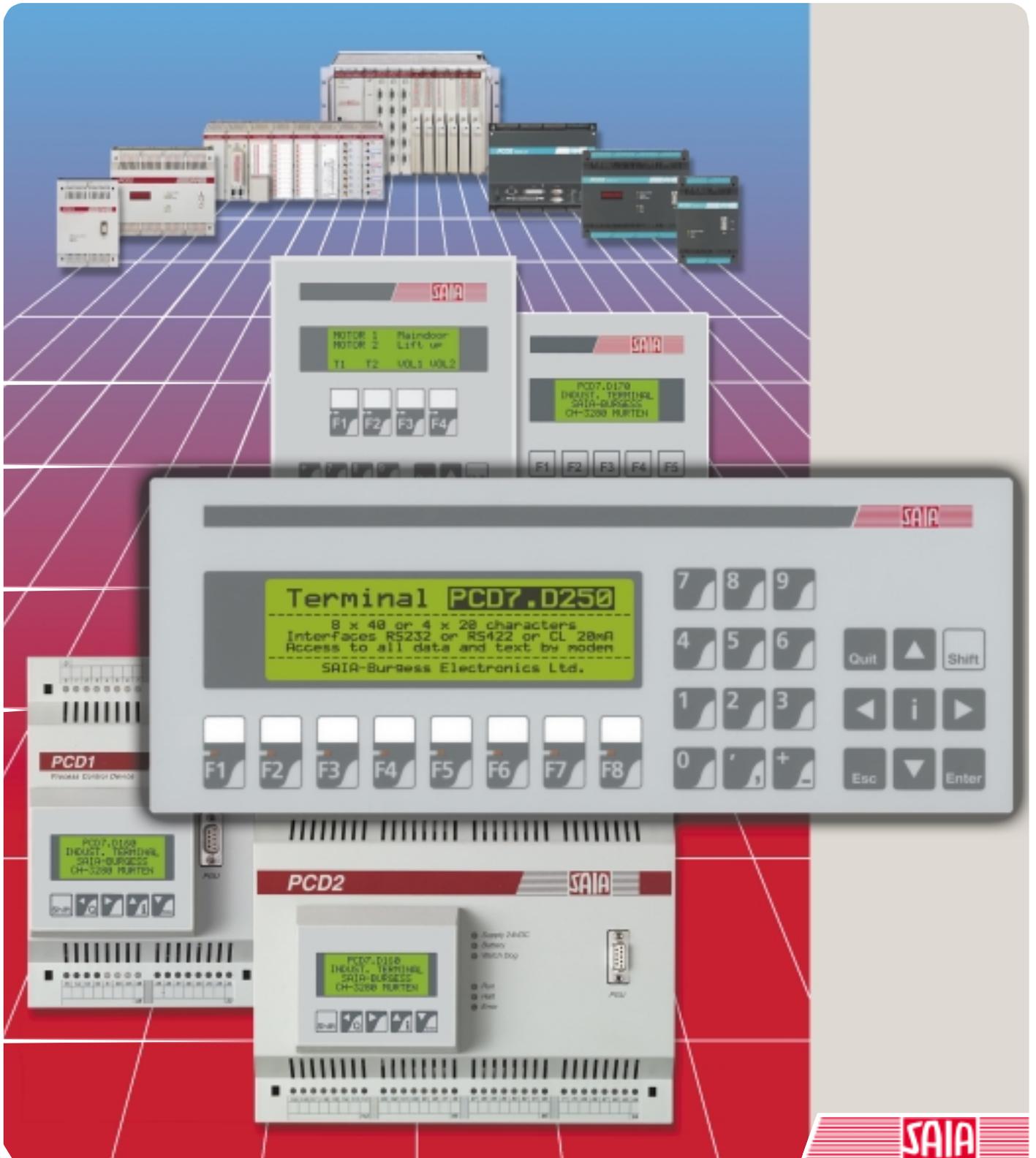


## SAIA® PCD Process Control Devices

## PCD7.D250 Industrie-Terminal Handbuch



---

## SAIA-Burgess Gesellschaften

<b>Schweiz</b>	SAIA-Burgess Electronics AG Freiburgstrasse 33 CH-3280 Murten ☎ 026 672 77 77, Fax 026 670 19 83	<b>Frankreich</b>	SAIA-Burgess Electronics Sàrl. 10, Bld. Louise Michel F-92230 Gennevilliers ☎ 01 46 88 07 70, Fax 01 46 88 07 99
<b>Deutschland</b>	SAIA-Burgess Electronics GmbH Daimlerstrasse 1k D-63303 Dreieich ☎ 06103 89 060, Fax 06103 89 06 66	<b>Niederlande</b>	SAIA-Burgess Electronics B.V. Hanzeweg 12c NL-2803 MC Gouda ☎ 0182 54 31 54, Fax 0182 54 31 51
<b>Österreich</b>	SAIA-Burgess Electronics Ges.m.b.H. Schallmooser Hauptstrasse 38 A-5020 Salzburg ☎ 0662 88 49 10, Fax 0662 88 49 10 11	<b>Belgien</b>	SAIA-Burgess Electronics Belgium Avenue Roi Albert 1er, 50 B-1780 Wemmel ☎ 02 456 06 20, Fax 02 460 50 44
<b>Italien</b>	SAIA-Burgess Electronics S.r.l. Via Cadamosto 3 I-20094 Corsico MI ☎ 02 48 69 21, Fax 02 48 60 06 92	<b>Ungarn</b>	SAIA-Burgess Electronics Automation Kft. Liget utca 1. H-2040 Budaörs ☎ 23 501 170, Fax 23 501 180

---

## Vertretungen

<b>Gross-britannien</b>	Canham Controls Ltd. 25 Fenlake Business Centre, Fengate Peterborough PE1 5BQ UK ☎ 01733 89 44 89, Fax 01733 89 44 88	<b>Portugal</b>	INFOCONTROL Electronica e Automatismo LDA. Praceta Cesário Verde, No 10 s/cv, Massamá P-2745 Queluz ☎ 21 430 08 24, Fax 21 430 08 04
<b>Dänemark</b>	Malthe Winje Automation AS Håndværkerbyen 57 B DK-2670 Greve ☎ 70 20 52 01, Fax 70 20 52 02	<b>Spanien</b>	Tecnosistemas Medioambientales, S.L. Poligono Industrial El Cabril, 9 E-28864 Ajalvir, Madrid ☎ 91 884 47 93, Fax 91 884 40 72
<b>Norwegen</b>	Malthe Winje Automasjon AS Haukelivn 48 N-1415 Oppegård ☎ 66 99 61 00, Fax 66 99 61 01	<b>Tschechische Republik</b>	ICS Industrie Control Service, s.r.o. Modranská 43 CZ-14700 Praha 4 ☎ 2 44 06 22 79, Fax 2 44 46 08 57
<b>Schweden</b>	Malthe Winje Automation AB Truckvägen 14A S-194 52 Upplands Väsby ☎ 08 795 59 10, Fax 08 795 59 20	<b>Polen</b>	SABUR Ltd. ul. Druzynowa 3A PL-02-590 Warszawa ☎ 22 844 63 70, Fax 22 844 75 20
<b>Suomi/ Finnland</b>	ENERGEL OY Atomitie 1 FIN-00370 Helsinki ☎ 09 586 2066, Fax 09 586 2046		
<b>Australien</b>	Siemens Building Technologies Pty. Ltd. Landis & Staefa Division 411 Ferntree Gully Road AUS-Mount Waverley, 3149 Victoria ☎ 3 9544 2322, Fax 3 9543 8106	<b>Argentinien</b>	MURTEN S.r.l. Av. del Libertador 184, 4° "A" RA-1001 Buenos Aires ☎ 054 11 4312 0172, Fax 054 11 4312 0172

---

## Kundendienst

<b>USA</b>	SAIA-Burgess Electronics Inc. 1335 Barclay Boulevard Buffalo Grove, IL 60089, USA ☎ 847 215 96 00, Fax 847 215 96 06
------------	---

**SAIA® Process Control Devices**

**Handbuch**

**Industrie-Terminal**

**PCD7.D250**

SAIA-Burgess Electronics AG 2000. Alle Rechte vorbehalten  
Ausgabe 26/770 D1 - 06.2000

Technische Änderungen vorbehalten

# Anpassungen

---

Handbuch: Industrie-Terminal PCD7.D250 - Ausgabe D1

Datum	Abschnitt	Seite	Beschreibung
01.12.2000	4.3	4-3	Serielle Schnittstelle COM 1: Übertragungs-Modus

# Inhalt

---

	Seite
<b>1. Anwendung</b>	
<b>2. Technische Daten</b>	
<b>3. Massbild</b>	
<b>4. Hardware</b>	
4.1 Stromversorgung / Anschlüsse	4-1
4.2 Systemprogramm	4-2
4.3 Serielle Schnittstelle COM 1: RS 232	4-3
4.3.1 Ohne Handshaking RTS/CTS oder mit Handshaking XON/XOFF	4-4
4.3.2 Mit Handshaking RTS/CTS	4-5
4.4 Serielle Schnittstelle COM 0: bestückt mit den Kommunikations-Modulen PCD7.F2..	4-6
4.4.1 RS 422 mit Kommunikationsmodul PCD7.F210	4-6
4.4.2 Stromschleife 20 mA (TTY) mit Kommunikationsmodul PCD7.F231	4-6
<b>5. Funktion</b>	
5.1 Einschaltvorgang	5-1
5.2 Die Tastatur	5-2
5.3 Setup/Test-Modus	5-4
5.3.1 Setup-Modus	5-5
5.3.2 Werks-Einstellung der Setup-Parameter	5-9
5.3.3 Demonstrations-Anzeige (Demo display)	5-9
5.3.4 Hardware-Test (Hardware tests)	5-9
5.3.5 Anzeige-Test (Display test)	5-9
5.3.6 Tastatur-Test (Keyboard test)	5-10
5.3.7 LED-Test (LED test)	5-10

	Seite
<b>6. Steuerbefehle über die serielle Schnittstelle</b>	
6.1 Konfigurationsbefehle	6-2
6.2 Cursor-Steuerung	6-5
6.3 Anzeige-Steuerung	6-8
6.4 LED-Steuerung	6-10
6.5 Sonstige Befehle	6-11
6.6 Übersicht aller Steuerfunktionen	6-13
<b>7. Die verschiedenen Charaktersätze</b>	
7.1 Erste ASCII-Tabelle (32...127 dez., 20...7F heza)	7-1
7.2 Erweiterte ASCII-Tabelle (128...255 dez., 80...FF heza)	7-1
7.2.1 CodePage 437 (Standard-Einstellung)	7-2
7.2.2 D100-kompatibel	7-4
<b>8. Programmbeispiele für die PCD</b>	
8.1 Hardware-Installation	8-1
8.2 Einfache Textausgabe	8-2
8.2.1 Einfache Textausgabe in BLOCTEC	8-2
8.2.2 Einfache Textausgabe in GRAFTEC	8-4
8.2.3 Textausgabe von Gross- und Kleinschrift in BLOCTEC	8-6
8.3 Ausgabe mehrerer unterschiedlicher Texte	8-7
8.3.1 Ausgabe von Texten in BLOCTEC	8-7
8.3.2 Ausgabe von Texten in GRAFTEC	8-9
8.4 Einzeltastenerkennung mit nachfolgender Aktion	8-13
8.4.1 Einzeltastenerkennung mit nachfolgender Aktion	8-14
8.4.2 Einzeltastenerkennung in BLOCTEC	8-17
8.4.3 Einzeltastenerkennung in GRAFTEC	8-20
8.5 Eingabe numerischer Werte	8-25
8.6 Funktionsblock INPUT	8-33
<b>9. Vergleich der wichtigsten Eigenschaften ..D202 ↔ ..D250</b>	
<b>10. Schnittstellen-Verbindungskabel RS 232</b>	



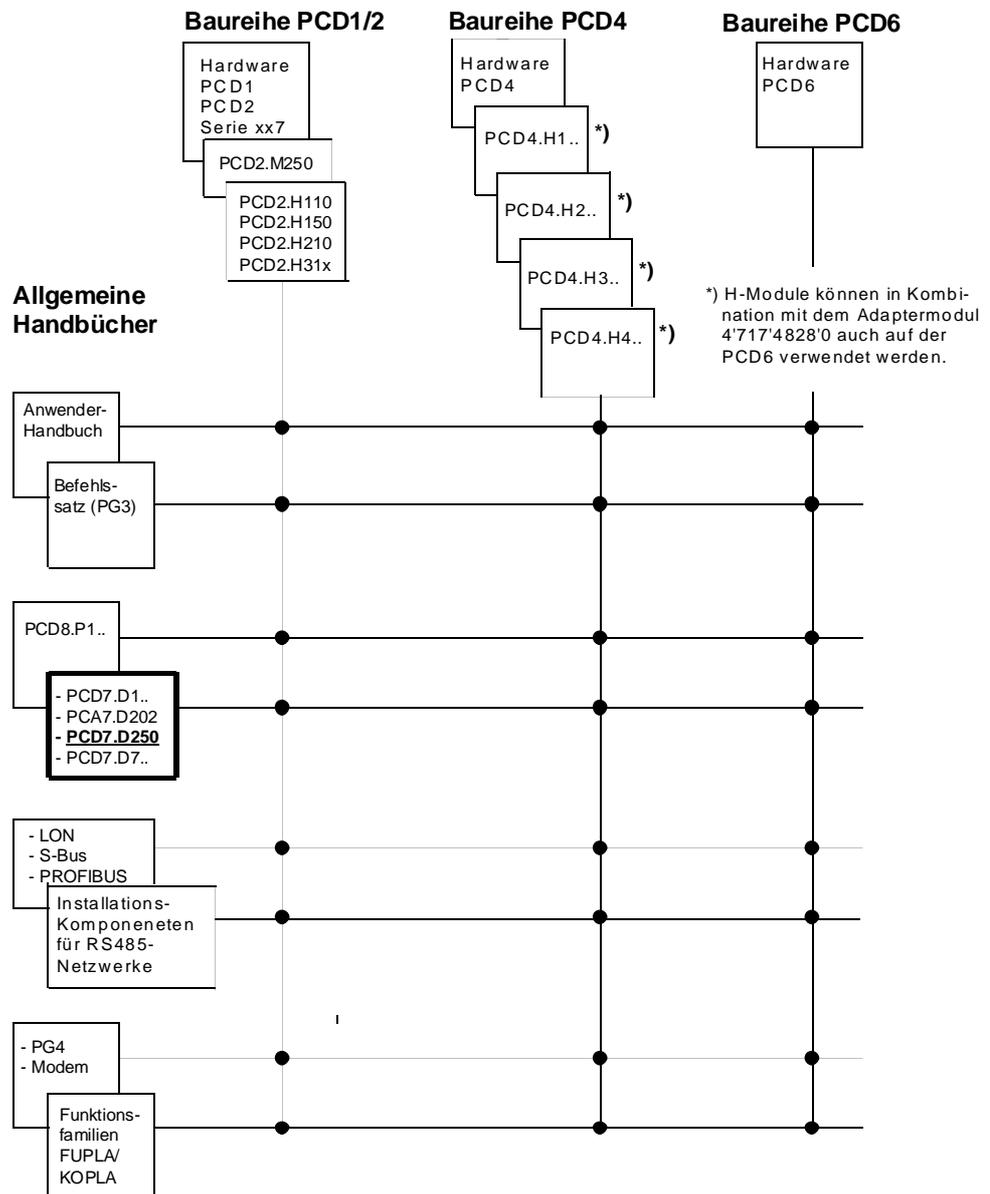
**Wichtiger Hinweis:**

Um den einwandfreien Betrieb von SAIA® PCD sicherstellen zu können, wurde eine Vielzahl detaillierter Handbücher geschaffen. Diese wenden sich an technisch qualifiziertes Personal, das nach Möglichkeit auch unsere Workshops erfolgreich absolviert hat.

Die vielfältigen Leistungen der SAIA® PCD treten nur dann optimal in Erscheinung, wenn alle in diesen Handbüchern aufgeführten Angaben und Richtlinien bezüglich Montage, Verkabelung, Programmierung und Inbetriebnahme genau befolgt werden.

Damit allerdings werden Sie zum grossen Kreis der begeisterten SAIA® PCD Anwendern gehören.

**Übersicht**



## Zuverlässigkeit und Sicherheit elektronischer Steuerungen

Die Firma SAIA-Burgess Electronics AG konzipiert, entwickelt und stellt ihre Produkte mit aller Sorgfalt her:

- Neuster Stand der Technik
- Einhaltung der Normen
- Zertifiziert nach ISO 9001
- Internationale Approbationen: z.B. Germanischer Lloyd, United Laboratories (UL), Det Norske Veritas, CE-Zeichen ...
- Auswahl qualitativ hochwertiger Bauelemente
- Kontrollen in verschiedenen Stufen der Fertigung
- In-Circuit-Tests

Die daraus resultierende hochstehende Qualität zeigt trotz aller Sorgfalt Grenzen. So ist z.B. mit natürlichen Ausfällen von Bauelementen zu rechnen. Für diese gibt die Firma SAIA-Burgess Electronics AG Garantie gemäss den "Allgemeinen Lieferbedingungen".

Der Anlagebauer seinerseits muss auch seinen Teil für das zuverlässige Arbeiten einer Anlage beitragen. So ist er dafür verantwortlich, dass die Steuerung datenkonform eingesetzt wird und keine Überbeanspruchungen, z.B. auf Temperaturbereiche, Überspannungen und Störfelder oder mechanischen Beanspruchungen auftreten.

Darüber hinaus ist der Anlagebauer auch dafür verantwortlich, dass ein fehlerhaftes Produkt in keinem Fall zu Verletzungen oder gar zum Tod von Personen bzw. zur Beschädigung oder Zerstörung von Sachen führen kann. Die einschlägigen Sicherheitsvorschriften sind in jedem Fall einzuhalten. Gefährliche Fehler müssen durch zusätzliche Massnahmen erkannt und hinsichtlich ihrer Auswirkung blockiert werden. So sind z.B. für die Sicherheit wichtige Ausgänge auf Eingänge zurückzuführen und softwaremässig zu überwachen. Es sind die Diagnoseelemente der PCD wie Watch-Dog, Ausnahme-Organisations-Blocks (XOB) sowie Test- und Diagnose-Befehle konsequent anzuwenden.

Werden alle diese Punkte berücksichtigt, verfügen Sie mit der SAIA®PCD über eine moderne und sichere programmierbare Steuerung, die Ihre Anlage über viele Jahre zuverlässig steuern, regeln und überwachen wird.

# 1. Anwendung

---

Die neuen Bedienterminal PCD7.D.. sind für den rauen Industrie-Einsatz konzipiert, wie er direkt an Produktionsmaschinen auftritt. In Kombination mit der intelligenten Textausgabe der SAIA® PCD lässt sich auf eine einfache Art eine Bedienerführung in Menü-Technik aufbauen.

Mit dem ..D250 stehen dem Anwender 2 verschiedene Schriftgrößen (für 8 x 40 bzw. 4 x 20 Charakter) zur Verfügung. Das ..D250 ist damit sowohl für die Gebäudeautomation als auch für die Industrie-Anwendung geeignet.

Anzeigen jeder Art über Temperatur, Druck, Stückzahl, Datum, Zeit usw. sowie Meldungen über Betriebszustände oder Alarmer lassen sich, dank neuester Technologie, über das extrem klare und hintergrundbeleuchtete LC-Display ausgeben.

Unter der abriebfesten Polyesterfolie befinden sich 29 Schnapptasten, welche es über die serielle Datenschnittstelle ermöglichen, geführt mit dem SAIA® PCD Menü, beliebige Betriebsdaten einzugeben oder Prozessfunktionen aufzurufen.

Unter der partiell durchsichtigen Frontfolie können Bezeichnungstreifen eingeschoben werden, was es dem Anwender auf eine einfache Art erlaubt, jederzeit seine individuelle Tastenbezeichnung zu realisieren und auch sein eigenes Label anzubringen.



**Schnellanleitung zur Handhabung des Terminals PCD7.D250**

In den folgenden Kapiteln sind die breiten Funktionsmöglichkeiten im Detail beschrieben, wie sie das Terminal ..D250 zur Verfügung stellt. In der praktischen Anwendung wird wohl nur ein kleiner Teil davon genutzt werden.

Um dem Einsteiger eine Kurzanleitung zur einfachen Ausgabe eines Textes zu geben, sei empfohlen, vor dem Einzelstudium aller Tests und Befehle ein Programmbeispiel aus Kapitel 8 praktisch zu vollziehen. Auf diese Weise wird sich zeigen, wie einfach das Arbeiten mit PCD und dem Terminal ..D250 in der Praxis ist.

