ACCESS TO DATAFIELD POSSIBILE  Net #002 3,* -( )
10001  = 0 E             COM_START
```

```
40400  = 00000 0000 .. PCS_DATA          Datafield
40401  = 00000 0000 ..
40402  = 40606 9E9E .. SYNC_1
40403  = 40606 9E9E .. SYNC_2
40404  = 00000 0000 .. F_KEYS
40405  = 00000 0000 ..
40406  = 00256 0100 ..
40407  = 00128 0080 ..
40408  = 00000 0000 ..
40409  = 00000 0000 ..
40410  = 00000 0000 ..
40411  = 00000 0000 ..
40412  = 00000 0000 ..
40413  = 04040 0FC8 .. COM_WORD1
40414  = 00000 0000 .. COM_WORD2
40415  = 00000 0000 .. MESS_BITS
40416  = 00000 0000 ..
40417  = 00000 0000 ..
40418  = 00000 0000 ..
40419  = 00000 0000 ..
40420  = 00000 0000 ..
40421  = 00000 0000 ..
40422  = 00000 0000 ..
40423  = 00000 0000 ..
40424  = 00000 0000 ..
40656  = 00000 0000 .. TIMEOUT
40657  = 00000 0000 .. SCRATCH

41750  = 00000 0000 .. INIT_VAL          Start presettings
41751  = 00000 0000 ..
41752  = 00000 0000 ..
41753  = 00000 0000 ..
41754  = 00000 0000 ..
41755  = 00000 0000 ..
41756  = 00000 0000 ..
41757  = 00000 0000 ..
41758  = 00000 0000 ..
41759  = 00000 0000 ..
41760  = 00000 0000 ..
41761  = 00000 0000 ..
41762  = 00000 0000 ..
41763  = 04040 0FC8 ..       PCS 090/95 command word
41764  = 00128 0080 ..             〃
41765  = 00000 0000 ..
41766  = 00000 0000 ..
41767  = 00000 0000 ..
41768  = 00000 0000 ..
41769  = 00000 0000 ..
41770  = 00000 0000 ..
41771  = 00000 0000 ..
41772  = 00000 0000 ..
41773  = 00000 0000 ..
41774  = 00000 0000 ..

41800  = 00000 0000 .. COFF_VAL          Values for communication loss
41801  = 00000 0000 ..
41802  = 00000 0000 ..
41803  = 00000 0000 ..
41804  = 00000 0000 ..
41805  = 00000 0000 ..
41806  = 00000 0000 ..
41807  = 00000 0000 ..
41808  = 00000 0000 ..
```

```
41809   = 00000 0000 ..
41810   = 00000 0000 ..
41811   = 00000 0000 ..
41812   = 00000 0000 ..
41813   = 04040 0FC8 ..
41814   = 00128 0080 ..
41815   = 00000 0000 ..
41816   = 00000 0000 ..
41817   = 00000 0000 ..
41818   = 00000 0000 ..
41819   = 00000 0000 ..
41820   = 00000 0000 ..
41821   = 00000 0000 ..
41822   = 00000 0000 ..
41823   = 00000 0000 ..
41824   = 00000 0000 ..