## 2. Technische Daten

---

### Funktionsdaten

Anzeige	LCD-Display, mit LED-Hintergrundbeleuchtung 8 x 40 und 4 x 20 Zeichen Höhe 3.7 mm bzw. 7.5 mm, plus Cursor  Zeichensatz: ASCII-Zeichen 32 bis 127 plus sprachabhängige Zusatzzeichen und IBM Extended Character Set 437
Tastatur	Folientastatur mit Schnappfunktion Zahlenblock mit 12 Tasten, Abstände 18 mm Steuerblock mit 9 Tasten, Abstände 18 mm 8 Funktionstasten, Abstände 20 mm, mit roten LED-Anzeigen und einschiebbarem Beschriftungsstreifen
Datenschnittstelle	Kommunikations-Schnittstellen (zur SAIA® PCD) für Textübergabe und Steuerfunktionen COM 1: RS 232 (fix) COM 0: für Kommunikations-Module PCD7.F2.. für RS 422 oder CL 20 mA Übertragungsgeschwindigkeit: 110...9600 bps

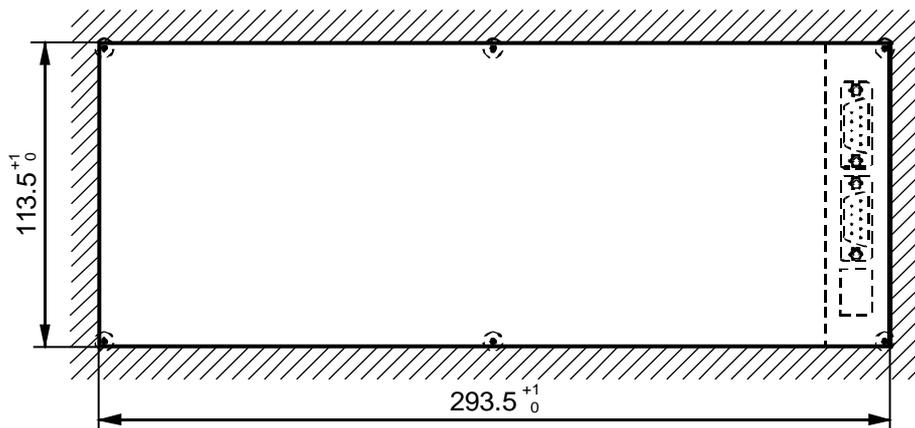
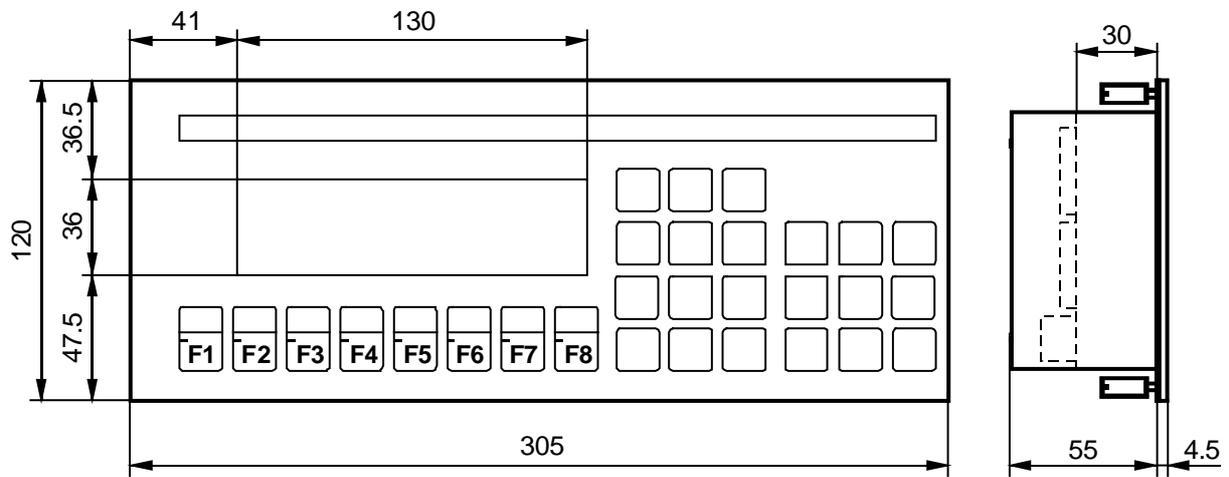
### Elektrische Daten

Speisespannung	24 VDC +30%/-20%, geglättet, mit Verpolungsschutz, oder  19 VAC ±15%, zweiweggleichgerichtet, mit Verpolungsschutz
Leistungsaufnahme	max. 320 mA bei 24 VDC
Anschluss	Stromversorgung über steckbare Schraubklemmen für Drähte von max. 2.5 mm <sup>2</sup>  Datenschnittstelle über 9-polige Sub-D-Buchse
EMV	Störemission: CE-Zeichen gemäss EN 50 081-1 Störimmunität: CE-Zeichen gemäss EN 50 082-2

**Allgemeine Daten**

Gehäuse	Alufrent mit Polyesterfolie, Schutzart frontseitig IP 65  Abdeckung rückseitig aus Alublech, Schutzart IP 30
Massbild	Abmessungen und Schalttafel Ausschnitt siehe Massbild Montage mit Schraubenbolzen
Umgebungstemperatur	Betrieb 0...50 °C Lagerung -25...+70 °C
Luftfeuchtigkeit	5...95% relative Feuchte ohne Betauung gemäss IEC 1131-2 bzw. DIN 40040, Klasse F
Mechanische Festigkeit	Vibration 10...57 Hz, 0.075 mm, bzw. 57...150 Hz, 1g, gemäss IEC 68-2-6

### 3. Massbild



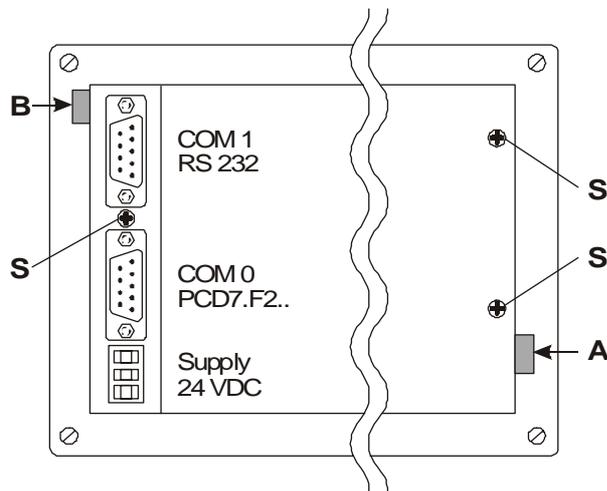
Einbauöffnung

Die Befestigung erfolgt mittels der mitgelieferten Schraubenbolzen

Notizen:

## 4. Hardware

### 4.1 Stromversorgung / Anschlüsse



Anschluss über die steckbare Schraubklemme für Drähte von max. 2.5 mm<sup>2</sup> (Litzen mit Kabelendhülsen bis 1.5 mm<sup>2</sup>).

24 VDC +30 % / -20 %, geglättet oder  
19 VAC +15 % zweiweggleichgerichtet,  
jeweils mit Verpolungsschutz.

Der Beschriftungsstreifen für die 8 Funktionstasten wird an der Stelle A eingeschoben, für ein anderes Label an der Stelle B.

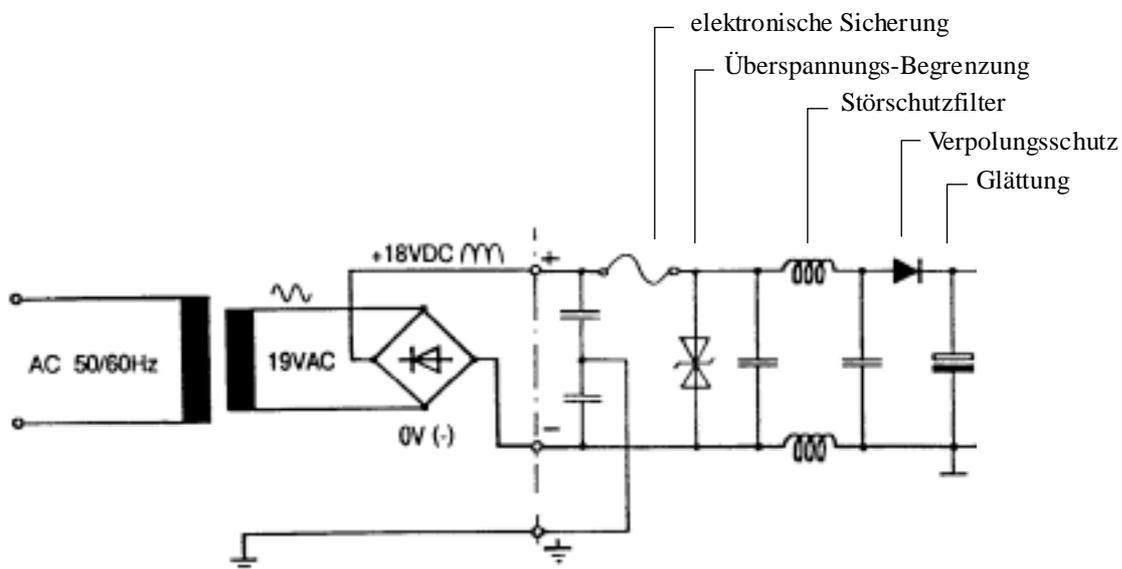


Eine gute Erdverbindung ist für den einwandfreien Betrieb unerlässlich! Aber auch die Deckelschrauben S müssen nach einem allfälligen Entfernen des Deckels kräftig festgeschraubt werden, damit wieder eine gute Masseverbindung hergestellt wird.

#### Stromversorgung mit zweiweggleichgerichteter Wechselspannung

##### Externe Speisung

##### PCD7.D250



## 4.2 Systemprogramm (Firmware)

---

Das Systemprogramm ist auf dem EPROM hinterlegt. Um allenfalls ein Update zu machen, ist der rückwärtige Deckel abzuheben und anschließend mit den 3 Schrauben wieder kräftig festzuschrauben.

### **4.3 Serielle Schnittstelle COM 1: RS 232 (Basisbestückung)**

9-polige D-Sub-Buchse (weiblich)

#### **Übertragungs-Modus**

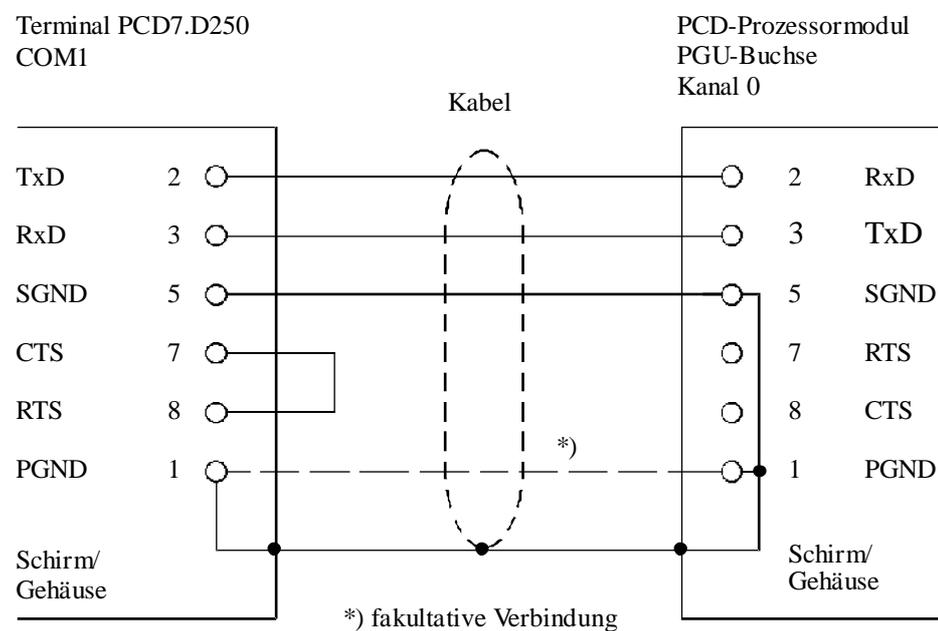
<b>1) Mit oder ohne HANDSHAKING</b>			
Übertragungsgeschwindigkeit	Typ	Handshaking	Überwachung
bis 9600 Baud	MC0	ohne	---
bis 9600 Baud	MC1	mit	RTS/CTS
bis 9600 Baud	MC2	mit	XON/XOFF
<b>2) Nur mit HANDSHAKING</b>			
Übertragungsgeschwindigkeit	Typ	Handshaking	Überwachung
19200 Baud	MC1	mit	RTS/CTS
19200 Baud	MC2	mit	XON/XOFF

### 4.3.1 Ohne Handshaking RTS/CTS oder mit Handshaking XON/XOFF

Hinweise gültig für alle Kommunikationskanäle der PCD:

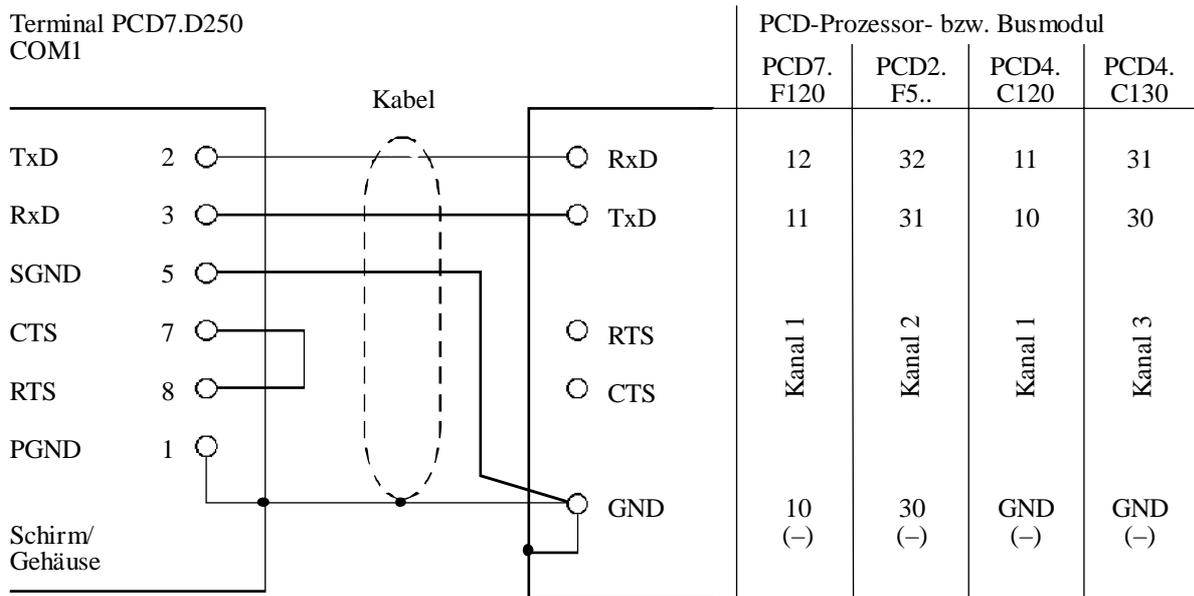
- Auf Terminal-Seite muss RTS mit CTS verbunden sein.
- Bei kleiner Auffrischrate (300...500 ms) kann bis 9600 Baud mit dem PCD-Kommunikationsmodus MC0 gearbeitet werden.
- Um ein Überlaufen des Eingangsbuffers zu vermeiden, wird jedoch empfohlen mit XON/XOFF-Handshaking (Modus MC2) zu arbeiten.

#### a) Terminal ..D250 zu PGU-Stecker der PCD



Für diese Verbindung kann das Kabel PCD7.K412 verwendet werden (siehe Kapitel 10).

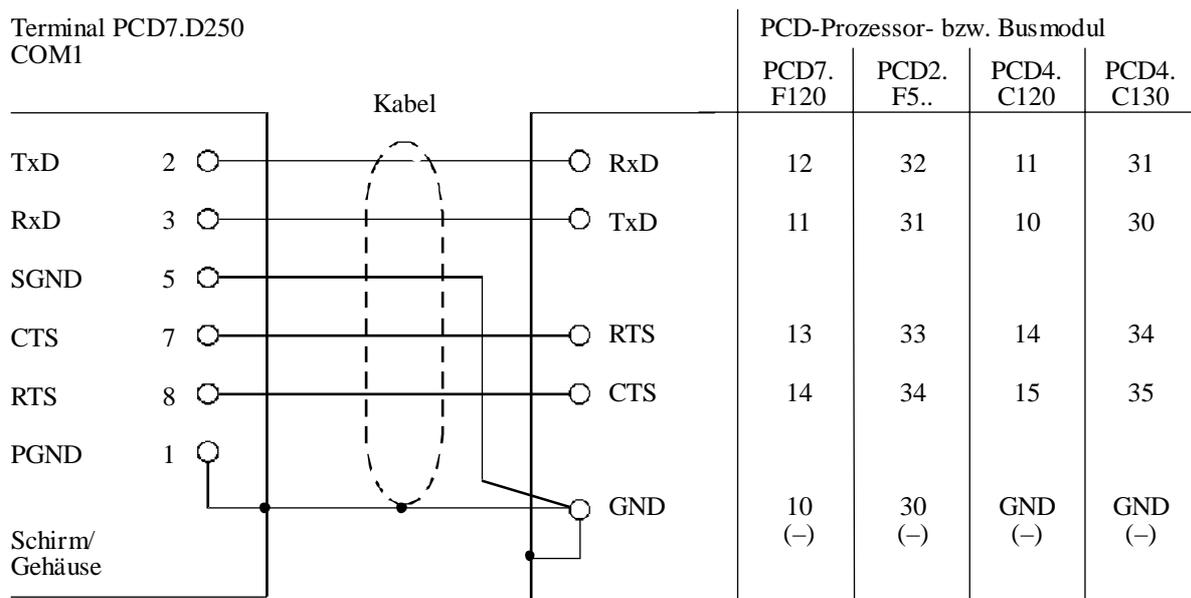
**b) Terminal ..D250 zu den PCD-Prozessoren Kanäle 1 bis 3**



Für diese Verbindung kann das Kabel PCD7.K422 verwendet werden (siehe Kapitel 10).

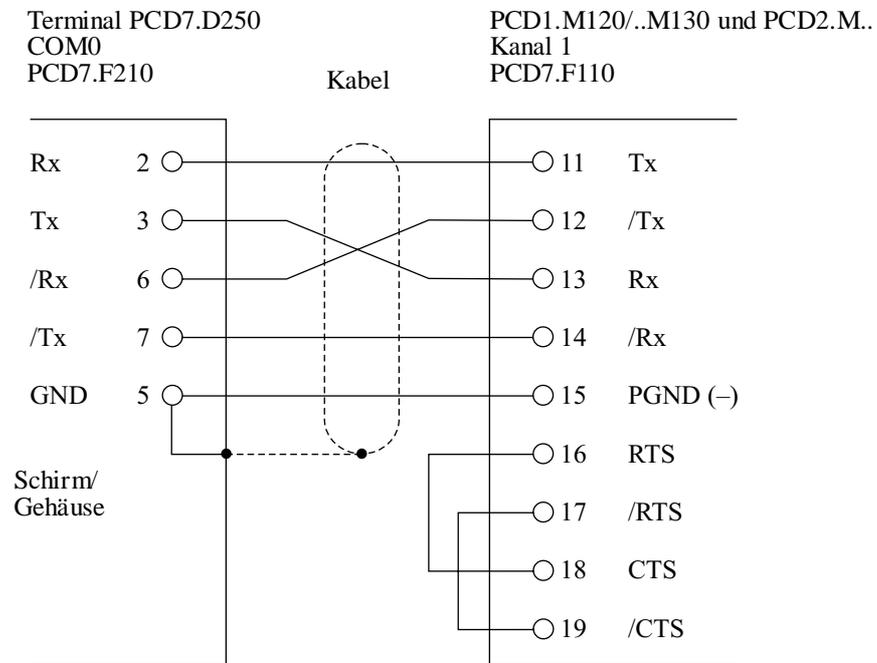
**4.3.2 Mit Handshaking RTS/CTS**

Der entsprechende Kommunikationskanal der PCD muss mit Modus MC1 assigniert werden.



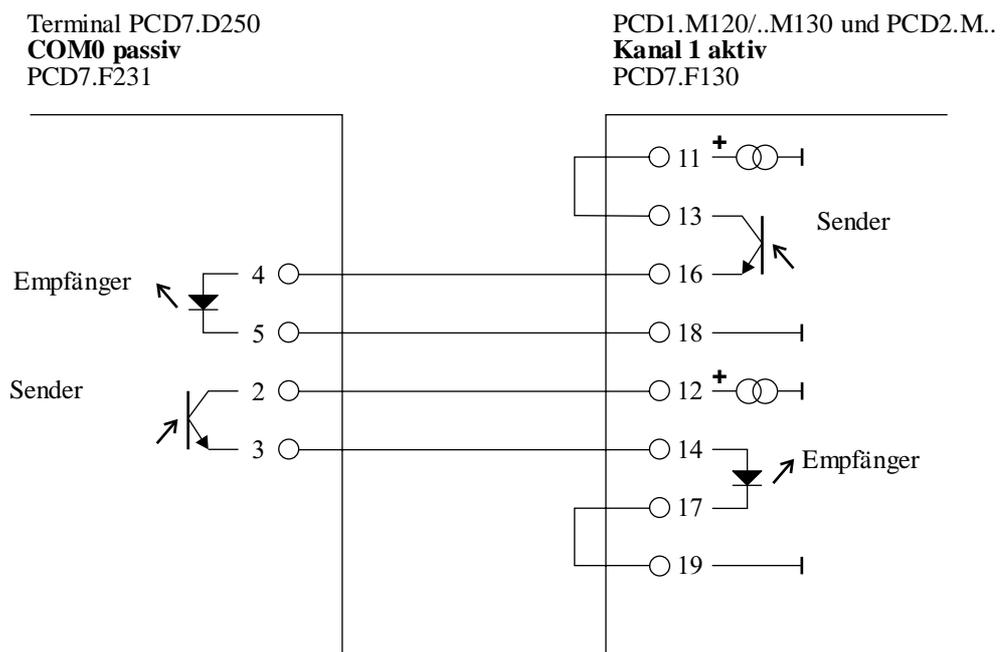
## 4.4 Serielle Schnittstelle COM 0: bestückt mit den Kommunikations-Modulen PCD7.F2..

### 4.4.1 RS 422 mit Kommunikationsmodul PCD7.F210



Jumper J1 geöffnet, Kommunikationsmodus MC0 oder MC2.