41850   = 32768 8000 .. K8000       Constants
41851   = 16384 4000 @. K4000
41852   = 08192 2000  . K2000
41853   = 04096 1000 .. K1000
41854   = 00000 0000 .. K0000
41855   = 65535 FFFF .. KFFFF
41856   = 00129 0081 .. K0081
```

Description of the „AEG250PX“ Expander driver for the PCS 900/PCS 950/ PCS 950c/PCS 9000 operating consoles. For the communication with an AEG A250 PLC.

Required devices and accessories:
The following products are required to operate a programmable controller with an already parameterized PCS (System Lauer company):

1. The PCS operating console itself (already parameterized).
2. The PCS 778 adapter cable for the PCS - programmable controller connection via RS-232C.
3. This manual.
4. Floppy disk and PCSPRO manual with AEG250PX driver (for the PCS side).
5. AEG master floppy disk with AEG250PX handling software for the A250.

Furthermore are necessary (AEG company):

6. An A250 controller.
7. DOLOG programming software.
8. An AEG adapter cable for the programming of the programmable controller.

.... as well as the power supplies for all components.

D1.1 Delimitation

The System Lauer operating consoles of the PCS topline series are designated in the following as PCS. The parameterization software for parameterizing the PCS is designated as PCSPRO.

The successful parameterization of the PCS, as described in the PCS 91/925/991 or 9091 manual, is assumed. This appendix relates exclusively to the use of a PCS together with an A250 controller of the AEG company. This controller is defined in the following as programmable controller, the programming software for the programmable controller as DOLOG, and the DOLOG program to be loaded as AEG250PX. The AEG-specific terms and the programming of the programmable controller with the DOLOG software are assumed as known.

This driver was developed on the A250 controller.

The PCS is connected to the programming interface of the controller.

Warning!
Use only the PCSPROPCSPRO^{WIN}/PCSPRO^{PLUS} software and/or PCS 9092 for the configuration. Other software packages can initiate malfunctions in the PCS and programmable controller.

D1.2 Loading of the AEG250PX driver into the THE PCS

During configuration of the PCS, both the user program with data and a selected driver are transferred. The driver is called „AEG250PX".

To configure a PCS-MIDI set DIL 7 according to the required baud rate („OFF" = 38.5 Kbaud, „ON" = 115 Kbaud), DIL 8 on „OFF" and DIL 9 on „ON". Connect PCS and PC via the PCS 733 programming cable. After you have selected the expander driver for the AEG A250 in the PCSPRO or PCS9092 programming software, the AEG250PX driver is loaded automatically. Before, you must inform the driver of the location of the send and receive buffer in the programmable controller. The MW 1256 start flag is defined as default. Take note during configuration that the length of the buffer is 100 words. If other values than the defaults are used, then the variables in the AEG250PX programmable controller handling software must also be adjusted correspondingly! This is described comprehensively in section 3.

D1.2.1 PCS midi driver variables

Seven variables can be set for the AEG250PX driver in the PCSPRO software under the menu item Project/Driver Parameter.... The content of the variables can be examined with the PCS offline menu.

1 AA variable: PCS Time-out time

The time-out time determines the maximum admissible time for the order processing of the synchronization word in the programmable controller. Default value for this time is 4000. This corresponds to 4000 ms. = 4.0 seconds. Admissible values for the variable are from 2000 to 19990. This corresponds to 2 to 19.99 seconds.

2 - 5 AC, AD, AE and AF variables: Baud rate and transfer type

The baud rate and interface, the PCS and the programmable controller use for communication are set with the PCS DIL-switches 5 and 6. Here, you select the same settings that you have selected in PCSPRO under the DRIVER VARIABLES menu item. The default settings are shown In the table below.

DIL5	DIL6	Variable	Preset values
off	off	AC	19200 Baud
on	off	AD	9600 Baud
off	on	AE	19200 Baud
on	on	AF	9600 Baud

All baud rates and transfer types that can be selected using the DIL-switches are shown in the following table.

Interface	Baud rate
RS 232	19200
RS 232	9600

6 AG variable: Send/receive buffer

Determines the first used flag word in the programmable controller for the data interchange area. Thereby, the variable has an offset function. Possible values are from 1 to 15000. The default setting for the flag word is 1256.

7 AJ variable: Task per package

Determines the number of the sub-packages for the data interchange. The default setting is AJ = 50. If AJ is reduced, then the transfer time of tasks with high priority is decreased (e.g. key tasks). Tasks with lower priority (e.g. present values) will be transferred correspondingly less often.

During configuration of the PCS, both the application program with data and a selected driver are transferred. The driver is called „AEG250PX.DRV".

D1.2.2 The PCS maxi driver variables

COM_TIMEOUT: PCS time-out time
The time-out time determines the maximum admissible time for the processing of the job package in the programmable controller. The default is 2000ms. Admissible values for the variable are 1000ms to 9999ms (= 1 to 9.9 sec.).

COM_DAT_PL: Start address of the I/ O buffer in the PLC
Determines the first used flag word in the programmable controller for the data interchange area. Thereby, the variable has an offset function. The possible values for flag words are 1 to 15000. The default setting is 1256. Therefore, flag word 1256.

COM_MAXLEN: Maximum number of the packages per communication cycle.
In a communication cycle, the PCS forms sub-packages for each task. The number of these sub-packages can be limited. This results in reducing the time for a communication cycle. This can be advantageous for fast jog operations (as far as possible without variable display). On the other hand, the refreshing of variables thereby takes longer. Limit this variable only if all other measures have not resulted in sufficient time savings. The possible values are between 1 and 32. Default value is 32.

Set the rotary switch at the rearside of the PCS to an arbitrary position 0..3.
The only baud rate is 19200 baud.

D.1.3 Connection of the PCS to the PLC

> **Warning!**
> Check the PCS function and also after parameterization and/or driver installation. All parameterized functions must be examined. Otherwise malfunctioning of the PCS and/or programmable controller are possible.

1. After you have configured the PCS, set the DIL-switches 8 and 9 at the rearside of the PCS to „OFF".
2. Supply operating voltage (19..33V) to the PCS. At least the ERR LED must light now.
3. Load the AEG250PX expander block into the programmable controller.
4. Connect the programming interface of the programmable controller to the PCS using the PCS 778 adapter cable.
5. Switch the programmable controller to „RUN". If the A 2.1 ERROR output lights, then the E2.1 restart input must be set.
6. Now, the ERR LED at the PCS must be deactivated. Idle text 0 appears on the PCS display. If this should not be the case, then continue reading under section D1.5.

D1.4 Trouble-shooting

The communication between programmable controller and PCS is monitored constantly by the two participants. The error indication on the programmable controller is made by setting the A 2.1 fault output. On the occurrence of communication faults, a corresponding fault text is shown in the display of the PCS and the ERR LED flashes.

Here, the most frequent faults are listed which occur during the first commissioning and permanent operation:

1. DIL switch no. 8 is set to ON. If this is the case, then the PCS enters a diagnosis routine after powering up. This is required only for test purposes. Remedy: set DIL switch 8 to OFF and restart the PCS (by cycling power or pressing the momentary reset push-button above the DIL switch).
2. The fault output „A 2.1" is set in the programmable controller. This is the case if no new order is written into the programmable controller receive buffer within 4 seconds.
   Has the E 2.1 restart input been shortly set at the programmable controller?
   Did you make the same definitions for the data block in programmable controller and in PCS?
   Have you used the correct cable?
   Is the cable defective?

3. The communication is active but after a certain time the following message appears on the display of the PCS:

   COMMUNICATION ERROR

   Has the E 2.1 restart input been shortly set at the programmable controller?
   Did you make the same definitions for the data block in programmable controller and in PCS?
   Have you used the correct cable?
   Is the cable defective?
   Is the PCS/programmable controller connection routed in a too noisy environment?
   Are the grounding relationships insufficient?
   Has the programmable controller been switched to STOP?

D1.5 Offlinemenu

A helpful diagnosis for the PCS 090/095/900 is the output of the PCS status on the display. This display can be accessed by the following combination: press the Help key after a PCS start (e.g. after a reset). The ERR LED must light constantly. This option is available no more after starting the communication (ERR LED is deactivated or flashes). By pressing the Help plus arrow keys you can display the PCS version, the data block version, and the driver version as well as the selected driver variables.

D1.7 References for the connection of the PCS to a PLC

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

Connect the cable screening to the central earthing point of the switching cabinet!
Ensure appropriate chassis groundings with regard to the PCS housing on the one hand and the programmable controller 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 interference 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.
Use a separate power supply for the PCS (24 V, 10 VA if the supply voltage is influenced by high interference. 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 equi-potential 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!

The used AEG250PX driver is an expander driver, i.e. it replaces the data area between programmable controller and PCS via job packages. A programmable controller program is required for this. Programmable controller and PCS communicate with each other using the BKOS protocol via RS-232C with 9600 or 19200 bauds, 8 bit, ODD parity and 1 stop bit.

D2.1 AEG250PX Expander block

D2.1.1 Expander block parameters

Determing the data area

A common data area must be defined for accessing the programmable controller and the PCS. This area must be physically available and defined in the programmable controller.

In the PCS, the send and receive area is defined via the AG driver variable. This send and receive buffer requires 100 reserved flag words that are located after the selected start address in the PCS. Furthermore, a data field with a size of 256 flag words is required in which the PCS data words are stored. In the programmable controller, the start addresses of the 2 data fields are defined in FB 201. A time-out can appear during the operation if the specifications of these areas are too small in the programmable controller!

Inputs:

RESET (E2.1): Input that admits a restart after loss communication.

Outputs:

COM_ERR (A2.1): Output that is active at communication loss.

Flag bits:

TIM_RES (M1.1): Flag to start the time-out timer. It is cyclically set and reset during active communication.

COM_OK (M1.2): Flag for communication was started (only for a programmable controller restart).

RESET_M (M1.3): Flag for time-out reset. Flag is set after set restart input and successful communication start.

Flag words:

P_DAT_OFF (MW1): This flag word points to the PCS data area starting address.

P_REC_OFF (MW2): This flag word points to the start address of the send and receive buffer.

PJOB_W (MW3): Pointer to the current orders or data in the receive buffer.

COUNT_W (MW4): Counter of all received data.

ADR_W (MW5): Flag word contains the current PCS data word number.

JOB_W (MW6): Flag word contains the current order instruction.

DW_COUNT_W: Flag word contains the number of data to be processed on the current order (MW7).

SYNC_W (MW8): This flag word contains a sent order number. The order number is used for the synchronization of the communication.

PSEND (MW 9): Pointer to the send buffer in which the data to be read are written.

TEMP1_W - TEMP4_W (MW10 - MW 13):
These flags are used as buffers.

D2.1.2 PB1

The enclosed PBS1 program block manages the data traffic between the A250 and a PCS 9000. Thereby, the PCS sends orders to the receive buffer of the programmable controller. The expander block transfers the data according to the received orders. The synchronization of the data traffic is made via an order number. The PCS writes a running number into the 1st. word of the receive buffer. After processing the orders, the expander block writes the inverted number into the 1st. word of the send buffer. At communication loss, this number is not written inverted. The PCS recognizes thus an interruption of the communication. For the data interchange, the expander requires a data area of 100 words as send and receive buffer. Furthermore, a data area of 256 flag words (PCS DATA BLOCK) is required for PCS data. The location of the data buffer is determined in FB201 that is selected by OB1 during the 1st. scan. The following FBs are selected in PB1:

FB202:
Recognizes and initializes a new job package.

FB200:
Processes the received job package and synchronizes the data interchange.
All received orders are processed successively.

Furthermore, communication faults in OB1 are recognized via a time-out timer. If the timer is not set anew by a missing job package, then the preset time (default value 4 sec.) runs down and the A2.1 output is set. If a time-out fault appears, then communication must be restarted via the E2.1 input.

D2.1.3 FB 200

FB200 has the task of selecting and executing the individual tasks from the received job package. FBs 210 to 213 are selected depending on the order. Using the COUNT_W data counter, the job package end is recognized and the FB 205 synchronization block is selected.

D2.1.4 FB201

This block initializes the location of the data areas. The P_DAT_OFF flag word (default value MW 1000) is initialized with the start address of the PCS data area. The PCS data area requires 256 flag words, whereby the location is driver-independent. The P_REC_OFF flag word initializes the location of the send and receive buffer. P_REC_OFF (default value MW 1256) must correspond to the address selected in the driver. The send and receive buffer requires 100 words.

D2.1.5 FB202

FB 202 recognizes a new job package and initializes the corresponding pointers that are necessary for package processing.

D2.1.6 FB205

FB 205 will be selected after the job packaget has been processed. Herewith, the counters are reset. The received synchronization word is inverted and entered into the send buffer.

D2.1.7 FB210

FB210 is called by FB200 at a selected reading order. Herewith, the data words of the PCS data block addressed by the ADR_W variable are written into the send buffer of the programmable controller. The number of consecutive data words is defined by the DW_COUNT_W variable.

D2.1.8 FB211

FB211 is called by FB200 at a selected writing order. Herewith, the data in the receive buffer are written into the PCS data block. The ADR_W variable indicates the position in the data block. The number of consecutive data words is defined by the DW_COUNT_W variable.

D2.1.9 FB212

FB212 is called by FB200 at a selected AND order. Herewith, the data in the receive buffer and the corresponding data in the PCS data block are logically „ANDed“ and are written back into the PCS data block. The ADR_W variable indicates the position in the data block. The number of consecutive data words is defined by the DW_COUNT_W variable.

D2.1.10 FB213

FB213 is called by FB200 at a selected OR order. Herewith, the data in the receive buffer and the corresponding data in the PCS data block are logically „ORed“ and are written back into the PCS data block. The ADR_W variable indicates the position in the data block. The number of consecutive data words is defined by the DW_COUNT_W variable.

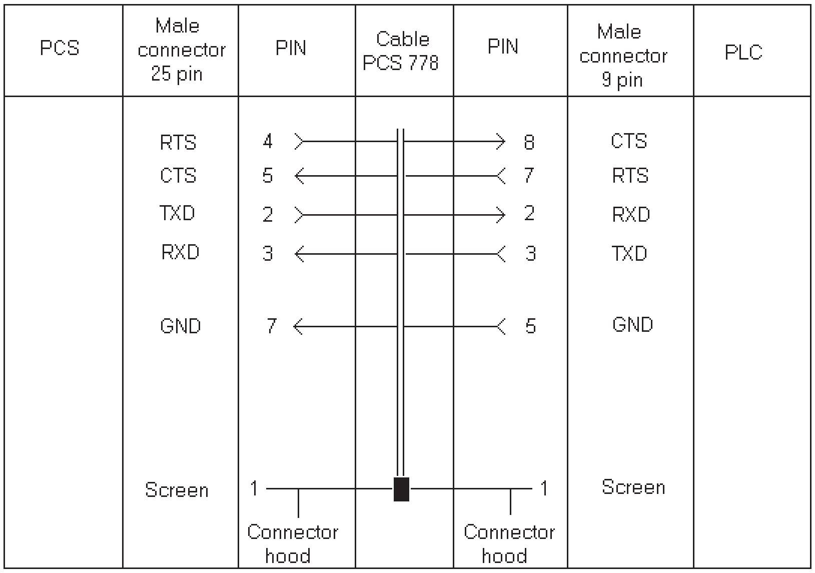
D2.2 Loading of the handling software

Warning!
Check the functioning of the handling software to avoid PCS and/or programmable controller malfunctions.

1. Connect the programmable controller and the PC via the programmable controller programming cable.
2. Run the DOLOG software on the PC.
3. Load the AEG250PX program.
4. Transfer the program into the programmable controller.
   Now, you can interface the PCS as described in section 2.4.

D3.1 PCS 778 adapter cable

AEG A250 - PCS connection via RS-232
The connection is made via the RS232 PG interface of the PLC.



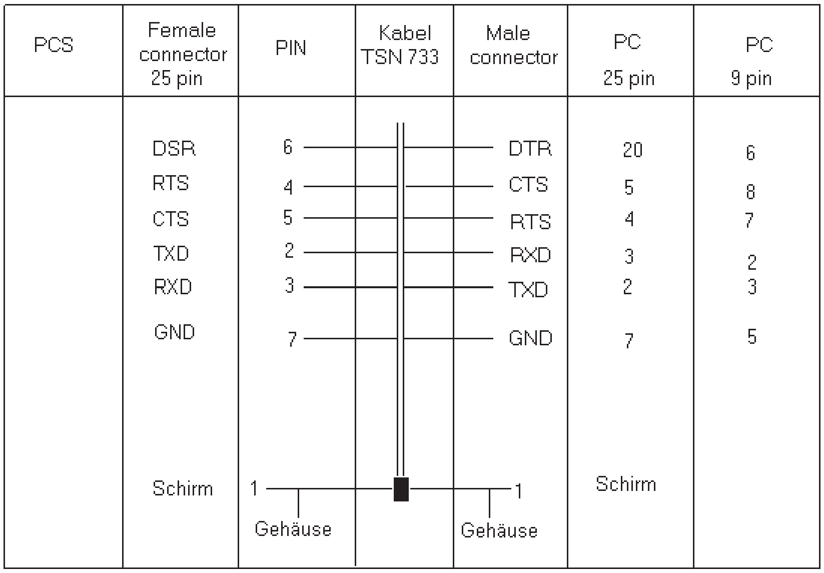
PCS	Male connector 25 pin	PIN	Cable PCS 778	PIN	Male connector 9 pin	PLC
	RTS	4		8	CTS	
	CTS	5		7	RTS	
	TXD	2		2	RXD	
	RXD	3		3	TXD	
	GND	7		5	GND	
	Screen	1 Connector hood		1 Connector hood	Screen	

D3.1.1 Screening

The screen should be connected on both sides to a metal coated connector case. The screen can also be connected to pin 1 when using non-metal coated connector cases. However, this is not recommended for error-technical reasons. There, the data lines should be covered as completely as possible by the screen! With grounding at both sides is to be noted however that possibly (because of ground potential shifts) a potential compensation wire is required having at least 10 times the screen cross section (reason: compensation currents should not flow through the cable screen if possible!). This applies particularly if PCS and programmable controller are not connected to the same earthing point. This is for example valid where the PCS and programmable controller are not housed in one switching cabinet!

D3.2 Programming cable PCS 733

PC - PCS Connection



PCS	Female connector 25 pin	PIN	Kabel TSN 733	Male connector	PC 25 pin	PC 9 pin
	DSR	6		DTR	20	6
	RTS	4		CTS	5	8
	CTS	5		RTS	4	7
	TXD	2		RXD	3	2
	RXD	3		TXD	2	3
	GND	7		GND	7	5
	Schirm	1		1	Schirm	
		Gehäuse		Gehäuse		

D3.3 Data transfer PCS - PLC

The data traffic with the controller is made in data packages. Each data package is provided with a checksum. The contents of the checksum is examined for possible faults by the programmable controller operating system and by the PCS. In addition, each read and write cycle is provided with a continuous order number.