### 4.4.2 Stromschleife 20 mA (TTY) mit Kommunikationsmodul PCD7.F231

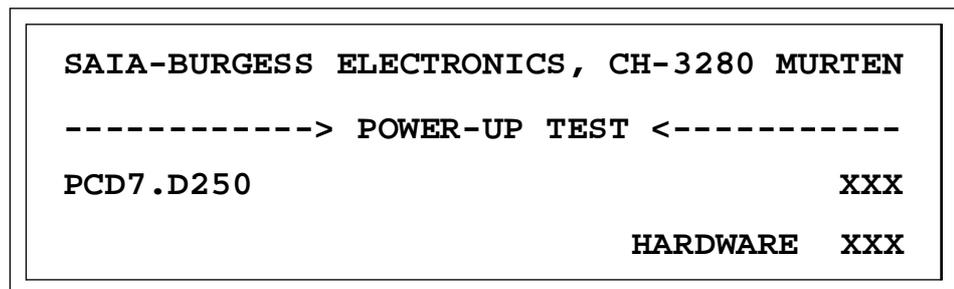


Kommunikationsmodus MC2 (XON/XOFF), bis 9600 Baud.

# 5. Funktion

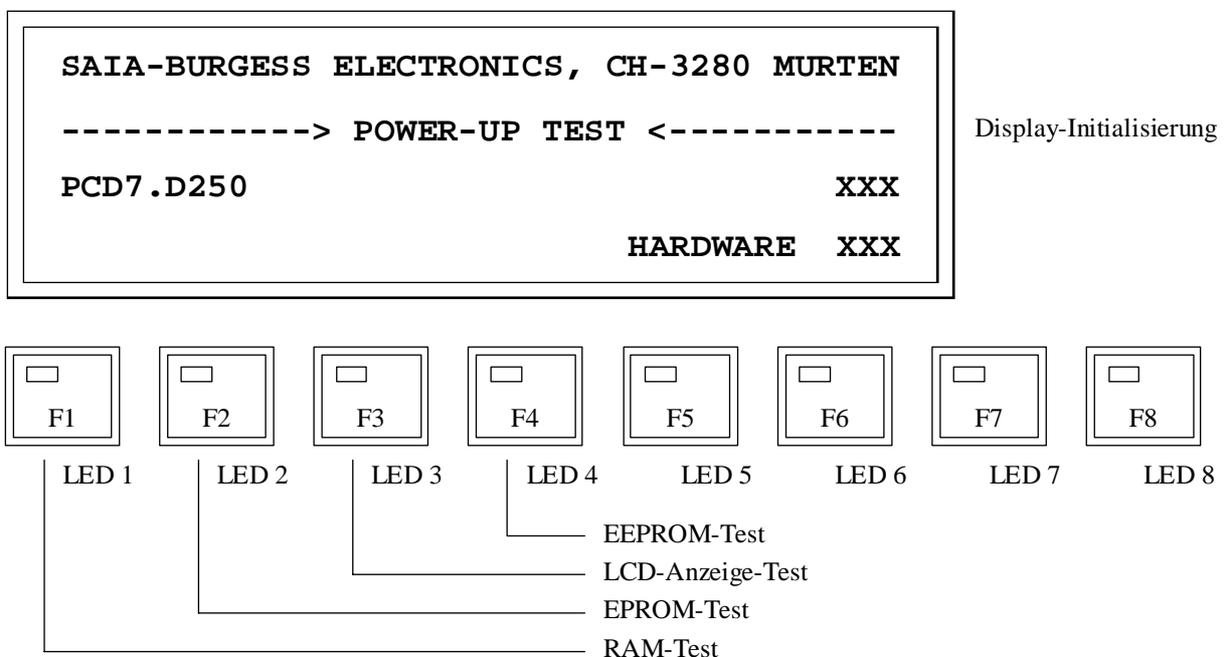
## 5.1 Einschaltvorgang

Nach dem Anlegen der Speisespannung führt das ..D250 einen Selbsttest durch. Während dieses Vorganges erscheint folgende Anzeige:



Der Test dauert ca. 3 Sekunden. Während dieser Zeit werden Befehle an der Datenschnittstelle ignoriert. Das Anwenderprogramm der PCD kann diese Zeit abwarten oder über den Befehl "POLL" (siehe Kapitel 6.5) die Bereitschaft des Terminals überprüfen.

Der Selbsttest gliedert sich in 6 Abschnitte, welche durch LED angezeigt werden:



Vorgängig der Einzeltests erfolgt der Test des Mikroprozessors, der durch kurzes Aufleuchten aller LEDs signalisiert wird.

Tritt ein Fehler auf, so wird dies im Display angezeigt. Falls der Mikroprozessor nicht arbeiten sollte, bleiben alle LEDs eingeschaltet.

## 5.2 Die Tastatur

Gleich bezeichnete Tasten sind beim ..D250 zum ..D202-Terminal kompatibel.



Folgende Codes werden ausgegeben:

Taste	Dez.	Hex	ASCII	Bemerkungen
F1	65	41	'A'	
F2	66	42	'B'	
F3	67	43	'C'	
F4	68	44	'D'	
F5	69	45	'E'	
F6	70	46	'F'	
F7	71	47	'G'	
F8	72	48	'H'	
0	48	30	'0'	
1	49	31	'1'	
2	50	32	'2'	
3	51	33	'3'	
4	52	34	'4'	
5	53	35	'5'	
6	54	36	'6'	
7	55	37	'7'	
8	56	38	'8'	
9	57	39	'9'	
+	43	2B	'+'	
-	45	2D	'-'	Shift + '+'
.	46	2E	'.'	
,	44	2C	','	Shift + '.'

Taste	Dez.	Hex	ASCII	Bemerkungen
i	105	69	'i'	Information
Quit	113	71	'q'	verlassen
Shift	-	-	-	keine Code-Ausgabe
Esc	27	1B	ESC	Escape
Enter	13	0D	CR	Carriage return (Enter)
↑	11	0B	VT	Pfeil nach oben
↓	5	05	ENQ	Pfeil nach unten
←	8	08	BS	Pfeil nach links
→	6	06	ACK	Pfeil nach rechts
Shift + F1	119	77	'w'	} Funktionstasten plus Shift-Taste
Shift + F2	120	78	'x'	
Shift + F3	121	79	'y'	
Shift + F4	122	7A	'z'	
Shift + F5	115	73	's'	
Shift + F6	116	74	't'	
Shift + F7	117	75	'u'	
Shift + F8	118	76	'v'	
Shift + 0	97	61	'a'	} Numerische Tasten plus Shift-Taste erzeugen Kleinbuchstaben von der ASCII-Tabelle
Shift + 1	98	62	'b'	
Shift + 2	99	63	'c'	
Shift + 3	100	64	'd'	
Shift + 4	101	65	'e'	
Shift + 5	102	66	'f'	
Shift + 6	103	67	'g'	
Shift + 7	104	68	'h'	
Shift + 8	106	6A	'j'	
Shift + 9	107	6B	'k'	
Shift + i	-	-	-	Keine Code-Ausgabe aber Wechsel zu "Setup/Test"-Modus
Shift + Quit bis Shift + →				} Gleicher Code wie ohne Shift

## 5.3 Setup/Test-Modus

Durch Betätigung der Tasten "**Shift + i**" wechselt das ..D250 in den Setup/Test-Modus und zwar unabhängig davon, ob das Terminal online oder offline betrieben wird. Alle Befehle, die an der Datenschnittstelle eintreffen, werden in diesem Modus ignoriert. Solange der Setup/Test-Modus aktiv ist, blinkt die LED Nr. 8.



Mit den Auf- und Ab-Pfeiltasten kann das entsprechende Menü gewählt werden.

Setup mode	Konfiguration des ..D250
Default setup	stellt die Konfiguration des ..D250 wieder auf Werks-Einstellung
Demo display	Demonstrations-Anzeige
Hardware tests	Hardware-Test
Display test	Test der Anzeige
Keyboard test	Test des Tastenfeldes
LED test	Test der LEDs

Mit Taste "**Enter**" werden die gewählten Setup-Parameter nullspannungssicher ins EEPROM geladen, bzw. der entsprechende Test wird ausgeführt. Verlassen des Setup/Test-Modus durch "**Quit**" oder "**Esc**".

Während des Setup/Test-Modus blinkt die LED Nr. 8.



**Achtung:** Solange sich das Terminal im Setup/Test-Modus befindet, soll über die serielle Schnittstelle nicht kommuniziert werden. Es besteht die Gefahr, dass Daten verloren gehen oder verändert werden, wenn kein Handshaking verwendet wird.

### 5.3.1 Setup-Modus

Er dient zur Festlegung der Terminal-Parameter. Diese werden im nullspannungssicheren EEPROM abgelegt. Die erste Anzeige enthält einen Hilfstext:

```

SETUP MODE
↑or↓ scrolls menu
←or→ changes data
[Ent] accepts, [Esc] aborts

```

Durch Betätigen einer beliebigen Taste gelangt man in das erste Menü:

```

SETUP MODE
Baudrate:
9600

```

Mit den Auf- und Ab-Pfeiltasten können die weiteren Menüs angewählt werden. Mit den Pfeiltasten links und rechts werden die gewünschten Parameter eingestellt.

Baudrate	110, 150, 300, 600, 1200, 2400, 4800, [9600], 19200
Data bits	[8], 7
Parity	[Even], Odd, None, Low
Stop bits	[1], 2
Handshaking	[None], RTS/CTS, XON/XOFF
Echo key to display	[No], Yes
Page/scroll mode	[Page], Scroll
Auto line feed	[No], Yes
Key auto-repeat	[No], All keys, All keys 2 speed, Arrow keys, Arrow keys 2 speed.
Character set	[CodePage 437], D100 compatible
Backlight	[On], Off
Contrast	0...15 [7]
Display mode	[8 x 40], 4 x 20
Serial port	[COM 1 (RS 232)], COM 0 (PCD7.F2..)

[ ] Die Werte in rechteckigen Klammern entsprechen der Werks-Einstellung, wie sie im System-EEPROM abgelegt sind.

Mit "**Enter**" werden die gewählten Parameter nullspannungssicher ins EEPROM geladen. Mit "**Quit**" oder "**Esc**" werden die neuen Parameter gelöscht, es bleiben die vorgängigen Parameter gültig. Die Einstellungen ab "Echo key to display" bis "Display mode" können vom Master her auch durch eine Escape-Sequenz verändert werden (siehe Kapitel 6.1).

**Kommunikations-Parameter**

(Baudrate, Daten-Bits, Parität und Stop-Bit)

Jeder übertragene Charakter besteht aus

- 1 Startbit
- 7 oder 8 Datenbits
- 1 Paritybit (oder keinem)
- 1 oder 2 Stopbits

Parität "High" kann auch durch folgende Einstellung simuliert werden:

Parität "None" mit 2 Stopbits entspricht Parität "High" mit 1 Stopbit.

**Handshaking**

[None]

Mit dem Handshaking kann der Datenfluss zwischen den Kommunikationspartnern gesteuert werden. Da das ..D250 über einen Empfangsbuffer von 512 Charaktern verfügt, ist ein Handshaking im allgemeinen nicht erforderlich.

Bei 19 200 Baud oder Kommunikation via Stromschleife 20 mA ist jedoch immer ein Handshaking (entweder mit RTS/CTS oder XON/XOFF) zu benützen.

RTS/CTS : Dieses Handshaking wird via Hardware über die entsprechenden Steuerleitungen bewirkt. Wird "None" gewählt, so sind am Terminal die Pins 8 und 7 (RTS/CTS) kurzzuschliessen (siehe Kapitel 4.3).

XON/XOFF : Dies sind die Software-Charakter, welche die gleiche Wirkung haben wie RTS/CTS, aber keine Steuerleitungen benötigen. Auch in diesem Fall sind am Terminal die Pins 8 und 7 kurzzuschliessen. Die PCD ist im Modus MC2 zu assignieren.

**Echo** (Echo key to display)

[No]

Bei "Yes" wird der an der Tastatur eingetippte Charakter sowohl gesendet als auch angezeigt. Ist dies erwünscht, so wird jedoch empfohlen, das Echo in der PCD durch den Modus MC3 zu erzeugen. Die PCD hat so die Möglichkeit, die Charakter vorgängig auf Gültigkeit zu prüfen.

**Page-/Scroll-Modus**

[Page]

**Page-Modus :** Befindet sich der Cursor auf der untersten Zeile und das Terminal erhält den Charakter LF, so springt er auf die oberste Zeile. Der Inhalt der Anzeige wird dabei nicht verändert.

**Scroll-Modus:** Befindet sich der Cursor auf der untersten Zeile und das Terminal erhält den Charakter LF, so wird die ganze Anzeige um eine Zeile nach oben gerollt. Die oberste Zeile verschwindet und die unterste Zeile wird leer. Der Cursor befindet sich dann auf der untersten Zeile in der gleichen Spalte.

**Automatischer Zeilenvorschub** (Auto line feed)

[No]

Bei automatischem Zeilenvorschub macht das Display nach Empfang eines CR automatisch ein LF (Zeilenvorschub).

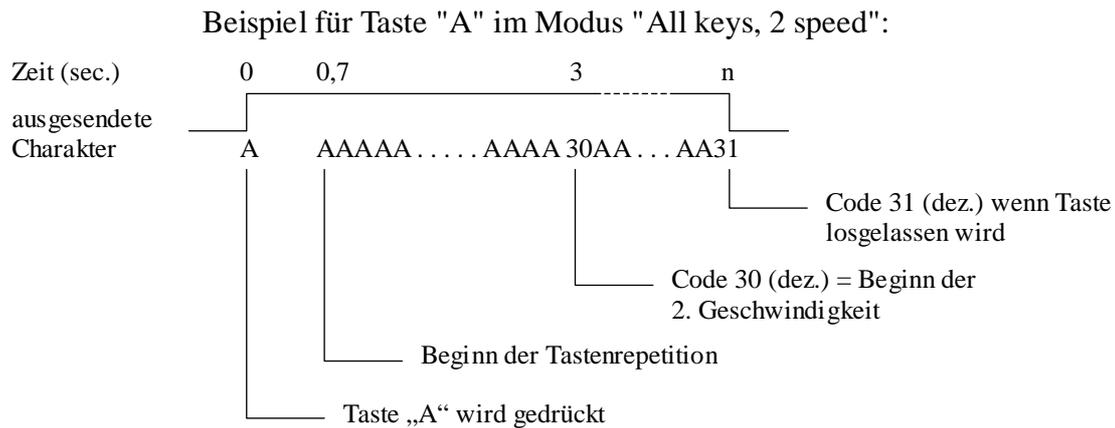
**Automatische Tastenrepetition** (Key auto-repeat)

[No]

Wird in diesem Modus eine Taste länger als 0.7 sec. betätigt, so wird sie automatisch repetiert und zwar mit einer Frequenz von 8 Zeichen pro Sekunde. In diesem Modus können folgende Varianten gewählt werden:

No	Keine Tastenrepetition (Werks-Einstellung)
All keys	Alle Tasten repetieren
All keys "2 speed"	Alle Tasten repetieren mit Signalisation "2 speed"
Arrow keys	Nur die Pfeiltasten repetieren
Arrow keys "2 speed"	Nur die Pfeiltasten repetieren mit Signalisation "2 speed"

Die "2 speed"-Funktion kann vom Master (z.B. einer PCD) so ausgewertet werden, dass nach einer gewissen Zeit z.B. die Inkrementations-Geschwindigkeit eines Wertes erhöht wird. Auf diese Weise können z.B. mit den Pfeiltasten grosse Veränderungen von Werten schneller erreicht werden. Wie das nachstehende Zeitdiagramm zeigt, repetiert z.B. der Charakter "A" nach 0.7 sec. mit einer Frequenz von 8 Zeichen pro Sekunde. Bleibt die Taste gedrückt, so wird nach total 3 sec. ein Steuercharakter (dez. 30) gesendet, der im Master z.B. zur Erhöhung einer Zählgeschwindigkeit ausgewertet wird. Der Charakter "A" wird weiterhin solange repetiert, bis die Taste losgelassen wird. In diesem Moment wird der Steuercharakter dez. 31 ausgegeben, der das Ende der "2 speed"-Phase anzeigt.



**Charakter-Modi 8 x 40 (4 x 20 off) oder 4 x 20** [8 x 40]

Im Setup kann nur entweder 8 x 40 (default) oder 4 x 20 gewählt werden. Eine Überlagerung beider Modi ist durch Steuerung via serielle Schnittstelle möglich (siehe Kapitel 6.2)

**Charakter Satz** (Character set) [CodePage 437]

Im Modus 8 x 40 oder 4 x 20 sind je 2 Charakter-Sätze verfügbar. Jeder Satz benützt die gleichen Charakter 32 bis 127 dez. Unterschiede liegen in den folgenden Charaktern 128 bis 255 dez. (siehe Tabellen in Kapitel 7).

**Hintergrundbeleuchtung** (Backlight) [On]

Normalerweise ist die Beleuchtung eingeschaltet. Um hohe Aufmerksamkeit zu erreichen (z.B. Alarm), kann die Beleuchtung durch entsprechende Escape-Sequenz auch blinken (siehe Kapitel 6.3).

**Anzeige-Kontrast** (Contrast) [7]

Der Anzeige-Kontrast kann in 16 Schritten 0...15 optimiert werden. 15 ist die dunkelste Einstellung.

**Kommunikationskanal** [COM1]

COM 1 ist fix mit RS 232 bestückt. Alternativ kann COM 0 mit einem der Module PCD7.F2.. bestückt werden. Beide Kanäle können nicht gleichzeitig betrieben werden.

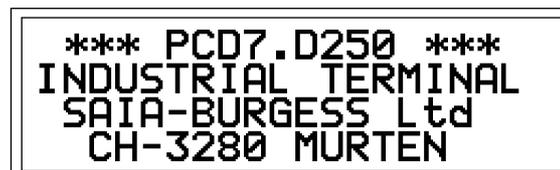
### 5.3.2 Werks-Einstellung der Setup-Parameter

Diese sind im System-EPROM hinterlegt und werden beim Einschaltvorgang ins nullspannungssichere EEPROM übertragen. Sie lauten:

Baudrate	9600
Data bits	8
Parity	Even
Stop bits	1
Echo key to display	No
Handshaking	None
Page/scroll mode	Page
Auto line feed	No
Key auto-repeat	No
Character set	CodePage 437
Backlight	On
Contrast	7 (medium)
Display mode	8 x 40
Serial port	COM 1 (RS 232)

### 5.3.3 Demonstrations-Anzeige (Demo display)

Es ist eine ruhende offline-Anzeige zu Demo-Zwecken. Mit Shift + F4 wird die Anzeige verlassen.



### 5.3.4 Hardware-Test (Hardware tests)

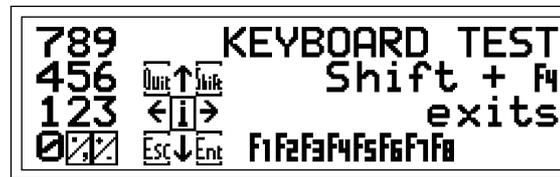
In diesem Menü läuft der gleiche Hardware-Test ab wie beim Einschaltvorgang, jedoch ununterbrochen. Damit lassen sich allfällige Fehler gut lokalisieren. Um den Testlauf zu verlassen, **muss die Speisespannung unterbrochen werden.**

### 5.3.5 Anzeige-Test (Display test)

Dies ist ein umfassender Test der LCD-Anzeige, des Charakter-Satzes, des LCD-Controllers sowie des internen RAM's. Durch Betätigen einer **beliebigen Taste** wird der Test verlassen.

### 5.3.6 Tastatur-Test (Keyboard test)

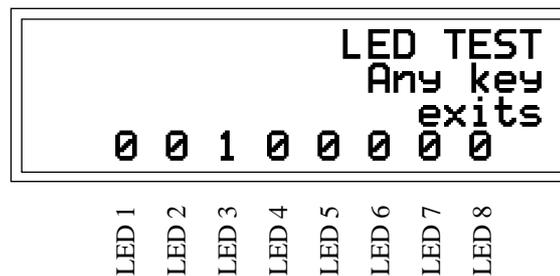
Die Anzeige gibt auf der linken Seite und unten die einzelnen Tasten wieder (in gleicher Anordnung wie auf der Tastatur). Bei Betätigung einer Taste wird das entsprechende Feld schwarz.



Mit **Shift + F4** kann der Test verlassen werden.

### 5.3.7 LED-Test (LED test)

Dieser Test erlaubt die individuelle Überprüfung jeder einzelnen LED. Mit einer Sequenz von 500 ms wird jede LED der Reihe nach aktiviert. Auf der Anzeige erscheint die zugehörige "1".



Mit Betätigung einer **beliebigen Taste** wird dieser Test verlassen.

## 6. Steuerbefehle über die serielle Schnittstelle

Einzelne Steuer-Charakter oder Escape-Sequenzen, bestehend aus 2 bis 4 Charaktern, bewirken im ..D250 verschiedene Funktionen. Diese Charakter können von der PCD unter Verwendung der PCD-Befehle STXD oder STXT gesendet werden.



**Achtung:** Einige Escape-Sequenzen verwenden den Charakter "@". Die PCD interpretiert dieses Zeichen im Modus C jedoch als Steuerzeichen.

Damit die PCD ein einzelnes "@" ausgibt, muss in ihrem Text "@@" geschrieben werden.

## 6.1 Konfigurierungsbefehle

---

Die Konfigurierung aus dem Setup-Modus kann durch die nachfolgenden Sequenzen modifiziert werden. Die Änderung ist jedoch nicht nullspannungssicher. Nach Wegnahme der Speisespannung ist wieder die ursprüngliche Konfiguration gemäss Setup-Modus gültig (ab EEPROM).