The PCS is the master during communication. It has the task of establishing the communication and of sending orders to the programmable controller. PCS and programmable controller communicate asynchronously via the serial interface (RS 232). Adjustable transmission baud rates are 19200 and/or 9600. 8 data bits are fixed pre-defined, ODD parity, 1 stop bit for communication via the programming unit interface.
Only the interchange of data packages is described in this manual.

Structure of the reading cycle:

PCS PROGRAMMABLE CONTROLLER
READ N BYTES →
← ANSWER N BYTES
...

(REPETITION CORRESPONDING TO THE DATA FIELD)
...
READ N BYTES →
← ANSWER N BYTES

Structure of the write cycle:

PCS		PROGRAMMABLE CONTROLLER
WRITE N BYTES	→	
	←	OK / ERROR
	...	
(REPETITION CORRESPONDING TO THE DATA FIELD)		
	...	
WRITE N BYTES	→	
	←	OK / ERROR
WRITE DW3	→	
	←	OK / ERROR

D3.4 PCS - PLC effective response times

The response time of the protocol depends extensively on the tasks executed in the PCS. If variables are displayed or even processed, then the communication cycle time is essentially higher than with text without variables. Also, the transfer of the message bit area and LED words influences this time. Limit these transfers. You can do this also in the running operation, e.g. to realize a key-driven jog operation. The response is likewise greatly dependent on the programmable controller scan time since the AEG-programmable controller processes the communication at the end of a cycle.

The following table presents a reference for the speed of communication. The so-called „key → LED time" is measured. That means the time in which a key is transferred to the programmable controller and an LED is set in the PCS. This time is made up of 2 communication cycles and a programmable controller scan time. The response, i.e. the time a key is signaled in the programmable controller is only half the indicated time! The measurements were made with a PCS950 and a low programmable controller scan time burden.

Key → LED [milliseconds]

Baud rate	Variable	Time [ms]
19200	NONE	550
9600	NONE	700
19200	8 flush	650
9600	8 flush	800
19200	7 non-flush	750
9600	7 non-flush	950

The programmable controller scan time must be added to the time value which is an average value. The variables are not consecutively stored but stored with at least one word distance in between. This is the worst case since for each variable a write or read order must be sent. For a fast communication, you should make sure that the variables are located one after another in the data word. Then, several variables can be sent in a write and/or read order.

D3.5 Speed optimization

The speed of the data transfer depends essentially on two criteria:

1. The enabled transfer functions in the command words and
2. The number of variables displayed on the indicated display page.
   The transfer times can be multiplied by a bad transfer organization!
   The following measures can be taken to speed up the data transfer:

You can change the refresh behavior of the PCS tasks with the AJ driver variable (PCS maxi: [COM_MAXPLEN]). A small AJ number provides for short communication cycles and thus for a fast interchange of the keys. Variables however, take longer. A large AJ number includes many tasks in a communication cycle and thus causes a fast refreshing of variables. Key transfers however, take longer. Take note, that with a small AJ number, the key LED test for a short keystroke does not function optimally since the erase action of the key is performed due to it's high priority before the LEDs can be read.

To 1.

Programmable controller program optimizations for the PCS 009/ 090/ 095:

Disable all functions not required in the command words via the programmable controller program. Thereby, the response on data, which are always transferred, is reduced.

- In addition, with data word 13 command word A, you can limit the number of message words with the DW13 bits 0..3.
  If for instance you require only 35 messages, it suffices to read the message bits of 3 words. This can be adjusted by writing xxxxxxxx xxxx0011 to DW13. Depending on need, this setting of the programmable controller can be changed (dynamically) at any time.
- By setting bit 7 of DW13 to a logical 0 you disable the reading (transfer) of the entire LED STATUS WORDS W10..11 for the PCS 009/ 090 and W24..25 for the PCS 095.
- By setting bit 6 of DW13 to a logical 0 you block the reading (transfer) of the display and memory behavior.
- Avoid frequent changing of the display text since this requires the transfer of the status words 6 to 9.
- You can dynamically change the amount of transferred data with your programmable controller program.
  If for instance, you want to realize a jog operation, go ahead as follows: disable all functions as described above. Select jog operation text without variables. After termination of the jog operation, the transfer functions are enabled again.

To 1.

Programmable controller program optimizations for the PCS 900/ 920/ 950:
Generally, the same principles are valid as for the PCS 090 but the command words are arranged differently. Consider especially however, the transfer of clock time and message words.
Disable all functions not required in the command words via the programmable controller program. Thereby, the transfer overhead of the data that are always transferred, is reduced.

- In addition, with data word 37 command word B, you can limit the number of message words with the bits 0..7.
  If for instance you require less than 128 messages, it suffices to read a message block of 8 words. This can be adjusted by writing xxxxxxxx 00000001 to DW37. Depending on need, this setting of the programmable controller can be changed (dynamically) at any time.
- By setting bit 4 of DW36 to a logical 0 in you block the reading (transfer) of the entire LED STATUS WORDS W20..27.
- By setting bit 7 of DW36 to a logical 0 you block the reading of the command words C, D and E.
- By setting bit 5 of DW36 to a logical 0 you block the transfer of the clock. This is especially important since the clock is transferred each second, and thus greatly burdens the communication. Therefore, enable the clock transfer only if you absolutely need it.
- By setting bit 6 of DW36 to a logical 0 you block the transfer of the date. This has only a slight meaning since the date is transferred only at changes. Therefore, once a day.
- Avoid frequent changing of the display text, since the status words 6 to 9 are transferred at changes.
- You can dynamically change the amount of transferred data with your programmable controller program.
  If for instance, you want to realize a jog operation, go ahead as follows: disable all functions as described above. Select jog operation text without variables. After termination of the jog operation, the transfer functions are enabled again.
- To relieve your programmable controller program, you can use (with the PCS 950) the soft key functions for switching idle texts, menus, etc.. You can always disable this option by redefining the soft key bar using the programmable controller program.

To 1.

Programmable controller program optimizations with the PCS maxi 9000/ 9100:
Generally, the same principles are valid as for the PCS 090 but the command words are arranged differently. Consider especially however, the transfer of clock time.
Disable all functions not required in the command words via the programmable controller program. Thereby, the transfer overhead of the data that are always transferred, is reduced.

- In addition, with data word 13 - transfer interlock - you can disable the following transfers:
  Clock time, date, messages, menus, LEDs and operating printer pages.

- You can dynamically change the amount of transferred data with your programmable controller program.
  If for instance, you want to realize a jog operation, go ahead as follows: disable all functions as described above. Select jog operation text without variables. After termination of the jog operation, the transfer functions are enabled again.
- To relieve your programmable controller program, you can use the soft key functions for switching idle texts, menus, etc.. You can always disable this option by redefining the soft key bar using the programmable controller program.

To 2.
PCSPRO program optimizations with the PCS 009/ 090/ 095/ 900/ 920/ 950:

- Display as few variables as possible on the shown display page because the amount of transferred data increases with the number of the variables.
- It is advantage to make sure that the variables are consecutively arranged if several of them should be shown on the same display page. Then, several variables can be sent in a write or read order and the communication rate increases.
  For instance, if the first variable in the display is located on DW50, then further variables should be located on data words 51,52,53.., etc..

To 2.
PCSPRO program optimizations with the PCS maxi 9000/ 9100:

- Generally, the same principles are valid as for the PCS micro/mini/midi. In addition, you should take note however, that each open window generates it's own self-contained orders. The communication is slowed down if, e.g. variables in the status window are constantly shown.

D3.6 Communication error

During PCS - programmable controller communication, the PCS operates as master and the programmable controller as slave. Thus, it is the task of the PCS to set up and monitor communication. Thereby, the correct communication is monitored between the programmable controller and the PCS. The following error message is provided for this purpose:

```
======== COMMUNICATION ERROR ============
                 TIMEOUT
```

In both, in the PCS and in the programmable controller there is a time monitoring function for the serial data exchange and sync. word processing. In the PCS, the time-out time totals 4 seconds as default value. After that an error message is displayed in the PCS and the error LED flashes. The PCS tries again to establish the communication in the background. If this is successful and word 3 is processed, then the above shown error message is erased. Example: Interruption of the adapter cables.

The data interchange is performed in data blocks and each block is examined for length, content and block check. If a fault is discovered, then the block is requested once more. With more repetitions in a sequence, the communication is interrupted and the above shown error message appears.

```
======== COMMUNICATION ERROR ============
      Timeout programmable controller
```

The status of the programmable controller is scanned during establishing the communication. Thus, the PCS can ascertain whether the programmable controller is in RUN mode. If the programmable controller in the STOP mode then the above-mentioned message appears. If the programmable controller is switched into RUN mode then the message is erased and the communication is resumed.

Printout of the AEG250PX handlingsoftware

```
Overview:
───OB1────PB1────SFB140
                      ┌──FB200──┬SFB139
                      │         ├──FB205──┬SFB138
                      │         │         └──SFB139
                      │         ├──FB210───SFB140
                      │         ├──FB211───SFB140
                      │         ├──FB212──┬SFB138
                      │         │         └──SFB139
                      │         └──FB213──┬SFB138
                      │                   └──SFB139
                      ├──FB201
                      └──FB202───SFB139
```

OB1:

```
                                                         Signal    Symbol
NETWORK:    1                                            ────────────────────
      :BA    PB1
      :***                                               PB1

NETWORK:    2                                            ────────────────────

      :BE
```

PB1:

```
NETWORK:    1
                                                         Signal    Symbol
                                                         ────────────────────
      :U    ZYKLUS1                                      SM119     ZYKLUS1
      :BAB  FB201 ;inits data buffers on first run FB201 FB201
NAME  :INIT                                              A2.1      COM_ERR
      :UN   COM_ERR
      :SPB  =NO_ERR
      :L    K 0         ;inits data words on             MW2       P_REC_OFF
      :=    SYNC_W      ;communication error             MW9       PSEND
      :L    P_REC_OFF                                    A2.1      COM_ERR
      :ADD  K 50
      :=    PSEND
      :BAB  LA6
NAME  :LA6                                               MW2       P_REC_OFF
IN    :     P_REC_OFF                                    MW9       PSEND
OUT   :     PSEND                                        A2.1      COM_ERR
      :U    COM_ERR     ;if com error jump to cycle end
      :UN   RESET
      :SPB  =ZY_END
NO_ERR:L    COUNT_W     ;all job in receive buffer done? MW4       COUNT_W
      :<>   K 0
      :SPB  =JOB
      :U    VLAUF       ;init timer-reset                SM1       VLAUF
      :U    COM_OK                                       M1.2      COM_OK
      :=    TIM_RES                                      M1.1      TIM_RES
      :BA   FB202       ;init the new received data      FB202
NAME  :JOB_INI
      :SP   =ZY_END     ;jump to timer reset
JOB   :BA   FB200       ;select the several jobs         FB200
NAME  :JOB_SEAR
      :U    VLAUF       ;communication ok                SM1       VLAUF
      :=    COM_OK                                       M1.2      COM_OK
      :UN   VLAUF                                        SM1       VLAUF
      :O    COM_ERR                                      A2.1      COM_ERR
      :=    TIM_RES     ;timer reset                     M1.1      TIM_RES
ZY_END:L    COUNT_W     ;process all jobs in             MW4       COUNT_W
      :<>   K 0         ;the received data
      :SPB  =JOB
      :=    TEMP4_W                                      MW13      TEMP4_W
```

```
NAME   :ANW_PRO
       :U     TIM_RES      ;timeout timer                  M1.1      TIM_RES
       :SE    T1                                           T1
       :DZB   1000MS
       :L     K 4
       :U     RESET        ;reset timeout output           E2.1      RESET
       :R     T1                                           T1
       :=     COM_ERR      ;timeout -> output a2.1         A2.1      COM_ERR
       :***
```

FB200:

```
NETWORK:     1
NAME   :JOB_SEAR                                           Signal    Symbol
       :***
NETWORK:     2
       :BA    LA5          ;actual job                     SFB139    LA5
NAME   :LA5
IN     :      PJOB_W                                       MW3       PJOB_W
OUT    :      TEMP3_W                                      MW12      TEMP3_W
       :L     TEMP3_W                                      MW12      TEMP3_W
       :=     ADR_W                                        MW5       ADR_W
       :=     JOB_W                                        MW6       JOB_W
       :=     DW_COUNT_W                                   MW7       DW_COUNT_W
       :L     ADR_W        ;data address data buffer       MW5       ADR_W
       :U     KH FF00
       :ROR   K 8
       :=     ADR_W                                        MW5       ADR_W
       :L     DW_COUNT_W   ;data counter                   MW7       DW_COUNT_W
       :U     KH F
       :=     DW_COUNT_W                                   MW7       DW_COUNT_W
       :L     JOB_W        ;actual job                     MW6       JOB_W
       :U     KH F0
       :=     JOB_W                                        MW6       JOB_W
       :L     JOB_W        ;read                           MW6       JOB_W
       :==    KH 10
       :SPB   =READ
       :L     JOB_W        ;write                          MW6       JOB_W
       :==    KH 20
       :SPB   =WRITE
       :L     JOB_W        ;and function                   MW6       JOB_W
       :==    KH 40
       :SPB   =AND
       :L     JOB_W        ;or function                    MW6       JOB_W
       :==    KH 80
       :SPB   =OR
READ   :BA    FB210                                        FB210
NAME   :READ_FUN
       :SP    =A_END
WRITE  :BA    FB211                                        FB211
NAME   :WRIT_FUN
       :SP    =A_END
AND    :BA    FB212                                        FB212
NAME   :AND_FUN
       :SP    =A_END
OR     :BA    FB213                                        FB213
NAME   :OR_FUN
A_END  :L     COUNT_W      ;if all jobs done               MW4       COUNT_W
       :<>    K 0
       :SPB   =B_END
       :BA    FB205        ;update sync word               FB205
NAME   :AUFT_END
B_END  :***
NETWORK:     3
       :BE
```

Printout of the AEG250PX handlingsoftware

D

FB201:

```
NETWORK:    1

NAME  :INIT
      :***
                                                       Signal    Symbol
NETWORK:    2                                          _____________________

      :L    K 1000      ;offset for 256 word pcs data  MW1       P_DAT_OFF
      :=    P_DAT_OFF
      :L    K 1256      ;offset 100 word I/O buffer    MW2       P_REC_OFF
      :=    P_REC_OFF   ;and receive buffer
      :***

NETWORK:    3

      :BE

FB202:

NETWORK:    1

NAME  :JOB_INI
      :***
                                                       Signal    Symbol
NETWORK:    2                                          _____________________

                                                       MW2       P_REC_OFF
     :L    P_REC_OFF   ;pointer to first received data MW3       PJOB_W
      :=    PJOB_W                                     MW2       P_REC_OFF
      :L    P_REC_OFF                                  MW9       PSEND
      :=    PSEND                                      MW9       PSEND
      :L    PSEND
      :ADD  K 51                                       MW9       PSEND
      :=    PSEND       ;pointer to send buffer        SFB139    LA5
      :BA   LA5
NAME  :LA5                                             MW3       PJOB_W
IN    :     PJOB_W                                     MW11      TEMP2_W
OUT   :     TEMP2_W                                    MW8       SYNC_W
      :L    SYNC_W      ;new syncword?                 MW11      TEMP2_W
      :==   TEMP2_W
      :SPB  =NW_END     ;no! -> no new jobs            SFB139    LA5
      :BA   LA5         ;count off all received data
NAME  :LA5                                             MW3       PJOB_W
IN    :     PJOB_W                                     MW4       COUNT_W
OUT   :     COUNT_W                                    MW4       COUNT_W
      :L    COUNT_W     ;store count
      :U    KH FF00
      :ROR  K 8                                        MW4       COUNT_W
      :=    COUNT_W                                    MW3       PJOB_W
      :L    PJOB_W      ;pointer to first job
      :ADD  K 1                                        MW3       PJOB_W
      :=    PJOB_W
NW_END:***                                             _____________________
NETWORK:    3