### Echo (Echo key to display)

Details siehe Kapitel 5.3.1

Befehl	ASCII	Dez .	Hex
Echo aus	ESC @ 0	27 64 48	1B 40 30
Echo ein	ESC @ 1	27 64 49	1B 40 31

### Page-/Scroll-Modus

Details siehe Kapitel 5.3.1

Befehl	ASCII	Dez .	Hex
Scroll-Modus	ESC @ 4	27 64 52	1B 40 34
Page-Modus	ESC @ 5	27 64 53	1B 40 35

### Automatischer Zeilenvorschub

Details siehe Kapitel 5.3.1

Befehl	ASCII	Dez .	Hex
Auto Zeilen- vorschub ein	ESC @ 2	27 64 50	1B 40 32
Auto Zeilen- vorschub aus	ESC @ 3	27 64 51	1B 40 33

### Automatische Tastenrepetition

Details siehe Kapitel 5.3.1

Befehl	ASCII	Dez .	Hex
Automat. Rep. aus	ESC A	27 65	1B 41
Automat. Rep. ein			
• alle Tasten	ESC B	27 66	1B 42
• nur Pfeiltasten	ESC C	27 67	1B 43
• alle Tasten"2 speed"	ESC D	27 68	1B 44
• Pfeiltasten"2 speed"	ESC E	27 69	1B 45

### Charakter-Modi

Per Default ist der Modus 8 x 40 gewählt. Mittels des angegebenen Steuerbefehls kann auf 4 x 20 umgeschaltet werden. Damit wird jedoch eine neue Bildschirmseite eröffnet.

Beide Bildschirmseiten bleiben jedoch in beiden Modi im Terminal gespeichert. Sie können mit dem Befehl "Transparent-Modus ein" überlagert angezeigt werden (siehe Beispiele).

Befehl	ASCII	Dez .	Hex
Select Modus 8 x 40 (default)	ESC @ M0	27 64 77 48	1B 40 4D 30
Select Modus 4 x 20	ESC @ M1	27 64 77 49	1B 40 4D 31

## Charakter-Satz

Zwei Charakter-Sätze sind wählbar. Details siehe Tabellen im Kapitel 7.

Befehl	ASCII	Dez .	Hex
CodePage 437 (default)	ESC @ J	27 64 74	1B 40 4A
D100-kompatibel	ESC @ F	27 64 70	1B 40 46

## Transparent-Modus

Mit diesem Modus lassen sich zwei Bildschirmseiten in Superposition (überlagert) zur Anzeige bringen. Damit können kleine (8 x 40) und grosse Charakter (4 x 20) auf dem gleichen Display dargestellt werden.



Befehl	ASCII	Dez .	Hex
Transparent-Modus aus	ESC @ M4	27 64 77 52	1B 40 4D 34
Transparent-Modus ein	ESC @ M5	27 64 77 53	1B 40 4D 35

Bezüglich inverser Darstellung siehe Kapitel 6.3.

## Hintergrundbeleuchtung und Kontrast

Details siehe Kapitel 6.3

## 6.2 Cursor-Steuerung

### Cursor nach oben, unten, links, rechts

Durch Senden eines einzelnen Charakters kann der Cursor nach oben, unten, links oder rechts verschoben werden.

Wird der Cursor ausserhalb des Anzeigefeldes befohlen, so kommt er automatisch auf der Gegenseite wieder zum Vorschein.

Befehl	ASCII	Dez .	Hex
Cursor nach oben	CTRL+K	11	0B
Cursor nach unten	CTRL+E	5	05
Cursor nach links	CTRL+H	8	08
Cursor nach rechts	CTRL+F	6	06

### Cursor-Positionierung

Diese Funktion benötigt neben dem Cursor-Adressierungscode (16 dez. bzw. 10 hex) die X-Adresse und die Y-Adresse zur Positionierung des Cursors.

Beide Adressen müssen mit einem Offset von 32 dez. bzw. 20 hex versehen werden. Wenn eine der beiden Adressen fehlerhaft ist, wird die Cursor-Position nicht verändert.

Die Adressbereiche für die Modi 8 x 40 und 4 x 20 sind entsprechend verschieden:

Code X \ Y Code	<32>	<33>	<34>	<35>	<36>	<37>	<38>	<39>	<40>	<41>	<42>	<43>	<44>	<45>	<46>	<47>	<48>	<49>	<50>	<51>	<52>	<53>	<54>	<55>	<56>	...	<70>	<71>
<32>	1	X																								...		
<33>	2			X																						...		
<34>	3																									...		
<35>	4														X											...		
<36>	5																									...		
<37>	6																									...	X	
<38>	7																									...		
<39>	8																									...		

Beispiele (Reihenfolge: **16 dez., Code X, Code Y**):

Cursorposition	Befehlsfolge Dezimal	Befehlsfolge Hex
1. Spalte, 1. Zeile	16 32 32	10 20 20
4. Spalte, 2. Zeile	16 35 33	10 23 21
16. Spalte, 4. Zeile	16 47 35	10 2F 23
39. Spalte, 6. Zeile	16 70 37	10 46 25

Hinweis: Um den Cursor in die x-Position 36 (entspricht ASCII \$), zu plazieren, muss im PCD-Text eingegeben werden: <36><36>.

Beispiel: TEXT xxxx "... 16 36 36 34 ..."  
x-pos y-pos

Werden beide Charaktergrößen auf dem gleichen Display verwendet, so ist deren Positionierung unterschiedlich. Die nachstehenden Bilder sollen helfen, die jeweils richtige Position zu finden.

(Das Positionierfeld für 4 x 20 Charakter ist kompatibel zum ..D202.)

**Positionierfeld für 8 x 40 Charakter**

Y Code	Code X	<32>	<33>	<34>	<35>	<36>	<37>	<38>	<39>	<40>	<41>	<42>	<43>	<44>	<45>	<46>	<47>	<48>	<49>	<50>	<51>	<52>	<53>	<54>	<55>	<56>	<57>	<58>	<59>	<60>	<61>	<62>	<63>	<64>	<65>	<66>	<67>	<68>	<69>	<70>	<71>							
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40							
<32>	1	X																																														
<33>	2			X																																												
<34>	3																																															
<35>	4															X																																
<36>	5																																															
<37>	6																																															X
<38>	7																																															
<39>	8																																															

**Positionierfeld für 4 x 20 Charakter**

Y Code	Code X	<32>	<33>	<34>	<35>	<36>	<37>	<38>	<39>	<40>	<41>	<42>	<43>	<44>	<45>	<46>	<47>	<48>	<49>	<50>	<51>	
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
<32>	1	X																				
<33>	2				X																	
<34>	3																					
<35>	4																	X				

**Cursor home**

Der Cursor wird auf die erste Spalte in der ersten Zeile positioniert: Home-Position. Der Anzeigehalt bleibt unverändert.

Befehl	ASCII	Dez .	Hex
Cursor home	CTRL+Z	26	1A

**Cursor ein/aus**

Dieser Zweicharakter-Befehl schaltet den Cursor ein oder aus.

Befehl	ASCII	Dez .	Hex
Cursor ein	ESC W	27 87	1B 57
Cursor aus	ESC T	27 84	1B 54

**Zeilenvorschub**

Der Cursor bewegt sich eine Zeile nach unten. Befindet sich dieser sich auf der untersten Zeile, wird die Anzeige gerollt (Scroll-Modus) oder der Cursor springt auf die oberste Zeile (Page-Modus). Die Spalten-Position bleibt dabei unverändert.

Befehl	ASCII	Dez .	Hex
Zeilenvorschub	LF	10	0A

**Return**

Der Cursor bewegt sich zurück auf die erste Spaltenposition in der momentanen Zeile. Ist der automatische Zeilenvorschub aktiviert, wird der Cursor auf den Zeilenanfang der nächsten Zeile positioniert.

Befehl	ASCII	Dez .	Hex
Return	CR	13	0D

**Lösche Charakter**

Der entsprechende Charakter links vom Cursor wird gelöscht (durch ein Space ersetzt). Der Cursor wird gleichzeitig um eine Stelle nach links verschoben. Vom Anfang einer Zeile wird der Cursor ans Ende der vorangehenden Zeile verschoben. In der Home-Position ist dieser Befehl unwirksam.

Befehl	ASCII	Dez .	Hex
Lösche Charakter	DEL	127	7F

## 6.3 Anzeige-Steuerung

---

### Anzeige löschen

Die gesamte Anzeige wird gelöscht (durch Spaces ersetzt). Der Cursor geht in die Home-Position.

Befehl	ASCII	Dez .	Hex
Anzeige löschen	CTRL+L	12	0C

### Speichern und Wiedergeben einer Anzeige

Es können 10 Bildschirminhalte (nummeriert von 0 bis 9), inkl. Cursor-Position, gespeichert und wieder aufgerufen werden. Wird ein Bildschirminhalt aufgerufen, der vorgängig nicht gespeichert wurde, erscheint ein leerer Bildschirm mit Cursor. Die gespeicherten Bildschirminhalte gehen beim Ausschalten des ..D250 verloren.

Befehl	ASCII	Dez .	Hex
Anzeige "n" speichern	ESC @ S n	27 64 83 n	1B 40 53 n
Anzeige "n" wiedergeben	ESC @ R n	27 64 82 n	1B 40 52 n
für "n" =	0...9	48..57	30...39

### Hintergrundbeleuchtung aus/ein

Normalerweise ist die Beleuchtung eingeschaltet. Um hohe Aufmerksamkeit zu erreichen (z.B. Alarm) kann die Beleuchtung auch blinken (alternierend aus/ein).

Befehl	ASCII	Dez .	Hex
Beleuchtung aus	ESC O	27 79	1B 4F
Beleuchtung ein	ESC L	27 76	1B 4C

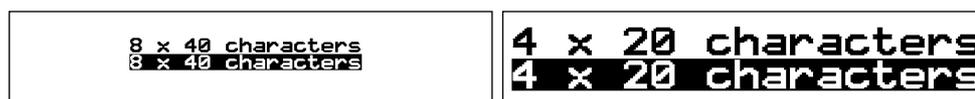
### Anzeige-Kontrast

Der Anzeige-Kontrast des LCD-Displays kann in 16 Schritten verändert werden (von 0 bis F).

Befehl	ASCII	Dez .	Hex
Hellste Einstellung	ESC @ D 0	27 64 68 48	1B 40 44 30
Mittlere Einstellung	ESC @ D 7	27 64 68 55	1B 40 44 37
Dunkelste Einstellung	ESC @ D F	27 64 68 70	1B 40 44 46

### Invertierte Charakter

Standardmässig werden die Charakter in schwarz auf hellem Grund dargestellt. Mit den nachfolgenden Befehlen kann die Darstellung (im laufenden Text) invertiert und wieder aufgehoben werden.



Weitere Möglichkeit:

Grosse-Charakter (4 x 20) positiv und invertiert sowie überlagert mit kleinen Charaktern (8 x 40) mit dem Befehl "Transparent-Modus", siehe dazu Kapitel 6.1.



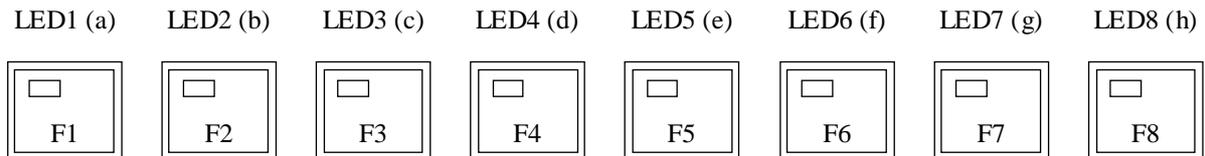
Befehl	ASCII	Dez .	Hex
Invertieren aus	ESC @ N 0	27 64 78 48	1B 40 4E 30
Invertieren ein	ESC @ N 1	27 64 78 49	1B 40 4E 31

**Einschränkung:** Im "Transparent-Modus" kann der Befehl für das Invertieren der Anzeige der Charakter 8 x 40 nicht aktiviert werden.

## 6.4 LED-Steuerung

---

Die 8 LEDs des ..D250 können durch eine 4-stellige Escape-Sequenz individuell aktiviert werden. Für LED1 wird der ASCII-Charakter "a", für LED8 der Charakter "h" verwendet. Mit "1" wird die LED ein-, mit "0" ausgeschaltet.



Befehl	ASCII	Dez .	Hex
LED1 einschalten	ESC @ a 1	27 64 97 49	1B 40 61 31
LED1 ausschalten	ESC @ a 0	27 64 97 48	1B 40 61 30
LED2 einschalten	ESC @ b 1	27 64 98 49	1B 40 62 31
LED2 ausschalten	ESC @ b 0	27 64 98 48	1B 40 62 30
LED3 einschalten	ESC @ c 1	27 64 99 49	1B 40 63 31
LED3 ausschalten	ESC @ c 0	27 64 99 48	1B 40 63 30
LED4 einschalten	ESC @ d 1	27 64 100 49	1B 40 64 31
LED4 ausschalten	ESC @ d 0	27 64 100 48	1B 40 64 30
LED5 einschalten	ESC @ e 1	27 64 101 49	1B 40 65 31
LED5 ausschalten	ESC @ e 0	27 64 101 48	1B 40 65 30
LED6 einschalten	ESC @ f 1	27 64 102 49	1B 40 66 31
LED6 ausschalten	ESC @ f 0	27 64 102 48	1B 40 66 30
LED7 einschalten	ESC @ g 1	27 64 103 49	1B 40 67 31
LED7 ausschalten	ESC @ g 0	27 64 103 48	1B 40 67 30
LED8 einschalten	ESC @ h 1	27 64 104 49	1B 40 68 31
LED8 ausschalten	ESC @ h 0	27 64 104 48	1B 40 68 30

## 6.5 Sonstige Befehle

---

### Tastatur sperren/freigeben

Nach Eingabe des Befehls "Tastatur sperren" gibt das ..D250 bei Betätigung der Tastatur keinen Charakter mehr aus.

Befehl	ASCII	Dez .	Hex
Tastatur sperren	ESC N	27 78	1B 4E
Tastatur freigeben	ESC Q	27 81	1B 51

### Restart warm/kalt

Mit "Restart warm" werden die vom Anwender mit der Tastatur eingegebenen Setup-Parameter (aus dem EEPROM) wieder aktiviert (wie beim normalen Einschalt-Vorgang).

Mit "Restart kalt" werden die Werkseinstellungen der Setup-Parameter (aus System EPROM) aktiviert (siehe Kapitel 5.3.2).

Befehl	ASCII	Dez .	Hex
Restart warm	ESC H	27 72	1B 48
Restart kalt	ESC @ G	27 64 71	1B 40 47

### Setup-Modus sperren/freigeben

Mit diesem Befehl kann verhindert werden, dass während des Betriebes der Setup unautorisiert verändert werden kann. Der Setup-Zugriff mit "Shift + i" ist nach erneuter Freigabe mit untenstehendem Befehl "Setup/Test-Modus freigeben" oder nach einem Befehl "Restart" oder auch, wenn die Speisung des Terminals aus- und wieder eingeschaltet wird, wieder möglich.

Befehl	ASCII	Dez .	Hex
Setup/Test-Modus sperren	ESC @ H	27 64 72	1B 40 48
Setup/Test-Modus freigeben	ESC @ I	27 64 73	1B 40 49

### Demonstrations-Anzeige und Hardware-Test

Diese Befehle führen die gleichen Funktionen, wie dies in den Kapiteln 5.3.3 bis 5.3.7 bereits beschrieben wurde, aus. Mit Befehl "POLL" kann festgestellt werden, wann der jeweilige Test beendet ist.

Befehl	ASCII	Dez .	Hex
Demonstrations- Anzeige	ESC J	27 74	1B 4A
Anzeige-Test	ESC @ A	27 64 65	1B 40 41
Tastatur-Test	ESC @ 9	27 64 57	1B 40 39
LED-Test	ESC @ L	27 64 76	1B 40 4C
Hardware-Test	ESC @ C	27 64 67	1B 40 43

### POLL

Mit dem Befehl "POLL" kann die PCD überprüfen, ob das Terminal richtig angeschlossen und funktionsbereit ist. Ist das ..D250 betriebsbereit, so antwortet es auf diesen Befehl mit "**SOH**" (1 dez., 01 hex). Wenn das ..D250 nicht betriebsbereit ist, so erfolgt keine Rückantwort. Damit kann die Steuerung jederzeit die Betriebsbereitschaft (auch beim Einschaltvorgang) überprüfen und im negativen Fall signalisieren.

Befehl	ASCII	Dez .	Hex
POLL	ESC @ B	27 64 66	1B 40 42
Antwort von ..D250 wenn betriebsbereit	SOH	1	01

## 6.6 Übersicht aller Steuerfunktionen

---

Befehl	ASCII	Dez .	Hex
Konfigurierung:			
Echo aus	ESC @ 0	27 64 48	1B 40 30
Echo ein	ESC @ 1	27 64 49	1B 40 31
Auto Zeilenvorschub ein	ESC @ 2	27 64 50	1B 40 32
Auto Zeilenvorschub aus	ESC @ 3	27 64 51	1B 40 33
Scroll-Modus	ESC @ 4	27 64 52	1B 40 34
Page-Modus	ESC @ 5	27 64 53	1B 40 35
Select Modus 8 x 40	ESC @ M0	27 64 77 48	1B 40 4D 30
Select Modus 4 x 20	ESC @ M1	27 64 77 49	1B 40 4D 31
Transparent-Modus aus	ESC @ M4	27 64 77 52	1B 40 4D 34
Transparent-Modus ein	ESC @ M5	27 64 77 53	1B 40 4D 35
CodePage 437	ESC @ J	27 64 74	1B 40 4A
D100-kompatibel	ESC @ F	27 64 70	1B 40 46
Automat. Tastenrep. aus	ESC A	27 65	1B 41
Automat. Tastenrep. ein			
• alle Tasten	ESC B	27 66	1B 42
• nur Pfeiltasten	ESC C	27 67	1B 43
• alle Tasten"2 speed"	ESC D	27 68	1B 44
• Pfeiltasten"2 speed"	ESC E	27 69	1B 45
Cursor-Steuerung:			
Cursor nach oben	CTRL+K	11	0B
Cursor nach unten	CTRL+E	5	05
Cursor nach links	CTRL+H	8	08
Cursor nach rechts	CTRL+F	6	06
Cursor home	CTRL+Z	26	1A
Cursor ein	ESC W	27 87	1B 57
Cursor aus	ESC T	27 84	1B 54
Zeilenvorschub	LF	10	0A
Return	CR	13	0D
Lösche Charakter	DEL	127	7F
Cursor Positionierung	ASCII	CTRL+P ' '+X	' '+Y
	Dezimal	16 32+X 32+Y	
	Hex	10 20+X 20+Y	

**Wichtig:** Damit die PCD-Steuerung den Charakter "@" ausgibt, muss im Befehltext "@@" programmiert werden.

## Anzeige-Steuerung:

Anzeige löschen	CTRL+L	12	0C
Anzeige"n" speichern	ESC @ S n	27 64 83 n	1B 40 53 n
Anzeige"n" wiedergeben	ESC @ R n	27 64 82 n	1B 40 52 n
Beleuchtung aus	ESC O	27 79	1B 4F
Beleuchtung ein	ESC L	27 76	1B 4C
Hellste Einstellung	ESC @ D 0	27 64 68 48	1B 40 44 30
Mittlere Einstellung	ESC @ D 7	27 64 68 55	1B 40 44 37
Dunkelste Einstellung	ESC @ D F	27 64 68 70	1B 40 44 46
Invertieren aus	ESC @ N 0	27 64 78 48	1B 40 4E 30
Invertieren ein	ESC @ N 1	27 64 78 49	1B 40 4E 31

## LED-Steuerung:

LED1 einschalten	ESC @ a 1	27 64 97 49	1B 40 61 31
LED1 ausschalten	ESC @ a 0	27 64 97 48	1B 40 61 30
(für alle anderen LEDs:	'b'=2, 'c'=3, 'd'=4, 'e'=5, 'f'=6,		
	'g'=7, 'h'=8)		

## Sonstige Befehle:

Tastatur sperren	ESC N	27 78	1B 4E
Tastatur freigeben	ESC Q	27 81	1B 51
Restart warm	ESC H	27 72	1B 48
Restart kalt	ESC @ G	27 64 71	1B 40 47
Setup-Modus sperren	ESC @ H	27 64 72	1B 40 48
Setup-Modus freigeben	ESC @ I	27 64 73	1B 40 49
Demonstrations-Anzeige	ESC J	27 74	1B 4A
Anzeige-Test	ESC @ A	27 64 65	1B 40 41
Tastatur-Test	ESC @ 9	27 64 57	1B 40 39
LED-Test	ESC @ L	27 64 76	1B 40 4C
Hardware-Test	ESC @ C	27 64 67	1B 40 43
POLL	ESC @ B	27 64 66	1B 40 42
..D250 antwortet mit	SOH	1	10

**Wichtig:** Damit die PCD-Steuerung den Charakter "@" ausgibt, muss im Befehltext "@@" programmiert werden.

## 7. Die verschiedenen Charakter-Sätze

Um die Kompatibilität zu früheren Terminal-Programmen zu gewährleisten, wird auch beim ..D250 der Zeichensatz "D100-kompatibel" berücksichtigt.

### 7.1 Erste ASCII-Tabelle (32...127 dezimal, 20...7F heza)

Diese Charakter sind für beide Charakter-Sätze gleich.

| Dez. Hex ASC |
|--------------|--------------|--------------|--------------|--------------|--------------|
| 32 20 SP     | 48 30 0      | 64 40 @      | 80 50 P      | 96 60 `      | 112 70 p     |
| 33 21 !      | 49 31 1      | 65 41 A      | 81 51 Q      | 97 61 a      | 113 71 q     |
| 34 22 "      | 50 32 2      | 66 42 B      | 82 52 R      | 98 62 b      | 114 72 r     |
| 35 23 #      | 51 33 3      | 67 43 C      | 83 53 S      | 99 63 c      | 115 73 s     |
| 36 24 \$     | 52 34 4      | 68 44 D      | 84 54 T      | 100 64 d     | 116 74 t     |
| 37 25 %      | 53 35 5      | 69 45 E      | 85 55 U      | 101 65 e     | 117 75 u     |
| 38 26 &      | 54 36 6      | 70 46 F      | 86 56 V      | 102 66 f     | 118 76 v     |
| 39 27 '      | 55 37 7      | 71 47 G      | 87 57 W      | 103 67 g     | 119 77 w     |
| 40 28 (      | 56 38 8      | 72 48 H      | 88 58 X      | 104 68 h     | 120 78 x     |
| 41 29 )      | 57 39 9      | 73 49 I      | 89 59 Y      | 105 69 i     | 121 79 y     |
| 42 2A *      | 58 3A :      | 74 4A J      | 90 5A Z      | 106 6A j     | 122 7A z     |
| 43 2B +      | 59 3B ;      | 75 4B K      | 91 5B [      | 107 6B k     | 123 7B {     |
| 44 2C ,      | 60 3C <      | 76 4C L      | 92 5C \      | 108 6C l     | 124 7C       |
| 45 2D -      | 61 3D =      | 77 4D M      | 93 5D ]      | 109 6D m     | 125 7D }     |
| 46 2E .      | 62 3E >      | 78 4E N      | 94 5E ^      | 110 6E n     | 126 7E →     |
| 47 2F /      | 63 3F ?      | 79 4F O      | 95 5F _      | 111 6F o     | 127 7F DEL   |

### 7.2 Erweiterte ASCII-Tabelle (128...255 dez., 80...FF heza)

Für die erweiterte ASCII-Tabelle stehen 2 Charaktersätze zur Verfügung:

- "CodePage 437", der universelle Zeichensatz, der alle sprachen-abhängigen Zeichen enthält
- "D100-kompatibel", der zum früheren Terminal ..D100 kompatibel ist.

**Hinweis:** Die auf der ..D250-Anzeige erscheinenden Zeichen können in Details leicht von den hier dargestellten Zeichen abweichen.

### 7.2.1 CodePage 437 (Default-Einstellung)

Dieser universelle Charakter-Satz enthält alle sprachenabhängigen Zeichen wie sie bei den kleinen Terminals ..D160/..D170 und ..D202 unter "Deutsch", "Englisch", "Französisch" und "Skandinavisch" definiert sind.

Dez. Hex ASC	Dez. Hex ASC	Dez. Hex ASC	Dez. Hex ASC
128 80 Ç	144 90 É	160 A0 á	176 B0 ☐
129 81 ü	145 91 æ	161 A1 í	177 B1 ☐
130 82 é	146 92 Æ	162 A2 ó	178 B2 ☐
131 83 â	147 93 ô	163 A3 ú	179 B3
132 84 ä	148 94 ö	164 A4 ñ	180 B4
133 85 à	149 95 ò	165 A5 Ñ	181 B5
134 86 â	150 96 û	166 A6 ª	182 B6
135 87 ç	151 97 ù	167 A7 °	183 B7 π
136 88 ê	152 98 ÿ	168 A8 ¿	184 B8 ¶
137 89 ë	153 99 Ö	169 A9 ←	185 B9
138 8A è	154 9A Ü	170 AA →	186 BA
139 8B ï	155 9B ø	171 AB ½	187 BB ¶
140 8C î	156 9C £	172 AC ¼	188 BC
141 8D ï	157 9D ¥	173 AD ¡	189 BD
142 8E Ä	158 9E Pts	174 AE «	190 BE
143 8F Å	159 9F f	175 AF »	191 BF ¶

Dez. Hex ASC	Dez. Hex ASC	Dez. Hex ASC	Dez. Hex ASC
192 C0 L	208 D0 ⚡	224 E0 α	240 F0 ≡
193 C1 ⊥	209 D1 ⚡	225 E1 β	241 F1 ±
194 C2 ⊤	210 D2 π	226 E2 Γ	242 F2 ≥
195 C3	211 D3 ⚡	227 E3 π	243 F3 ≤
196 C4 —	212 D4 ⚡	228 E4 Σ	244 F4
197 C5 †	213 D5 F	229 E5 σ	245 F5
198 C6 †	214 D6 π	230 E6 μ	246 F6 ÷
199 C7	215 D7 †	231 E7 τ	247 F7 ≈
200 C8 ⚡	216 D8 †	232 E8 Φ	248 F8 °
201 C9 ⚡	217 D9 J	233 E9 Θ	249 F9 ·
202 CA ⚡	218 DA Γ	234 EA Ω	250 FA ·
203 CB ⚡	219 DB ■	235 EB δ	251 FB √
204 CC	220 DC ■	236 EC ∞	252 FC n
205 CD =	221 DD	237 ED φ	253 FD ²
206 CE †	222 DE	238 EE ε	254 FE ■
207 CF ±	223 DF ■	239 EF ∩	255 FF

CodePage 437

Darstellung der Zeichen auf dem Display im Modus 4 x 20 Charakter

	32	48	64	80	96	112	128	144	160	176	192	208	224	240	Dez
	ø	á	â	ã	ä	å	ç	é	á	â	ã	ä	å	æ	≡
!	1	A	Q	a	q		ü	æ	í					þ	±
"	2	B	R	b	r		é	æ	ó					γ	∞
#	3	C	S	c	s		â	ô	ú					π	∞
\$	4	D	T	d	t		â	ö	ñ					Σ	∞
%	5	E	U	e	u		â	ó	ñ					∫	∞
&	6	F	V	f	v		â	ó	ñ					∫	∞
'	7	G	W	g	w		ç	ú	ñ					∫	∞
<	8	H	X	h	x		ç	ú	ñ					∫	∞
>	9	I	Y	i	y		ç	ú	ñ					∫	∞
*	:	J	Z	j	z		ç	ú	ñ					∫	∞
+	;	K	L	k	l		ç	ú	ñ					∫	∞
,	<	L	\	l	l		ç	ú	ñ					∫	∞
-	=	M	J	m	j		ç	ú	ñ					∫	∞
.	>	N	^	n	→		ç	ú	ñ					∫	∞
/	?	O	_	o	←		ç	ú	ñ					∫	∞
	47	63	79	95	111	127	143	159	175	191	207	223	239	255	Dez

## 7.2.2 D100-kompatibel

Dez. Hex ASC	Dez. Hex ASC	Dez. Hex ASC	Dez. Hex ASC
128 80	144 90	160 A0	176 B0
129 81	145 91	161 A1 □	177 B1
130 82	146 92	162 A2	178 B2
131 83	147 93	163 A3	179 B3
132 84	148 94	164 A4	180 B4
133 85	149 95	165 A5	181 B5
134 86	150 96	166 A6	182 B6
135 87	151 97	167 A7	183 B7
136 88	152 98	168 A8	184 B8
137 89	153 99	169 A9	185 B9
138 8A	154 9A	170 AA	186 BA
139 8B	155 9B	171 AB	187 BB
140 8C	156 9C	172 AC	188 BC
141 8D	157 9D	173 AD	189 BD
142 8E	158 9E	174 AE Σ	190 BE
143 8F	159 9F	175 AF	191 BF

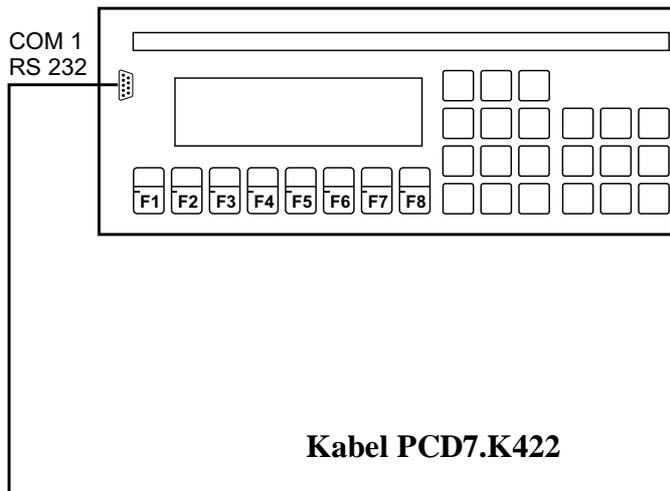
Dez. Hex ASC	Dez. Hex ASC	Dez. Hex ASC	Dez. Hex ASC
192 C0	208 D0	224 E0 α	240 F0
193 C1	209 D1	225 E1 ä	241 F1
194 C2	210 D2	226 E2 ß	242 F2
195 C3	211 D3	227 E3	243 F3
196 C4 Ä	212 D4	228 E4 ä	244 F4 Ω
197 C5 Å	213 D5	229 E5 å	245 F5
198 C6 Æ	214 D6 Ö	230 E6 æ	246 F6 ö
199 C7	215 D7	231 E7	247 F7 π
200 C8	216 D8 Ø	232 E8	248 F8 Ø
201 C9	217 D9	233 E9	249 F9
202 CA	218 DA	234 EA	250 FA
203 CB	219 DB □	235 EB x	251 FB
204 CC	220 DC Ü	236 EC φ	252 FC ü
205 CD	221 DD	237 ED	253 FD
206 CE	222 DE	238 EE	254 FE
207 CF	223 DF □	239 EF Ö	255 FF ■

## 8. Programmbeispiele für die PCD

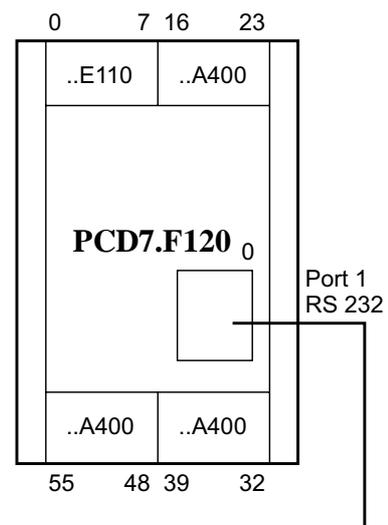
### 8.1 Hardware-Installation

Die folgenden Beispiele basieren auf der nachstehenden Hardware-Installation:

#### PCD7.D250



#### PCD1.M120/..M130 oder PCD2.M1..



Serielle Schnittstelle Nr. 1 : RS 232 (PCD7.F120)  
 Kabel (PCD7.K422) : angeschlossen für Modus MC0  
 (ohne RTS/CTS)  
 Setup am ..D250 (EPROM) : "Default Setup"-Einstellung  
 (siehe Kap. 5.3)

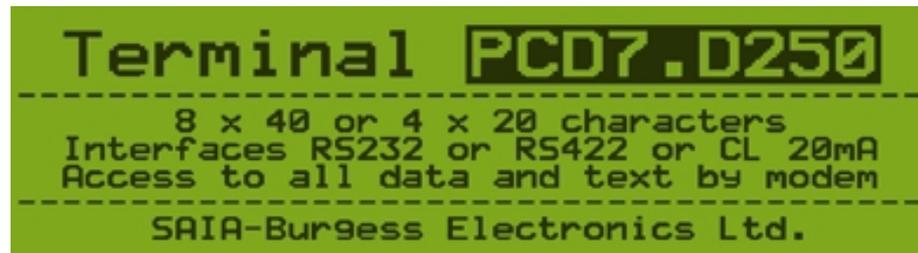
#### Hinweise:

- Zur Kontrolle und Diagnose der seriellen Datenübertragung werden sogenannte Diagnoseflags verwendet. Um diese auf einfache Weise sichtbar zu machen, werden in den Beispielen Ausgänge verwendet. Es ist deshalb wichtig, dass die drei eingezeichneten Ausgangsmodule ..A400 (oder ..A410) an den entsprechenden Plätzen vorhanden sind.
- Für die gleichzeitige Übertragung mehrerer Bildschirmseiten wird empfohlen mit RTS/CTS (MC1) oder XON/XOFF (MC2) zu arbeiten. Dies vermeidet, dass der Eingangsbuffer des Terminals überlaufen kann. Bitte beachten, dass dies auch im "Setup" des Terminals eingestellt werden muss!

## 8.2 Einfache Textausgabe

Durch Schliessen des Schalters, angeschlossen an Eingang 0, soll ein einfacher Text in Grossschrift (4 x 20), durch Schliessen von Eingang 1 ein Text in Kleinschrift ausgegeben werden.

- 8.2.1 Dieses Anwenderprogramm ist in BLOCTEC strukturiert
- 8.2.2 Dieses Anwenderprogramm ist in GRAFTEC strukturiert
- 8.2.3 Durch Schliessen von Eingang 0 soll ein kombinierter Text in Gross- und Kleinschrift ausgegeben werden



### 8.2.1 Einfache Textausgabe in BLOCTEC

```

;-----+
;
; User program example 8.2.1 for the industrial terminal PCD7.D250
; =====
; The program is structured in BLOCTEC
;
; File:          NDEMO21.SRC
;
; Creation:     16.01.97          U.Jäggi
; Modified:    03.02.00          C. Bruegger
;
;-----+

```

```

TEXT    1      "<12>"                ; Clear display
          "<27><64><77><49>"          ; Display mode:4 x 20
          "<27><84>"                ; Cursor off
          "      INDUSTRIAL      "
          "    CONTROL-TERMINAL  "
          "      PCD7.D250       "
          "Display mode:4 x 20 "

```

```

TEXT    10     "<12>"                ; Clear display
          "<27><64><77><48>"          ; Display mode:8 x 40
          "<27><84>"                ; Cursor off
          "<10>"                    ; LF
          "#####"
          "#      INDUSTRIAL      #"
          "#    CONTROL-TERMINAL  #"
          "#      PCD7.D250       #"
          "#    Display mode:8 x 40 #"
          "#####"

```

```

TEXT    100    "UART:9600,8,E,1;MODE:MC0;DIAG:016,R100"
;

```

```

;-----
; Coldstart
;-----
XOB      16
SASI     1           ; Assignment RS232 interface n°1
          100        ; Text 100
EXOB

;-----
; Mainprogram
;-----
COB      0
          0
STH      I          0           ; for display 4 x 20
DYN      F          0
ANL      O          22        ; Text busy flag
CPB      H          0           ; Send text
STH      I          1           ; for display 8 x 40
DYN      F          1
ANL      O          22        ; Text busy flag
CPB      H          1           ; Send text
ECOB

;=====
PB       0           ; Send text
STXT    1           ; Interface 1
          1           ; Text 1
EPB

PB       1           ; Send text
STXT    1           ; Interface 1
          10          ; Text 10
EPB

;-----

```

### 8.2.2 Einfache Textausgabe in GRAFTEC

```

;-----+
;
; User program example 8.2.2 for the industrial terminal PCD7.D250
; =====
; The program is structured in GRAFTEC
;
; File:          NDEMO22.SRC
;
; Creation:     29.01.97          U.Jäggi
; Modified:    03.02.00          C. Bruegger
;
;-----+

```

```

TEXT    1      "<12>"                ; Clear display
          "<27><64><77><49>"          ; Display mode:4 x 20
          "<27><84>"                ; Cursor off
          "      INDUSTRIAL      "
          " CONTROL-TERMINAL    "
          "      PCD7.D250      "
          "Display mode:4 x 20 "

```

```

TEXT    10     "<12>"                ; Clear display
          "<27><64><77><48>"          ; Display mode:8 x 40
          "<27><84>"                ; Cursor off
          "<10>"                    ; LF
          "#####"
          "#          INDUSTRIAL          #"
          "#          CONTROL-TERMINAL    #"
          "#          PCD7.D250          #"
          "#          Display mode:8 x 40  #"
          "#####"

```

```

TEXT    100    "UART:9600,8,E,1;MODE:MC0;DIAG:016,R100"

```

---

```

;-----+
; Coldstart
;-----+
XOB      16
SASI     1          ; Assignment interface n°1
          100       ; Text 100
EXOB

```

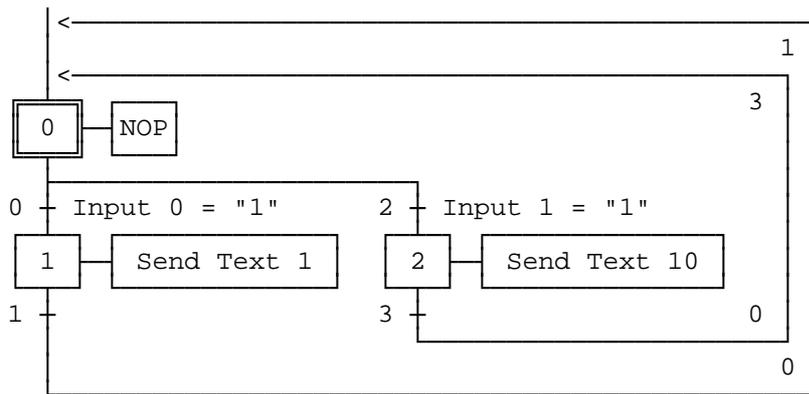
---

```

;-----+
; Mainprogram
;-----+
COB      0
          0
CSB      0
ECOB

```

---



```

SB      0
;-----
IST     0           ; NOP
        I 1
        I 3
        O 0
        O 2
Acc     1

EST

;-----

ST      1           ; Send Text 1
        I 0
        O 1
STXT    1
        1
EST     1           ; 1
;-----

ST      2           ; Send Text 10
        I 2
        O 3
STXT    1
        10
EST     2           ; 2
;=====

TR      0           ; Input 0 = "1"
        I 0           ; NOP
        O 1

STH     I      0
DYN     F      0
ANL     O      22    ; Text busy
ETR     ; 0
;-----

TR      1
        I 1
        O 0           ; NOP
ETR     ; 1
;-----

TR      2           ; Input 1 = "1"
        I 0           ; NOP
        O 2

STH     I      1
DYN     F      1
ANL     O      22    ; Text busy
ETR     ; 2
;-----

TR      3
        I 2
        O 0           ; NOP
ETR     ; 3

ESB

;-----
    
```

### 8.2.3 Textausgabe von Gross- und Kleinschrift in BLOC TEC

```

;-----+
;
; User program example 8.2.3 for the industrial terminal PCD7.D250
; =====
; The program is structured in BLOC TEC
;
; File:          NDEMO23.SRC
;
; Creation:     15.02.00          C. Bruegger
;
;-----+

```

```

TEXT    1      "<27><64><77><49>"      ; Display mode:4 x 20
           "<12>"                    ; Clear display
           "<27><84>"                  ; Cursor off
           "<27><64><78><48>"          ; Invert mode off
           " TERMINAL "
           "<27><64><78><49>"          ; Invert mode on
           "PCD7.D250"
           "<27><64><78><48>"          ; Invert mode off
           "<27><64><77><53>"          ; Transparent mode on
           "<27><64><77><48>"          ; Display mode:8 x 40
           "<12>"                    ; Clear display
           "<27><84>"                  ; Cursor off
           "<5><5>"                    ; Cursor down 2 x
           "-----"
           "      8 x 40 or 4 X 20 characters      "
           " Interfaces RS232 or RS422 or CL 20mA  "
           " Access to all data and text by modem  "
           "-----"
           "      SAIA -Burgess Electronics AG      "

```

```

TEXT    100    "UART:9600,8,E,1;MODE:MC0;DIAG:016,R100"

```

```

;-----+

```

```

;-----+
; Coldstart
;-----+
XOB      16
SASI     1          ; Assignment RS232 interface n°1
          100       ; Text 100
EXOB

```

```

;=====+

```

```

;-----+
; Mainprogram
;-----+
COB      0
          0
STH      I          0
DYN      F          0
ANL      O          22      ; Text busy flag
CPB      H          0      ; Send text

```

```

ECOB

```

```

;=====+

```

```

PB       0          ; Send text
STXT     1          ; Interface 1
          1          ; Text 1
EPB

```

```

;-----+

```

## 8.3 Ausgabe mehrerer unterschiedlicher Texte

Durch Schliessen der Schalter, angeschlossen an den Eingängen 0, 1 und 2, sollen folgende Texte ausgegeben werden:

Input 0 : ein einfacher Text wird ausgegeben.

Input 1 : ein Text mit dem Zustand der Eingänge 4 und 5 wird ausgegeben.

Input 2 : ein Text mit dem Zustand der Eingänge 6 und 7 wird ausgegeben.

Die Anzeigen erfolgen im Modus 4 x 20.

- 8.3.1 Dieses Anwenderprogramm ist in BLOCTEC strukturiert
- 8.3.2 Dieses Anwenderprogramm ist in GRAFTEC strukturiert <sup>\*)</sup>

\*) Bei diesem Programm wird in der Kaltstartroutine XOB16 mit dem POLL-Befehl die Funktionsbereitschaft des Terminals überprüft. Damit werden auch die Einschaltvorgänge von Steuerung und Terminal miteinander koordiniert.