      :BE
```

D

Printout of the AEG250PX handlingsoftware

FB205:

```
NETWORK:     1

NAME   :AUFT_END
       :***

NETWORK:     2

       :L     P_REC_OFF
       :=     PJOB_W       ;pointer sync word receive
       :L     P_REC_OFF
       :=     PSEND
       :L     PSEND
       :ADD   K 50
       :=     PSEND        ;pointer sync word send
       :BA    LA5          ;received sync word
NAME   :LA5
IN     :      PJOB_W
OUT    :      COUNT_W
       :BA    LA5          ;store received sync word
NAME   :LA5
IN     :      PJOB_W
OUT    :      SYNC_W
       :L     COUNT_W
       :U     KH FF        ;select sync byte
       :=     COUNT_W
       :L     COUNT_W
       :X     KH FF        ;invert sync byte
       :=     COUNT_W
      :BA    LA4         ;invert sync word to send buff
NAME   :LA4
IN     :      COUNT_W
OUT    :      PSEND
       :L     K 0          ;counter = 0
       :=     COUNT_W
       :***

NETWORK:     3

       :BE
```

Signal	Symbol
MW2	P_REC_OFF
MW3	PJOB_W
MW2	P_REC_OFF
MW9	PSEND
MW9	PSEND
MW9	PSEND
SFB139	LA5
MW3	PJOB_W
MW4	COUNT_W
SFB139	LA5
MW3	PJOB_W
MW8	SYNC_W
MW4	COUNT_W
MW4	COUNT_W
MW4	COUNT_W
MW4	COUNT_W
SFB138	LA4
MW4	COUNT_W
MW9	PSEND
MW4	COUNT_W

FB210:

```
NETWORK:     1

NAME   :READ_FUN
       :***

NETWORK:     2

       :L     PJOB_W       ;actual job
       :ADD   K 1
       :=     PJOB_W       ;next job
       :L     COUNT_W      ;decrement data counter
       :SUB   K 1
       :=     COUNT_W
       :L     P_DAT_OFF    ;pointer actual pcs data
       :ADD   ADR_W
       :=     ADR_W
RD_LP  :BA    LA6          ;data to send buffer
```

Signal	Symbol
MW3	PJOB_W
MW3	PJOB_W
MW4	COUNT_W
MW4	COUNT_W
MW1	P_DAT_OFF
MW5	ADR_W
MW5	ADR_W
SFB140	LA6

```
NAME  :LA6
IN    :     ADR_W                                           MW5       ADR_W
OUT   :     PSEND                                           MW9       PSEND
      :L    PSEND       ;increment send pointer             MW9       PSEND
      :ADD  K 1
      :=    PSEND                                           MW9       PSEND
      :L    ADR_W       ;increment data pointer             MW5       ADR_W
      :ADD  K 1
      :=    ADR_W                                           MW5       ADR_W
      :L    DW_COUNT_W  ;decrement data word counter        MW7       DW_COUN_W
      :DEC
      :=    DW_COUNT_W                                      MW7       DW_COUN_W
      :L    DW_COUNT_W                                      MW7       DW_COUN_W
      :==   K 0
      :SPZ  =RD_LP      ;read until data counter <> 0
      :=    DW_COUNT_W                                      MW7       DW_COUN_W
      :***
NETWORK:    3
      :BE
```

FB211:

```
NETWORK:    1

NAME  :WRIT_FUN
      :***

NETWORK:    2                                           Signal    Symbol

      :L    COUNT_W     ;update data word counter           MW4       COUNT_W
      :SUB  K 1
      :SUB  DW_COUNT_W                                      MW7       DW_COUN_W
      :=    COUNT_W                                         MW4       COUNT_W
      :L    P_DAT_OFF   ;pointer actual pcs data            MW1       P_DAT_OFF
      :ADD  ADR_W                                           MW5       ADR_W
      :=    ADR_W                                           MW5       ADR_W
      :L    PJOB_W      ;actual job                         MW3       PJOB_W
      :ADD  K 1
      :=    PJOB_W      ;pointer to received data           MW3       PJOB_W
WR_LP :BA   LA6         ;received data to data buffer       SFB140    LA6
NAME  :LA6
IN    :     PJOB_W                                          MW3       PJOB_W
OUT   :     ADR_W                                           MW5       ADR_W
      :L    PJOB_W                                          MW3       PJOB_W
      :ADD  K 1
      :=    PJOB_W      ;pointer to next data               MW3       PJOB_W
      :L    ADR_W       ;increment data pointer             MW5       ADR_W
      :ADD  K 1
      :=    ADR_W                                           MW5       ADR_W
      :L    DW_COUNT_W  ;decrement counter received data    MW7       DW_COUN_W
      :SUB  K 1
      :=    DW_COUNT_W                                      MW7       DW_COUN_W
      :SPN  =WR_LP      ;write until data counter <> 0
      :***
NETWORK:    3

      :BE
```

FB212:

```
NETWORK:    1

NAME  :AND_FUN
      :***

NETWORK:    2                                            Signal    Symbol

      :L    COUNT_W     ;update data word counter        MW4       COUNT_W
      :SUB  K 1
      :SUB  DW_COUNT_W                                   MW7       DW_COUN_W
      :=    COUNT_W  ¦                                   MW4       COUNT_W
      :L    P_DAT_OFF   ;pointer actual pcs data         MW1       P_DAT_OFF
      :ADD  ADR_W                                        MW5       ADR_W
      :=    ADR_W                                        MW5       ADR_W
      :L    PJOB_W      ;actual job                      MW3       PJOB_W
      :ADD  K 1
      :=    PJOB_W      ;pointer to received data        MW3       PJOB_W
AND_LP:BA   LA5         ;received data                   SFB139    LA5
NAME  :LA5
IN    :     PJOB_W                                       MW3       PJOB_W
OUT   :     TEMP1_W                                      MW10      TEMP1_W
      :BA   LA5         ;plc data                        SFB139    LA5
NAME  :LA5
IN    :     ADR_W                                        MW5       ADR_W
OUT   :     TEMP2_W                                      MW11      TEMP2_W
      :L    TEMP2_W     ;and function between plc data   MW11      TEMP2_W
      :U    TEMP1_W     ;and received data               MW10      TEMP1_W
      :=    TEMP1_W                                      MW10      TEMP1_W
      :BA   LA4         ;actual data to data buffer      SFB138    LA4
NAME  :LA4
IN    :     TEMP1_W                                      MW10      TEMP1_W
OUT   :     ADR_W                                        MW5       ADR_W
      :L    ADR_W       ;increment data pointer          MW5       ADR_W
      :ADD  K 1
      :=    ADR_W                                        MW5       ADR_W
      :L    PJOB_W      ;pointer to next data            MW3       PJOB_W
      :ADD  K 1
      :=    PJOB_W                                       MW3       PJOB_W
     :L    DW_COUNT_W  ;decrement counter received data  MW7       DW_COUN_W
      :SUB  K 1
      :=    DW_COUNT_W                                   MW7       DW_COUN_W
      :SPN  =AND_LP     ;until datacounter <> 0
      :***
NETWORK:    3