### 8.3.1 Ausgabe mehrerer unterschiedlicher Texte in BLOCTEC

```

;-----+
;
; User program example 8.3.1 for the industrial terminal PCD7.D250
; =====
; The program is structured in BLOCTEC
;
; File:          NDEMO31.SRC
;
; Creation:     16.01.97          U.Jäggi
; Modified:    03.02.00          C. Bruegger
;
;-----+

```

```

TEXT      1      "<12>"                                ; Clear display
;                                     ; cursor off
;                                     " <27><84>"
;                                     "  Main menu I0<10><13>  "
;                                     "  Display status  "
;                                     "  Input 4,5   : I1  "
;                                     "  Input 6,7   : I2  "
TEXT      2      "<12>"
;                                     "  Status <10><13>  "
;                                     "  Input 4 : $i0004<10><13>  "
;                                     "  Input 5 : $i0005<10><13>  "
;                                     "  Main menu I0  "
TEXT      3      "<12>"
;                                     "  Status <10><13>  "
;                                     "  Input 6 : $i0006<10><13>  "
;                                     "  Input 7 : $i0007<10><13>  "
;                                     "  Main menu I0  "
TEXT      4      "<27><64><77><49>"                    ; Display mode:4 x 20
TEXT      100    "UART:9600,8,E,1;MODE:MC0;DIAG:O16,R100"
;

```

```

;-----
; Coldstart
;-----
XOB      16
SASI     1           ; Assignment RS232 interface
          100        ; Text 100
STXT     1
          4           ; Text 4: mode 4 x 20
EXOB

;-----
; Mainprogram
;-----
COB      0
          0

;-----
STH      I          0
DYN      F          0
ANL      O          22      ; Text busy flag
CFB      H          0      ; Send text
          1           ; Text 1

;-----
STH      I          1
DYN      F          1
ANL      O          22      ; Text busy flag
CFB      H          0      ; Send text
          2           ; Text 2

;-----
STH      I          2
DYN      F          2
ANL      O          22      ; Text busy flag
CFB      H          0      ; Send text
          3           ; Text 3
ECOB

;=====
FB        0           ; Send text
STXT     1           ; Interface 1
          =          1      ; Textnumber
EFB

;-----

```

### 8.3.2 Ausgabe mehrerer unterschiedlicher Texte in GRAFTEC

```

;-----+
;
; User program example 8.3.2 for the industrial terminal PCD7.D250
; =====
; The program is structured in GRAFTEC
;
; File:          NDEMO32.SRC
;
; Creation:     16.01.97          U.Jäggi
; Modified:    03.02.00          C. Bruegger
;
;-----+

TEXT    1      "<12>"                ; Clear display
                "<27><84>"            ; Cursor off
                "  Main menue I0<10><13>  "
                "  Display status      "
                "  Input 4,5   : I1  "
                "  Input 6,7   : I2  "

TEXT    2      "<12>"                "  Status <10><13>  "
                "  Input  4  :$i0004<10><13>  "
                "  Input  5  :$i0005<10><13>  "
                "  Main menue I0  "

TEXT    3      "<12>"                "  Status <10><13>  "
                "  Input  6  :$i0006<10><13>  "
                "  Input  7  :$i0007<10><13>  "
                "  Main menue I0  "

TEXT    4      "<27><64><77><49>"                ; Display mode:4 x 20

TEXT    10     "<ESC>@B"                ; Poll command

TEXT    100    "UART:9600,8,E,1;MODE:MC0;DIAG:O16,R100"
;-----+

;-----+
; Coldstart
;-----+

XOB     16
SASI    1                ; Assignment interface n°1
        100              ; Text 100

;-----+

```

```

termpoll:  stxt  1          ; START OF THE POLL COMMAND
           sth   0 22
           jr    h -1

           acc   h          ; (ld T is accu dependent)
           ld   T 0         ; start short receive timeout
           sth   2          ; (must be min. 10 mS)
           sth   3

termwait:  sth   0 16       ; character received?
           jr    h termok   ; yes
           sth   T 0
           jr    h termwait ; loop for timeout period
           jr    termpoll   ; Terminal not ready, repeat the poll

termok:    srxd  1          ; read the character
           R 1
           cmp  R 1         ; SOH character ?
           sth  1
           jr   z termready ; yes, Terminal is ready
           jr   termpoll    ; no, repeat the poll

termready: ld   R 1         ; clear receive register
           sth  0          ; END OF THE POLL COMMAND

;

```

---

```

           STXT  1          ; Display mode : 4 x 20
           sth   4

           sth   0 22
           jr    h -1

           STXT  1          ; Text 1: Main menu
           sth   1
           sth   0 22
           jr    h -1

           EXOB

;

```

```

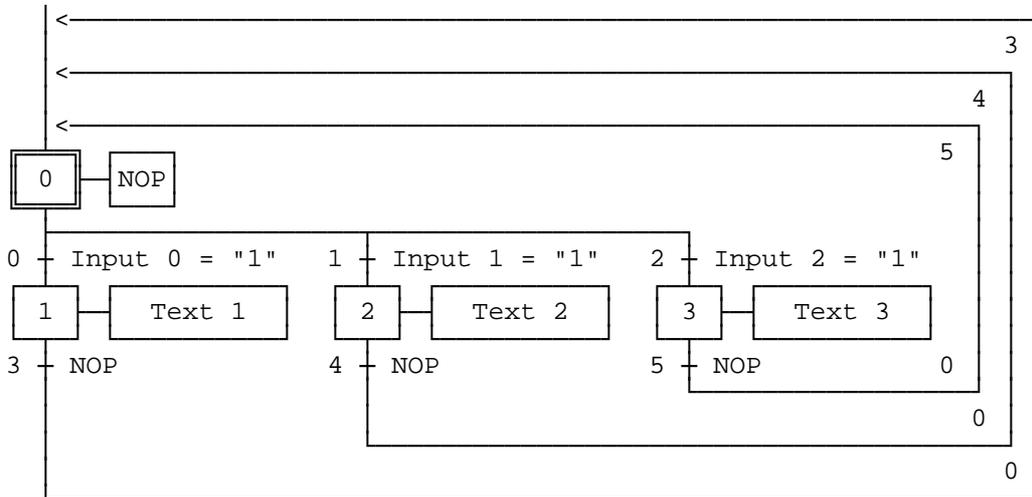
;-----
; Mainprogram
;-----

```

```

COB    0
      0
CSB    0
ECOB

```



```

SB    0

```

```

;-----

```

```

IST    0          ; NOP
      I          3      ; NOP
      I          4      ; NOP
      I          5      ; NOP
      O          0      ; Input 0 = "1"
      O          1      ; Input 1 = "1"
      O          2      ; Input 2 = "1"
EST

```

```

;-----

```

```

ST     1          ; Text 1
      I          0      ; Input 0 = "1"
      O          3      ; NOP
STXT   1
      1
EST

```

```

;-----

```

```

ST     2          ; Text 2
      I          1      ; Input 1 = "1"
      O          4      ; NOP
STXT   1
      2
EST

```

```

;-----

```

```

ST     3          ; Text 3
      I          2      ; Input 2 = "1"
      O          5      ; NOP
STXT   1
      3
EST

```

```

;-----

```

```

TR      0          ; Input 0 = "1"
        I          ; NOP
        O          ; Text 1
        0          ;
STH     I          ;
DYN     F          ;
ANL     O          ; Text busy
        0          ;
        22         ;
;
-----
TR      1          ; Input 1 = "1"
        I          ; NOP
        O          ; Text 2
        0          ;
        2          ;
STH     I          ;
DYN     F          ;
ANL     O          ; Text busy
        0          ;
        22         ;
ETR
;
-----
TR      2          ; Input 2 = "1"
        I          ; NOP
        O          ; Text 3
        0          ;
        3          ;
STH     I          ;
DYN     F          ;
ANL     O          ; Text busy
        0          ;
        22         ;
ETR
;
-----
TR      3          ; NOP
        I          ; Text 1
        0          ;
        1          ;
        0          ;
ETR
;
-----
TR      4          ; NOP
        I          ; Text 2
        0          ;
        2          ;
        0          ;
ETR
;
-----
TR      5          ; NOP
        I          ; Text 3
        0          ;
        3          ;
        0          ;
ETR
;
ESB
;
-----

```

## 8.4 Einzeltastenerkennung mit nachfolgender Aktion

---

Nach Betätigen der Funktionstasten F1, F2 ,F3 und F4 werden folgende Texte ausgegeben:

Taste F1 : ein einfacher Text wird ausgegeben.

Taste F2 : ein Text mit dem Zustand der Eingänge 0 bis 7 sowie der Ausgänge 16 bis 23 wird ausgegeben.

Taste F3 : ein Text mit dem BCD-Wert der Schalter, angeschlossen an den Eingängen 0 bis 7, wird ausgegeben.

Taste F4 : ein Text mit Datum, Woche und Zeit wird ausgegeben.

Bei Betätigung der Funktionstasten wird der entsprechende Text nur einmal zum Terminal gesendet. Falls ein Wert auf dem Terminal zyklisch aufgefrischt werden soll, so sind bei der Textausgabe die folgenden Punkte zu beachten damit eine stabile Anzeige erreicht wird:

- Cursor ausschalten
- Am Textanfang keinen Steuercode "12" (Anzeige löschen) senden.

- 8.4.1 Dieses Anwenderprogramm enthält Sprünge \*)
- 8.4.2 Dieses Anwenderprogramm ist in BLOC TEC strukturiert \*)
- 8.4.3 Dieses Anwenderprogramm ist in GRAF TEC strukturiert \*)

\*) Bei diesen Programmen wird in der Kaltstartroutine XOB16 mit dem POLL-Befehl die Funktionsbereitschaft des Terminals überprüft. Damit werden auch die Einschaltvorgänge von Steuerung und Terminal miteinander koordiniert.

### 8.4.1 Einzeltastenerkennung mit nachfolgender Aktion (Programm enthält Sprünge)

```

;+-----+
;
; User program example 8.4.1 for the industrial terminal PCD7.D250
; =====
; The program contains jumps
;
; File:          NDEMO41.SRC
;
; Creation:     29.01.97          U.Jäggi
; Modified:    07.02.00          C. Bruegger
;
;+-----+

TEXT    1      "<12>"                ; Clear display
                "<27><84>"            ; Cursor off
                "  Main menu   F1<10><13>"
                "  I/O     0..23 F2<10><13>"
                "  BCD-Switch F3<10><13>"
                "  Date/Time  F4"

TEXT    2      "<12>"                ; Clear display
                "  Input Status<10><13>"
                "  I0..7   :$I000<10><13>"
                "  O16..23:$O0016<10><13>"
                "  Main menu  F1"

TEXT    3      "<12>"                ; Clear display
                "  BCD-Value I0...7 <10><13>"
                "  ----- <10><13>"
                "  Value   : $R0010 <10><13>"
                "  Main menu  F1"

TEXT    4      "<12>"                ; Clear display
                "  Date   : $D<10><13>"
                "  Week  : $W<10><13>"
                "  Time  : $H<10><13>"
                "  Main menu  F1"

TEXT    5      "<27><64><77><49>"      ; Display mode:4 x 20

TEXT    10     "<ESC>@B"              ; Poll command

TEXT    100    "UART:9600,8,E,1;MODE:MC0;DIAG:O16,R100"

; Symboldefinitions
; =====
; Diagnostic outputs serial interface
; -----
RBSY    EQU    0      16          ; Receiver Busy
RFUL    EQU    0      RBSY+1     ; Receive Buffer Full
RDIA    EQU    0      RBSY+2     ; Receiver Diagnostic
TBSY    EQU    0      RBSY+3     ; Transmitter Busy
TFUL    EQU    0      RBSY+4     ; Transmit Buffer Full
TDIA    EQU    0      RBSY+5     ; Transmitter Diagnostic
XBSY    EQU    0      RBSY+6     ; Text Busy
NEXE    EQU    0      RBSY+7     ; Not Executed
; -----
; Function/Program blocks
; -----
READ    EQU    FB     0          ; Read character
SEND    EQU    FB     1          ; Send text
COMPARE EQU    PB     0          ; Compare received character
; -----
; Register
; -----
RBUF_R  EQU    R      1000
;

```

```

;-----
; Coldstart
;-----
XOB      16
SASI     1          ; Assignment interface n°1
          100       ; Text 100

;-----
;
termpoll: stxt      1          ; START OF THE POLL COMMAND
          10
          sth      0 22
          jr       h -1

          acc      h          ; (ld T is accu dependent)
          ld       T 0        ; start short receive timeout
          4        ; (must be min. 10 mS)
          5

termwait: sth      0 16        ; character received?
          jr       h termok   ; yes
          sth      T 0
          jr       h termwait ; loop for timeout period
          jr       termpoll   ; Terminal not ready, repeat the poll

termok:   srxd     1          ; read the character
          R 1
          cmp      R 1        ; SOH character ?
          1
          jr       z termready ; yes, Terminal is ready
          jr       termpoll   ; no, repeat the poll

termready: ld      R 1        ; clear receive register
          0          ; END OF THE POLL COMMAND

;-----
;
          STXT     1          ; Display mode : 4 x 20
          5

          sth      0 22
          jr       h -1

          STXT     1          ; Main menu
          1
          sth      0 22
          jr       h -1

          EXOB

;-----
;

```

```

;-----
; Main program
;-----
COB      0
          0
STH      0      RBSY      ; Receiver busy
ANL      0      XBSY      ; Text busy
JR        L      END        ; If RBSY = low then do nothing
SRXD     1      ; Interface 1
          R      RBUF_R     ; Receive buffer register
;-----
; Compare received character
CMP      R      RBUF_R
          K      65        ; F1
ACC      Z
JR        L      F2
STXT     1      ; Interface 1
          1      ; Text 1
JR        END
;-----
F2:      CMP      R      RBUF_R
          K      66        ; F2
ACC      Z
JR        L      F3
STXT     1      ; Interface 1
          2      ; Text 2
JR        END
;-----
F3:      CMP      R      RBUF_R
          K      67        ; F3
ACC      Z
JR        L      F4
STXT     1      ; Interface 1
          3      ; Text 3
JR        END
;-----
F4:      CMP      R      RBUF_R
          K      68        ; F4
ACC      Z
JR        L      END
STXT     1      ; Interface 1
          4      ; Text 4
JR        END
;-----
END:     DIGI     2
          I      0
          R      10
;-----
          ECOB

```

### 8.4.2 Einzeltastenerkennung mit nachfolgender Aktion in BLOC TEC

```

;+-----+
;
; User program example 8.4.2 for the industrial terminal PCD7.D160/170
; =====
; The program is structured in BLOC TEC
;
; File:          NDEMO42.SRC
;
; Creation:     29.01.97          U.Jäggi
; Modified:    07.07.00          C.Bruegger
;
;+-----+

TEXT    1      "<12>"                ; Clear display
                "<27><84>"            ; Cursor off
                "  Main menu   F1<10><13>"
                "  I/O   0..23 F2<10><13>"
                "  BCD-Switch F3<10><13>"
                "  Date/Time  F4"

TEXT    2      "<12>"                ; Clear display
                "  Input Status <10><13>"
                "  I0..7   :$I0000 <10><13>"
                "  O16..23:$O0016 <10><13>"
                "  Main menu  F1"

TEXT    3      "<12>"                ; Clear display
                "  BCD-Value I0...7 <10><13>"
                "  ----- <10><13>"
                "  Value   : $R0010 <10><13>"
                "  Main menu  F1"

TEXT    4      "<12>"                ; Clear display
                "  Date   : $D<10><13>"
                "  Week  : $W<10><13>"
                "  Time  : $H<10><13>"
                "  Main menu F1"

TEXT    5      "<27><64><77><49>"        ; Display mode:4 x 20

TEXT    10     "<ESC>@B"                ; Poll command

TEXT    100    "UART:9600,8,E,1;MODE:MC0;DIAG:O16,R100"

; Symboldefinitions
;=====
; Diagnostic outputs serial interface
;-----
RBSY    EQU    0      16                ; Receiver Busy
RFUL    EQU    0      RBSY+1           ; Receive Buffer Full
RDIA    EQU    0      RBSY+2           ; Receiver Diagnostic
TBSY    EQU    0      RBSY+3           ; Transmitter Busy
TFUL    EQU    0      RBSY+4           ; Transmit Buffer Full
TDIA    EQU    0      RBSY+5           ; Transmitter Diagnostic
XBSY    EQU    0      RBSY+6           ; Text Busy
NEXE    EQU    0      RBSY+7           ; Not Executed
;-----
; Function/Program blocks
;-----
READ    EQU    FB     0                ; Read character
SEND    EQU    FB     1                ; Send text
COMPARE EQU    PB     0                ; Compare received character
;-----
; Register
;-----
RBUF_R  EQU    R      1000
;

```

```

;-----
; Coldstart
;-----
XOB      16
SASI     1          ; Assignment interface n°1
          100       ; Text 100

;-----
;
termpoll: stxt      1          ; START OF THE POLL COMMAND
          10
          sth      0 22
          jr       h -1

          acc      h          ; (ld T is accu dependent)
          ld       T 0        ; start short receive timeout
          6        ; (must be min. 10 mS)
          7

termwait: sth      0 16        ; character received?
          jr       h termok   ; yes
          sth      T 0
          jr       h termwait ; loop for timeout period
          jr       termpoll   ; Terminal not ready, repeat the poll

termok:   srxd     1          ; read the character
          R 1
          cmp      R 1        ; SOH character ?
          1
          jr       z termready ; yes, Terminal is ready
          jr       termpoll   ; no, repeat the poll

termready: ld      R 1        ; clear receive register
          0          ; END OF THE POLL COMMAND

;-----
;
          STXT     1          ; Display mode : 4 x 20
          5

          sth      0 22
          jr       h -1

          STXT     1          ; Main menu
          1
          sth      0 22
          jr       h -1

          CFB      SEND

          EXOB     1

;-----
;

```

```

;-----
; Main program
;-----
COB      0
         0
STH      0      RBSY      ; Receiver busy
ANL      0      XBSY      ; Text busy
CFB      H      READ      ; Read character
         R      RBUF_R    ; Receive buffer register
CPB      H      COMPARE   ; Compare received character
;-----
; Read BCD-Switch
DIGI     2
         I      0
         R      10
;-----
;=====
PB       COMPARE      ; Compare received character
;-----; Key = F1 ?
CMP      R      RBUF_R
         K      65      ; F1
ACC      Z
CFB      H      SEND      ; Send text
         1      ; Text 1
;-----; Key = F2 ?
CMP      R      RBUF_R
         K      66      ; F2
ACC      Z
CFB      H      SEND      ; Send text
         2      ; Text 2
;-----; Key = F3 ?
CMP      R      RBUF_R
         K      67      ; F3
ACC      Z
CFB      H      SEND      ; Send text
         3      ; Text 3
;-----; Key = F4 ?
CMP      R      RBUF_R
         K      68      ; F4
ACC      Z
CFB      H      SEND      ; Send text
         4      ; Text 4
;-----
EPB
;-----
FB       READ      ; Read character
SRXD     1
         =      1      ; Interface 1
EFB
;-----
FB       SEND      ; Send text
STXT     1
         =      1      ; Interface 1
         ; Textnumber
EFB
;-----

```

### 8.4.3 Einzeltastenerkennung mit nachfolgender Aktion in GRAFTEC

```

;+-----+
;
; User program example 8.4.3 for the industrial terminal PCD7.D250
; =====
; The program is structured in GRAFTEC
;
; File:          NDEMO43.SRC
;
; Creation:      29.01.97          U.Jäggi/T.Hofer
; Modified:      07.02.00          C. Bruegger
;
;+-----+

TEXT      1      "<12>"                                ; Clear display
              "<27><84>"                                ; Cursor off
              "  Main menu    F1<10><13>"
              "  I/O      0..23 F2<10><13>"
              "  BCD-Switch F3<10><13>"
              "  Date/Time   F4"

TEXT      2      "<12>"                                ; Clear display
              "  Input Status <10><13>"
              "  I0..7    :$I0000 <10><13>"
              "  O16..23:$O0016 <10><13>"
              "  Main menu  F1"

TEXT      3      "<12>"                                ; Clear display
              "  BCD-Value I0...7 <10><13>"
              "  ----- <10><13>"
              "  Value   : $R0010 <10><13>"
              "  Main menu  F1"

TEXT      4      "<12>"                                ; Clear display
              "  Date   : $D<10><13>"
              "  Week  : $W<10><13>"
              "  Time  : $H<10><13>"
              "  Main menu F1"

TEXT      5      "<27><64><77><49>"                    ; Display mode:4 x 20

TEXT      10     "<ESC>@B"                            ; Poll command

TEXT      100    "UART:9600,8,E,1;MODE:MC0;DIAG:O16,R100"

; Symboldefinitions
; =====
; Diagnostic outputs serial interface
;-----
RBSY      EQU    0      16          ; Receiver Busy
RFUL      EQU    0      RBSY+1     ; Receive Buffer Full
RDIA      EQU    0      RBSY+2     ; Receiver Diagnostic
TBSY      EQU    0      RBSY+3     ; Transmitter Busy
TFUL      EQU    0      RBSY+4     ; Transmit Buffer Full
TDIA      EQU    0      RBSY+5     ; Transmitter Diagnostic
XBSY      EQU    0      RBSY+6     ; Text Busy
NEXE      EQU    0      RBSY+7     ; Not Executed
;-----
; Function/Program blocks
;-----
READ      EQU    FB     0          ; Read character
SEND      EQU    FB     1          ; Send text
COMPARE   EQU    PB     0          ; Compare received character
;-----
; Register
;-----
RBUF_R    EQU    R      1000
;

```

```

;-----
; Coldstart
;-----
XOB      16
SASI     1          ; Assignment interface n°1
          100       ; Text 100

;-----
;
termpoll: stxt      1          ; START OF THE POLL COMMAND
          10
          sth      0 22
          jr       h -1

          acc      h          ; (ld T is accu dependent)
          ld       T 0        ; start short receive timeout
          8         ; (must be min. 10 mS)
          9

termwait: sth      0 16        ; character received?
          jr       h termok   ; yes
          sth      T 0
          jr       h termwait ; loop for timeout period
          jr       termpoll   ; Terminal not ready, repeat the poll

termok:   srxd     1          ; read the character
          R 1
          cmp      R 1        ; SOH character ?
          1
          jr       z termready ; yes, Terminal is ready
          jr       termpoll   ; no, repeat the poll

termready: ld      R 1        ; clear receive register
          0         ; END OF THE POLL COMMAND

;-----
;
          STXT     1          ; Display mode : 4 x 20
          5

          sth      0 22
          jr       h -1

          STXT     1          ; Text 1: Main menu
          1
          sth      0 22
          jr       h -1

          EXOB

;-----
;

```

```

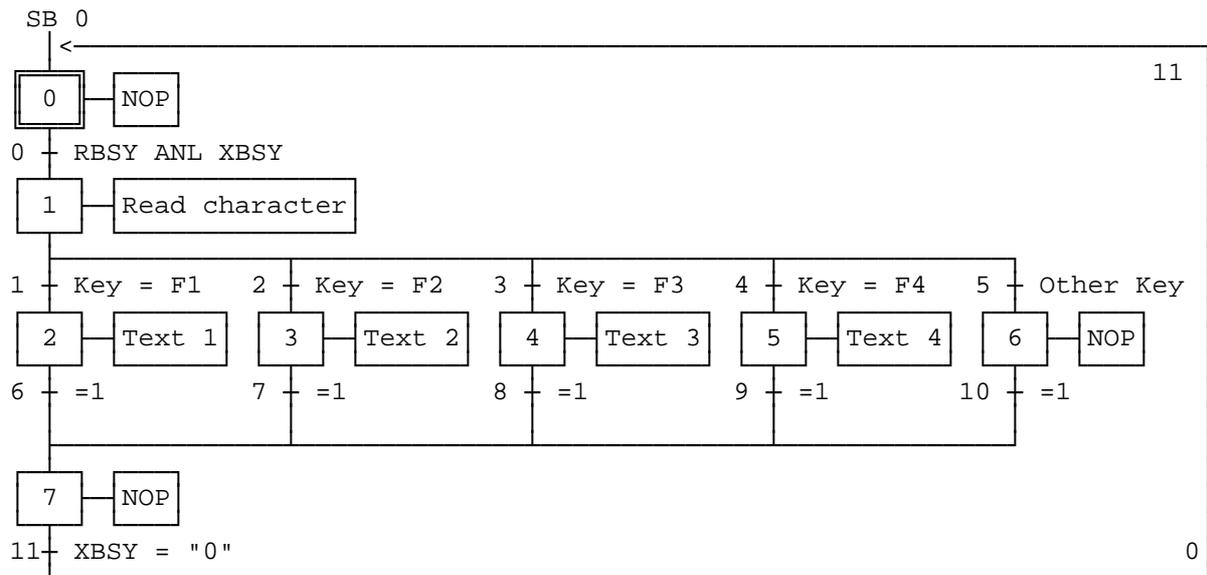
;-----
; Main program
;-----
COB    0
        0
CSB    0
;

```

```

; Read BCD-Switch
DIGI   2
        I    0
        R    10
ECOB
;

```



```

SB    0
;
IST    0          ; NOP
        I    11      ; XBSY = "0"
        O    0          ; RBSY ANL XBSY
EST
;

```

```

ST    1          ; Read character
        I    0          ; RBSY ANL XBSY
        O    1          ; Key = F1
        O    2          ; Key = F2
        O    3          ; Key = F3
        O    4          ; Key = F4
        O    5          ; Other key
SRXD  1
        R          RBUF_R
EST
;

```

```

ST    2          ; Text 1
        I    1          ; Key = F1
        O    6          ; =1
STXT  1          ; send
        1          ; text 1
EST
;

```

```

      ST      3          ; Text 2
      I      2          ; Key = F2
      O      7          ; =1
      STXT   1          ; send
      EST    2          ; text 2
;
-----
      ST      4          ; Text 3
      I      3          ; Key = F3
      O      8          ; =1
      STXT   1          ; send
      EST    3          ; text 3
;
-----
      ST      5          ; Text 4
      I      4          ; Key = F4
      O      9          ; =1
      STXT   1          ; send
      EST    4          ; text 4
;
-----
      ST      6          ; NOP
      I      5          ; Other key
      O     10          ; =1
      EST
;
-----
      ST      7          ; NOP
      I      6          ; =1
      I      7          ; =1
      I      8          ; =1
      I      9          ; =1
      I     10          ; =1
      O     11          ; XBSY = "0"
      EST
;
-----
      TR      0          ; RBSY ANL XBSY
      I      0          ; NOP
      O      1          ; Read character
      STH    0      16   ; Receiver busy
      ANL    0      22   ; Text busy
      ETR
;
-----
      TR      1          ; Key = F1
      I      1          ; Read character
      O      2          ; Text 1
      CMP    R      RBUF_R
      K      65         ; F1
      ACC    Z
      ETR
;
-----
      TR      2          ; Key = F2
      I      1          ; Read character
      O      3          ; Text 2
      CMP    R      RBUF_R
      K      66         ; F2
      ACC    Z
      ETR
;
-----

```

```

TR      3
I      1
O      4
CMP    R      RBUF_R
      K      67
ACC
ETR
; Key = F3
; Read character
; Text 3

;
-----
TR      4
I      1
O      5
CMP    R      RBUF_R
      K      68
ACC
ETR
; Key = F4
; Read character
; Text 4

; F4

;
-----
TR      5
I      1
O      6
ETR
; Other key
; Read character
; NOP

;
-----
TR      6
I      2
O      7
ETR
; =1
; Text 1
; NOP

;
-----
TR      7
I      3
O      7
ETR
; =1
; Text 2
; NOP

;
-----
TR      8
I      4
O      7
ETR
; =1
; Text 3
; NOP

;
-----
TR      9
I      5
O      7
ETR
; =1
; Text 4
; NOP

;
-----
TR      10
I      6
O      7
ETR
; =1
; NOP
; NOP

;
-----
TR      11
I      7
O      0
STL   O      22
ETR
; XBSY = "0"
; NOP
; NOP
; Text busy

;
-----
ESB

```

## 8.5 Eingabe numerischer Werte

---

Menügeführt soll der Inhalt eines Registers und Zählers via Terminal verändert werden.

Bedingungen:

- Für das Register sollen Werte mit oder ohne negatives Vorzeichen unter Verwendung des Festkommaformates eingegeben werden können.
- Für den Zähler sollen nur positive Werte ohne Dezimalpunkt eingegeben werden können.

Um diese Funktion zu realisieren, wurde der universelle Funktionsblock **"INPUT"** entwickelt. Eine detaillierte Beschreibung des Funktionsblockes befindet sich im Anschluss an die Programmlistings.

### Eingabe numerischer Werte

```

;-----
;
; User program example 8.5 for the industrial terminal PCD7.D250
; =====
; Input of numerical parameters
;
; File:          DEMO.SRC
; Version:      1.0
;
; Creation:     21.01.93      U.Jäggi
; Modified:    08.02.00      C. Bruegger
;-----

```

```

RBSY_F      EQU      0 32          ; Receiver Busy
             PUBL     RBSY_F
XBSY_F      EQU      0 38          ; Text Busy
             PUBL     XBSY_F
SIGN        EQU      0 48          ; Sign input
IN_BUSY     EQU      0 49          ; Input busy
             PUBL     IN_BUSY
DIGIT       EQU      R 0           ; Number of digits
X_POS       EQU      R 1           ; X-position
Y_POS       EQU      R 2           ; Y-position
DECIMAL     EQU      R 3           ; Number of decimal places
             DOC      R 500
DIAG_R      EQU      R 999         ; Diagnostic register
             DOC      R 1000
             DOC      C 100
MAIN        EQU      TEXT 0        ; Main menue
IN_TXT_R    EQU      TEXT 1        ; Input text register
IN_TXT_C    EQU      TEXT 2        ; Input text counter
DISP        EQU      TEXT 10

ASSIGN      EQU      TEXT 999      ; Assignment of the serial interfac
CHAN_N      EQU      1             ; Number of serial channel
             PUBL     CHAN_N
             DOC      COB 0
             DOC      XOB 16
INPUT       EQU      FB 0          ; Functionblock input
             PUBL     INPUT

TEXT        ASSIGN      "UART:9600,8,E,1;"
             "MODE:MC0;"
             "DIAG: ",RBSY_F.T," ",DIAG_R.T," "

TEXT        DISP        "<27><64><77><49>"          ; Display mode:4 x 20

TEXT        MAIN        "<12>"                ; Clear display
             "<27><84>"                ; Cursor off
             "=="PARAMETER INPUT=="
             "-----"
             "MODIFY REGISTER [F1]"
             "MODIFY COUNTER [F2]"

TEXT        IN_TXT_R    "<12>"
             "R-Value : $%00.3d$R0500<10><13>"
             "ACCEPT VALUE [CR]"
             "MODIFY VALUE [F1]"
             "MAIN MENU [F2]"

TEXT        IN_TXT_C    "<12>"
             "C-Value : $C0100<10><13>"
             "ACCEPT VALUE [CR]"
             "MODIFY VALUE [F1]"
             "MAIN MENU [F2]"

```

```

;-----
; Coldstart
;-----

XOB      16

SASI     CHAN_N      ; Assingation RS232 interface
         ASSIGN      ; Text 100
ACC      H
RES      IN_BUSY    ; Reset input busy flag

STXT     CHAN_N
         DISP

EXOB

;=====

;-----
; Main program
;-----

COB      0
         0

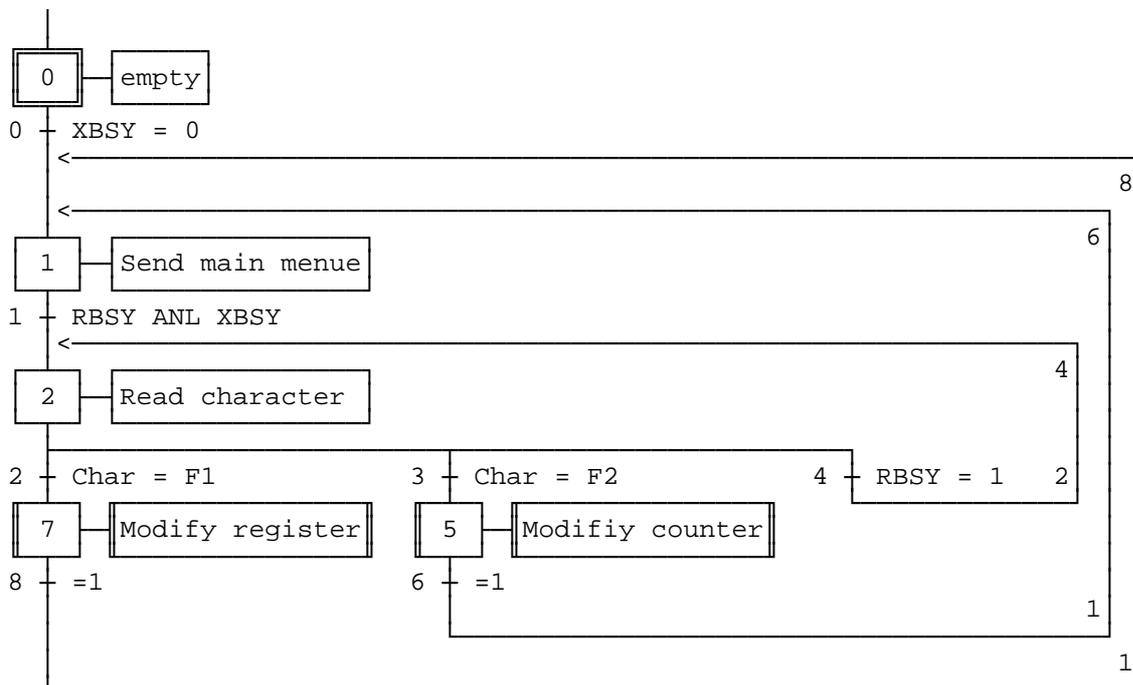
CSB      0          ; Call communication SB 0

ECOB

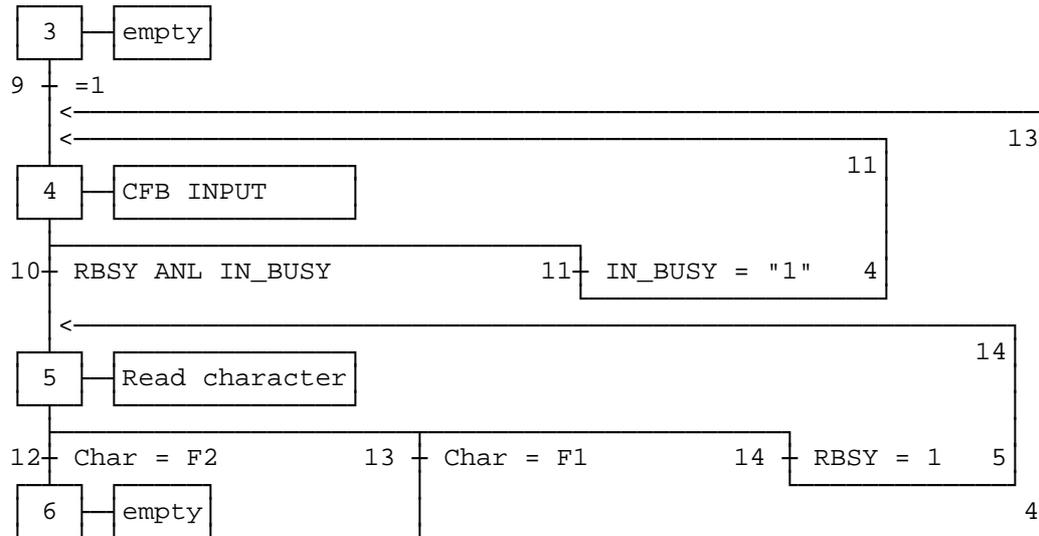
;-----

```

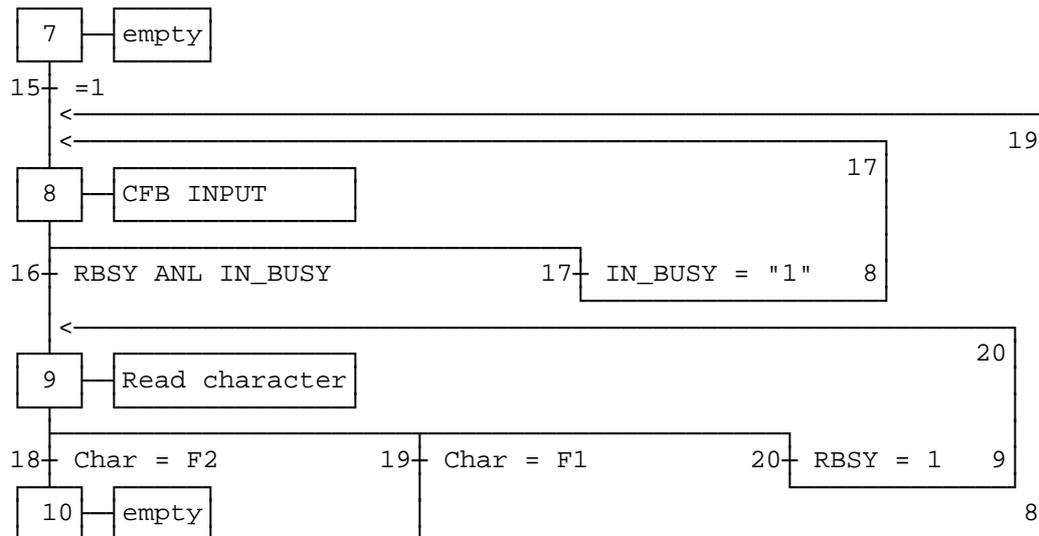
SB 0



Page Nb 5, Modify counter



Page Nb 7, Modify register



```

;
;          SB      0
;
;-----
;          IST      0          ; empty
;          O 0      ; XBSY = 0
;          EST
;
;-----
;          ST       1          ; Send main menue
;          I 0      ; XBSY = 0
;          I 8      ; =1
;          I 6      ; =1
;          O 1      ; RBSY ANL XBSY
;          STXT    CHAN_N      ; Send
;          MAIN     ; the main menue
;          EST
;
;-----
;

```

```

ST      2          ; Read character
        I 1        ; RBSY ANL XBSY
        I 4        ; RBSY = 1
        O 2        ; Char = F1
        O 3        ; Char = F2
        O 4        ; RBSY = 1
SRXD    CHAN_N    ; Read character
        R 1000    ; from the receive buffer
EST
;
-----
ST      3          ; empty
        I 3        ; Char = F2
        O 9        ; =1
EST
;
-----
ST      4          ; CFB INPUT
        I 9        ; =1
        I 11       ; IN_BUSY = "1"
        I 13       ; Char = F1
        O 10       ; RBSY ANL IN_BUSY
        O 11       ; IN_BUSY = "1"
RES     SIGN      ; Sign input not allowed
LD      X_POS     ; X-position
        42
LD      Y_POS     ; Y-position
        32
LD      DIGIT     ; Number of digits
        9
LD      DECIMAL   ; Number of decimal places
        0
CFB     INPUT     ; D100 input
        IN_TXT_C  ; Input text counter
        C 100    ; Counter to be modified
        DIGIT    ; Number of digits
        DECIMAL  ; Number of decimal places
        X_POS    ; X-position
        Y_POS    ; Y-position
        SIGN     ; Sign input yes/no (1/0)
EST
;
-----
ST      5          ; Read character
        I 10       ; RBSY ANL IN_BUSY
        I 14       ; RBSY = 1
        O 12       ; Char = F2
        O 13       ; Char = F1
        O 14       ; RBSY = 1
SRXD    CHAN_N    ; Read character
        R 1000    ; from the receive buffer
EST
;
-----
ST      6          ; empty
        I 12       ; Char = F2
        O 6        ; =1
EST
;
-----
ST      7          ; empty
        I 2        ; Char = F1
        O 15       ; =1
EST
;
-----

```

```

ST      8          ; CFB INPUT
        I 15       ; =1
        I 17       ; IN_BUSY = "1"
        I 19       ; Char = F1
        O 16       ; RBSY ANL IN_BUSY
        O 17       ; IN_BUSY = "1"
SET     SIGN      ; Sign input allowed
LD      X_POS     ; X-position
        42
LD      Y_POS     ; Y-position
        32
LD      DIGIT     ; Number of digits
        9
LD      DECIMAL   ; Number of decimal places
        3
CFB     INPUT     ; D100 input
        IN_TXT_R  ; Input text register
        R 500     ; Register to be modified
        DIGIT    ; Number of digits
        DECIMAL  ; Number of decimal places
        X_POS    ; X-position
        Y_POS    ; Y-position
        SIGN     ; Sign input yes/no (1/0)
EST
;-----.

ST      9          ; Read character
        I 16       ; RBSY ANL IN_BUSY
        I 20       ; RBSY = 1
        O 18       ; Char = F2
        O 19       ; Char = F1
        O 20       ; RBSY = 1
SRXD   CHAN_N    ; Read character
        R 1000    ; from the receive buffer
EST
;-----.

ST      10         ; empty
        I 18       ; Char = F2
        O 8        ; =1
EST
;=====
TR      0          ; XBSY = 0
        I 0        ; empty
        O 1        ; Send main menu
STL    XBSY_F
ETR
;-----.

TR      1          ; RBSY ANL XBSY
        I 1        ; Send main menu
        O 2        ; Read character
STH    RBSY_F
ANL    XBSY_F
ETR
;-----.

TR      2          ; Char = F1
        I 2        ; Read character
        O 7        ; empty
CMP    R 1000
        K 65       ; F1
ACC    Z
ETR
;-----.

```

```

TR      3          ; Char = F2
        I 2        ; Read character
        O 3        ; empty
CMP     R 1000
        K 66      ; F2
ACC     Z
ETR
;
-----
TR      4          ; RBSY = 1
        I 2        ; Read character
        O 2        ; Read character
STH     RBSY_F
ETR
;
-----
TR      5          ; Modify counter
        I 3        ; empty
        O 6        ; empty
ETR
;
-----
TR      6          ; =1
        I 6        ; empty
        O 1        ; Send main menu
ETR
;
-----
TR      7          ; Modify register
        I 7        ; empty
        O 10       ; empty
ETR
;
-----
TR      8          ; =1
        I 10       ; empty
        O 1        ; Send main menu
ETR
;
-----
TR      9          ; =1
        I 3        ; empty
        O 4        ; CFB INPUT
ETR
;
-----
TR      10         ; RBSY ANL IN_BUSY
        I 4        ; CFB INPUT
        O 5        ; Read character
STH     RBSY_F
ANL     IN_BUSY
ETR
;
-----
TR      11         ; IN_BUSY = "1"
        I 4        ; CFB INPUT
        O 4        ; CFB INPUT
STH     IN_BUSY
ETR
;
-----
TR      12         ; Char = F2
        I 5        ; Read character
        O 6        ; empty
CMP     R 1000
        K 66      ; F2
ACC     Z
ANL     XBSY_F
ETR
;
-----

```

```

TR      13                ; Char = F1
        I 5                ; Read character
        O 4                ; CFB INPUT
CMP     R 1000
        K 65                ; F1
ACC     Z
ANL    XBSY_F
ETR

;
-----
TR      14                ; RBSY = 1
        I 5                ; Read character
        O 5                ; Read character
STH    RBSY_F
ETR

;
-----
TR      15                ; =1
        I 7                ; empty
        O 8                ; CFB INPUT
ETR

;
-----
TR      16                ; RBSY ANL IN_BUSY
        I 8                ; CFB INPUT
        O 9                ; Read character
STH    RBSY_F
ANL    IN_BUSY
ETR

;
-----
TR      17                ; IN_BUSY = "1"
        I 8                ; CFB INPUT
        O 8                ; CFB INPUT
STH    IN_BUSY
ETR

;
-----
TR      18                ; Char = F2
        I 9                ; Read character
        O 10               ; empty
CMP     R 1000
        K 66                ; F2
ACC     Z
ANL    XBSY_F
ETR

;
-----
TR      19                ; Char = F1
        I 9                ; Read character
        O 8                ; CFB INPUT
CMP     R 1000
        K 65                ; F1
ACC     Z
ANL    XBSY_F
ETR

;
-----
TR      20                ; RBSY = 1
        I 9                ; Read character
        O 9                ; Read character
STH    RBSY_F
ETR

ESB

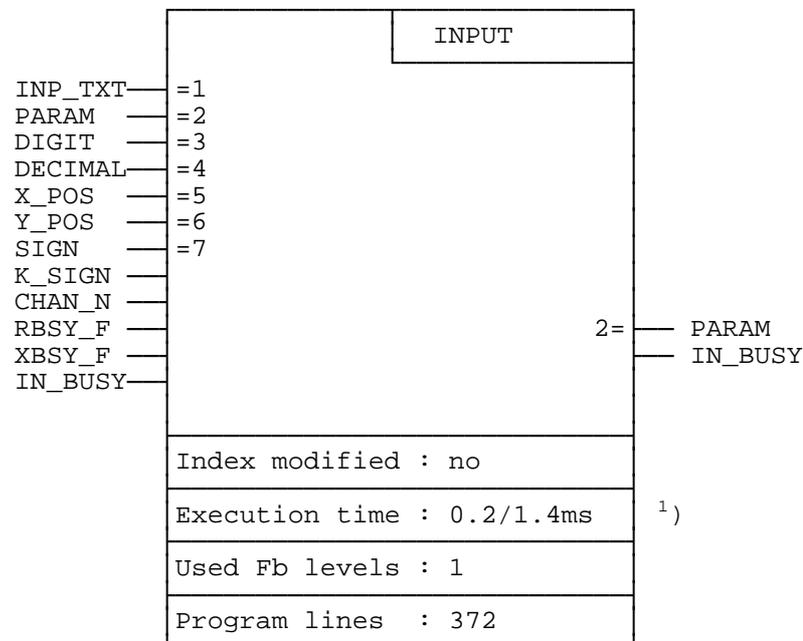
;
-----

```

## 8.6 Funktionsblock: INPUT

---

Parametereingabe mit dem Industrieterminal PCD7.D250:



- 1) 0.2 ms: RBSY\_F = "0" (kein Zeichen im Empfangsbuffer)  
 1.4 ms: RBSY\_F = "1" (Ziffer 0..9 wird vom Empfangsbuffer gelesen und verarbeitet.)