      :BE
```

FB213:

```
NETWORK:     1

NAME   :OR_FUN
       :***

NETWORK:     2                                           Signal    Symbol

      :L    COUNT_W     ;update data word counter        MW4       COUNT_W
      :SUB  K 1
      :SUB  DW_COUNT_W                                   MW7       DW_COUN_W
      :=    COUNT_W                                      MW4       COUNT_W
      :L    P_DAT_OFF   ;pointer actual pcs data         MW1       P_DAT_OFF
      :ADD  ADR_W                                        MW5       ADR_W
      :=    ADR_W                                        MW5       ADR_W
      :L    PJOB_W      ;actual job                      MW3       PJOB_W
      :ADD  K 1                                          MW3       PJOB_W
      :=    PJOB_W      ;pointer to received data        SFB139    LA5
OR_LP :BA   LA5         ;received data
NAME  :LA5                                               MW3       PJOB_W
IN    :     PJOB_W                                       MW10      TEMP1_W
OUT   :     TEMP1_W                                      SFB139    LA5
      :BA   LA5         ;plc data
NAME  :LA5                                               MW5       ADR_W
IN    :     ADR_W                                        MW11      TEMP2_W
OUT   :     TEMP2_W                                      MW11      TEMP2_W
      :L    TEMP2_W     ;or function between plc data    MW10      TEMP1_W
      :O    TEMP1_W     ;and received data               MW10      TEMP1_W
      :=    TEMP1_W                                      SFB138    LA4
      :BA   LA4         ;actual dat to data buffer
NAME  :LA4                                               MW10      TEMP1_W
IN    :     TEMP1_W                                      MW5       ADR_W
OUT   :     ADR_W                                        MW5       ADR_W
      :L    ADR_W       ;increment data pointer
      :ADD  K 1                                          MW5       ADR_W
      :=    ADR_W                                        MW3       PJOB_W
      :L    PJOB_W      ;pointer to next data
      :ADD  K 1                                          MW3       PJOB_W
      :=    PJOB_W                                       MW7       DW_COUN_W
      :L    DW_COUNT_W  ;decrement counter received data
      :SUB  K 1                                          MW7       DW_COUN_W
      :=    DW_COUNT_W
      :SPN  =OR_LP      ;until data counter <> 0¦
      :***
NETWORK:     3

      :BE
```

D5.1 Ideograms and symbols

The following symbols and ideograms are used in this manual.

⚠

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

D5.2. Safty related information

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

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

Modifications reserved

D5.3 Quality and Support

Quality is the most important factor in our company. From the electronic component to the manufactured device, quality is completely tested by qualified personal. For this purpose, national and international test standards (ISO, TÜV, VDE, CE, Germanischer Lloyd) are applied. Each PCS is tested to 100% at different temperatures (5 ... 55°C) and test voltages (19 ... 33 VDC) and submitted to a permanent test under worst case conditions during 48 hours. This assures a maximum of quality!

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

- Qualified application support by qualified sales engineers.
- Our support is available to you every day by word and deed. Use our direct info line, if you have questions concerning the PCS topline
- Intensive and practice-orientated training for our products. Either in our training center or, after agreement, in your company.
- You do not only receive demo devices, but you are also supported during your first application by our specialists.
- Update service for our software.

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

Who for what to task?

Whenever you need us we are there for you: dynamically, creatively and tremendously efficiently. Simply select the appropriate support and hotline number!

By the way...

... you can daily access the newest product informations via

Mailbox:	07022/9660225
CompuServe:	100565,1525
Internet:	100565.1525@compuserve.com

Distribution

Sales manager total	Henzler				
	07022/9660 240				
Sales manager Germany	Raif				
	07022/9660 242				
Sale processing	Gröger	Mönkemeier	Lehner		
	07022/9660 241	07022/9660 244	07022/9660 243		
Order processing	Feiler	Estner-Lenz			
	07022/9660 260	07022/9660 261			
Information processing	Koop	Maly			
	07022/9660 123	07022/9660 123			

Technical Support SIC

Support manager	Schauwecker				
	07022/9660 220				
Support		Arndt	Gekeler	Hauber	Kolbus
		07022/9660 226	07022/9660 221	07022/9660 223	07022/9660 222
Training & Exercise			Gekeler (EASYWARE)		Kolbus (LCA+PCS)
			07022/9660 221		07022/9660 222
responsible for product line					
LCA starline	■	■	■	■	■
LCA standard	■	■	■		■
PCS topline	■	■	■	■	■
PCS classic	■	■	■		■
VPC exclusiv, VPC compact	■		■		
VPC EASYWARE	■		■	■	
TeleService TSN	■				■
responsible for net & bus					
INTERBUS	■	■			■
PROFIBUS	■	■			■
ArcNET	■	■			■
responsible for PLC driver					
ABB		■			■
AEG	■	■			■
Allen Bradley				■	■
B&R	■			■	
Bosch	■	■			
Cegelec				■	■
Crouzet		■	■		
Eberle					■
Festo	■	■	■		■
GE Fanuc	■				■
Hitachi			■	■	
IPC	■				
IZUMI IDEC			■	■	
KLM	■				■
Matsushita		■			
Mitsubishi				■	■
OMRON	■		■		
PC	■	■			
Philips	■		■		
Saia				■	■
Samsung		■			
Siemens	■	■	■	■	■
Sprecher & Schuh		■			
Sulzer	■				
Teco					■
Telemecanique			■		■
Toshiba					■

Index