### Funktionsbeschreibung

Mit diesem Funktionsblock kann der Inhalt eines Registers oder Zählers via Industrieterminal PCD7.D250 verändert werden. Die Eingabe des negativen Vorzeichen und eines Dezimalpunktes wird unterstützt.

**Liste der Ein- und Ausgänge**

Symbol	Beschreibung	Parameter	Daten			Adresse
			Typ	Format	Wert	
INP_TXT	Input Text Eingabe Text	yes	X	Text	beliebig	0...3999
PARAM	Input parameter Eingabe Register oder Zähler	yes	R/C	Integer	- 2 147 483 648... + 2 147 483 647	0...4095
DIGIT	Number of digits Anzahl Eingabestellen	yes	R	Integer	1...11	0...4095
DECIMAL	Number of decimal places Anzahl Nachkommastellen	yes	R	Integer	0,1...10	0...4095
X_POS	Cursor x-position	yes	R	Integer	32...51	0...4095
Y_POS	Cursor y-position	yes	R	Integer	32...35	0...4095
SIGN	Sign yes/no (1/0)	yes	F/I/O	Binary	0/1	0...8191
K_SIGN	Sign key ASCII-Code der Vorzeichenaste	no	K	ASCII	0...255	-
CHAN_N	Serial channel number Schnittstellenummer	no	K	Number	0...3	-
RBSY_F	Receive busy flag	no	F/O	Binary	0/1	0...8191
XBSY_F	Text busy flag	no	F/O	Binary	0/1	0...8191
IN_BUSY	Input Busy Eingabe Busy Flag	no	F/O	Binary	0/1	0...8191

Intern verwendete und reservierte Elemente mit symbolischem Namen:

Von dem Fb werden intern 7 Arbeitsregister und 6 Arbeitsflag verwendet. Diese Register und Flag enthalten während der Eingabe eines Parameters Zwischenergebnisse und dürfen deshalb nur für diesen FB verwendet werden. In der FB-Datei müssen nur die Basisadressen der verwendeten Elemente definiert werden.

Symbol	Beschreibung	Daten		Adresse
		Typ	Format	
WORK_R	Base address of 7 used work registers Basisadresse von 7 Arbeitsregistern	R	Integer	0...4089 (+6)
WORK_F	Base address of 6 used work flags Basisadresse von 6 Arbeitsflag	F	Binary	0...8186 (+5)

**Tastenzuweisung:**

Da anwendungsspezifische Tastenbezeichnungen verwendet werden können, kann den Tasten durch die Verwendung von Symbolen ein beliebiger ASCII-Code zugewiesen werden. Die voreingestellten Symbolzuweisungen entsprechen der Standard-Tastenbezeichnung.

Die Zuweisung der numerischen Tasten (0...9) darf nicht verändert werden, da sonst die Umrechnung des ASCII-Wertes in den entsprechenden Dezimal-Wert einen Fehler ergibt.

Symbol	Beschreibung	Daten		Wert
		Typ	Format	
K_BS	Backspace key	K	ASCII	0...255
K_CR	Carriage return key	K	ASCII	0...255
K_DP	Decimalpoint key	K	ASCII	0...255
K_SIGN	Negative sign key	K	ASCII	0...255
K_0	0 key	K	ASCII	48
K_1	1 key	K	ASCII	49
K_2	2 key	K	ASCII	50
K_3	3 key	K	ASCII	51
K_4	4 key	K	ASCII	52
K_5	5 key	K	ASCII	53
K_6	6 key	K	ASCII	54
K_7	7 key	K	ASCII	55
K_8	8 key	K	ASCII	56
K_9	9 key	K	ASCII	57

**Aufruf des Funktionsblockes**

```

CFB      INPUT      ; Input
          INP_TXT    ; Input text
R        PARAM      ; Parameter
R        DIGIT      ; Number of digits
R        DECIMAL    ; Number of decimal places
R        X_POS      ; X-position
R        Y_POS      ; Y-position
F        Sign       ; Sign input yes/no (1/0)

```

### Detaillierte Beschreibung der Ein/Ausgänge

- Eingabetext "INP\_TXT":

Der Eingabetext wird beim ersten Aufruf des FB über die Schnittstelle ausgegeben. Der zu ändernde Parameter (Register/Zähler) wird vom FB nur durch die Ausgabe des Eingabetextes am Terminal angezeigt und kann deshalb mittels Sondertext im Eingabetext ausgegeben werden. Ansonsten kann der Eingabetext von beliebiger Grösse und Inhalt sein. Das Ausgabeformat des Parameters kann beliebig gewählt werden. Es empfiehlt sich jedoch für die Ausgabe und Eingabe des Parameters das gleiche Format zu wählen.

Beispiel:

```
TEXT    INP_TXT "<12>"                                ; Clear display
        "PARAMETER INPUT<10><13>"
        "=====<10><13>"
        "Value : $%00.3d$",PARAM.04T,"<10><13>"
        "Accept value [CR]"
```

- Eingabe Parameter "PARAM":

Als Eingabe-Parameter muss das/der zu ändernde Register/Zähler angegeben werden.

- Eingabe Stellen "DIGIT":

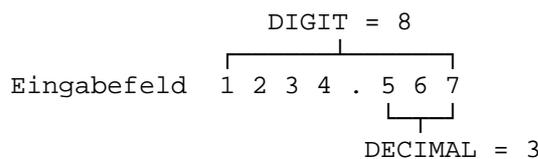
Mit diesem Wert ist die Grösse des Eingabefeldes für den Parameter definiert. Der Wert in dem Register definiert die Anzahl Stellen inklusive negatives Vorzeichen und Dezimalpunkt. Die maximale Anzahl Stellen wird während der Eingabe überwacht und nötigenfalls begrenzt.

Bemerkung: das letzte Digit (Cursor X-Position 51) der Anzeigenszeile darf für das Eingabefeld nicht verwendet werden.

- Nachkommastellen "DECIMAL":

Definiert die Anzahl Nachkommastellen. Es wird das Festkommaformat angewendet. Soll für die Eingabe das Format ohne Dezimalpunkt (Normalformat) zur Anwendung kommen, so muss der Wert 0 in das Register geladen werden. Die Anzahl der Kommastellen werden während der Eingabe überwacht und nötigenfalls begrenzt.

Beispiel:



- Cursorposition "X\_POS"/"Y\_POS":

Definiert die Position für die Eingabe der ersten Ziffer des Eingabeparameters.

- Vorzeichen "SIGN":  
Mit diesem Flag kann die Eingabe des negativen Vorzeichens zugelassen, respektive gesperrt werden.  
"SIGN" = 0 → Eingabe des negativen Vorzeichen ist gesperrt.  
"SIGN" = 1 → Eingabe des negativen Vorzeichen ist möglich.
- Vorzeichen Taste "K\_SIGN":  
Definiert den ASCII-Code für die Minuszeichentaste. Eingabe des positiven Vorzeichens ist nicht zugelassen.
- Schnittstellennummer "CHAN\_N":  
Definiert die Nummer der seriellen Schnittstelle. Die Schnittstelle muss vor dem Aufruf des FB im Mode C assigniert sein. (Mögliche Assignierungsmodi für das Terminal PCD7.D250 : MC0, MC1 und MC2)
- Diagnoseflag "RBSY\_F"/"XBSY\_F" der seriellen Schnittstelle:  
Die Adressen für das RBSY\_F und XBSY\_F müssen mit der Adresse der Diagnoseflags übereinstimmen, welche mit der SASI-Instruktion definiert wurden.
- Eingabe Busy Flag "IN\_BUSY":  
Vor dem ersten Aufruf des FB muss das Input Busy Flag "IN\_BUSY" Null sein. Andernfalls wird der FB nicht korrekt funktionieren.  
(→ "IN\_BUSY" Flag zurücksetzen im XOB 16). Das Flag wird beim ersten Aufruf des FB hoch gesetzt. Nachdem vom Terminal ein Carriage Return empfangen wurde, wird das Flag zurückgesetzt.

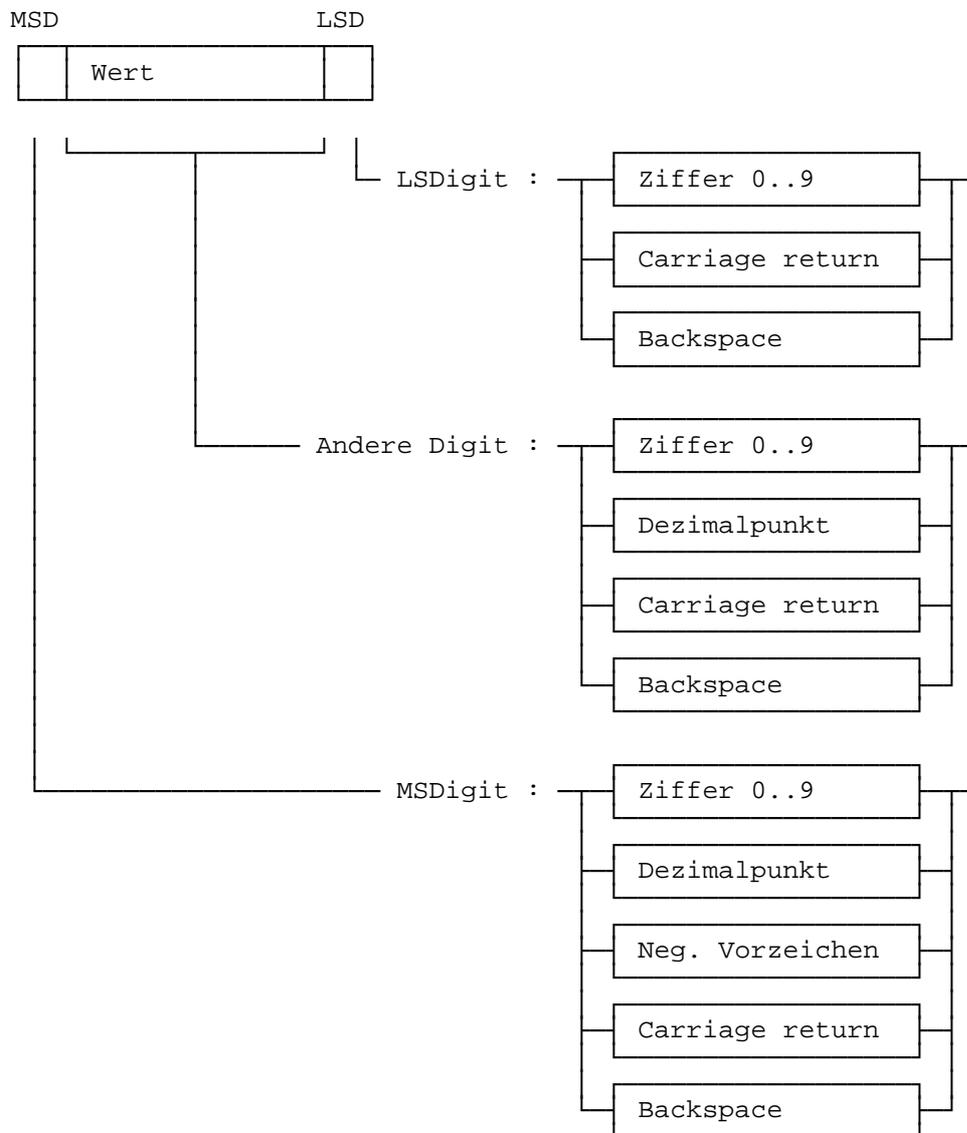
### **Eingabe/Änderung eines Parameters**

Beim ersten Aufruf des FB wird der Eingabetext ausgegeben, das "IN\_BUSY" gesetzt und der Cursor entsprechend der FB-Parameter "X\_POS" und "Y\_POS" positioniert. In der Folge kann über die Tastatur eine Zahl (mit Minuszeichen) eingegeben werden. Ist die erste gedrückte Taste eine Ziffer, das Vorzeichen oder der Dezimalpunkt, so wird das Eingabefeld für den Parameter gelöscht.

Die Grösse des Eingabefeldes ist limitiert durch die maximale Anzahl Stellen (definiert mit FB-Parameter "DIGIT") der Zahl. Die maximale Anzahl Stellen der Zahl wird vom FB während der Eingabe überwacht und nötigenfalls begrenzt. Sobald die Return (CR) Taste gedrückt wurde, wird der eingegebene Wert im Register/Zähler "PARAM" abgespeichert, das "IN\_BUSY"-Flag zurückgesetzt und die Eingabe somit abgeschlossen.

Der FB muss während der Eingabe eines Wertes (solange das "IN\_BUSY"-Flag hoch ist) vom Anwenderprogramm zyklisch aufgerufen werden.

Eingabeformat des numerischen Wertes am Terminal PCD7.D250:



Das folgende Beispiel soll das Funktionsprinzip der Eingabe zeigen.

Die FB-Parameter enthalten folgende Werte:

```
Eingabetext  "INP_TXT"  : "<12>" ; Clear display
                "PARAMETER INPUT<10><13>"
                "=====<10><13>"
                "Value : $%00.3d$",PARAM.04T,"<10><13>"
                "Accept value [CR]"

Register  "PARAM"      : 567890
Register  "DIGIT"      : 8
Register  "DECIMAL"    : 3
Register  "X_POS"      : 40
Register  "Y_POS"      : 34
```

Als Minuszeichen wird die Minustaste definiert:

```
K_SIGNEQUK 45 ; Negative sign key
```

Nach dem ersten Aufruf des FB erscheint der folgende Text auf der Anzeige:

```
PARAMETER INPUT
=====
Value : 567.890
Accept value [CR]
```

Jetzt wird der Parameter gemäss nachfolgender Tabelle eingegeben. In der Tabelle ist nur das Eingabefeld des Parameters dargestellt, da die übrige Anzeige während der Eingabe unverändert erhalten bleibt.

Gedrückte Taste	ASCII Code dez.	Anzeige des Eingabefeld (Max. 8 Stellen)	Register/ Zähler "PARAM"	Input Busy Flag "IN_BUSY"
(1. FB-Aufruf)		567.890	567890	0
3	51	3_	567890	1
5	53	35_	567890	1
7	55	357_	567890	1
<-	8	35_	567890	1
<-	8	3_	567890	1
<-	8	_	567890	1
<-	8	567.890	567890	1
1	49	1_	567890	1
2	50	12_	567890	1
3	51	123_	567890	1
4	52	1234_	567890	1
.	54	1234._	567890	1
7	55	1234.7_	567890	1
8	56	1234.78_	567890	1
9	57	1234.789	567890	1
4	52	1234.784	567890	1
<-	8	1234.78_	567890	1
<-	8	1234.7_	567890	1
CR	13	1234.7	1234700	0
(1. FB-Aufruf)		1234.700	1234700	0
-	45	_	1234700	1
8	56	-8_	1234700	1
4	52	-84_	1234700	1
6	54	-846_	1234700	1
CR	13	-846	-846000	0

### Verwendung des Funktionsblockes im Anwenderprogramm

Der FB befindet sich in der Datei D1\_INP.SRC. Die Datei enthält nebst dem FB auch alle Symboldefinition, welche zu dessen Benützung notwendig sind.

Alle Symbole, welche global im Anwenderprogramm verwendet werden, sind in der Datei als EXTN definiert und müssen in einer andern Anwenderdatei definiert werden. Auf diese Weise muss die Datei D1\_INP.SRC nur einmal assembliert und anschliessend mit den andern Anwenderdateien gelinkt werden.

Globale Symbole: INPUT, CHAN\_N, IN\_BUSY, RBSY\_F, XBSY\_F

Soll der FB durch die Assemblerdirective \$INCLUDE in das Anwenderprogramm eingebunden werden, so müssen die EXTN-Definition gelöscht, respektive durch lokale Symboldefinitionen ersetzt werden.

## 9. Vergleich der wichtigsten Eigenschaften PCD7.D202 ↔ PCD7.D250

Funktionen	..D202	..D250
Front		
Display Abmessungen Front Funktionstasten Einschiebestreifen Basisprint Schnittstellen	4 x 20 Charakter 141 x 181 mm 4 mit LED über Funktionstasten = Nr. 1: RS 232	8 x 40 / 4 x 20 Charakter (umschaltbar) 305 x 120 mm 8 mit LED über Funktionstasten und SAIA-Label = Nr. 1: RS 232 oder Nr. 0: für Modul ..F2.. (RS 422 oder TTY)
Befehlssatz / Setup	..D202	..D202 plus: – Umschalten Modus 8 x 40 oder 4 x 20 – Cursor-Positionierung erweitert (8 x 40) – Umschaltung Schnittstelle 1 → 0 – ganzer Zeichensatz IBM CodePage 437 – Anfang und Ende invertierte Schrift – Transparent Modus
FBs Editor-Tool (Dialog Library)	..D202 alte und neue Version	..D202 modifiziert HMI-Editor in Vorbereitung

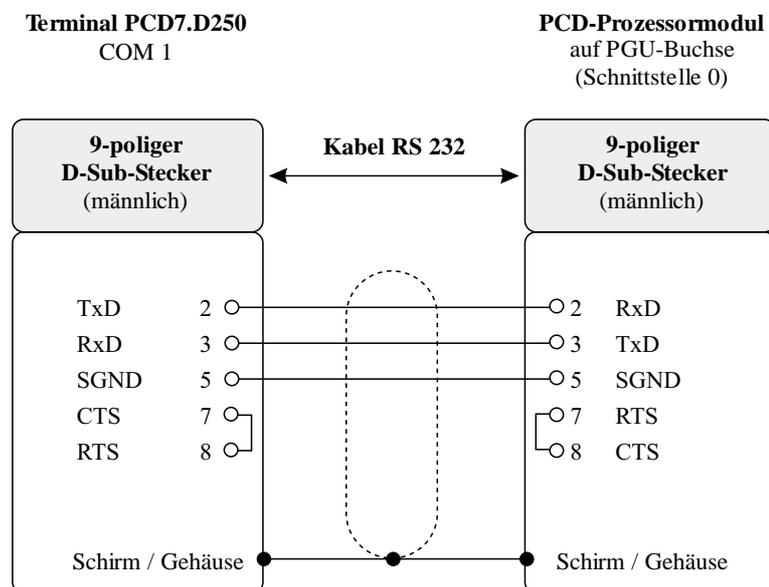
Notizen:

## 10. Schnittstellen-Verbindungskabel RS 232

Die Kabel sind doppelt abgeschirmt und weisen metallische Steckergehäuse auf. Standardlänge 2.5 m.

**Typ PCD7.K412:** Betriebskabel RS 232 ohne RTS/CTS

Einsatz zwischen ..D250 (COM 1) und der PGU-Buchse (Kanal 0) aller PCD-Prozessormodule.

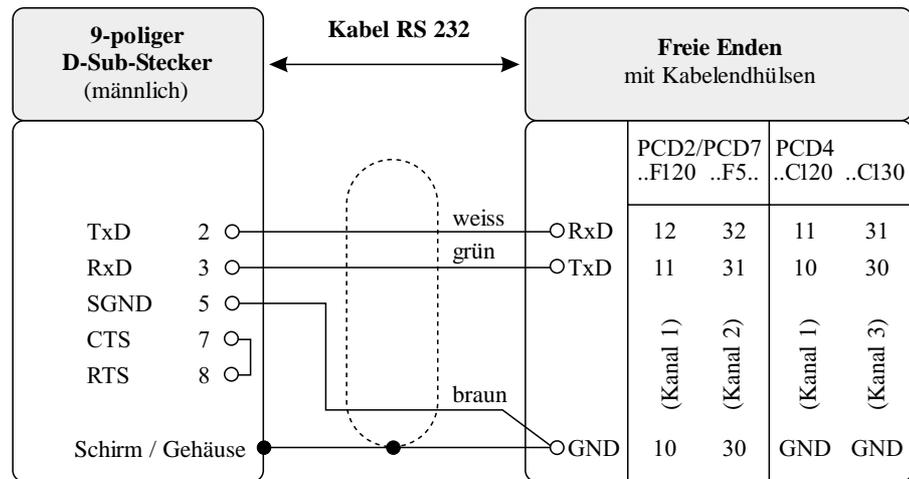


**Typ PCD7.K422:** Betriebskabel RS 232 ohne RTS/CTS

Einsatz zwischen ..D250 (COM 1) und dem Basismodul PCD1 und PCD2 bzw. dem Busmodul von PCD4. Freie Kabelenden für Schraubanschlüsse.

**Terminal PCD7.D250**  
COM 1

**PCD-Prozessor- bzw. Busmodul**  
auf PCD1, PCD2 und PCD4  
(Schraubklemmen)



Absender:

Firma  
Abteilung  
Name  
Adresse

Tel.

Datum

An:

SAIA-Burgess Electronics AG  
Bahnhofstrasse 18  
CH-3280 Murten (Schweiz)  
<http://www.saia-burgess.com>

GB: Electronic Controllers

Industrie-Terminal PCD7.D250

Falls Sie Vorschläge zu SAIA® PCD zu machen oder Fehler in diesem Handbuch gefunden haben, sind wir Ihnen für einen kurzen Bericht dankbar.

Ihre Vorschläge: