



SAIA® PCD

Prozess-Steuergeräte

Industrielle Bedienterminals und Displaymodule PCD 7.D1.. PCA2.D1..





SAIA[®] Prozess-Steuergeräte

Handbuch

**Industrielle Bedienterminals
und Displaymodule**

SAIA AG 1993 Alle Rechte vorbehalten

Ausgabe 26 / 738 D1

Stand 02/93

Technische Änderungen vorbehalten

Verkaufspreis sFr. 50.-

Inhaltsverzeichnis

1. Bedienterminal PCD7.D1..

1.1	Anwendung	1-1
1.2	Technische Daten	1-2
1.3	Typenübersicht und Anschlussarten	1-3
1.4	Massbild	1-4
1.5	Hardware	1-5
	1.5.1 Rückansicht	1-5
	1.5.2 Stromversorgung	1-6
	1.5.3 Serielle Schnittstellen	1-7
	1.5.4 Funktion der DIP-Schalter	1-11
1.6	Die Tastatur	1-16
	1.6.1 Standard-Tastatur	1-16
	1.6.2 Anwendungsspezifische Tastenbezeichnungen	1-16
1.7	Die Anzeige	1-18
	1.7.1 Zeichensatz	1-18
	1.7.2 Einschaltvorgang	1-19
	1.7.3 Steuerfunktionen	1-21
1.8	Programmbeispiele PCD2/4/6 mit ..D100 und D110 für Kommunikation ohne Protokoll	1-25
1.9	Das GMP-Protokoll auf RS485 Bus	1-59
1.10	Programmbeispiele PCD2/4/6 mit ..D110 für Kommunikation mit GMP-Protokoll	1-63
1.11	Das S-Bus Protokoll auf den Schnittstellen RS422 und RS485	1-79
1.12	Programmbeispiele PCD2/4/6 mit ..D110 für Kommunikation mit S-Bus Protokoll	1-84

2. PCA2.D12 Display-Modul mit 4 Ziffern

3. PCA2.D14 Display-Modul mit 2x6 Ziffern



Wichtiger Hinweis:

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

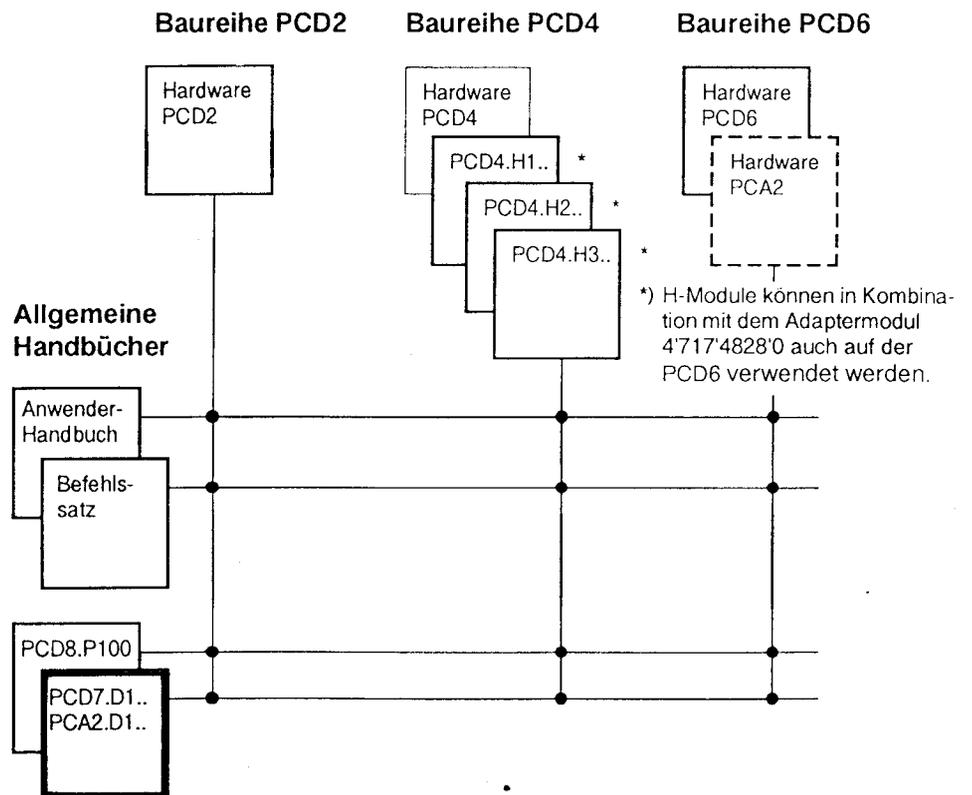
SAIA° PCD sind qualitativ hochwertige elektronische Geräte, welche während ihrer Entstehung bei der Entwicklung, bei der Auswahl und Prüfung der Komponenten, bei den "In circuit tests", beim "Burn in" und bei den verschiedenen Funktionskontrollen strengen Prüfungen unterzogen werden.

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

Dann allerdings werden auch Sie zum grossen Kreis der begeisterten SAIA° PCD Anwender gehören.

Sollten Sie aber Verbesserungsvorschläge zur Technik oder zu diesem Handbuch haben, so lassen Sie uns das bitte wissen. Sie verwenden dann am einfachsten eine Kopie des letzten Blattes dieses Handbuches.

Übersicht



1. Industrielle Bedienterminals PCD7.D100 und D110

1.1 Anwendung

Das neue Bedienterminal PCD7.D1.. ist für den rauen Industrie-Einsatz konzipiert, wie er direkt an Produktionsmaschinen auftritt. In Kombination mit der intelligenten Textausgabe der SAIA®PLC (Baureihen PCA und PCD) lässt sich auf eine einfache Art eine Bedienerführung in Menü-Technik aufbauen.

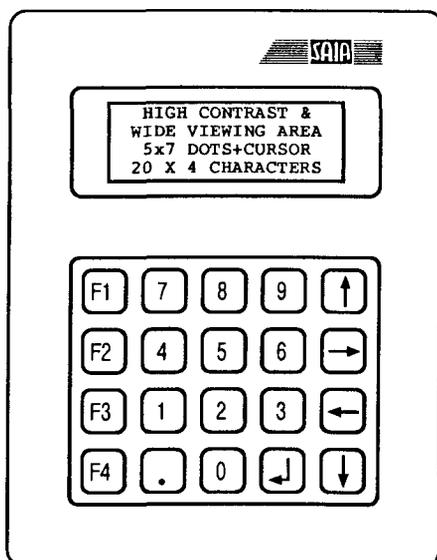
Anzeigen jeder Art über Temperatur, Druck, Stückzahl, Datum, Zeit usw. sowie Meldungen über Betriebszustände oder Alarmer lassen sich mit dem, dank neuester Technologie, extrem klaren und hintergrundbeleuchteten LC-Display ausgeben.

Unter der abriebfesten Polyesterfolie befinden sich 20 Kurzhubtasten, welche es über die seriellen Datenschnittstellen ermöglichen, in der SAIA®PLC Menü-Wahlen vorzunehmen oder beliebige Betriebsdaten einzugeben.

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

Für den universellen Einsatz stehen zwei verschiedene Ausführungen zur Verfügung (siehe auch Kap. 1.3):

- PCD7.D100 mit Schnittstelle RS232 bzw. Stromschleife 20 mA für Punkt-Punkt Kommunikation
- PCD7.D110 mit Schnittstelle RS422/485 für Punkt-Punkt Kommunikation oder für Direktanschluss an ein RS485-Netz mit den Protokollen GMP oder S-Bus.

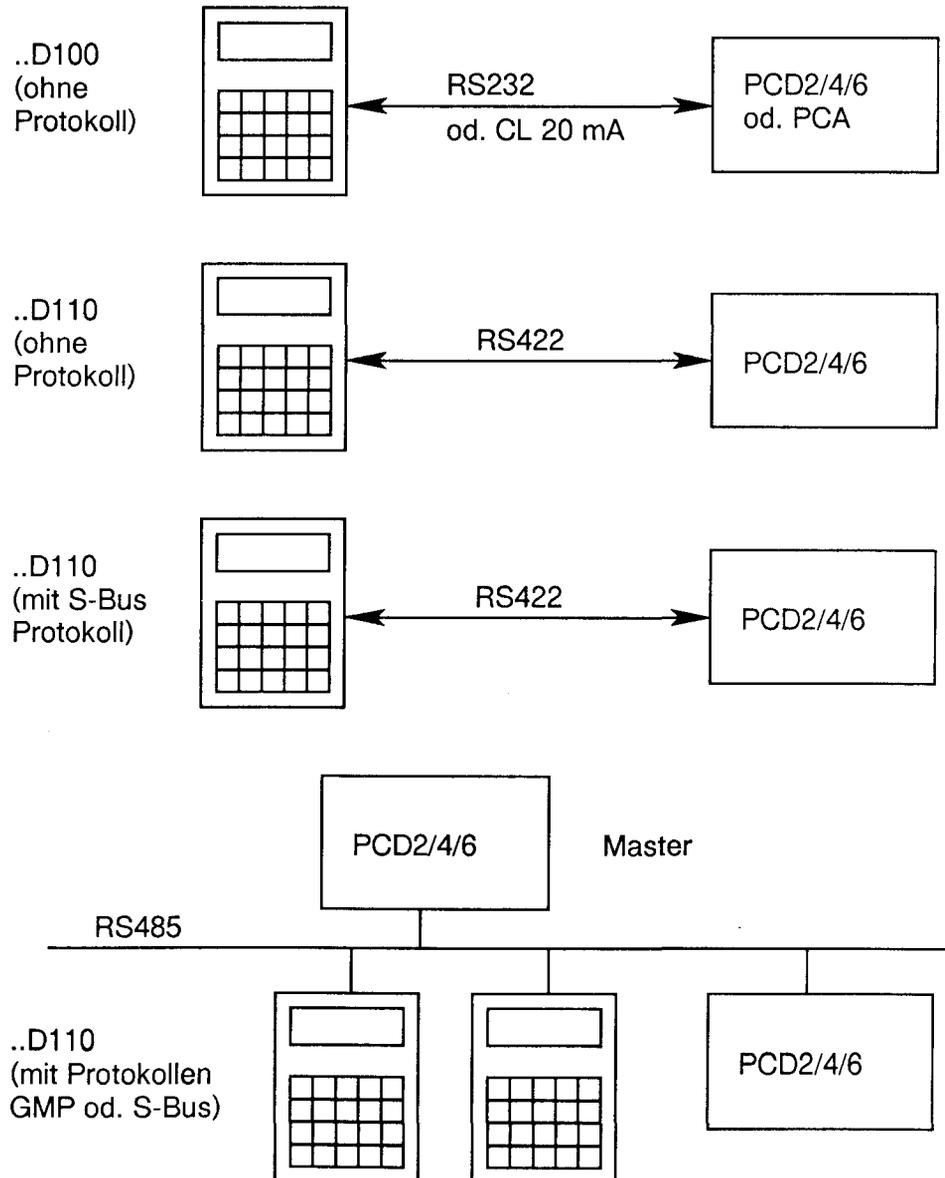


1.2 Technische Daten

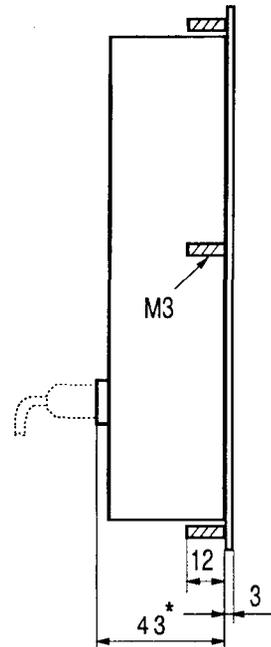
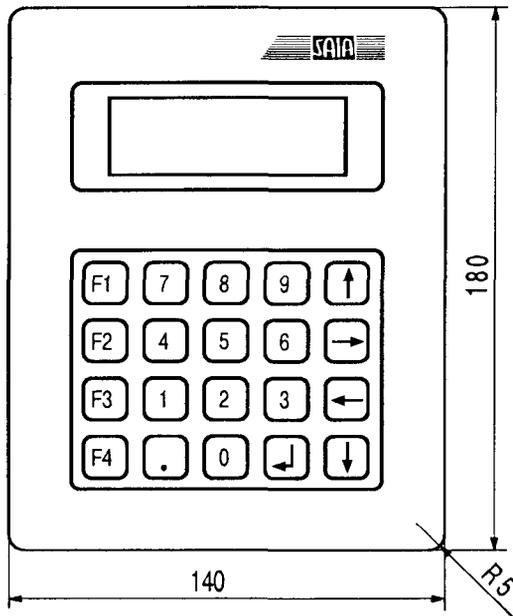
Gehäuse	Metallgehäuse für Fronttafeleinbau
Tasten	20 Kurzhubtasten mit Abdeckung aus Polyesterfolie, Tastenbezeichnungen austauschbar
Schutzklasse	Frontplatte IP65 (Moosgummidichtung)
Anzeige	4x20 Charakter mit Cursor, LC-Display supertwisted mit LED-Hintergrundbeleuchtung
LED-Hintergrundbeleuchtung	Lebensdauer 100'000 h
Frontfarbe	kieselgrau (RAL 7032)
Serielle Schnittstelle	Typ PCD7.D100: RS232 mit RTS/CTS und Stromschleife 20 mA (passiv) Beide Schnittstellen sind gleichzeitig vorhanden. Typ PCD7.D110: RS422 / RS485 Punkt-Punkt und netzwerkfähig (GMP und S-Bus)
Baudrate	150 bis 9600 bps , 8 Bit und Parität
DIP-Schalter	ermöglichen neben der Wahl der Übertragungsparameter auch: Vollduplex, Halbduplex, XON/XOFF-Protokoll, automatisches CR/LF und die Wahl des Page- bzw. Scroll-Modus sowie Stationsnummer bei Typ D110
Speisung	24 V DC $\pm 20\%$ geglättet 18 V DC $\pm 15\%$ aus zweiweggleichgerichteter Wechselspannung
Stromaufnahme	max. 200 mA
Klima	Lagertemperatur -20°C ... +70°C Betriebstemperatur 0°C ... +50°C
Störfestigkeit	Speisung 2 kV nach IEC 801-4 Datenleitungen 1 kV nach IEC 801-4 Gehäuse 4 kV nach IEC 801-2 Solide Erdung des Gehäuses erforderlich.

1.3 Typenübersicht und Anschlussarten

Typ	Beschreibung
PCD7.D100	Bedienterminal mit 1 seriellen Schnittstelle, alternativ betreibbar als RS232 oder als Stromschleife 20 mA
PCD7.D110	Bedienterminal mit Schnittstelle RS422 / RS485 für Punkt-Punkt- oder Netzwerkanschluss
PCD7.D1..Z	Sonderausführungen bezüglich Beschriftung auf Anfrage

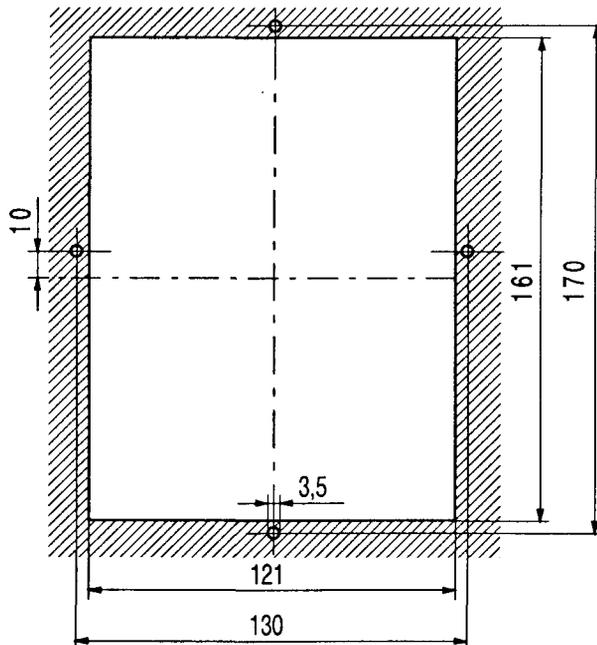


1.4 Massbild



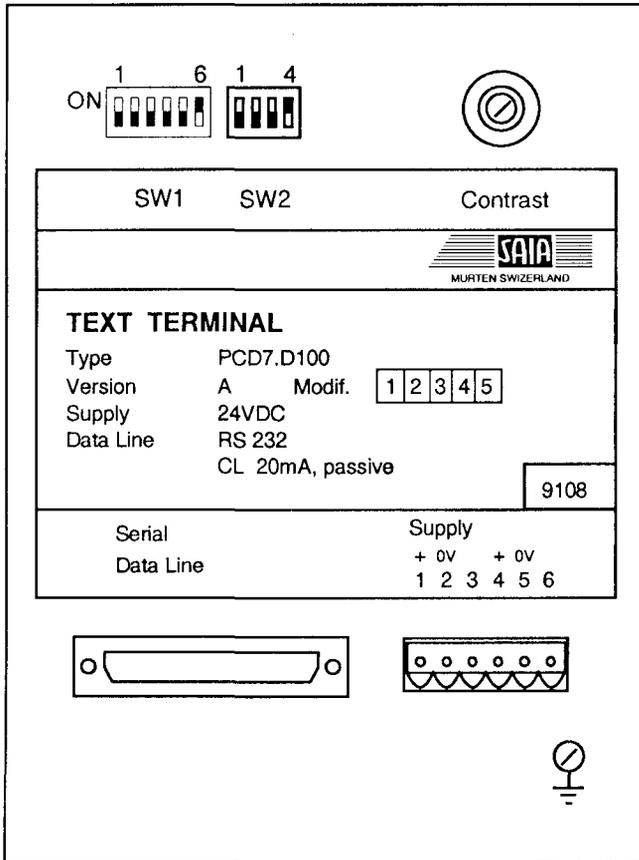
*) ohne Sub-D Gegenstecker

Schalttafelausschnitt



1.5 Hardware

1.5.1 Rückansicht



PCD7.D100

- DIP-Schalter SW1 und SW2

SW1 Vorwahl von:

- Baudrate
- Parität
- Vollduplex / Halbduplex

SW2 Vorwahl von:

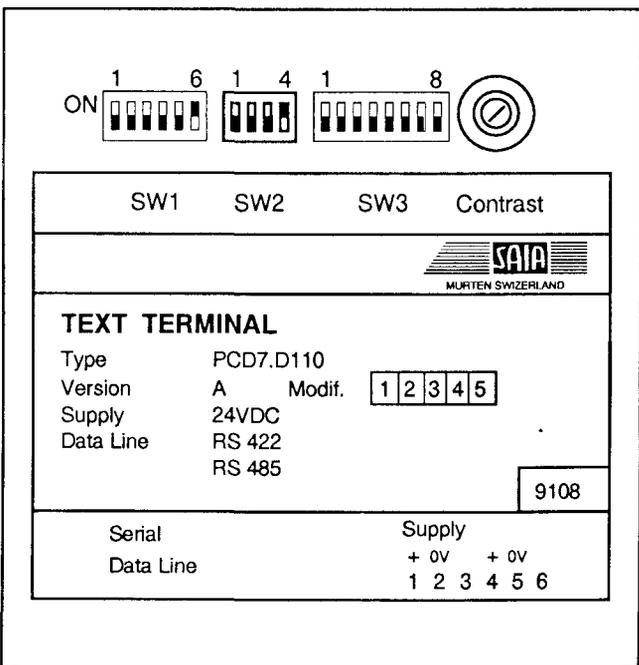
- Page- / Scroll-Modus
- Zeilenvorschub
- Autorepeat

- Potentiometer zur Einstellung des Kontrastes

- Serial Data Line
Kommunikations-Schnittstelle über 25 polige Standard Sub-D-Buchse

- Supply
Speisung über steckbare Schraubklemmen

- Erdschraube M4



PCD7.D110

Zusätzlich bzw. in Abänderung zu Typ ..D100 sind zu erkennen:

- DIP-Schalter SW3 zur Vorwahl von
 - Teilnehmeradresse
 - Schnittstelle RS422 oder RS 485
 - Art des Protokolles
 Details siehe Kap. 1.5.4

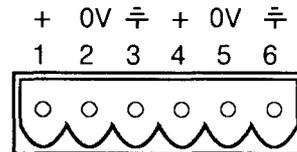
- Serial Data Line
- RS422/485

1.5.2 Stromversorgung

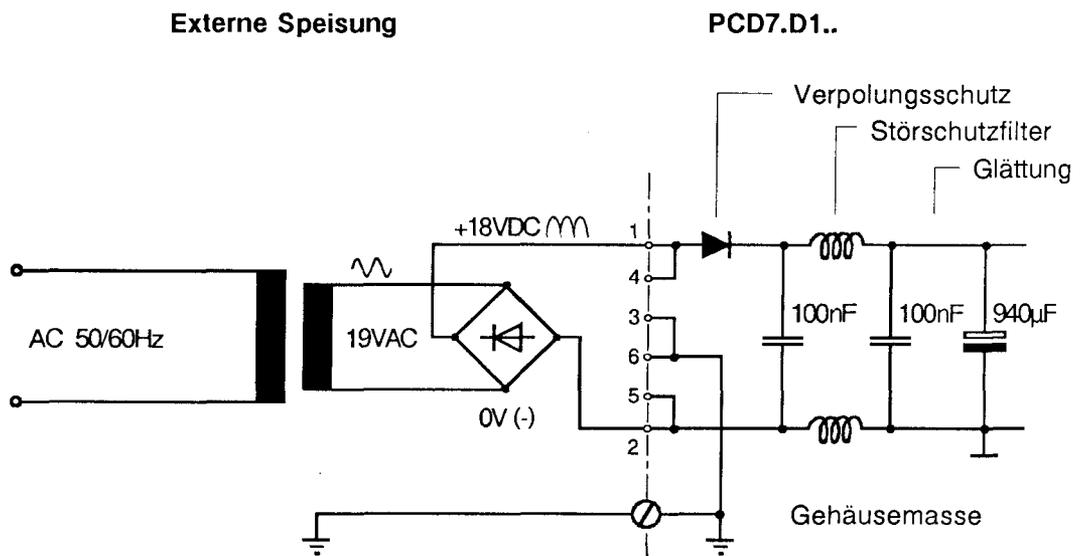
24 VDC \pm 20% geglättet oder
18 VDC \pm 15% aus zweiweggleichgerichteter Wechselspannung
50/60 Hz

Beide Stromversorgungen entsprechen auch den zulässigen Toleranzen für die Baureihen PCD4 und PCD6.

Der **Anschluss-Stecker** ist mit Schraubklemmen für Drähte und Litzen bis 1.5mm^2 ausgelegt.



Stromversorgung mit doppelgleichgerichteter Wechselspannung



1.5.3 Serielle Schnittstellen

PCD7.D100 mit Schnittstellen RS232C und Stromschleife 20 mA

Beide Schnittstellen sind gleichzeitig vorhanden und auf verschiedene Pin der 25-poligen D-Sub-Buchse geführt.

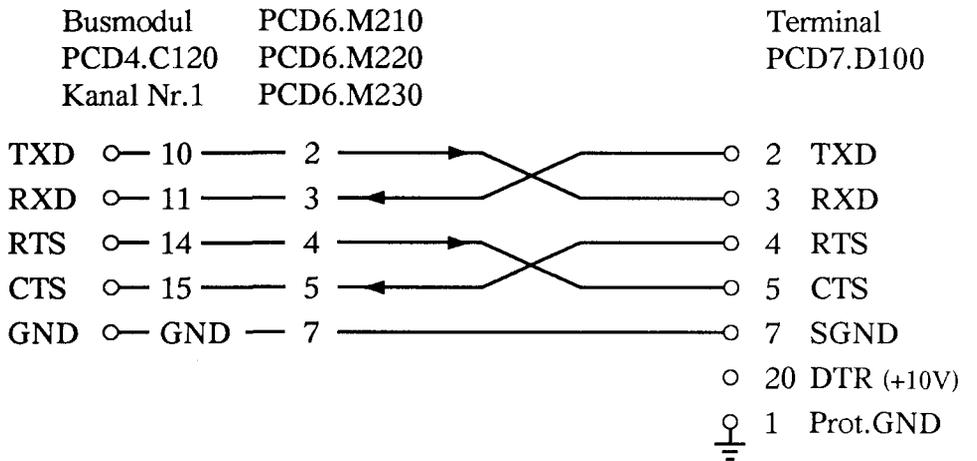
RS232

Terminal
Pin - Nr.

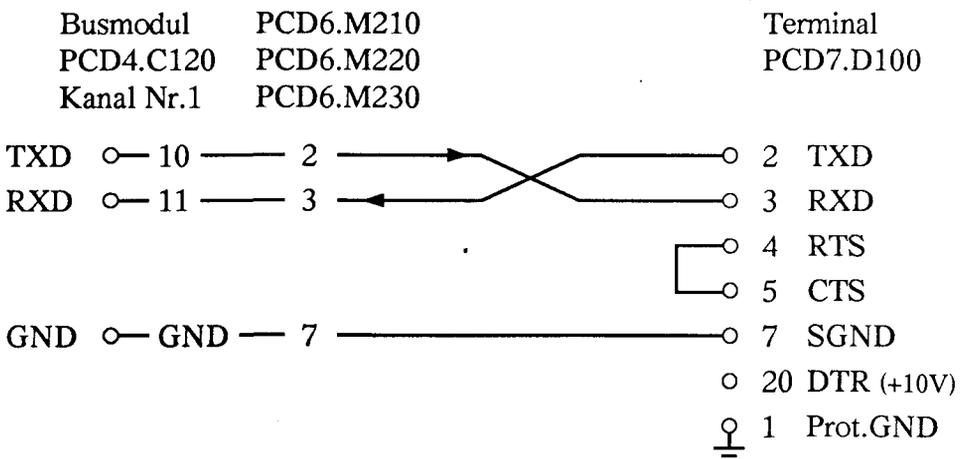
Funktion

2	TXD	Transmit Data	Sendedaten
3	RXD	Receive Data	Empfangsdaten
4	RTS	Request To Send	Sendeaufforderung
5	CTS	Clear To Send	Sendebereitschaft
20	DTR	+10VDC	+10VDC-Ausgang
7	SGND	Signal Ground	Signalerde
1	PGND	Protection Ground	Schutzerde

Verbindungskabel PCD zu PCD7.D100 für Modus MC1:
(mit Steuerung RTS/CTS)



Verbindungskabel PCD zu PCD7.D100 für Modus MC0:
(ohne RTS/CTS-Steuerung)

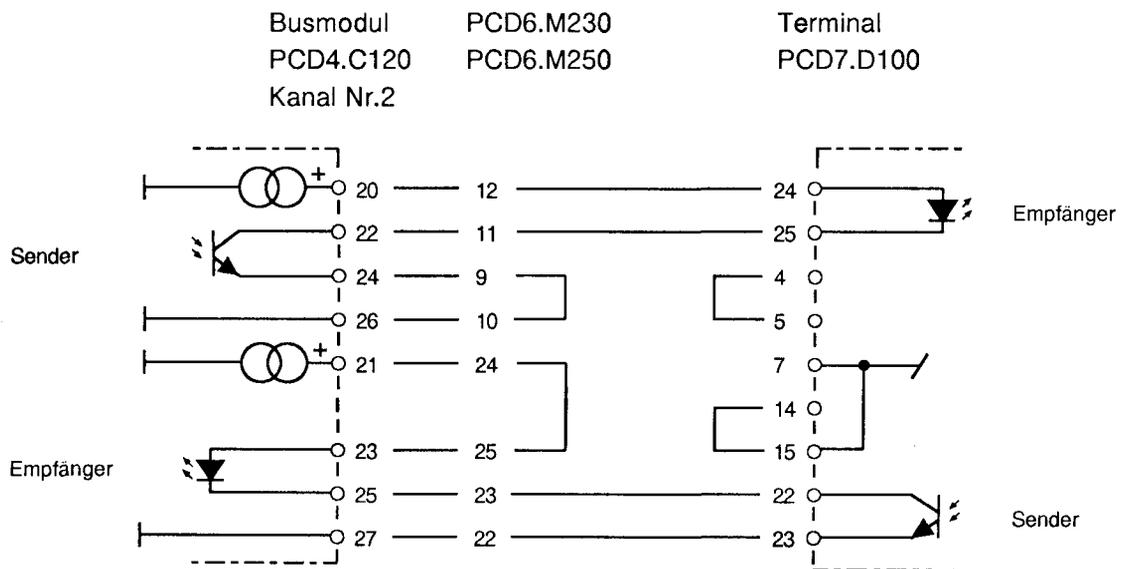


Stromschleife 20 mA

Terminal Pin - Nr.		Funktion	
22	TA	Transmitter Anode	} Sender
23	TC	Transmitter Cathode	
24	RA	Receiver Anode	} Empfänger
25	RC	Receiver Cathode	

Bei Stromschleife 20mA ist die max. Baudrate auf 9600 bps begrenzt.

Verbindungskabel PCD zu PCD7.D100 für Modus MC0 oder MC 2:

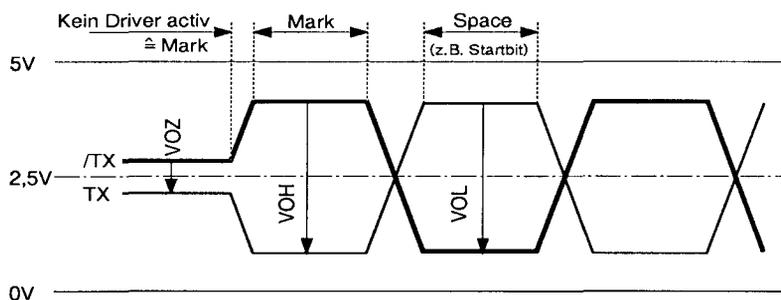


PCD7.D110 mit Schnittstelle RS422

Vom DIP-Schalter SW3 muss sich DIP 6 in der Stellung "off" befinden (siehe Kap. 1.5.4). Die 25-polige D-Sub-Buchse weist dann folgende Belegung auf:

Pin - Nr.		Funktion	
2	TX	Transmit Data	Sendedaten
9	/TX	Transmit Data	Sendedaten
4	RX	Receive Data	Empfangsdaten
11	/RX	Receive Data	Empfangsdaten
3	RTS	Request To Send	Sendeaufforderung
10	/RTS	Request To Send	Sendeaufforderung
5	CTS	Clear To Send	Sendebereitschaft
12	/CTS	Clear To Send	Sendebereitschaft
1	PGND	Protection Ground	Schutzerde

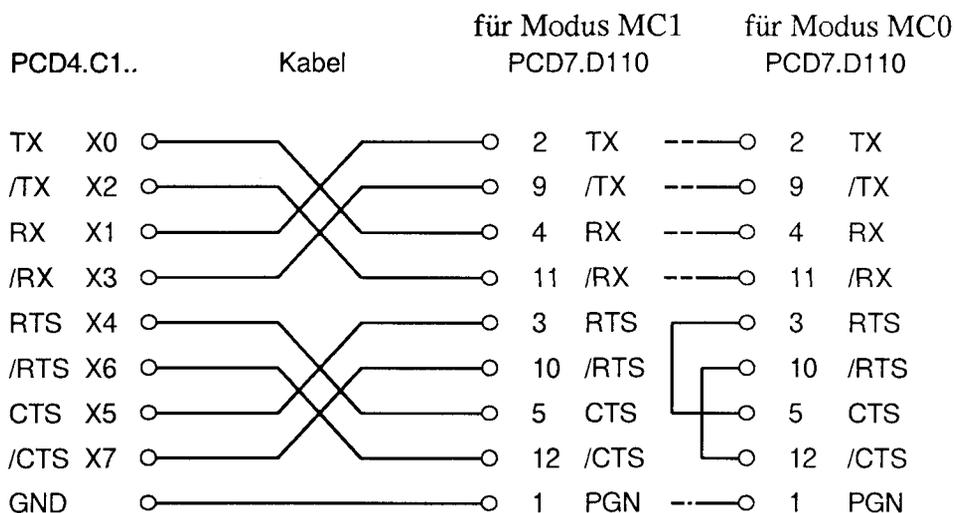
Signalverlauf /TX zu TX:



VOH = 2 V min. (mit Last) ... 5 V max. (ohne Last)

VOL = 0 V min. ... 0,5V max.

Anschlussbeispiel für RS422



PCD7.D110 mit Bus-Schnittstelle RS485

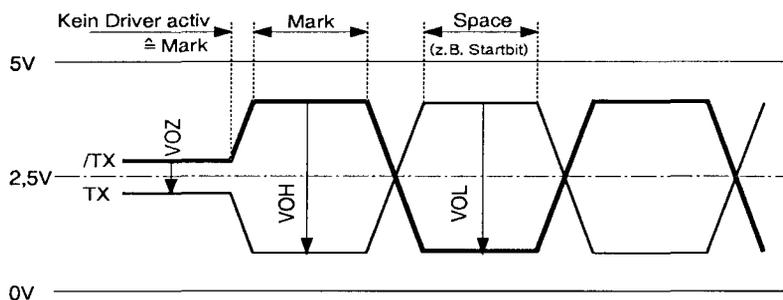
Vom DIP-Schalter SW3 muss sich DIP 6 in der Stellung "on" befinden (siehe Kap. 1.5.4).

Von der 25-poligen D-Sub-Buchse werden lediglich folgende Pin benützt:

Pin - Nr.	Funktion
2	RX-TX
9	/RX-/TX
1	Schutzerde

} Daten

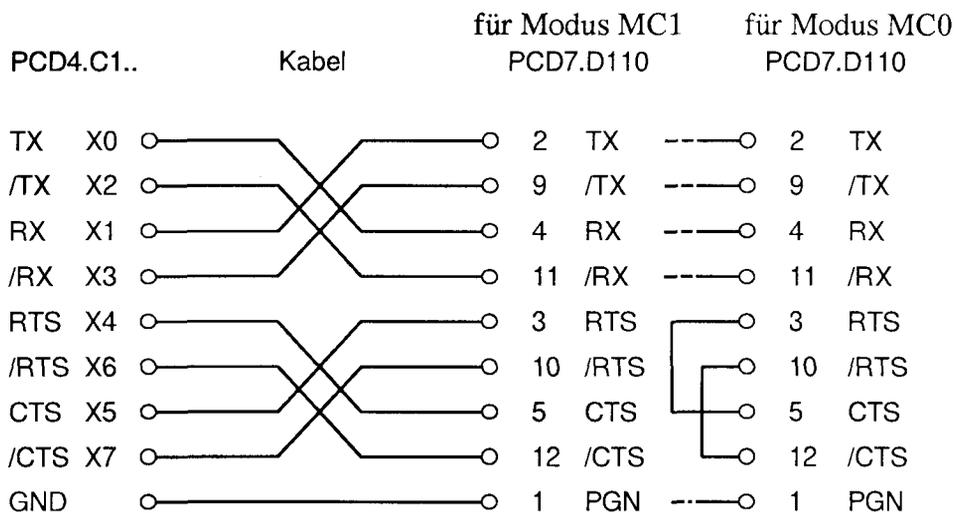
Signalverlauf /TX zu TX:



VOH = 2 V min. (mit Last) ... 5 V max. (ohne Last)

VOL = 0 V min. ... 0,5V max.

Anschlussbeispiel für RS422



PCD7.D110 mit Bus-Schnittstelle RS485

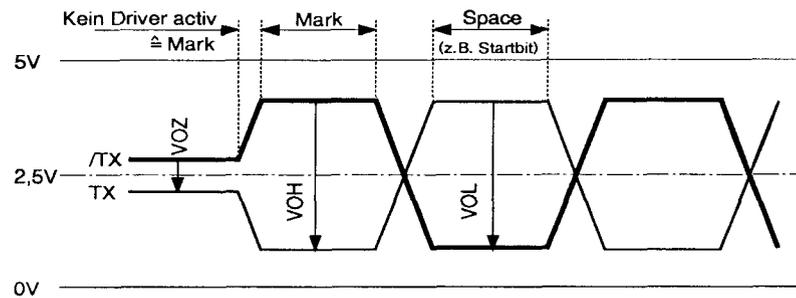
Vom DIP-Schalter SW3 muss sich DIP 6 in der Stellung "on" befinden (siehe Kap. 1.5.4).

Von der 25-poligen D-Sub-Buchse werden lediglich folgende Pin benutzt:

Pin - Nr.	Funktion
2	RX-TX
9	/RX-/TX
1	Schutzerde

} Daten

Signalverlauf /TX zu TX:

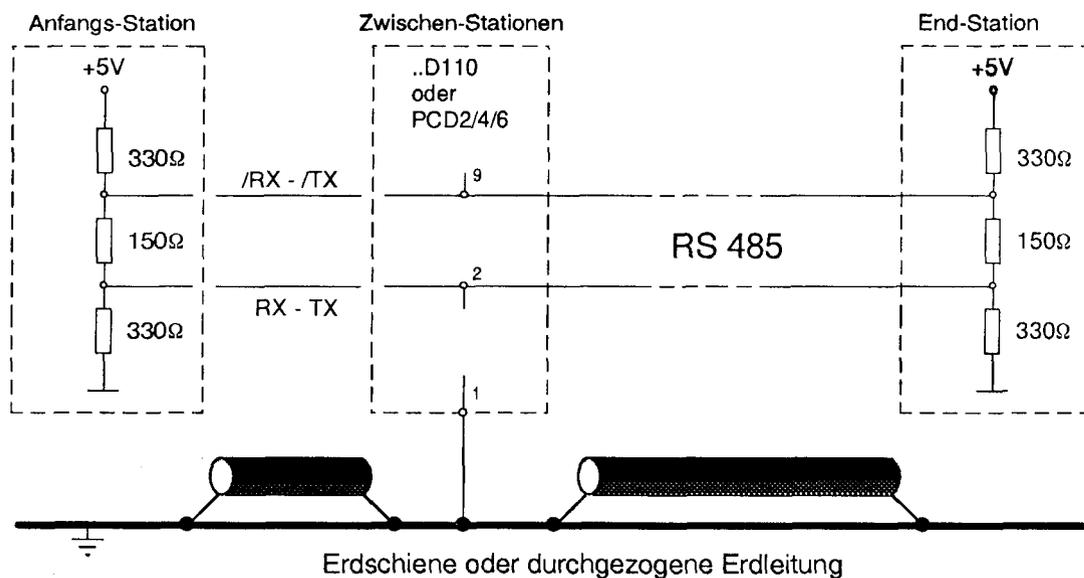


VOH = 2 V min. (mit Last) ... 5 V max. (ohne Last)

VOL = 0 V min. ... 0,5V max.

Anschluss und Verlegung der Bus-Leitungen RS485

Kabel: Litze von mind. 0,5mm², 2 adrig verdreht und abgeschirmt.
Segmentlänge für 32 Stationen max. 1200m.

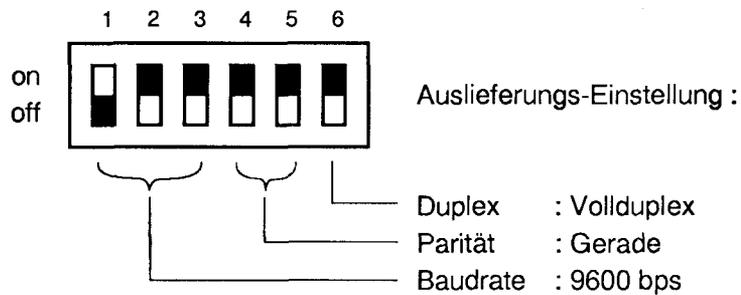


Anfangs- und Endstation muss je eine PCD2/4/6 mit entsprechenden Abschlusswiderständen oder eine Termination-Box Typ PCD7.T160 sein.

Als Zwischenstationen können ..D110 und PCD2/4/6 beliebig gemischt werden (S-Bus),. Bitte Erdungskonzept beachten !

1.5.4 Funktion der DIP - Schalter SW1, SW2 und SW3

SW1



- Baudrate

Baudrate bps	Schalterstellungen von SW1		
	DIP 1	DIP 2	DIP 3
150	off	off	off
300	on	off	off
600	off	on	off
1200	on	on	off
2400	off	off	on
4800	on	off	on
9600	off	on	on

Hinweis: Die DIP-Schalter für die Baudrate werden nur beim Einschalten des Terminals gelesen.

- Charakterformat und Parität

Das Charakterformat ist fest. Es besteht aus 11 Bits, davon 1 Startbit, 8 Datenbits, 1 Paritätsbit und 1 Stopbit.

Start Bit	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7	Bit 8	Parity Bit	Stop Bit
-----------	-------	-------	-------	-------	-------	-------	-------	-------	------------	----------

Die Parität kann mit den Schaltern 4 und 5 wie folgt festgelegt werden:

Sendeparität		Empfangsparität		SW1	
				DIP4	DIP5
tief	(low)	-	-	off	off
hoch	(high)	-	-	off	on
ungerade	(odd)	ungerade	(odd)	on	off
gerade	(even)	gerade	(even)	on	on

Alle verstümmelt empfangenen Zeichen oder Zeichen mit Paritätsfehler werden durch einen schwarzen Block ersetzt.

- Halb- / Vollduplex

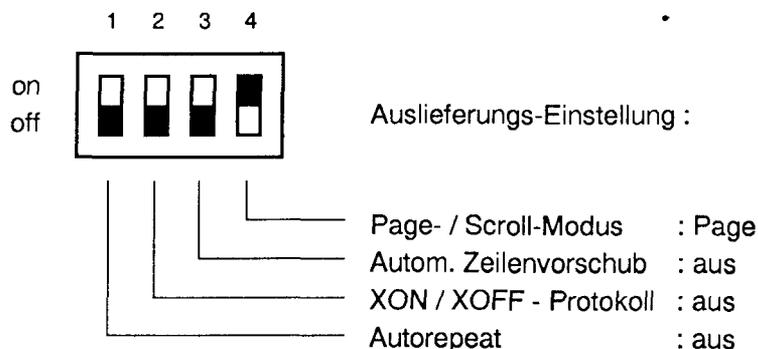
Betriebsart	DIP 6
Halbduplex	off
Vollduplex	on

In der Betriebsart **Halbduplex** werden an der Tastatur eingegebene Zeichen auch auf dem Display angezeigt.

In der Betriebsart **Vollduplex** werden die eingetippten Zeichen nicht automatisch auf das Display gebracht, da gleichzeitig gesendet und empfangen werden kann.

Wird eine Anzeige der eingegebenen Zeichen auch bei Vollduplex gewünscht, so kann dies über die Echofunktion (PCD-Modus MC 3) erfolgen.

SW 2



- Autorepeat

Autorepeat ein :	DIP 1 on
Autorepeat aus :	DIP 1 off

Ist Autorepeat ein, so wird eine länger betätigte Taste automatisch repetiert.

- XON / XOFF - Protokoll

XON / XOFF ein:	DIP 2 on
XON / XOFF aus :	DIP 2 off

Das XON / XOFF - Protokoll steuert den Datenfluss zwischen Terminal und SPS. Sollte nämlich das Terminal die empfangenen Daten nicht mehr schnell genug abarbeiten können, so wird im Eingangspuffer zwischengepuffert. Um ein Überlaufen dieses Puffers zu vermeiden, sendet das Terminal der SPS einen XOFF - Code. Die SPS muss daraufhin das Senden unterbrechen, bis das Terminal die Daten im Eingangspuffer abgearbeitet hat und der SPS einen XON-Code sendet. Nur möglich bei Assignierung der PCD in Modus MC2.

- Automatischer Zeilenvorschub

Automatischer Zeilenvorschub :	DIP 3 on
Kein autom. Zeilenvorschub :	DIP 3 off

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

- Page- / Scroll - Modus

Page - Modus :	DIP 4	on
Scroll - Modus :	DIP 4	off

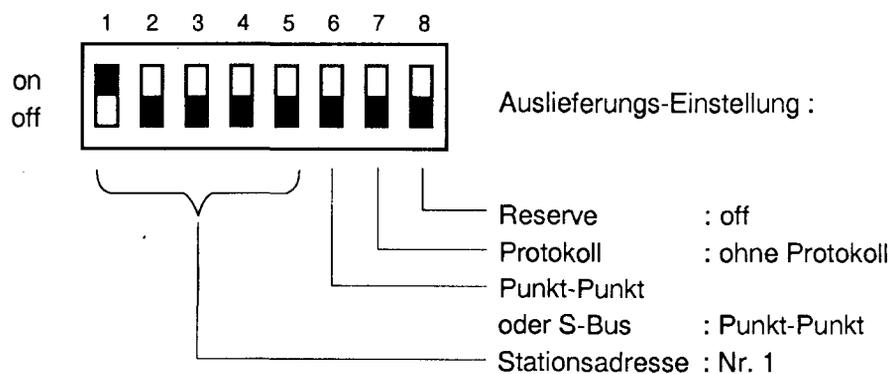
Page-Modus : Befindet sich der Cursor auf der untersten Zeile und das Terminal erhält den Charakter <LF>, so springt er auf die oberste Zeile. Der Inhalt der Anzeige wird dabei nicht verändert.

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.

Wichtig:

Der Scroll-Modus benötigt im ..D100 und ..D110 viel Prozessor-Zeit. Er ist daher nur in Punkt-Punkt-Kommunikation unter Verwendung des Hardware-Handshaking (mit den Steuerleitungen) zulässig. Für alle übrigen Fälle ist der Page-Modus zu verwenden.

SW3



- Stationsadresse

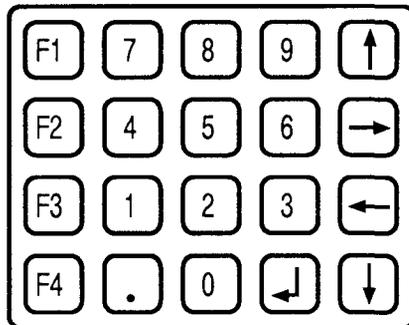
Adresse	DIP 1	2	3	4	5 (MSB)
1	on	off	off	off	off
2	off	on	off	off	off
3	on	on	off	off	off
:					
:					
31	on	on	on	on	on

- Schnittstelle und Protokoll

Funktion	DIP 6	DIP 7
RS422 ohne Protokoll (Pt.-Pt.)	off	off
RS422 mit S-Bus Protokoll (Pt.-Pt.)	off	on
RS485 mit GMP-Protokoll (Bus)	on	off
RS485 mit S-Bus Protokoll (Bus)	on	on

1.6 Die Tastatur

1.6.1 Standard - Tastenbezeichnung



Code - Tabelle (bezüglich Standardtastatur)

Taste	Dez. Code	Hex. Code	Taste	Dez. Code	Hex. Code
0	<48>	30H	F1	<65>	41H
1	<49>	31H	F2	<66>	42H
2	<50>	32H	F3	<67>	43H
3	<51>	33H	F4	<68>	44H
4	<52>	34H	↑	<11>	0BH
5	<53>	35H	→	<6>	06H
6	<54>	36H	←	<8>	08H
7	<55>	37H	↓	<5>	05H
8	<56>	38H	.	<46>	2EH
9	<57>	39H	↙	<13>	0DH

1.6.2 Anwendungsspezifische Tastenbezeichnungen

Die besondere Konstruktion des Terminals PCD7.D1.. erlaubt es, auf einfache Art die Tastenbezeichnung der jeweiligen Anwendung anzupassen.

Nachdem die rückwärtige Abdeckung weggeschraubt wurde, werden die unter die Frontfolie eingeschobenen Bezeichnungstreifen sichtbar. Die Standardstreifen können durch individuell bezeichnete Streifen ersetzt werden. Jeder Verpackung liegen 4 Streifen bei, auf welchen lediglich die Ziffern aufgedruckt sind. Die übrigen 8 Tasten lassen sich anwendungsbezogen beschriften.

Beschriftungsbeispiele:

F1	7	8	9	F5
F2	4	5	6	F6
F3	1	2	3	F7
F4	.	0	←	F8

START	7	8	9	TIMER
STOP	4	5	6	TEMP 1
TEST	1	2	3	TEMP 2
RESET	.	0	←	ERASE

Die Streifen sind der Länge nach zu trennen und mit Vorteil auf der linken Seite leicht die Ecken zu schneiden.

Sollte der Streifen beim Einschieben an den Kurzhubtasten anstehen, so kann dies durch Betätigen der Taste und gleichzeitiges Schieben leicht überwunden werden.

Es ist genügend Zwischenraum für die Streifen vorhanden. Bitte die Stehbolzen der gedruckten Schaltungen nicht lösen, dies erleichtert das Einschieben nicht.

1.7 Die Anzeige

**HIGH CONTRAST &
WIDE VIEWING AREA
5 x 7 DOTS + CURSOR
20 x 4 CHARACTERS**

Die wichtigsten Daten des modernen LC - Displays gehen aus obiger Abbildung hervor. Die Anzeige ist hintergrundbeleuchtet mit einer langlebigen LED - Beleuchtung (100'000 h). Diese Beleuchtung kann über Steuerzeichen ein- und ausgeschaltet werden.

1.7.1 Zeichensatz

Er setzt sich zusammen aus den sichtbaren Zeichen, den Zeichen für die Cursor - Steuerung sowie einigen Steuerzeichen.

HEX	0	1	2	3	4	5	6	7	A	C	D	E	F
0	0 ₀	Cursor pos. 16	SP 32	0 48	@ 64	P 80	` 96	p 112	160	192	208	α 224	240
1			! 33	1 49	A 65	Q 81	a 97	q 113	□ 161			ä 225	
2			" 34	2 50	B 66	R 82	b 98	r 114				β 226	
3			# 35	3 51	C 67	S 83	c 99	s 115					
4			\$ 36	4 52	D 68	T 84	d 100	t 116		Ä 196		ä 228	Ω 244
5	Cursor ↓ 5		% 37	5 53	E 69	U 85	e 101	u 117		Å 197		å 229	
6	Cursor → 6		& 38	6 54	F 70	V 86	f 102	v 118		Æ 198	Ö 214	æ 230	ö 246
7			' 39	7 55	G 71	W 87	g 103	w 119					π 247
8	Cursor ← 8		(40	8 56	H 72	X 88	h 104	x 120			Ø 216		φ 248
9) 41	9 57	I 73	Y 89	i 105	y 121					
A	Line Feed 10	Cursor home 26	* 42	: 58	J 74	Z 90	j 106	z 122					
B	Cursor ↑ 11	ESC 27	+ 43	; 59	K 75	[91	k 107	{ 123			□ 219	x 235	
C	Clear Screen 12		, 44	< 60	L 76	\ 92	l 108	124			Ü 220	Φ 236	ü 252
D	Return 13		- 45	= 61	M 77] 93	m 109	} 125					
E			. 46	> 62	N 78	^ 94	n 110	→ 126	Ε 174				
F			/ 47	? 63	O 79	_ 95	o 111	Rubout Del 127			□ 223	Ö 239	■ 255

Hinweis: In den leeren Feldern A0H bis FFH befinden sich japanische Zeichen.

1.7.2 Einschaltvorgang

Nach Anlegen der Speisespannung führt das PCD7.D1.. einen Selbsttest aus und schreibt folgende Meldung :

SAIA CH-3280 MURTEN
PCD7.D1.. V002

V002 ist die Bezeichnung für die Firmwareversion. Diese wird bei Ergänzungen mutiert.

Diese Anzeige bleibt solange stehen bis diese durch das Kommando "Anzeige löschen" oder durch Überschreiben entfernt wird.

Was tun, wenn das Terminal nicht funktioniert ?

Bitte machen Sie die nachfolgenden einfachen Tests, um einen möglichen Fehler zu lokalisieren und ihn ggf. zu beheben.

a.) Die Anzeige bleibt leer

Mögliche Ursachen :

- Kontrast falsch eingestellt;
- Falsche Stromversorgung oder Stromversorgung an den falschen Pins angeschlossen.

Lösung :

- Kontrast richtig einstellen;
- Stromversorgungsanschluss überprüfen.

b.) Verstümmelte oder falsche Zeichen werden dargestellt

Mögliche Ursache :

- Falsche Baudrate oder Datenformat von der SPS
- Nach Änderung der DIP-Schalter am Terminal wurde kein Einschaltvorgang gemacht

Lösung :

- Baudrate / Datenformat an der SPS und dem Terminal überprüfen.
- Speisung vom Terminal aus- und wieder einschalten

c.) Die Tastatur scheint nicht zu funktionieren

Mögliche Ursache :

- CTS wird nicht bedient (dauernd low)
- Das Terminal befindet sich im Vollduplex - Modus und die SPS sendet keine Echos.
- Ursachen gemäss Punkt b.

Lösung :

- CTS verbinden mit RTS (von PCD oder D1..)
- Echo der Zeichen ermöglichen bei der SPS oder
- Terminal in Halbduplex schalten (SW1 / DIP 6 off)
- Lösungen gemäss Punkt b.

d.) Text wird unvollständig dargestellt

Mögliche Ursache :

- Textbusy (XBSY) zwischen mehreren Telegrammen wurde im PCD-Programm nicht abgewartet.
- Scroll-Modus eingestellt ohne dass Steuerleitungen bedient sind.

Lösung :

- Textbusy (XBSY) auswerten
- Mit SW 2 Page-Modus einstellen oder Steuerleitungen bedienen.

Sollte das Gerät danach immer noch nicht richtig arbeiten, so wenden Sie sich bitte an die Lieferfirma oder die nächstgelegene SAIA-Verkaufsniederlassung.

1.7.3 Steuerfunktionen

In diesem Abschnitt werden die verschiedenen Steuerfunktionen beschrieben, welche über die serielle Schnittstelle empfangen werden können und eine der folgenden Funktionen ausführen.

Cursorsteuerung

	Dez.	Hex.	ASCII
Cursor nach unten	<5>	05H	CTRL/E

Der Cursor bewegt sich um eine Zeile nach unten. Erreicht er die unterste Zeile und folgt ein weiteres 'Cursor-unten' -Kommando, so wird der Cursor in die gleiche Spalte der obersten Zeile positioniert.

Cursor nach rechts	<6>	06H	CTRL/F
---------------------------	-----	-----	--------

Mit diesem Befehl wird der Cursor um eine Spalte nach rechts bewegt. Erreicht er die letzte Spalte und folgt ein weiteres 'Cursor -rechts' Kommando, so wird er auf den Anfang der gleichen Zeile positioniert.

Cursor nach links	<8>	08H	CTRL/H
--------------------------	-----	-----	--------

Mit diesem Befehl wird der Cursor um eine Spalte nach links bewegt. Erreicht er die erste Spalte und folgt ein weiteres 'Cursor-links' -Kommando, so wird er ans Ende der gleichen Zeile positioniert.

Cursor nach oben	<11>	0BH	CTRL/K
-------------------------	------	-----	--------

Mit diesem Befehl wird der Cursor eine Zeile nach oben bewegt. Erreicht er die oberste Zeile und folgt ein weiteres 'Cursor-oben' -Kommando, so wird er in die gleiche Spalte der untersten Zeile positioniert.

Return	<13>	0DH	RETURN
---------------	------	-----	--------

Der Cursor bewegt sich zurück auf die erste Spaltenposition in der momentanen Zeile.

Ist der automatische Zeilenvorschub aktiviert (SW2, DIP3), wird der Cursor auf den Zeilenanfang der nächsten Zeile positioniert, wobei die Anzeige ggf. gerollt bzw. seitenweise weitergeschaltet wird (SW2, DIP4).

Zeilenvorschub	<10>	0AH	CTRL/J
-----------------------	------	-----	--------

Der Cursor bewegt sich eine Zeile nach unten. Befindet sich der Cursor auf der letzten Zeile, wird die Anzeige gerollt bzw. seitenweise weitergeschaltet. (SW2, DIP4).

Cursor-Positionierung **<16>** **10H** **CTRL/P**

Diese Funktion benötigt neben dem Cursor-Adressierungscode (<16> bzw. 10H) die X-Adresse und die Y-Adresse zur Positionierung des Cursors. Beide Adressen müssen mit einem Offset von <32> bzw. 20H versehen werden. Wenn eine der beiden Adressen fehlerhaft ist, wird die Cursor-Position nicht verändert.

Beispiele (Reihenfolge : <16>, Code X, Code Y) :

Cursorposition	Befehlsfolge Dezimal	Befehlsfolge Hex
1. Spalte, 1.Zeile	<16> <32> <32>	10H 20H 20H
4. Spalte, 2.Zeile	<16> <35> <33>	10H 23H 21H
16. Spalte, 4.Zeile	<16> <47> <35>	10H 2FH 23H

Y Code	X	<32>	<33>	<34>	<35>	<36>	<37>	<38>	<39>	<40>	<41>	<42>	<43>	<44>	<45>	<46>	<47>	<48>	<49>	<50>	<51>
		<32>	1	x																	
<33>	2				x																
<34>	3						X														
<35>	4																x				

→ <16><36><36><34>

Cursorpositionierung mit Textausgabe (Befehl STXT) im Modus C:
Gemäss ASCII-Tabelle entspricht die Cursorposition <36> dem "\$"-Zeichen, welches in Sondertexten verwendet wird um die PCD-Daten auszugeben. Soll in einem Text ein "\$" (<36>) ausgegeben werden, so ist zu schreiben "\$\$". Siehe dazu auch Befehlssatz für die PCD-Familie Seite 8.21.

Beispiel: TEXT xxxx "...^{X-Pos.}<16>^{Y-Pos.}<36><36><34>...."

Cursor ausschalten <27> <84> **1BH 54H** **ESC T**

Dieser Befehl schaltet den Cursor grundsätzlich aus.

Cursor einschalten <27> <87> **1BH 57H** **ESC W**

Mit dieser Funktion lässt sich der Cursor wieder einschalten.

Cursor home <26> **1AH** **CTRL/Z**

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

Anzeige

Anzeige löschen	<12>	0CH	CTRL/L
------------------------	-------------------	------------	---------------

Die gesamte Anzeige wird gelöscht. Der Cursor geht in die Home-Position.

Zeichen löschen	<127>	7FH	DEL
------------------------	--------------------	------------	------------

Der Cursor bewegt sich eine Position nach links und entfernt das Zeichen an dieser Stelle. Er stoppt jedoch auf der Home-Position.

Testbild	<27> <73>	1BH 49H	ESC I
-----------------	------------------------------	----------------	--------------

Diese Funktion füllt die Anzeige mit 'E's und dient der Kontrasteinstellung während der Fertigung.

Demonstrationsanzeige	<27> <74>	1BH 4AH	ESC J
------------------------------	------------------------------	----------------	--------------

Diese Funktion zeigt die Leistungsmerkmale des Terminals und dient daher für Demonstrationszwecke.

LCD - Hintergrundbeleuchtung

Beleuchtung einschalten	<27> <76>	1BH 4CH	ESC L
--------------------------------	------------------------------	----------------	--------------

Beleuchtung ausschalten	<27> <79>	1BH 4FH	ESC O
--------------------------------	------------------------------	----------------	--------------

Sonstige Kommandos

Escape	<27>	1BH	ESC
---------------	-------------------	------------	------------

Hiermit wird eine ESCape-Befehlsfolge eingeleitet.

Firmware rücksetzen	<27> <72>	1BH 48H	ESC H
----------------------------	------------------------------	----------------	--------------

Die Firmware des Terminals wird zurückgesetzt und der Hardware-Selbsttest ausgeführt.

Tastatur sperren	<27> <78>	1BH 4EH	ESC N
-------------------------	------------------------------	----------------	--------------

Tastatur freigeben	<27> <81>	1BH 51H	ESC Q
---------------------------	------------------------------	----------------	--------------

Übersicht aller Steuerfunktionen

Steuerfunktion	Dezimal	HEX	ASCII
Anzeige löschen	<12>	0CH	CTRL/L
Cursor-Positionierung	<16>	10H	CTRL/P
Cursor ausschalten	<27> <84>	1BH 54H	ESC T
Cursor einschalten	<27> <87>	1BH 57H	ESC W
Cursor home	<26>	1AH	CTRL/Z
Cursor nach links	<8>	08H	CTRL/H
Cursor nach oben	<11>	0BH	CTRL/K
Cursor nach rechts	<6>	06H	CTRL/F
Cursor nach unten	<5>	05H	CTRL/E
Demo-Anzeige	<27> <74>	1BH 4AH	ESC J
Escape	<27>	1BH	ESC
Return	<13>	0DH	RET
Firmware rücksetzen	<27> <72>	1BH 48H	ESC H
Testbild	<27> <73>	1BH 49H	ESC I
Zeichen löschen	<127>	7FH	DEL
Zeilenvorschub	<10>	0AH	CTRL/J
Beleuchtung ein	<27> <76>	1BH 4CH	ESC L
Beleuchtung aus	<27> <79>	1BH 4FH	ESC O
Tastatur sperren	<27> <78>	1BH 4EH	ESC N
Tastatur freigeben	<27> <81>	1BH 51H	ESC Q

Einzeltastenerkennung mit nachfolgender Aktion

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

Taste F1 : ein einfacher Text wird ausgegeben.

Taste F2 : ein Text mit dem Zustand der Eingänge 0 bis 15 wird ausgegeben.

Taste F3 : ein Text mit dem Wert des BCD-Schalters, angeschlossen an die Eingänge 16..31, 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.

Beispiel 8.4.1: Das Anwenderprogramm enthält Sprünge.

Beispiel 8.4.2: Das Anwenderprogramm ist in BLOCTEC strukturiert.

Beispiel 8.4.3: Das Anwenderprogramm ist in GRAFTEC strukturiert.

Eingabe numerischer Werte

Menuegeführt soll der Inhalt eines Registers und Zählers via das 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.

```

;
;
; User program example 8.2.1 for the industrial terminal PCD7.D1..
; =====
; The program is structured in BLOC TEC
;
; File : DEMO21.SRC
;
; Creation: 03.09.91 U.Jäggi
;
;
;

```

```

TEXT 1 "<12>" ; Clear display
      "<27><84>" ; Cursor off
      "# INDUSTRIAL #"
      "# CONTROL-TERMINAL #"
      "# PCD7.D100 #"
      "#####"

```

```

TEXT 100 "UART:9600,8,E,1;MODE:MC0;DIAG:O32,R100"

```

```

;-----
; Coldstart
;-----
XOB 16
SASI 1 ; Assignment RS232 interface
      100 ; Text 100
EXOB
;-----
; Mainprogram
;-----
COB 0
      0
STH I 0
DYN F 0
ANL O 38 ; Text busy flag
CPB H 0 ; Send text
ECOB

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

```

```

;
;
; User program example 8.2.2 for the industrial terminal PCD7.D1..
; =====
; The program is structured in GRAFTEC
;
; File :      DEMO22.SRC
;
; Creation:   03.09.91      U.Jäggi
;
;

```

```

TEXT    1      "<12>"          ; Clear display
          "<27><84>"          ; Cursor off
          "# INDUSTRIAL #"
          "# CONTROL-TERMINAL #"
          "# PCD7.D100 #"
          "#####"

```

```

TEXT    100    "UART:9600,8,E,1;MODE:MC0;DIAG:O32,R100"

```

```

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

```

```

XOB      16
SASI     1          ; Assignment RS232 interface
          100       ; Text 100

```

```

EXOB

```

```

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

```

```

COB      0
          0
CSB      0
ECOB

```

```

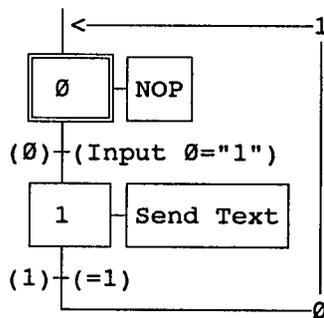
;-----

```

```

SB      0

```



```
SB      0
;-----
IST      0                      ;NOP
EST
;-----
ST       1                      ;Send Text
STXT      1
          1
EST
;-----
TR       0                      ;Input 0="1"
STH      I      0
DYN      F      0
ANL      O      38              ; Text busy
ETR
;-----
TR       1                      ;=1
ETR
;-----
ESB
```

```

;
;
; User program example 8.3.1 for the industrial terminal PCD7.D1..
; -----
; The program is structured in BLOC TEC
;
; File : DEMO31.SRC
;
; Creation: 03.09.91 U.Jäggi
;
;

```

```

TEXT 1      "<12>"                ; Clear display
           "<27><84>"            ; cursor off
           "Main menu [I0]<10><13>"
           "Display status      "
           "Input 8,9 : [I1]  "
           "Input 10,11 : [I2] "

TEXT 2      "<12>"
           "Status <10><13>"
           "Input 8 : $i0008<10><13>"
           "Input 9 : $i0009<10><13>"
           "Main menu [I0]"

TEXT 3      "<12>"
           "Status <10><13>"
           "Input 10 : $i0010<10><13>"
           "Input 11 : $i0011<10><13>"
           "Main menu [I0]"

TEXT 100    "UART:9600,8,E,1;MODE:MC0;DIAG:O32,R100"

```

```

;-----
; Coldstart
;-----
XOB          16
SASI         1          ; Assingation RS232 interface
           100         ; Text 100

EXOB

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

;-----
STH   I     0
DYN   F     0
ANL   O     38      ; Text busy flag
CFB   H     0      ; Send text
           1      ; Text 1

;-----
STH   I     1
DYN   F     1
ANL   O     38      ; Text busy flag
CFB   H     0      ; Send text
           2      ; Text 2

;-----
STH   I     2
DYN   F     2
ANL   O     38      ; Text busy flag
CFB   H     0      ; Send text
           3      ; Text 3

ECOB

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

EFB

```

```

;
;
; User program example 8.3.2 for the industrial terminal PCD7.D1..
; =====
; The program is structured in GRAFTEC
;
; File : DEMO32.SRC
;
; Creation: 03.09.91 U.Jäggi
;
;

```

```

TEXT 1      "<12>"          ; Clear display
            "<27><84>"      ; Cursor off
            "Main menu [I0]<10><13>"
            "Display status  "
            "Input 8,9 : [I1]  "
            "Input 10,11 : [I2]  "

TEXT 2      "<12>"
            "Status <10><13>"
            "Input 8 : $i0008<10><13>"
            "Input 9 : $i0009<10><13>"
            "Main menu [I0]"

TEXT 3      "<12>"
            "Status <10><13>"
            "Input 10 : $i0010<10><13>"
            "Input 11 : $i0011<10><13>"
            "Main menu [I0]"

TEXT 100    "UART:9600,8,E,1;MODE:MC0;DIAG:032,R100"

```

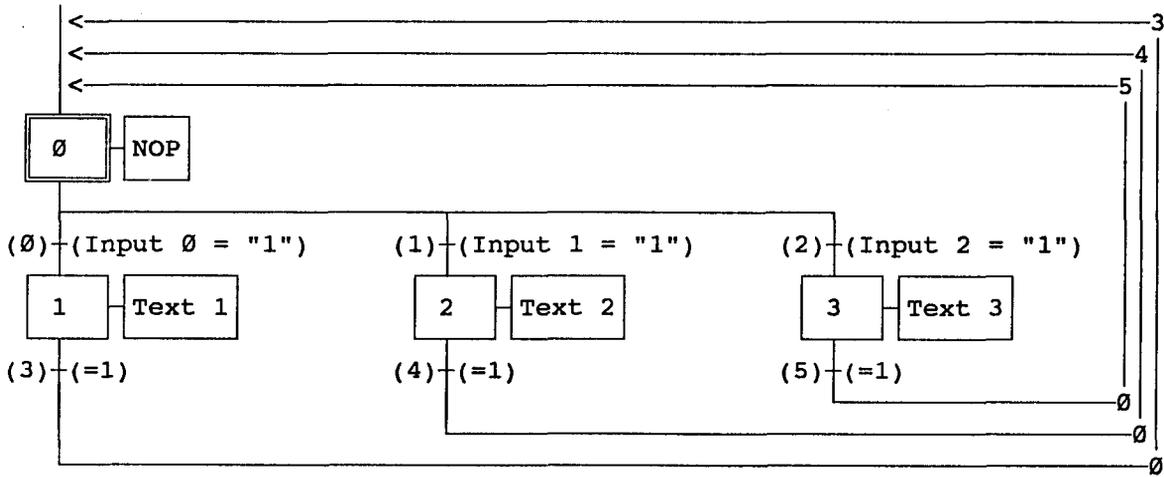
```

;-----
; Coldstart
;-----
XOB          16
SASI         1           ; Assigination RS232 interface
              100        ; Text 100

EXOB
;-----
; Mainprogram
;-----
COB          0
              0
CSB          0
ECOB

```

SB 0



ESB

```

SB      0
;-----
IST      0                      ;NOP
EST
;-----
ST       1                      ;Text 1
STXT      1
          1
EST
;-----
ST       2                      ;Text 2
STXT      1
          2
EST
;-----
ST       3                      ;Text 3
STXT      1
          3
EST
;-----
TR       0                      ;Input 0 = "1"
STH      I      0
DYN      F      0
ANL      O      38              ; Text busy
ETR
;-----
TR       1                      ;Input 1 = "1"
STH      I      1
DYN      F      1
ANL      O      38              ; Text busy
ETR
;-----
TR       2                      ;Input 2 = "1"
STH      I      2
DYN      F      2
ANL      O      38              ; Text busy
ETR
;-----
TR       3                      ;=1
ETR
;-----
TR       4                      ;=1
ETR
;-----
TR       5                      ;=1
ETR
;-----
ESB

```

```

;
;
; User program example 8.4.1 for the industrial terminal PCD7.D1..
; =====
; The program contains jumps
;
; File : DEMO41.SRC
;
; Creation: 03.09.91 U.Jäggi
;

```

```

TEXT 1      "<12>"                ; Clear display
            "<27><84>"            ; Cursor off
            "Main menu [F1]<10><13>"
            "Input 0..15 [F2]<10><13>"
            "BCD-Switch [F3]<10><13>"
            "Date/Time [F4]"

TEXT 2      "<12>"                ; Clear display
            "Input Status      "
            "I 0..7 : $I0000<10><13>"
            "I 8..15 : $I0008<10><13>"
            "Main menu [F1]"

TEXT 3      "<12>"                ; Clear display
            "BCD-Switch (I16..31)"
            "-----"
            "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 100    "UART:9600,8,E,1;MODE:MC0;DIAG:O32,R100"

; Symboldefinitions
; =====
; Diagnostic outputs serial interface
; -----
RBSY EQU O 32 ; Receiver Busy
RFUL EQU O RBSY+1 ; Receive Buffer Full
RDIA EQU O RBSY+2 ; Receiver Diagnostic
TBSY EQU O RBSY+3 ; Transmitter Busy
TFUL EQU O RBSY+4 ; Transmit Buffer Full
TDIA EQU O RBSY+5 ; Transmitter Diagnostic
XBSY EQU O RBSY+6 ; Text Busy
NEXE EQU O 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 RS232 interface
              100       ; Text 100
STXT         1          ; Interface 1
              1          ; Text 1
SOCL         1          ; nötig weil Kabel für MC1-Mode
              0
EXOB
;-----
; Main program
;-----
COB          0
              0
STH          O          RBSY          ; Receiver busy
ANL          O          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
;-----                               ; Read BCD-Switch
END:        DIGI         4
              I          16
              R          10
;-----
ECOB

```

```

;
;
; User program example 8.4.2 for the industrial terminal PCD7.D1..
; =====
; The program is structured in BLOC TEC
;
; File : DEMO42.SRC
;
; Creation: 03.09.91 U.Jäggi
;
;

```

```

TEXT 1 "<12>" ; Clear display
      "<27><84>" ; Cursor off
      "Main menu [F1]<10><13>"
      "Input 0..15 [F2]<10><13>"
      "BCD-Switch [F3]<10><13>"
      "Date/Time [F4]"

```

```

TEXT 2 "<12>" ; Clear display
      "Input Status "
      "I 0..7 : $I0000<10><13>"
      "I 8..15 : $I0008<10><13>"
      "Main menu [F1]"

```

```

TEXT 3 "<12>" ; Clear display
      "BCD-Switch (I16..31)"
      "-----"
      "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 100 "UART:9600,8,E,1;MODE:MC0;DIAG:O32,R100"

```

; Symboldefinitions

; =====

; Diagnostic outputs serial interface

; -----

```

RBSY EQU O 32 ; Receiver Busy
RFUL EQU O RBSY+1 ; Receive Buffer Full
RDIA EQU O RBSY+2 ; Receiver Diagnostic
TBSY EQU O RBSY+3 ; Transmitter Busy
TFUL EQU O RBSY+4 ; Transmit Buffer Full
TDIA EQU O RBSY+5 ; Transmitter Diagnostic
XBSY EQU O RBSY+6 ; Text Busy
NEXE EQU O 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           ; Assingation RS232 interface
              100        ; Text 100
CFB          SEND
              1
SOCL         1           ; nötig weil Kabel für MC1-Mode
              0
EXOB
;-----
; Main program
;-----
COB          0
              0
STH          O          RBSY           ; Receiver busy
ANL          O          XBSY           ; Text busy
CFB          H          READ           ; Read character
              R          RBUF_R        ; Receive buffer register
CPB          H          COMPARE        ; Compare received character
;-----; Read BCD-Switch
DIGI         4
              I          16
              R          10
;-----
ECOB

```

```

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          ; Interface 1
           =          1
EFP

;-----
FB          SEND          ; Send text
STXT       1          ; Interface 1
           =          1          ; Textnumber
EFP

```

```

;
;
; User program example 8.4.3 for the industrial terminal PCD7.D1..
; =====
; The program is structured in GRAFTEC.
;
; File : DEMO43.SRC
;
; Creation: 28.10.91 U.Jäggi
;
;

```

```

TEXT 1      "<12>"                ; Clear display
           "<27><84>"            ; Cursor off
           "Main menu [F1]<10><13>"
           "Input 0..15 [F2]<10><13>"
           "BCD-Switch [F3]<10><13>"
           "Date/Time [F4]"

TEXT 2      "<12>"                ; Clear display
           "Input Status      "
           "I 0..7 : $I0000<10><13>"
           "I 8..15 : $I0008<10><13>"
           "Main menu [F1]"

TEXT 3      "<12>"                ; Clear display
           "BCD-Switch (I16..31)"
           "-----"
           "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 100    "UART:9600,8,E,1;MODE:MC0;DIAG:O32,R100"

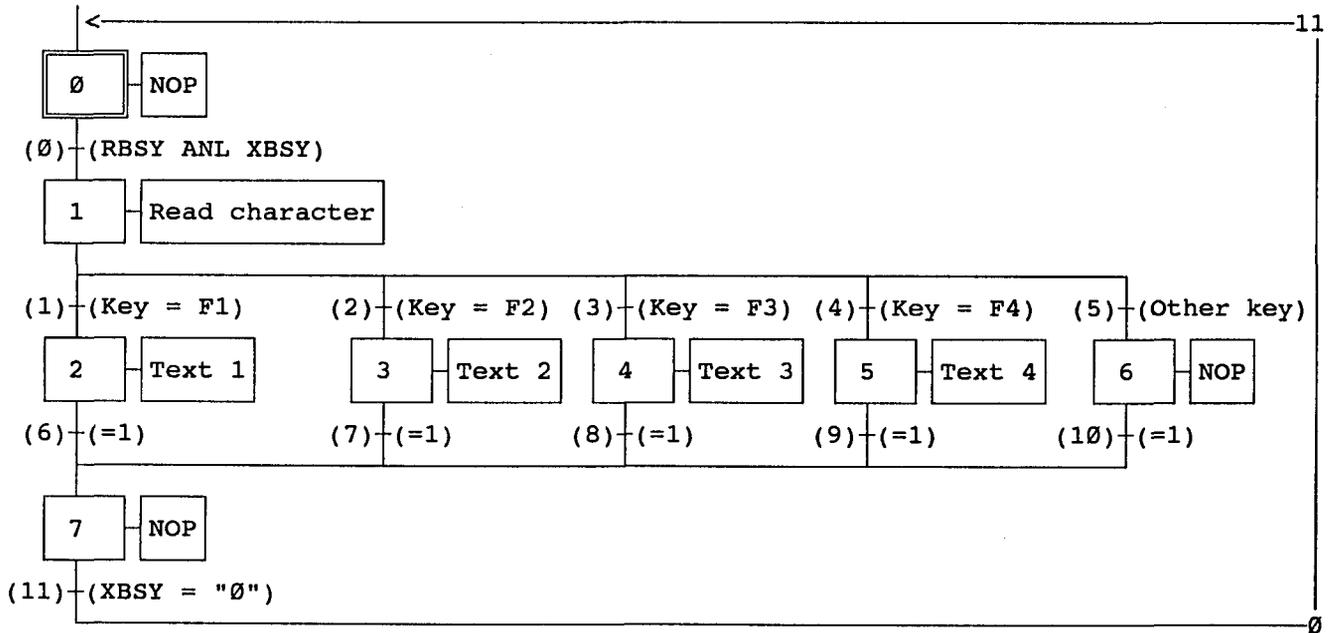
; Symboldefinitions
; =====
; Diagnostic outputs serial interface
; -----
RBSY EQU O 32 ; Receiver Busy
RFUL EQU O RBSY+1 ; Receive Buffer Full
RDIA EQU O RBSY+2 ; Receiver Diagnostic
TBSY EQU O RBSY+3 ; Transmitter Busy
TFUL EQU O RBSY+4 ; Transmit Buffer Full
TDIA EQU O RBSY+5 ; Transmitter Diagnostic
XBSY EQU O RBSY+6 ; Text Busy
NEXE EQU O RBSY+7 ; Not Executed
; -----
; Register
; -----
RBUF_R EQU R 1000

```

```

;-----
; Coldstart
;-----
XOB          16
SASI         1           ; Assigination RS232 interface
              100        ; Text 100
STXT         1
              1
EXOB
;-----
; Main program
;-----
COB          0
              0
CSB          0
;-----; Read BCD-Switch
DIGI         4
              I          16
              R          10
;-----
ECOB
    
```

SB 0



ESB

```

SB      0
;-----
IST      0                      ;NOP
EST
;-----
ST       1                      ;Read character
SRXD      R      1      RBUF_R
EST
;-----
ST       2                      ;Text 1
STXT      1                      ; send
          1                      ; text 1
EST
;-----
ST       3                      ;Text 2
STXT      1                      ; send
          2                      ; text 2
EST
;-----
ST       4                      ;Text 3
STXT      1                      ; send
          3                      ; text 3
EST
;-----
ST       5                      ;Text 4
STXT      1                      ; send
          4                      ; text 4
EST
;-----
ST       6                      ;NOP
EST
;-----
ST       7                      ;NOP
EST
;-----

```

```

TR      0                ;RBSY ANL XBSY
STH     0      32        ; Receiver busy
ANL     0      38        ; Text busy
ETR
;-----
TR      1                ;Key = F1
CMP     R      RBUF_R
        K      65        ; F1
ACC     Z
ETR
;-----
TR      2                ;Key = F2
CMP     R      RBUF_R
        K      66        ; F2
ACC     Z
ETR
;-----
TR      3                ;Key = F3
CMP     R      RBUF_R
        K      67        ; F3
ACC     Z
ETR
;-----
TR      4                ;Key = F4
CMP     R      RBUF_R
        K      68        ; F4
ACC     Z
ETR
;-----
TR      5                ;Other key
ETR
;-----
TR      6                ;=1
ETR
;-----
TR      7                ;=1
ETR
;-----
TR      8                ;=1
ETR
;-----
TR      9                ;=1
ETR
;-----
TR      10               ;=1
ETR
;-----
TR      11               ;XBSY = "0"
STL     0      38        ; Text busy
ETR
;-----
ESB

```

```

;
;
; User program example 8.5 for the industrial terminal PCD7.D1..
; =====
;
; Input of numerical parameters
;
; File : DEMO.SRC
;
; Creation: 03.09.91 U.Jäggi
;
;
;
RBSY_F EQU O 32 ; Receiver Busy
XBSY_F EQU O 38 ; Text Busy
SIGN EQU O 46 ; Sign input
IN_BUSY EQU O 47 ; Input 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
DIAG_R EQU R 999 ; Diagnostic register
MAIN EQU TEXT 0 ; Main menu
IN_TXT_R EQU TEXT 1 ; Input text register
IN_TXT_C EQU TEXT 2 ; Input text counter
ASSIGN EQU TEXT 999 ; Assign. of the serial interf.
INPUT EQU FB 0 ; Functionblock input
CHAN_N EQU 1 ; Number of serial channel

PUBL CHAN_N ; Number of serial channel
PUBL RBSY_F ; Receiver Busy
PUBL XBSY_F ; Text Busy
PUBL IN_BUSY ; Input Busy
PUBL INPUT ; D100 input

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

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          ; Cold start

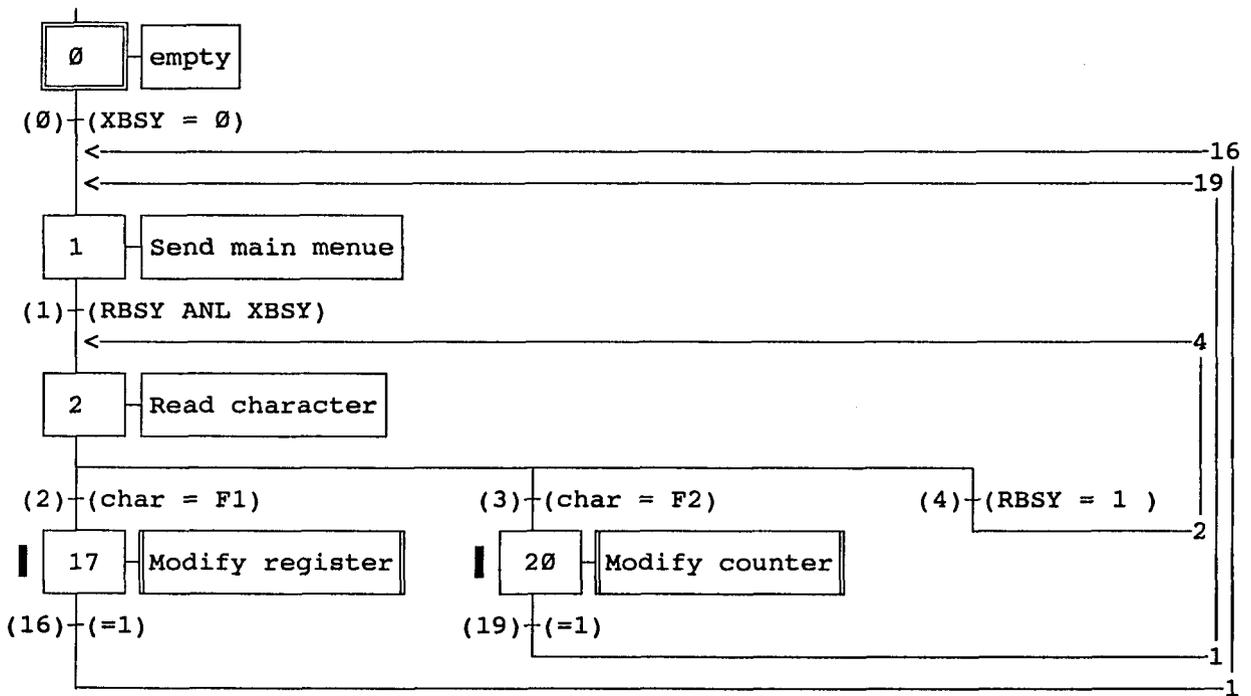
SASI     CHAN_N      ; Assingation RS232 interface
         ASSIGN      ; Text 100
ACC      H
RES      IN_BUSY    ; Reset input busy flag
    
```

```

EXOB
;-----
; Mainprogram
;-----
COB      0          ; Main program
         0
CSB      0          ; Call communication SB
    
```

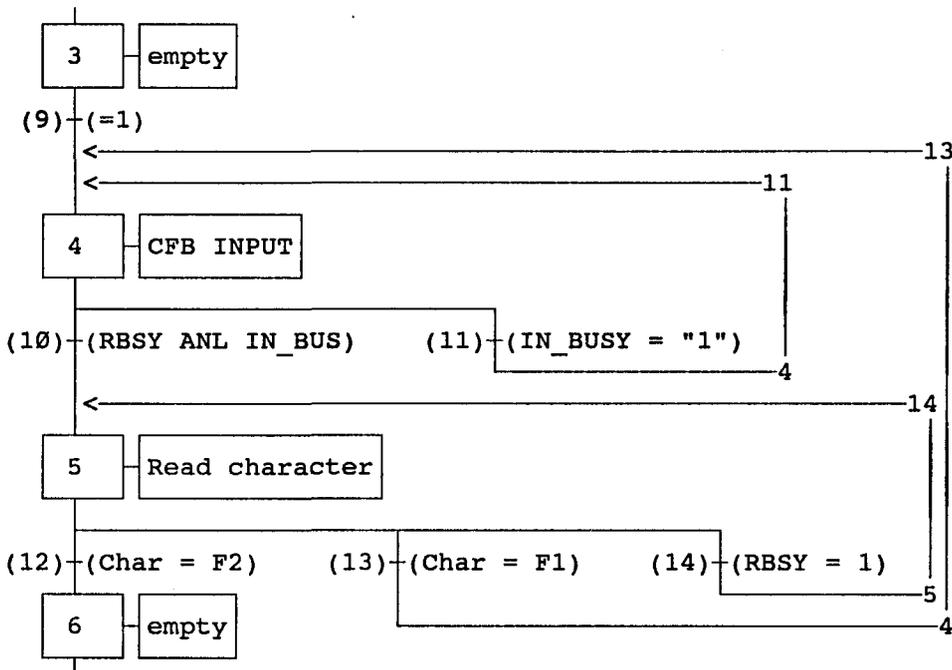
ECOB

SB 0

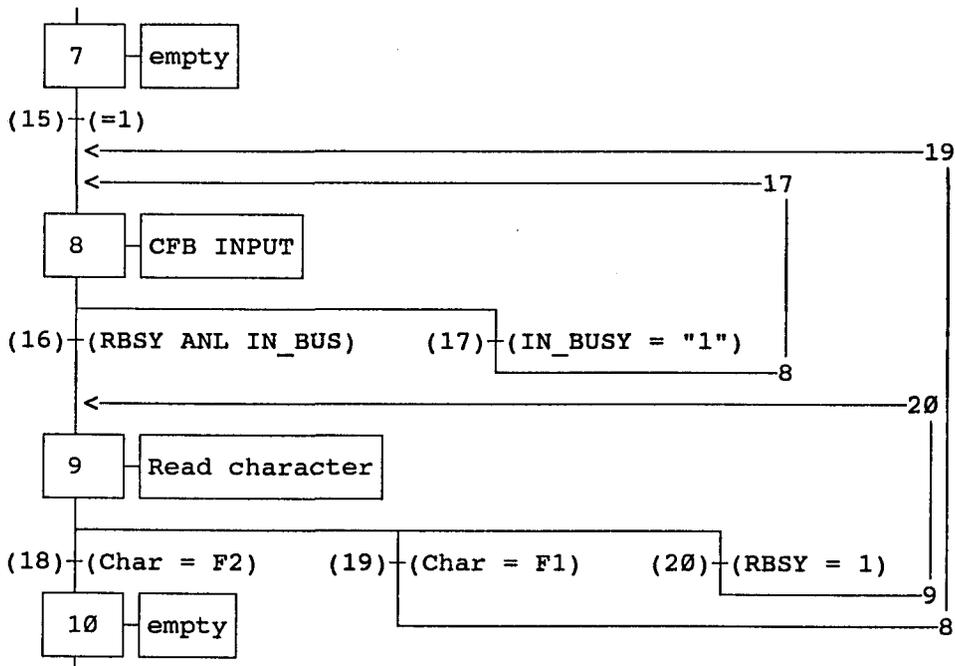


ESB

PAGE-NB: 5 Modify counter



PAGE-NB: 7 Modify register



```

;-----
; Coldstart
;-----
XOB      16          ; Cold start

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

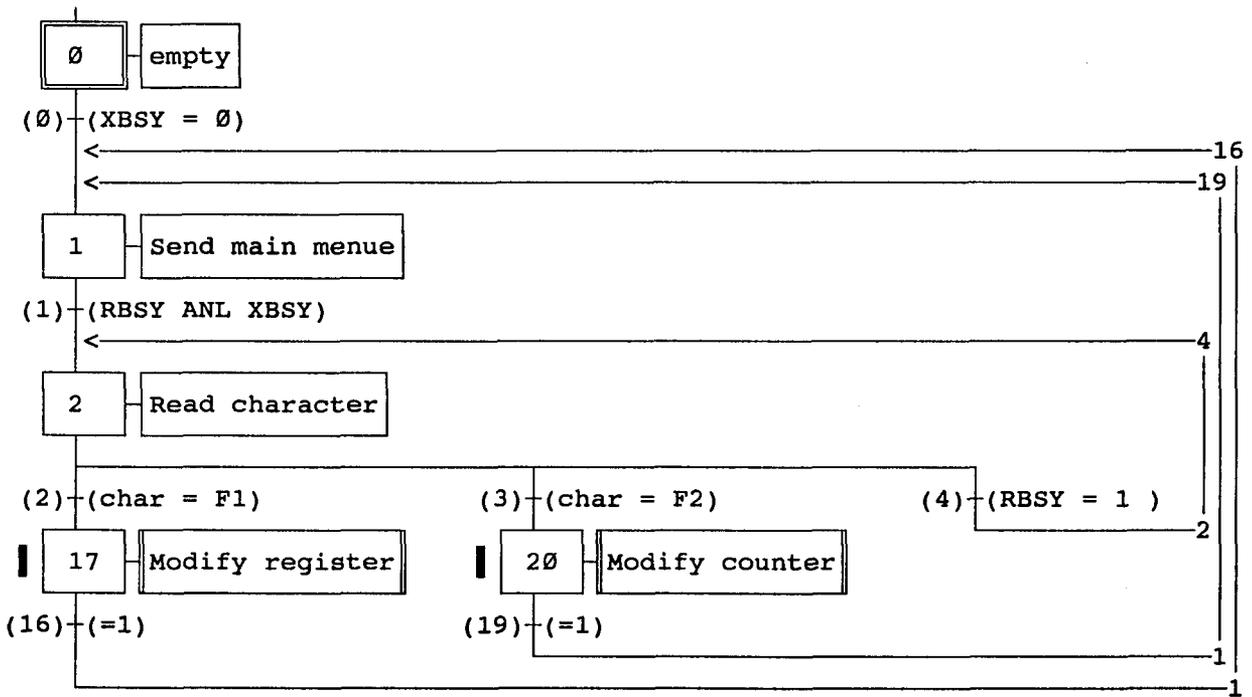
EXOB

;-----
; Mainprogram
;-----
COB      0          ; Main program
        0

CSB      0          ; Call communication SB

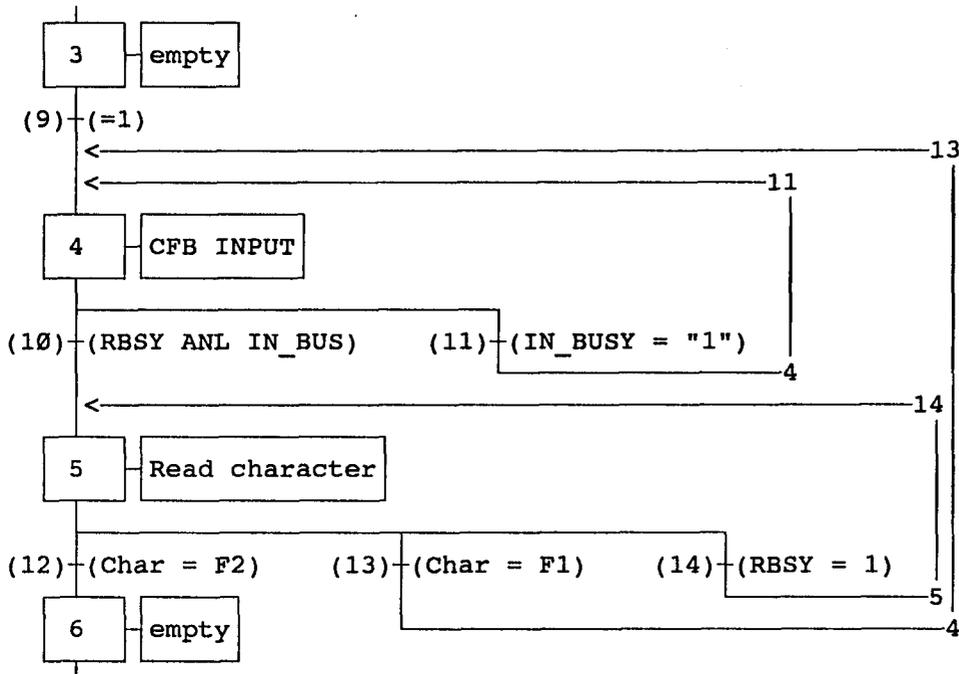
ECOB

SB       0
    
```

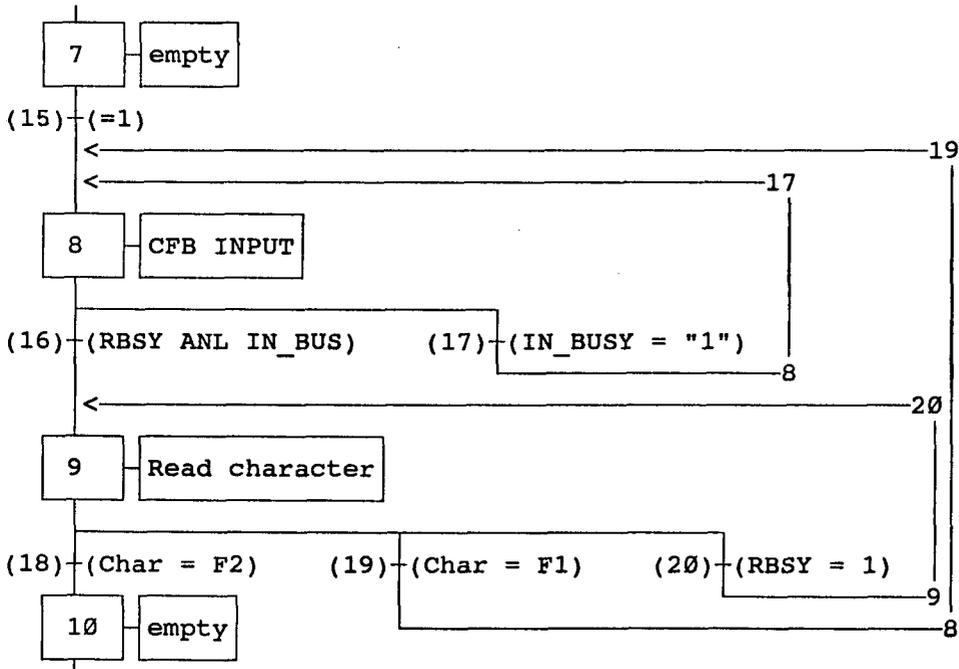


ESB

PAGE-NB: 5 Modify counter



PAGE-NB: 7 Modify register



```

SB      0
;-----
IST     0                                ;empty
EST
;-----
ST      1                                ;Send main menu
STXT          CHAN_N                    ; Send
                MAIN                      ; the main menu
EST
;-----
ST      2                                ;Read character
SRXD     CHAN_N                          ; Read character
                R 1000                     ; from the receive buffer
EST
;-----
ST      3                                ;empty
EST
;-----
ST      4                                ;CFB INPUT
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
SRXD     CHAN_N                          ; Read character
                R 1000                     ; from the receive buffer
EST
;-----
ST      6                                ;empty
EST
;-----

```

```

ST      7                                ;empty
EST
;-----
ST      8                                ;CFB INPUT
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
        4
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
SRXD    CHAN_N                             ; Read character
        R 1000                             ; from the receive buffer
EST
;-----
ST      10                               ;empty
EST
;-----

```

```

TR      0                ;XBSY = 0
STL                    XBSY_F
ETR
;-----
TR      1                ;RBSY ANL XBSY
STH                    RBSY_F
ANL                    XBSY_F
ETR
;-----
TR      2                ;char = F1
CMP                    R 1000
                    K 65                ; F1
ACC                    Z
ETR
;-----
TR      3                ;char = F2
CMP                    R    1000
                    K    66                ; F2
ACC                    Z
ETR
;-----
TR      4                ;RBSY = 1
STH                    RBSY_F
ETR
;-----
TR      5                ;Modify counter
ETR
;-----
TR      6                ;=1
ETR
;-----
TR      7                ;Modify register
ETR
;-----
TR      8                ;=1
ETR
;-----
TR      9                ;=1
ETR
;-----
TR      10               ;RBSY ANL IN_BUSY
STH                    RBSY_F
ANL                    IN_BUSY
ETR
;-----
TR      11               ;IN_BUSY = "1"
STH                    IN_BUSY
ETR
;-----

```

```

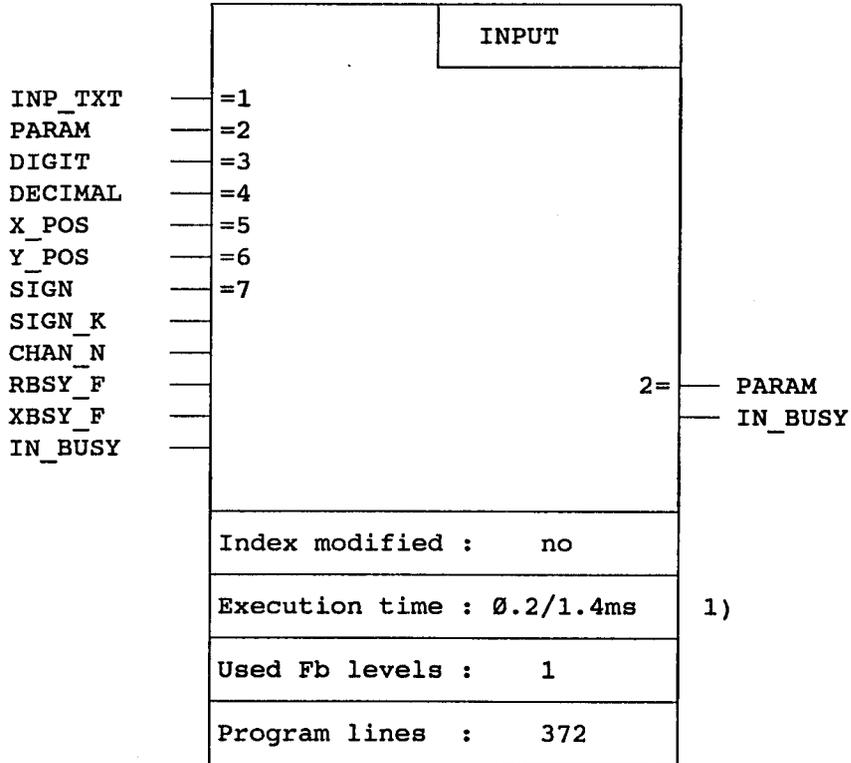
TR      12                                ;Char = F2
CMP     R      1000
        K      66                        ; F2
ACC     Z
ANL     XBSY_F
ETR
;-----
TR      13                                ;Char = F1
CMP     R 1000
        K 65                            ; F1
ACC     Z
ANL     XBSY_F
ETR
;-----
TR      14                                ;RBSY = 1
STH     RBSY_F
ETR
;-----
TR      15                                ;=1
ETR
;-----
TR      16                                ;RBSY ANL IN_BUSY
STH     RBSY_F
ANL     IN_BUSY
ETR
;-----
TR      17                                ;IN_BUSY = "1"
STH     IN_BUSY
ETR
;-----
TR      18                                ;Char = F2
CMP     R      1000
        K      66                        ; F2
ACC     Z
ANL     XBSY_F
ETR
;-----
TR      19                                ;Char = F1
CMP     R 1000
        K 65                            ; F1
ACC     Z
ANL     XBSY_F
ETR
;-----
TR      20                                ;RBSY = 1
STH     RBSY_F
ETR
;-----
ESB

```

Funktionsblock:

INPUT

Parametereingabe mit dem Industrieterminal PCD7.D1



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

Funktionsbeschreibung

Mit diesem Funktionsblock kann der Inhalt eines Registers oder Zählers via das Industrieterminal PCD7.D100 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	-2147483648 ...+2147483647	0..4095
DIGIT	Number of digits Anzahl Eingabestellen	yes	R	Integer	1..11	0..4095
DECIMAL	Number of decimal places Anzahl Nachkomma- stellen	yes	R	Integer	0,1..10	
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
SIGN_K	Sign key ASCII-Code der Vorzeichentaste	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_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)
    
```

Detallierte 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.

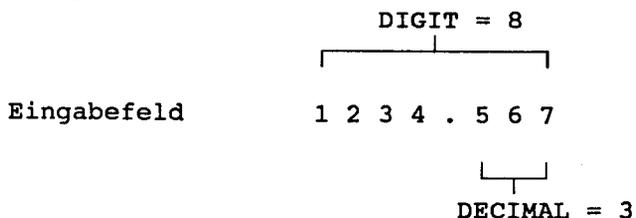
Bemerkung: das letzte Digit (Cursor X-Position 51) der Anzeigzeile darf nicht verwendet für das Eingabefeld.

■ 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 Taste "SIGN_K":

Definiert den ASCII-Code für die Minuszechentaste.
Eingabe des positiven Vorzeichens ist nicht zugelassen.

■ 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.

■ Schnittstellenummer "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.D100 : MC0, MC1 und MC2)

■ Diagnoseflag "RBSY_F"/"XBSY_F" der seriellen Schnittstelle:

Die Adressen für das RBSY_F und XBSY_F müssen übereinstimmen mit der Adresse der Diagnoseflag, 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.

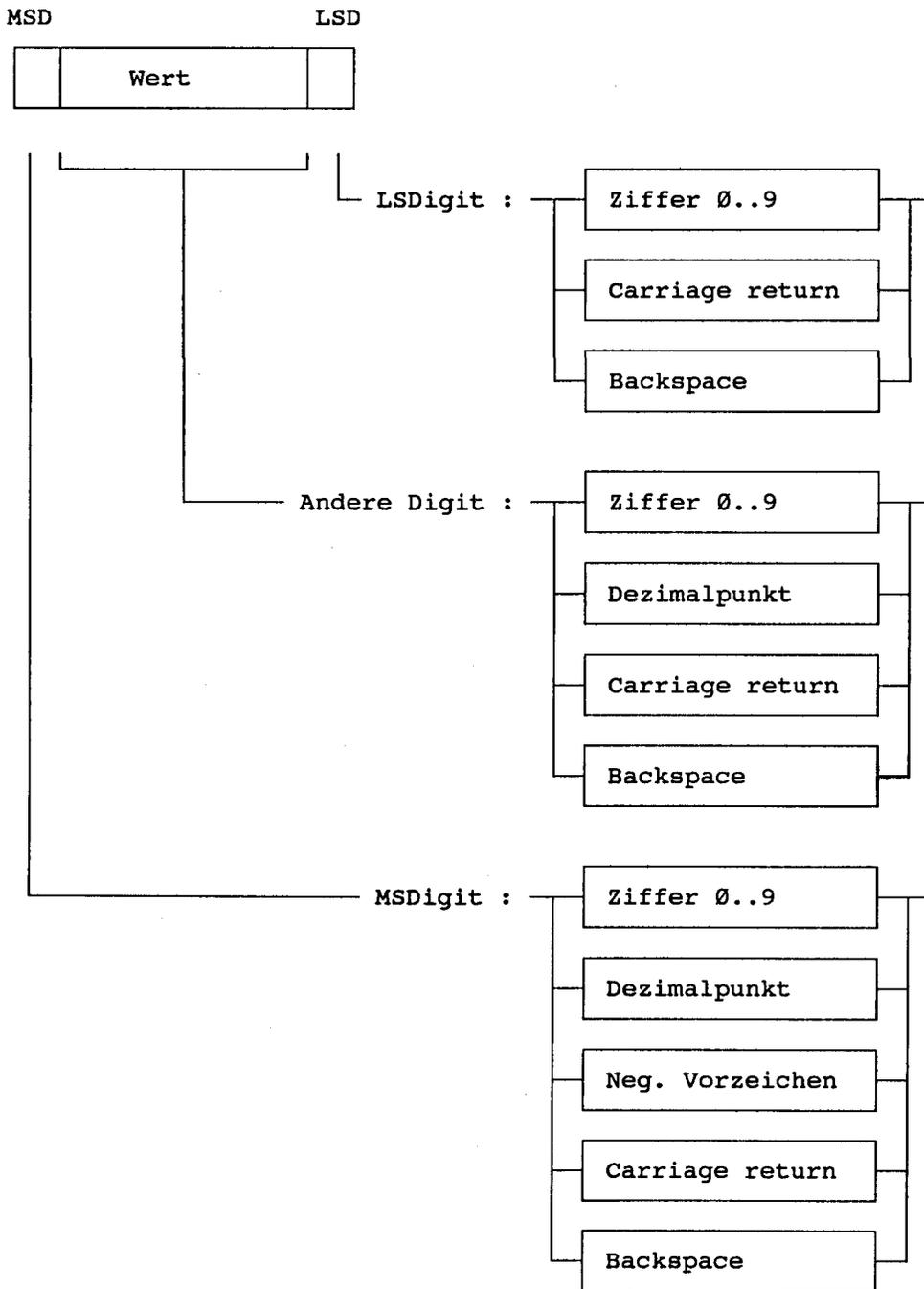
Eingabe/Änderung eines Parameters:

Beim ersten Aufruf des FB wird der Eingabetext ausgegeben, das "IN_BUSY" gesetzt und der Cursor wird 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 Carriage 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.D100:



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 Funktionstaste 4 (F4) definiert.

```
SIGN_K EQU      K          65          ; 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"
-	-	-	-	0
(1. FB-Aufruf)	-	567.890	567890	1
3	51	3_	567890	1
5	53	35_	567890	1
7	55	357_	567890	1
<-	8	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
5	53	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
-	-	-	-	0
(1. FB-Aufruf)	-	1234.700	1234700	1
F4	68	_	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 EXTERNAL 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

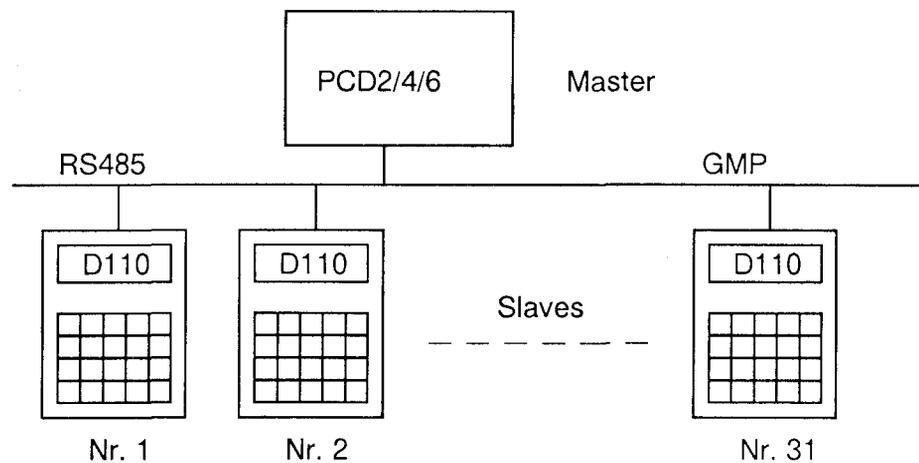
Soll der FB durch die Assemblerdirective \$INCLUDE in das Anwenderprogramm eingebunden werden, so müssen die EXTERNAL Definition gelöscht, respektive ersetzt werden durch lokale Symboldefinitionen.

1.9 Das GMP-Protokoll auf RS485 Bus

GMP ist ein einfaches Multidrop-Protokoll zum Datenaustausch zwischen einem Master (PCD2/4/6) und 2 bis 31 Slave-Terminals PCD7.D110.

Durch einfache Textstrings mit Steuerzeichen (ausgehend von der PCD) kann die Verbindung zu einem Terminal aktiviert werden und anschliessend Daten transparent ausgetauscht werden.

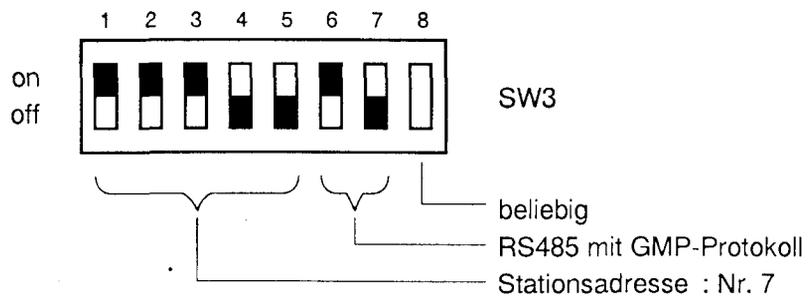
Der Datenaustausch erfolgt über den Zweidrahtbus RS485 (siehe Kap. 1.5.3).



Einstellungen am Terminal ...D110

Mit dem DIP-Schalter SW3 werden die Stationsnummer und die Kommunikations-Funktion eingestellt.

Beispiel:



Schnittstellenassignierung auf der PCD2/4/6

Es wird eine Schnittstelle RS422/485 gewählt und diese im SASI-Befehl mit dem Modus MC4 assigniert.

Beispiel für PCD4.M120 auf Basismodul PCD4.C130:

```
SASI  1      ; Assignierung Kanal 1 (RS422/485)
      10     ; Parameter im Text 10

TEXT  10    "UART:4800,8,E,1;"
      "MODE:MC4;"
      "DIAG:F8000,R4000"
```

Aktivierung bzw. Desaktivierung der Terminals ..D110

- Aktivierung des Terminals mit Adresse "ADDR" (ADDR=1.. 31):

Darstellungsart	
Charakter	ESC (SPACE + ADDR)
Hex	1BH (20H + ADDRH)
Dezimal	<27> <32 + ADDR>

- Wird ADDR = 0 verwendet, so werden **alle** Terminals gleichzeitig aktiviert:

Dezimal	<27> <32>
---------	-----------

- Desaktivierung aller Terminals (ein einzelnes Terminal kann nicht desaktiviert werden):

00H bzw. <0>

Textausgabe auf Display des ..D110

Nachdem das entsprechende Terminal aktiviert wurde, ist die Verbindung von PCD zu D110 transparent für eine beliebige Textausgabe inkl. aller Steuerzeichen nach Kap. 1.7.3 (wie Typ D100).

Beispiel: Ausgabe des Textes "SAIA PCD" an die Terminals Nr. 7 und Nr. 9.

```

TEXT 20  "<27><39><27><41>SAIA PCD"
STXT 1    ; Textausgabe über Kanal 1
      20  ; des Textes Nr. 20
    
```

XBSY abfragen

```

LD    R    10  ; Register mit Null laden
      0
STXD  1    ; ASCII-Null senden = Desaktivierung aller
      R    10  ; Terminals
    
```

Lesen der Tastatureingaben durch die PCD

Eingaben auf die Terminal-Tastatur werden im D110 zwischengespeichert (max. 12 Charakter). Über eine ESCape-Sequenz wird das Terminal aufgefordert, diese Daten zu senden. Dazu muss aber das entsprechende Terminal aktiviert sein. Nach seiner Desaktivierung (<1ms später) sendet das Terminal seinen Bufferinhalt an die PCD.

- Befehl an das Terminal, seinen Tastatur-Buffer zu senden:

Darstellungsart		
Charakter	ESC	P
Hex	1BH	50H
Dezimal	<27>	<80>

- Desaktivierung aller Terminals:

00H bzw. <0>

- Antworttelegramm von D110 --> PCD:

<27><32 + ADDR> Daten aus Tastatur-Buffer <0>

Falls der Buffer leer war wird zurückgesendet:

<27><32><0>

Beispiel: Lesen des Tastatur-Buffers von Terminal Nr. 6

TEXT 30 "<27><38>" ; Terminal 6 aktivieren
" <27><80>" ; Lesebefehl

STXT 1 ; Textausgabe über Kanal 1
30 ; des Textes Nr. 30

XBSY abfragen

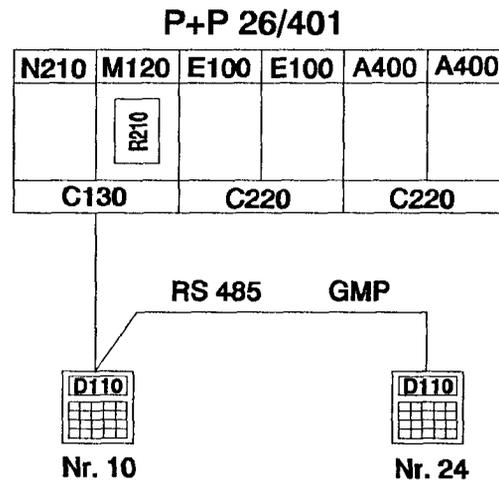
LD R 10 ; Register mit Null laden
0

STXD 1 ; ASCII-Null senden = Desaktivierung und
R 10 ; Auslösung des Antworttelegramms D110 --> PCD

1.10 Programmbeispiele PCD2/4/6 mit ..D110 für Kommunikation mit GMP-Protokoll

Hardware-Installation

Die folgenden Beispiele basieren auf der Hardware der Workshopmodelle P+P 26/401 (V-PCX 20) für die PCD4. Alle Beispiele können jedoch genauso gut auch auf der PCD2 oder PCD6 mit der entsprechenden Konfiguration angewendet werden.



Beispiel 1

Einzelastenerkennung mit nachfolgender Aktion

Am RS485-Netzwerk sind zwei ..D110 angeschlossen. Nach Betätigung der Funktionstasten F1, F2 und F3 können an beiden Terminal, unabhängig voneinander, die folgenden Texte ausgegeben werden:

- Taste F1 : ein einfacher Text wird ausgegeben.
- Taste F2 : ein Text mit dem Zustand der Eingänge 0 bis 15 wird ausgegeben.
- Taste F3 : ein Text mit dem Wert des BCD-Schalters, angeschlossen an die Eingänge 16..31, 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.

Beispiel 2

Eingabe numerischer Werte

An beiden Terminal kann gleichzeitig, menügeführt der Inhalt eines Registers verändert werden. Dazu wird im Programm der Funktionsblock "INPGMP" verwendet, welcher die Editierung eines Parameters via das ..D110 unterstützt. Ein detaillierte Beschreibung des Funktionsblockes befindet sich im Anschluss an die Programm listings.

```

;
;
;   User program example 1 for the industrial terminal PCD7.D110
;   =====
;
;   File :      EX01GMP.SRC
;
;   Creation:  13.01.93      U.Jäggi
;
;
;
;
RBSY_F      EQU      0 32      ; Receiver Busy
XBSY_F      EQU      0 38      ; Text Busy
;-----
DIAG_R      EQU      R 999      ; Diagnostic register
;-----
MAIN        EQU      TEXT 0      ; Main menu
I_STAT      EQU      TEXT 1      ; Displays input 0..15
BCD_SW      EQU      TEXT 2      ; Displays BCD switches
DAT_TI      EQU      TEXT 3      ; Displays date and time
ON_10       EQU      TEXT 4      ; Activates terminal 10
ON_24       EQU      TEXT 5      ; Activates terminal 24
R_BUF_10    EQU      TEXT 6      ; Request keybd buf. of term. 10
R_BUF_24    EQU      TEXT 7      ; Request keybd buf. of term. 24
ASSIGN      EQU      TEXT 999    ; Assignment of the ser. interf.
;-----
CHAN_N      EQU      1          ; Number of serial channel
;-----

TEXT    ASSIGN    "UART:9600,8,E,1;"
                "MODE:MC4;"
                "DIAG:",RBSY_F.T,";",DIAG_R.T,""

TEXT    MAIN      "<12>"          ; Clear display
                "<27><84>"      ; Cursor off
                "Main menu      [F1]<10><13>"
                "Input 0..15    [F2]<10><13>"
                "BCD-Switch     [F3]<10><13>"
                "Date/Time      [F4]"

TEXT    I_STAT    "<12>"          ; Clear display
                "Input Status      "
                "I 0..7 : $I0000<10><13>"
                "I 8..15 : $I0008<10><13>"
                "Main menu [F1]"

TEXT    BCD_SW    "<12>"          ; Clear display
                "BCD-Switch (I16..31)"
                "-----"
                "Value : $R0010<10><13>"
                "Main menu [F1]"

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

```

```

TEXT    ON_10      "<27><42>"           ; Terminal 10
TEXT    ON_24      "<27><56>"           ; Terminal 24

TEXT    R_BUF_10   "<27><42>"           ; Terminal 10
                "<27><42><27>P"       ; Request buffer term. 10

TEXT    R_BUF_24   "<27><56>"           ; Terminal 24
                "<27><56><27>P"       ; Request buffer term. 24
    
```

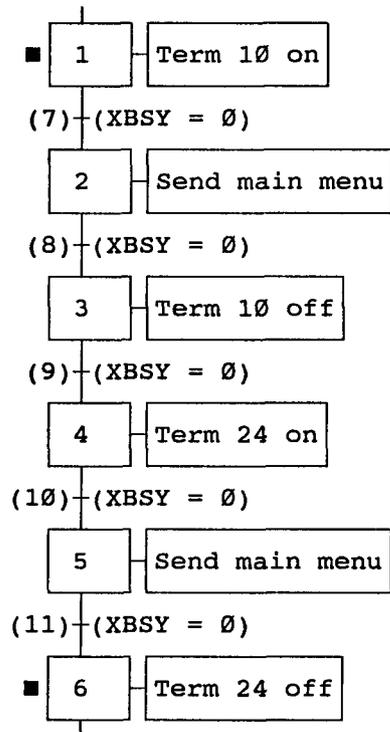
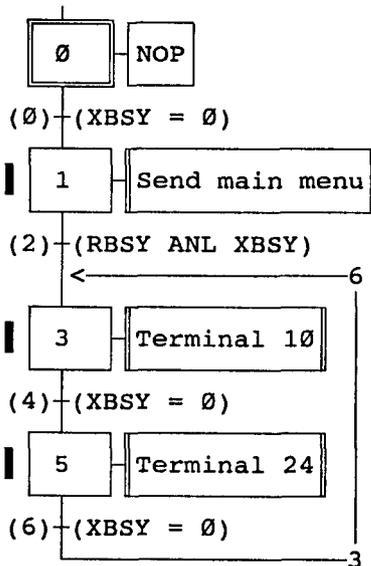
```

XOB      16          ; Cold start
;-----
DEFTB    1           ; Time base = 1/100s
SASI     CHAN_N      ; Assignment RS485 interface
        ASSIGN      ; Text 100
;-----
EXOB

COB      0           ; Main program
        0
;-----
CSB      0           ; Call communication SB
;-----
ECOB
    
```

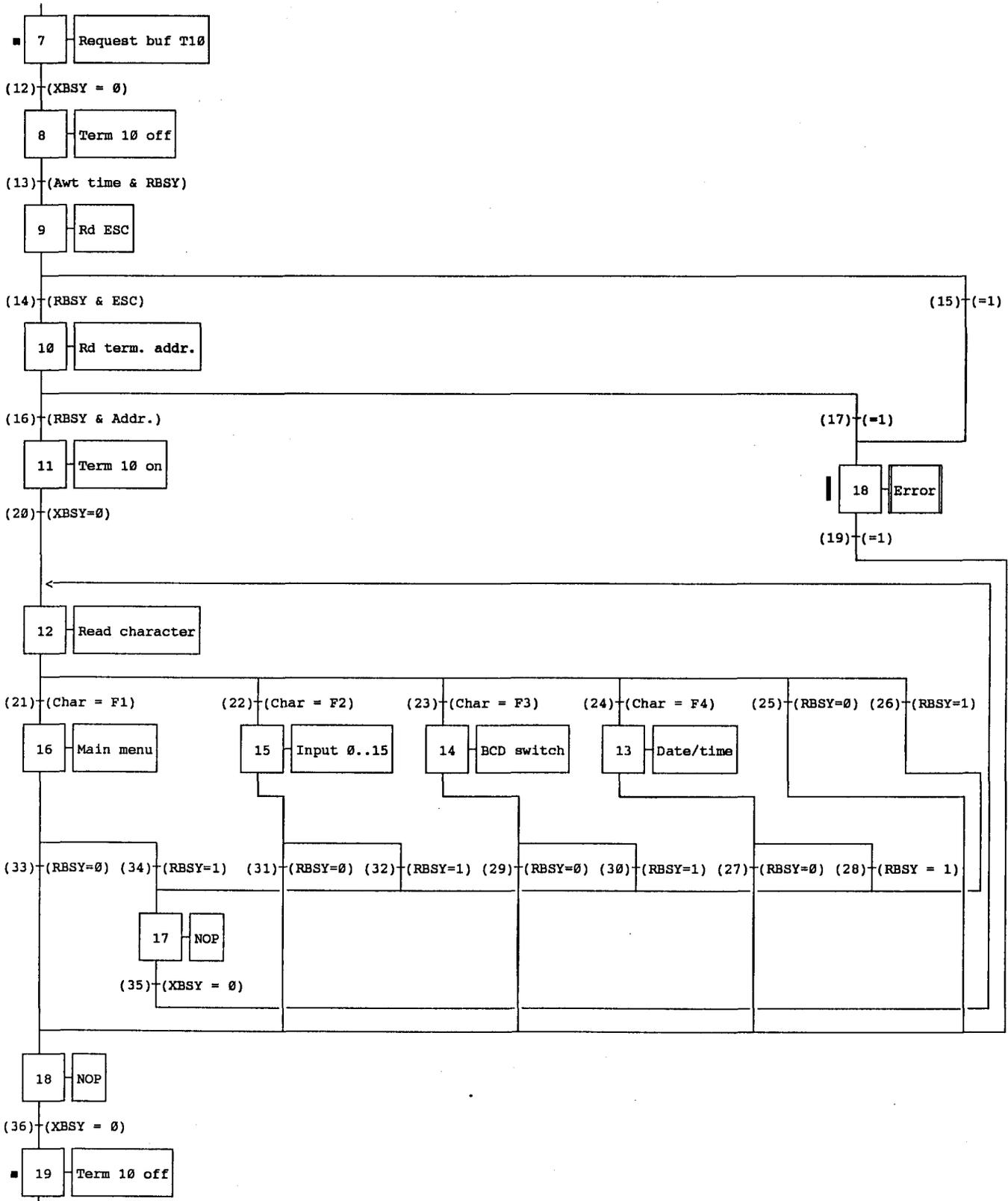
SB-NUMBER: 0
PAGE-NB: 0

SB-NUMBER: 0
PAGE-NB: 1 Send main menu

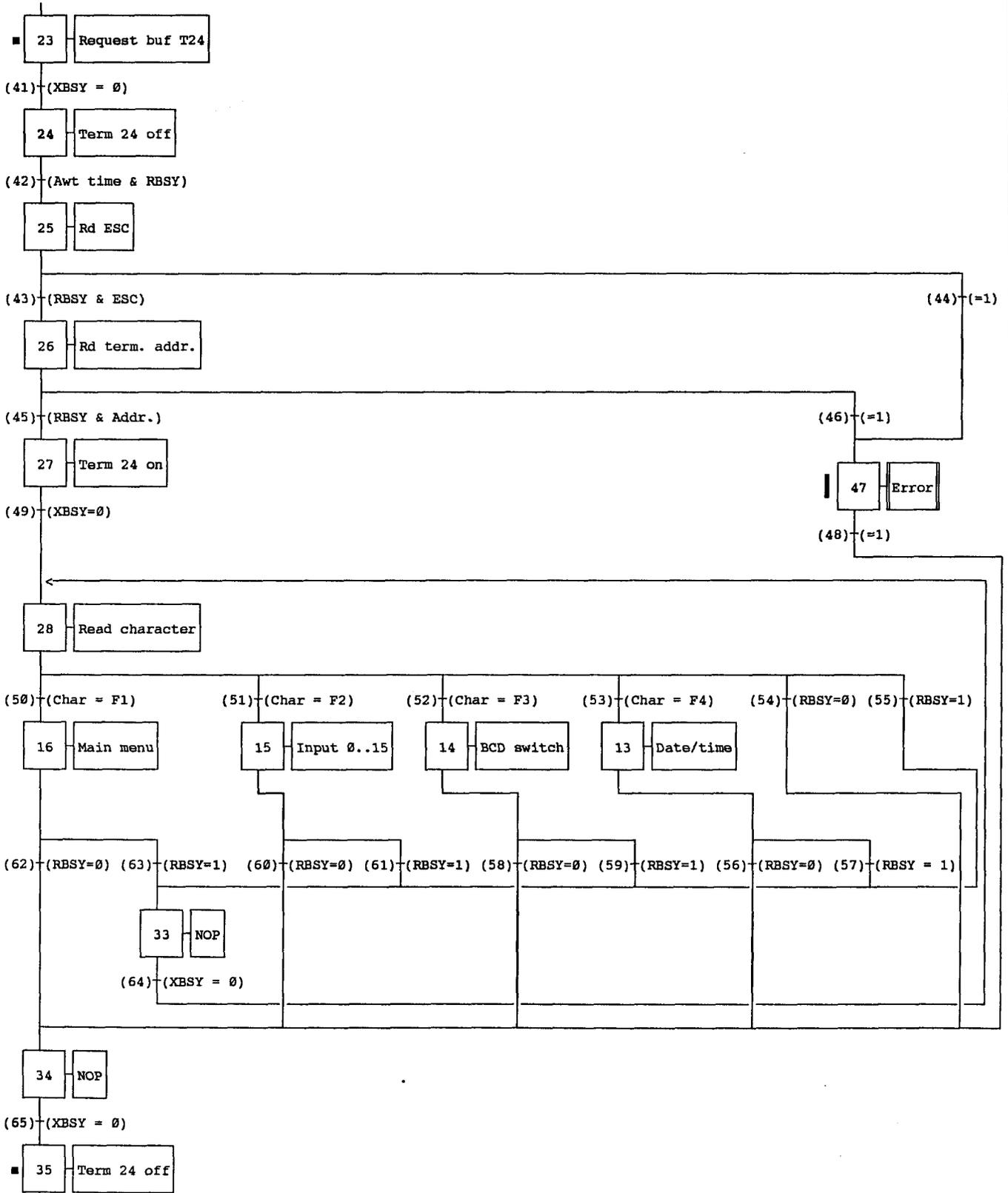


SB-NUMBER: 0

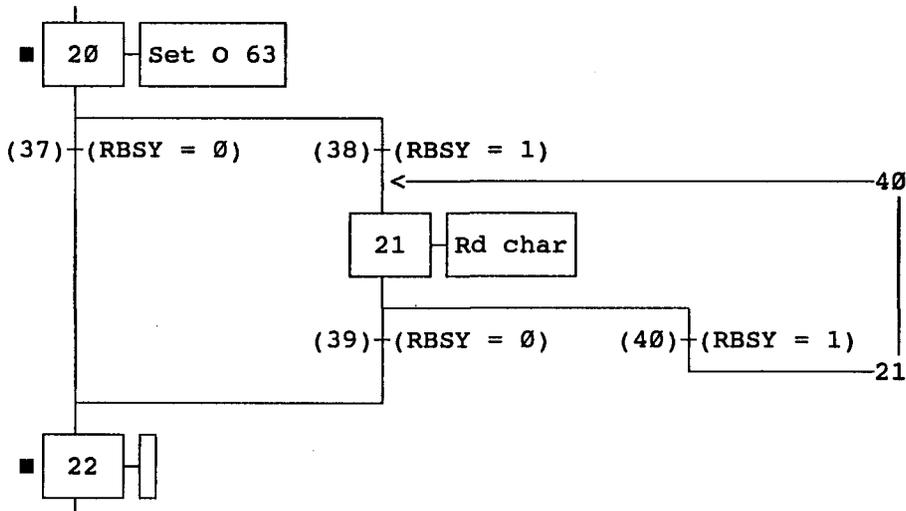
PAGE-NB: 3 Terminal 10



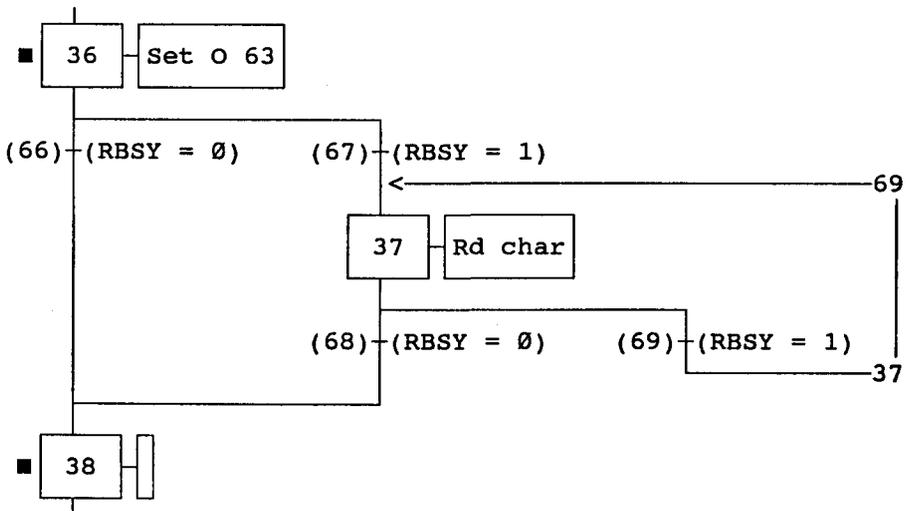
SB-NUMBER: 0
PAGE-NB: 5 Terminal 24



SB-NUMBER: Ø
 PAGE-NB: 18 Error



SB-NUMBER: Ø
 PAGE-NB: 47 Error




```

TEXT    ASSIGN    "UART:9600,8,E,1;"
                    "MODE:MC4;"
                    "DIAG:",RBSY_F.T,";",DIAG_R.T,""

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

TEXT    INTXT10  "<12>"
                    "R-Value : $%00.3d$R0500<10><13>"
                    "ACCEPT VALUE   [CR]"
                    "MODIFY VALUE   [F1]"
                    "MAIN MENU     [F2]"

TEXT    INTXT24  "<12>"
                    "R-Value : $%00.3d$R0501<10><13>"
                    "ACCEPT VALUE   [CR]"
                    "MODIFY VALUE   [F1]"
                    "MAIN MENU     [F2]"

TEXT    ON_10    "<27><42>"            ; Terminal 10
TEXT    ON_24    "<27><56>"            ; Terminal 24

TEXT    R_BUF_10 "<27><42>"            ; Terminal 10
                    "<27><42><27>P"    ; Request buffer term. 10

TEXT    R_BUF_24 "<27><56>"            ; Terminal 24
                    "<27><56><27>P"    ; Request buffer term. 24

XOB     16                ; Cold start
;-----
DEFTB   1                ; Time base = 1/100s
SASI    CHAN_N           ; Assignment RS485 interface
        ASSIGN          ; Text 100

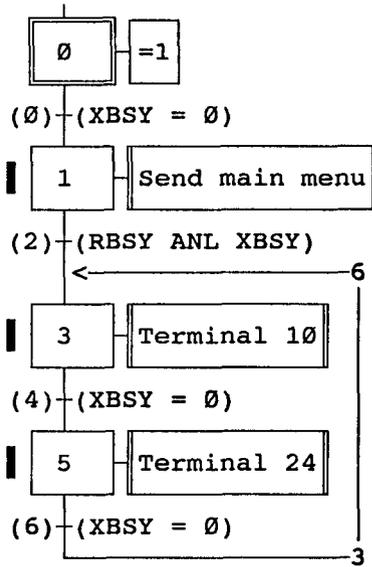
ACC     H
RES     INBSY10          ; Reset input busy flag term. 10
RES     INBSY24          ; Reset input busy flag term. 24
;-----
EXOB

COB     0                ; Main program
        0

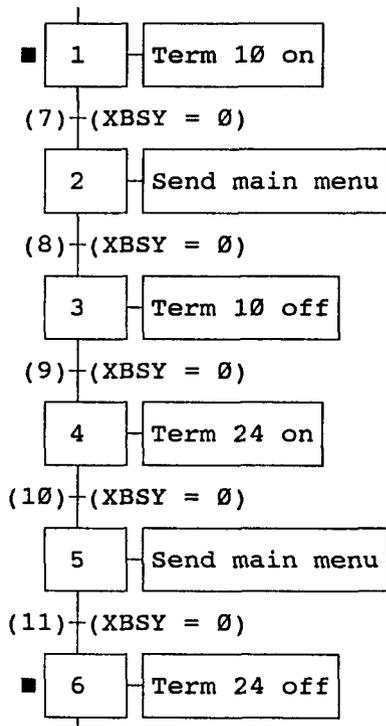
;-----
CSB     0                ; Call communication SB
;-----
ECOB

```

SB-NUMBER: 0
PAGE-NB: 0

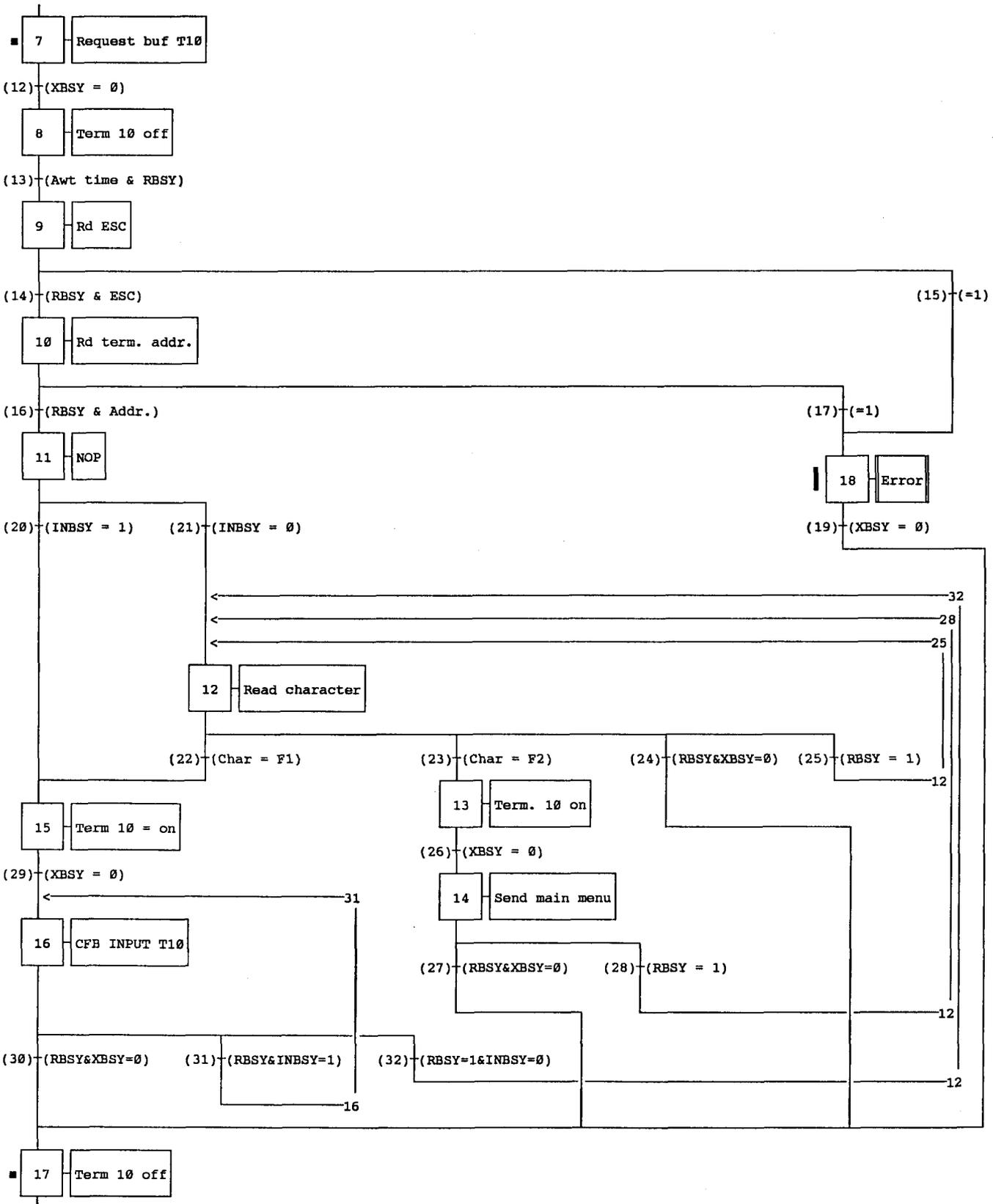


SB-NUMBER: 0
PAGE-NB: 1 Send main menu



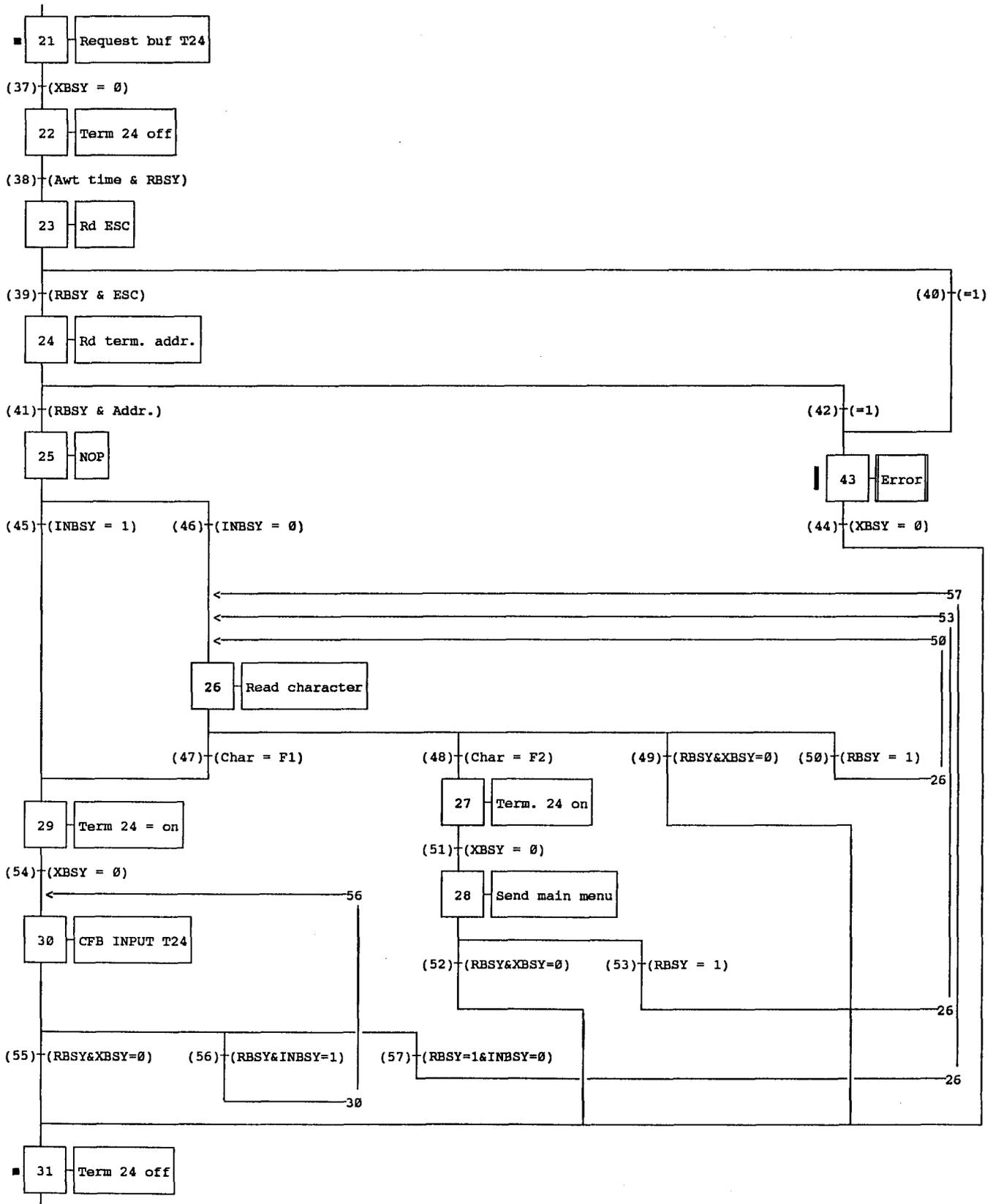
SB-NUMBER: 0

PAGE-NB: 3 Terminal 10

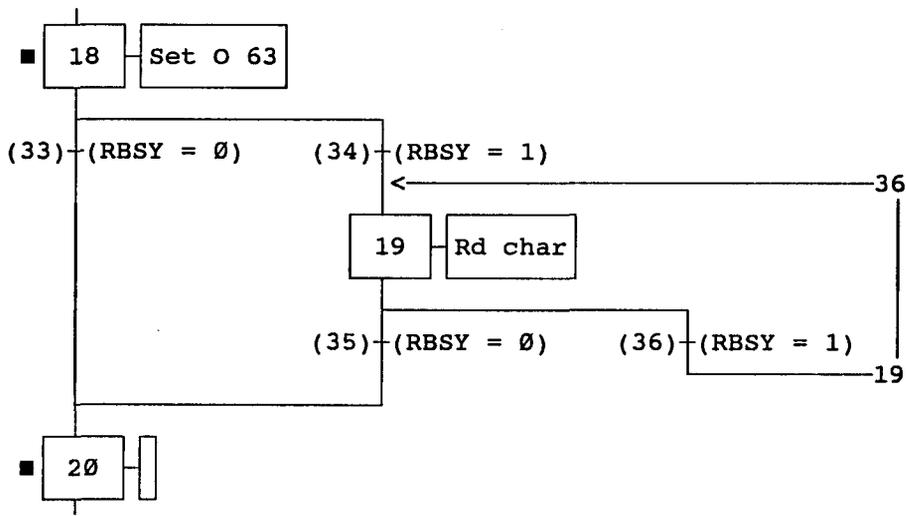


SB-NUMBER: 0

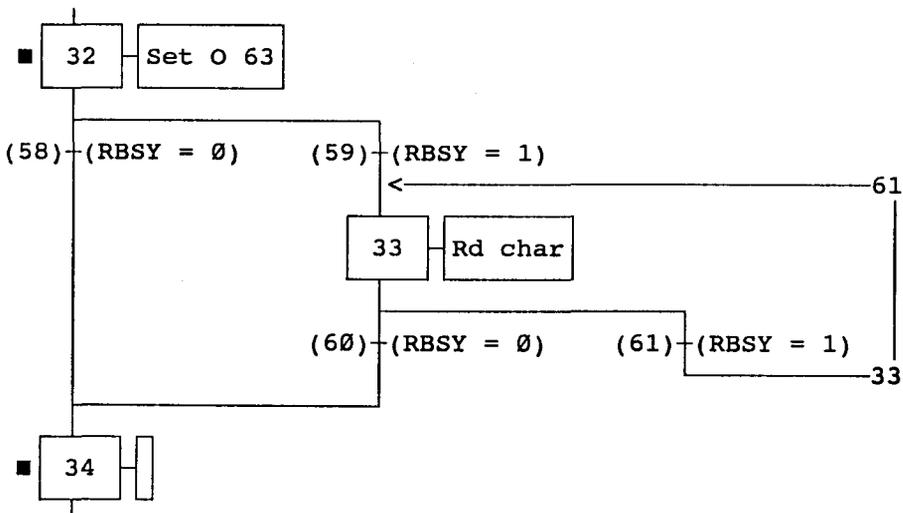
PAGE-NB: 5 Terminal 24



SB-NUMBER: Ø
 PAGE-NB: 18 Error



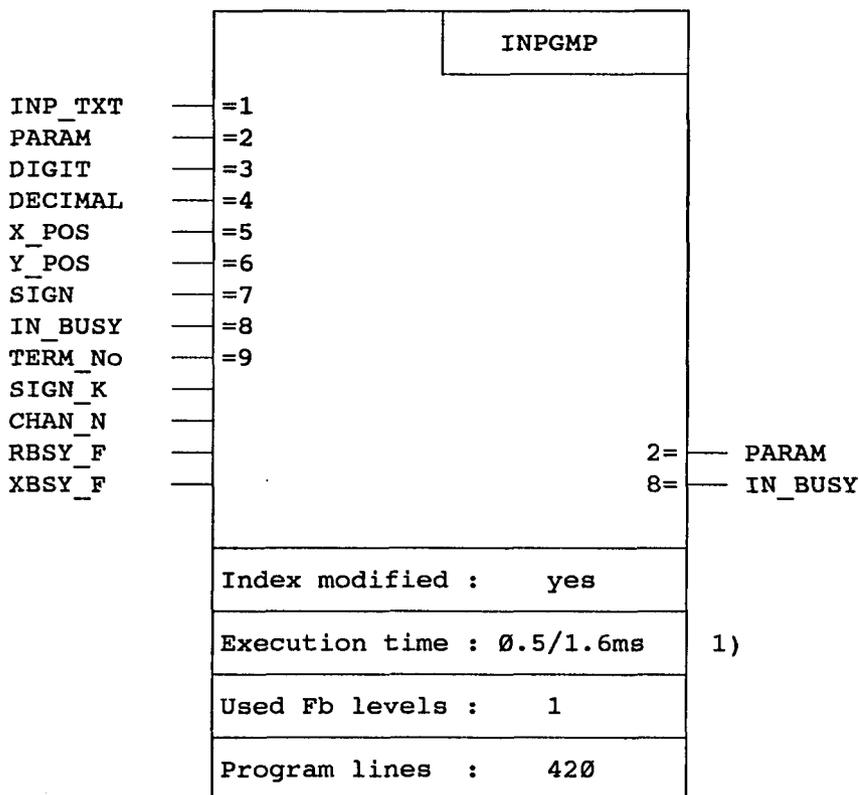
SB-NUMBER: Ø
 PAGE-NB: 43 Error



Funktionsblock:

INPGMP

Parametereingabe am Terminal PCD7.D110 via RS 485-Bus mit GMP-Protokoll



- 1) 0.5ms : RBSY_F = "0" (kein Zeichen im Empfangsbuffer)
 1.6ms : RBSY_F = "1" (Ziffer 0..9 wird vom Empfangsbuffer gelesen und verarbeitet.)

Funktionsbeschreibung

Mit dem Funktionsblock kann der Inhalt eines Registers oder Zählers via das Industrieterminal PCD7.D110 verändert werden. Die Eingabe des negativen Vorzeichen und eines Dezimalpunktes wird unterstützt. Für die Kommunikation wird das GMP-Protokoll im Modus MC4 verwendet. Dadurch können gleichzeitig an mehreren, am RS485-Bus angeschlossenen, ..D110 Parameter eingegeben und verändert werden.

Der FB unterstützt die gleichen Funktionen wie der FB "INPUT" für die Parametereingabe mit dem PCD7.D100, musste jedoch erweitert werden, um die Kommunikation im GMP-Protokoll über den RS485-Bus zu ermöglichen. In der Folge sind deshalb auch nur die erwähnten Erweiterungen detailliert beschrieben. Die Beschreibung des FB "INPUT" befindet sich in Kapitel 1.8 in diesem Handbuch.

Liste der Ein- und Ausgänge

Symbol	Beschreibung	Parameter	Typ	Daten Format	Wert	Adresse
INP_TXT	Input Text Eingabe Text	yes	X	Text	beliebig	0..3999
PARAM	Input parameter Eingabe Register oder Zähler	yes	R/C	Integer	-2147483648 ...+2147483647	0..4095
DIGIT	Number of digits Anzahl Eingabestellen	yes	R	Integer	1..11	0..4095
DECIMAL	Number of decimal places Anzahl Nachkomma- stellen	yes	R	Integer	0,1..10	
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
IN_BUSY	Input Busy Eingabe Busy Flag	yes	F/O	Binary	0/1	0..8191
TERM_No	Terminal number Nummer des Terminals	yes	K	Integer	1..31	-
SIGN_K	Sign key ASCII-Code der Vorzeichentaste	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

■ Eingabe Busy Flag "IN_BUSY":

Das Flag zeigt an, wann die Eingabe/Editierung eines Parameters abgeschlossen ist. Vor dem ersten Aufruf des FB's 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. Sobald vom Terminal ein Carriage Return empfangen wird, ist die Eingabe abgeschlossen und das Flag wird zurückgesetzt. Da an mehreren Terminals gleichzeitig Werte eingegeben werden können, muss beim Aufruf des FB's für jedes Terminal ein separates "IN_BUSY" Flag verwendet werden.

■ Nummer des Terminals "TERM_No":

Mit diesem Parameter werden die Terminal spezifischen Arbeits-Flag und -Register im Datenspeicher adressiert. Um diese Adressierung vorzunehmen, müssen alle am RS485-Bus installierten Terminals durchnummeriert werden von 1 bis 31. Beim Aufruf des FB, wird die Terminalnummer mit diesem Parameter übergeben.

Vorsicht:

Diese Nummer ist nicht identisch mit der Terminaladresse, welche mittels DIL-Schalter am ..D110 eingestellt wird. Terminal-Adresse und -Nummer können deshalb auch unterschiedlich gewählt werden.

Aufruf des Funktionsblockes

```

CFB      INPGMP      ; 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)
F  IN_BUSY; Input busy flag
          TERM_No   ; Terminal number
    
```

Beispiel: FB-Aufruf für Terminal Nr. 2

```

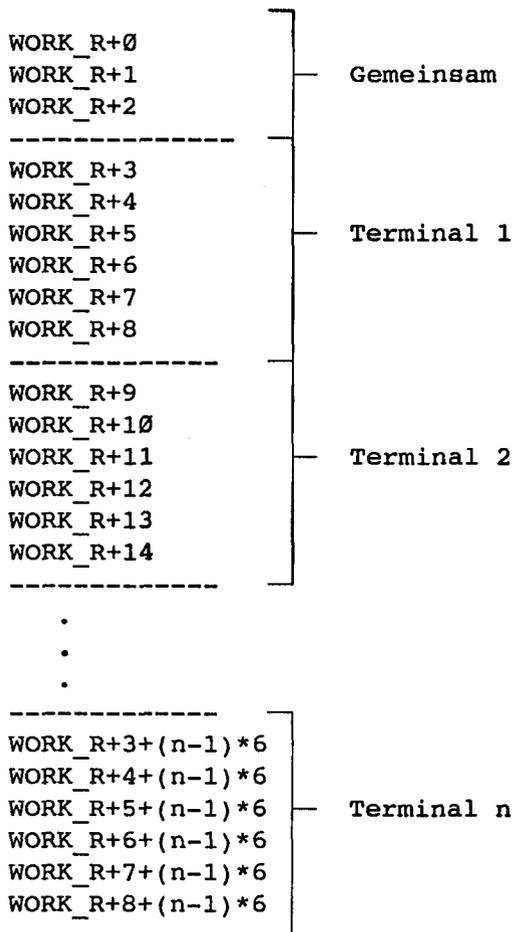
CFB      1          ; Input
X  100    ; Input text
R  50     ; Parameter
R  10     ; Number of digits
R  11     ; Number of decimal places
R  20     ; X-position
R  21     ; Y-position
F  2000   ; Sign input yes/no (1/0)
O  32     ; Input busy flag
          2          ; Terminal number
    
```

Intern verwendete und reservierte Elemente mit symbolischem Namen:

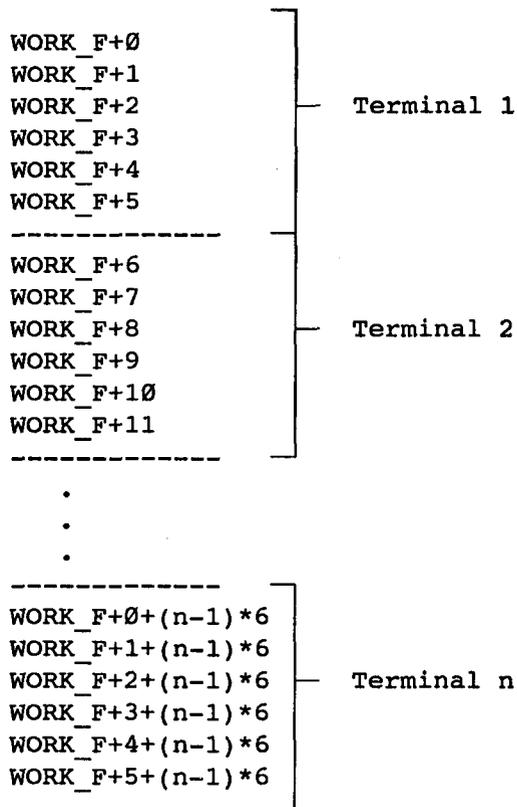
Von dem FB werden intern für jedes Terminal eine Anzahl Arbeits-Register und -Flag verwendet. Diese Register und Flag enthalten während der Eingabe eines Parameters Zwischenergebnisse und dürfen deshalb nur für diesen FB verwendet werden. In der FB-Datei müssen lediglich die Basisadressen der verwendeten Elemente definiert werden.

Symbol	Beschreibung	Daten		Adresse
		Typ	Format	
WORK_R	Base address of used work registers Basisadresse der Arbeitsregister	R	Integer	0..4087 (+8)
WORK_F	Base address of used work flags Basisadresse der Arbeitsflag	F	Binary	0..8186 (+5)

Verwendung der Register:



Verwendung der Flag:



Verwendung des Funktionsblockes im Anwenderprogramm

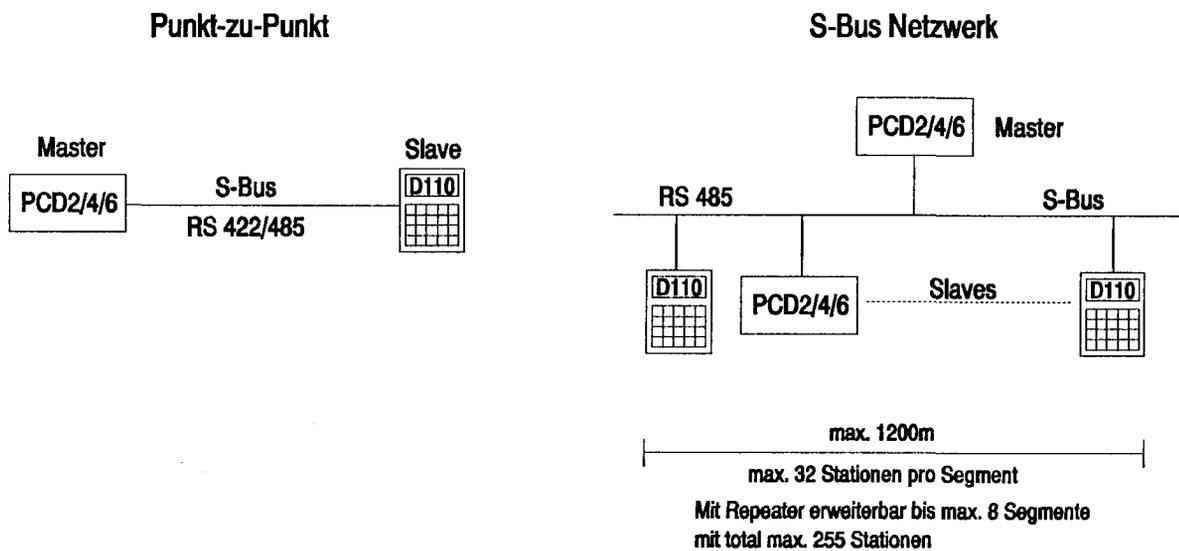
Der FB befindet sich in der Datei D1GMPFB.SRC. Die Datei enthält nebst dem FB auch alle Symboldefinitionen, welche zu dessen Benützung notwendig sind. Alle Symbole, welche global im Anwenderprogramm verwendet werden, sind in der Datei als EXTERNAL definiert und müssen in einer andern Anwenderdatei definiert werden. Auf diese Weise muss die Datei D1GMPFB.SRC nur einmal assembliert und anschliessend mit den andern Anwenderdateien gelinkt werden.

Globale Symbole: INPGMP, CHAN_N, RBSY_F, XBSY_F

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

1.11 Das S-Bus Protokoll auf den Schnittstellen RS 422 und RS 485

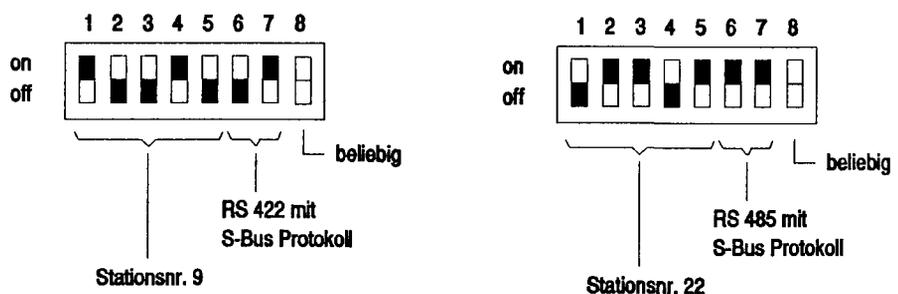
Mit der RS422-Schnittstelle wird das S-Bus Protokoll für eine Punkt zu Punkt-Verbindung angewendet, um Daten zwischen einer Master-Station (PCD2/4/6) und einem Slave-Terminal PCD7.D110 auszutauschen. Mit der RS485-Schnittstelle kann ein Master/Slave-Netzwerk, bestehend aus einer Master-Station (PCD2/4/6) und mehreren Slave-Stationen (..D110 gemischt mit PCD2/4/6) aufgebaut werden. Pro Netzwerk können bis zu max. 31 Slave-Terminals PCD7.D110 angeschlossen werden.



Einstellungen am Terminal ..D110

Mit dem DIP-Schalter SW3 werden die Stationsnummer und die Kommunikations-Funktion eingestellt.

Beispiele:



Schnittstellenassignierung in der PCD2/4/6 (Master-Station)

Es wird eine Schnittstelle RS422/485 gewählt und diese mit dem SASI-Befehl im Modus SM1 assigniert.

Beispiel für PCD4.M120 auf Busmodul PCD4.C130:

```
SASI      1      ; Assignierung Kanal 1 (RS485)
          999    ; Parameter im Text 999

TEXT      999    "UART:9600;"
          "MODE:SM1,R4095;"
          "DIAG:F100,R4000"
```

Textausgabe auf Display des ..D110

Texte können im S-Bus Protokoll nicht direkt gesendet werden. Texte müssen deshalb vorgängig in Register transferiert werden, bevor sie anschliessend zum ..D110 gesendet und dort angezeigt werden können. Dabei werden auch alle in den Registern enthaltenen Steuerzeichen verarbeitet.

Format des Sende-Befehls:

```
STXM[X]   Kanal      ; Kanal-Nummer (0..3)
          Anzahl     ; Anzahl Register (1..32)
          Quelle (i) ; Basisadr. Quellregister (R0..4095)
          Ziel (i)  ; Basisadr. Zielregister im ..D110 (fix R10)
```

Beispiel:

```
STXM      1          ; Der Inhalt der
          21         ; Register 100 bis R120
          R 100      ; wird am
          R 10       ; adressierten ..D110 angezeigt
```

Befehls-Reihenfolge, um Texte oder Daten am ..D110 anzuzeigen:

1. Sende-Register mit ASCII-Code des Anzeigetextes (Text und/oder Daten) laden.
2. TBSY-Flag der Schnittstelle abfragen.
3. Terminal adressieren und gewünschte Anzahl Register mit STXM-Befehl übertragen.

Text in Sende-Register kopieren:

Mit dem Befehl GET kann ein definierter Text vom Anwenderspeicher in Register kopiert werden.

Format:

```
GET[X]    Quell-Element (i)    ; Text 0...7999
          Ziel-Element         ; Sende-Register 0...4095
```

Beispiel: Text 120 soll in Register mit Basisadresse 500 kopiert werden.

```
TEXT 120  "<12>"                ; Löscht die Anzeige
          "DIES IST EIN TEXT"
```

```
GET      X 120                ; Kopiere Text 120
          R 500                ; in Register 500 und folgende
```

Resultat:

Register	ASCII-Wert	Hex.-Wert
500	DIE	12687369
501	S IS	83327383
502	T EI	84326973
503	N TE	78328469
504	XT	88840000

(Die Register 500-504 waren vor der Ausführung des Befehls alle = 0)

Wichtig:

Es ist zu beachten, dass die Senderegister mit der höchsten Adresse (in diesem Beispiel Register 504) vor der Ausführung des Befehls mit dem Wert Null geladen werden, da beim kopieren des Textes im letzten Register nur zwei Byte überschrieben werden. Der Wert Null (ASCII-NUL) hat beim ..D110 keine Funktion und wird deshalb ignoriert.

Die Ausgabe von formatierten Daten wird mit dem Befehl GET nicht unterstützt. D.h. bei der Ausführung des Befehls werden Sondertexte (z.B. \$Rnnnn) nicht als solche erkannt und die Daten deshalb nicht aufbereitet im ASCII-Format in die Sende-Register gespeichert.

Daten in Sende-Register kopieren

Daten sind in der PCD im binären Format gespeichert. Um beispielsweise einen Registerinhalt oder den Zustand eines Einganges am ..D110 anzuzeigen, müssen die Daten vom Binär- in das ASCII-Format gewandelt und in die Sende-Register kopiert werden.

Dazu stehen folgende Funktionsblöcke zur Verfügung:

BASCIR	Binär → ASCII -Umwandlung für die Ausgabe von Register, Timer oder Zähler
BASCIF	Binär → ASCII -Umwandlung für die Ausgabe von Flag, Ein- und Ausgängen
TI_DAT	Binär → ASCII -Umwandlung für die Ausgabe von Datum und Zeit

Siehe dazu auch detaillierte Beschreibung der Funktionsblöcke in Kapitel 1.12

Lesen der Tastatureingaben durch die PCD

Eingaben auf die Terminal-Tastatur werden im ...D110 zwischengespeichert. Der Tastaturpuffer verfügt über eine Kapazität von 12 Byte, was 12 ASCII-Charaktern entspricht. Mit einem SRXM-Befehl kann der gesamte Puffer (3 Register), nur 8 Charakter (2 Register) oder auch nur 4 Charakter (1 Register) gelesen werden.

Format des Lese-Befehls:

SRXM[X]	Kanal	; Kanalnummer (0..3)
	Anzahl	; Anzahl Register (1..3)
	Quelle (i)	; Basisadr. Quellregister im ..D110 (fix R 0)
	Ziel (i)	; Basisadr. Empfangsregister PCD (R0..4095)

Beispiel: der gesamte Tastaturpuffer soll gelesen werden

SRXM	1	; Lesen des gesamten
	3	; Tastaturpuffers und
	R 0	; speichern in
	R 20	; Register 20 bis 22

In den Empfangsregistern kommt folgendes Format zur Anwendung:

Register	MSByte			LSByte
	3	2	1	0
R_BUF+0	1. Charakter	2. Charakter	3. Charakter	4. Charakter
R_BUF+1	5. Charakter	6. Charakter
R_BUF+2	letzter Char.	ASCII-NUL	ASCII-NUL

Charakter = ASCII-Code einer gedrückten Taste

Die letzte gedrückte Taste in den Empfangsregistern ist markiert durch nachfolgende ASCII-NUL (= Dez. 0). Um Tastatureingaben auszuwerten, werden also die Empfangsregister verarbeitet bis ein ASCII-Nul entdeckt wird. Dazu steht folgender Funktionsblock zur Verfügung:

P_CHAR Empfangener Tastaturpuffer verarbeiten

Siehe dazu auch detaillierte Beschreibung der Funktionsblöcke in Kapitel 1.12

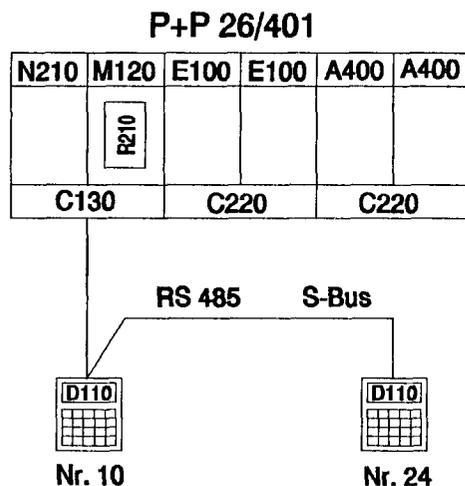
Befehls-Reihenfolge, um Tastatureingaben zu lesen und zu verarbeiten:

1. TBSY-Flag der Schnittstelle abfragen.
2. Tastaturpuffer mit SRXM-Befehl vom ..D110 lesen.
3. Funktionsblock P_CHAR aufrufen um Tasteneingaben auszuwerten.

1.12 Programmbeispiele PCD2/4/6 mit ..D110 für Kommunikation mit S-Bus Protokoll

Hardware-Installation

Die folgenden Beispiele basieren auf der Hardware der Workshopmodelle P+P 26/401 (V-PCX 20) für die PCD4. Alle Beispiele können jedoch genauso gut auch auf der PCD2 oder PCD6 mit der entsprechenden Konfiguration angewendet werden.



Beispiel

Einzeltastenerkennung mit nachfolgender Aktion

Am S-Bus Netzwerk sind zwei ..D110 angeschlossen. Nach Betätigung der Funktionstasten F1, F2 und F3 können an beiden Terminal, unabhängig voneinander, die folgenden Texte ausgegeben werden:

- Taste F1 : ein einfacher Text wird ausgegeben.
- Taste F2 : ein Text mit dem Zustand der Eingänge 0 bis 15 wird ausgegeben.
- Taste F3 : ein Text mit dem Wert des BCD-Schalters, angeschlossen an die Eingänge 16..31, 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.

```

;
;   User program example for the terminal PCD7.D110 with S-Bus
;   =====
;
;   File:      EXSBUS.SRC
;
;   Creation:  20.01.93  U.Jäggi
;
;=====
; Symboldefinitions
; =====
;-----
R_BUF  EQU    R 0    ; Base address of 3 receive registers
CHAR   EQU    R 3    ; Processed ASCII charakter
ST_NO  EQU    R 4    ; Register for the slave station number
WORK_R EQU    R 5    ; Base address of 3 work registers
TRANSM EQU    R 8    ; Base address transmit registers (max. 31)
R_BUSY EQU    F 0    ; Receive busy flag
T_D    EQU    F 0    ; Time/date flag

; Diagnostic outputs serial interface
;-----
RBSY   EQU    O      32      ; Receiver Busy
RFUL   EQU    O      RBSY+1  ; Receive Buffer Full
RDIA   EQU    O      RBSY+2  ; Receiver Diagnostic
TBSY   EQU    O      RBSY+3  ; Transmitter Busy
TFUL   EQU    O      RBSY+4  ; Transmit Buffer Full
TDIA   EQU    O      RBSY+5  ; Transmitter Diagnostic
XBSY   EQU    O      RBSY+6  ; Text Busy
NEXE   EQU    O      RBSY+7  ; Not Executed

; Global symbols:
; -----
EXTN BASCIR      ; Process receive registers
EXTN P_CHAR     ; Binary --> ASCII conversion
EXTN P_CUR      ; Position cursor
EXTN RES_R      ; Reset registers
EXTN BASCIF     ; Binary --> ASCII conversion F/I/O
EXTN TI_DAT     ; Time/date conversion

```

```

; Text definitions
; -----
TEXT      1      "<12>"                ; Clear display
              "<27><84>"                ; Cursor off
              "Main menu   [F1]<10><13>"
              "Input 0..15 [F2]<10><13>"
              "BCD-Switch [F3]<10><13>"
              "Date/Time  [F4]"

TEXT      2      "<12>"                ; Clear display
              "Input Status           "
              "I 0..7   : <10><13>"
              "I 8..15 : <10><13>"
              "Main menu [F1]"

TEXT      3      "<12>"                ; Clear display
              "BCD-Switch (I16..31)"
              "-----"
              "Value   : <10><13>"
              "Main menu [F1]"

TEXT      4      "<12>"                ; Clear display
              "Date   : <10><13>"
              "Week   : <10><13>"
              "Time   : <10><13>"
              "Main menu [F1]"

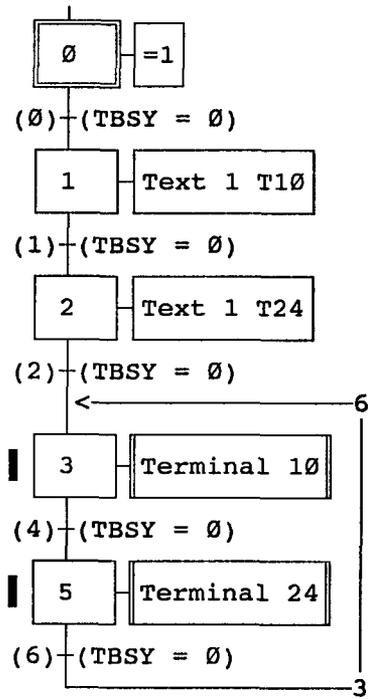
TEXT      999    "UART:9600,500;MODE:SM1,R4;DIAG:032,R999"

;-----
; Coldstart
;-----
XOB          16
SASI         1          ; Assigination RS485 interface
              999      ; Text 999
EXOB

;-----
; Main program
;-----
COB          0
              0
CSB          0          ; D110 communication
ECOB

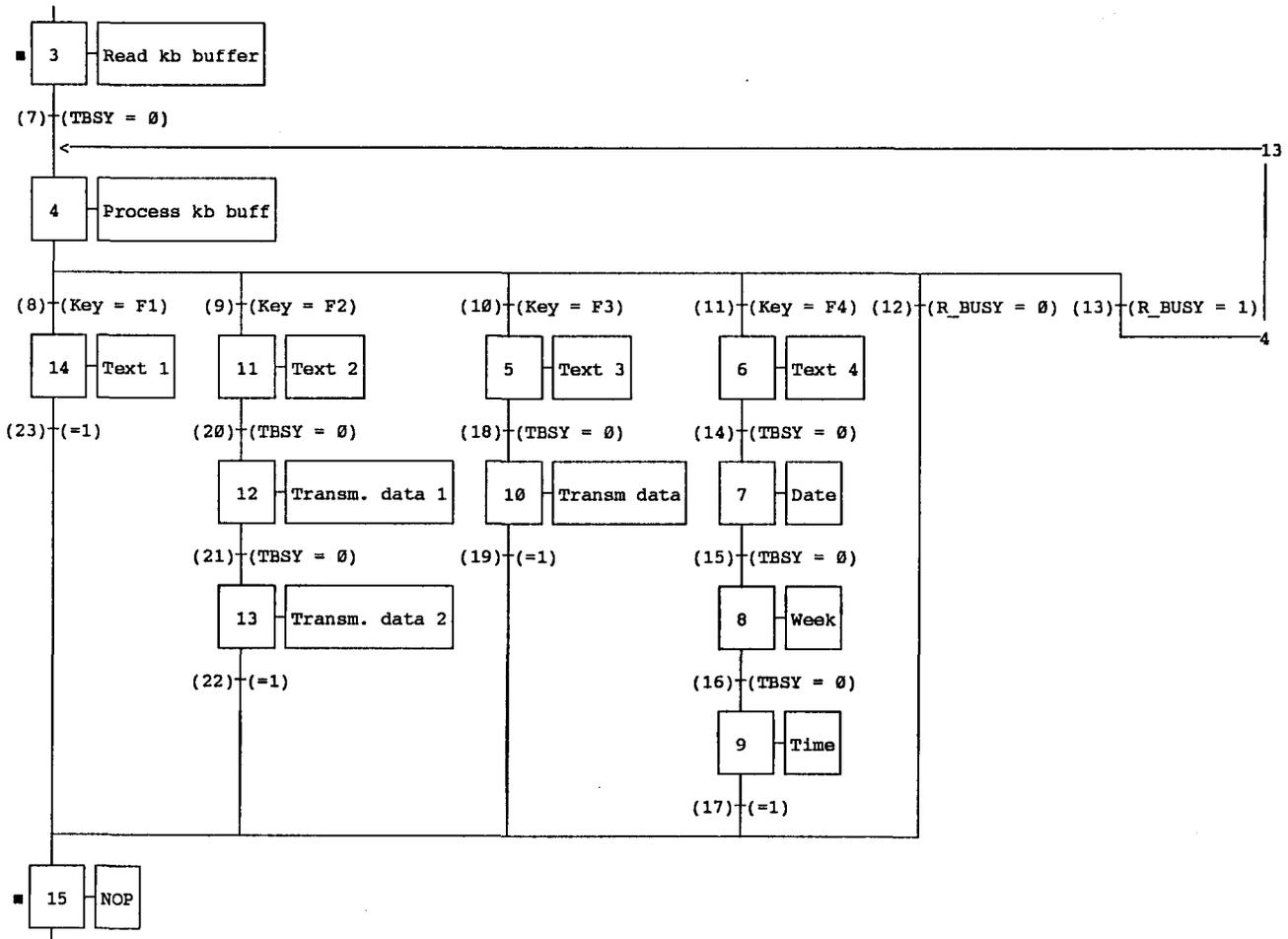
```

SB-NUMBER: Ø
PAGE-NB: Ø



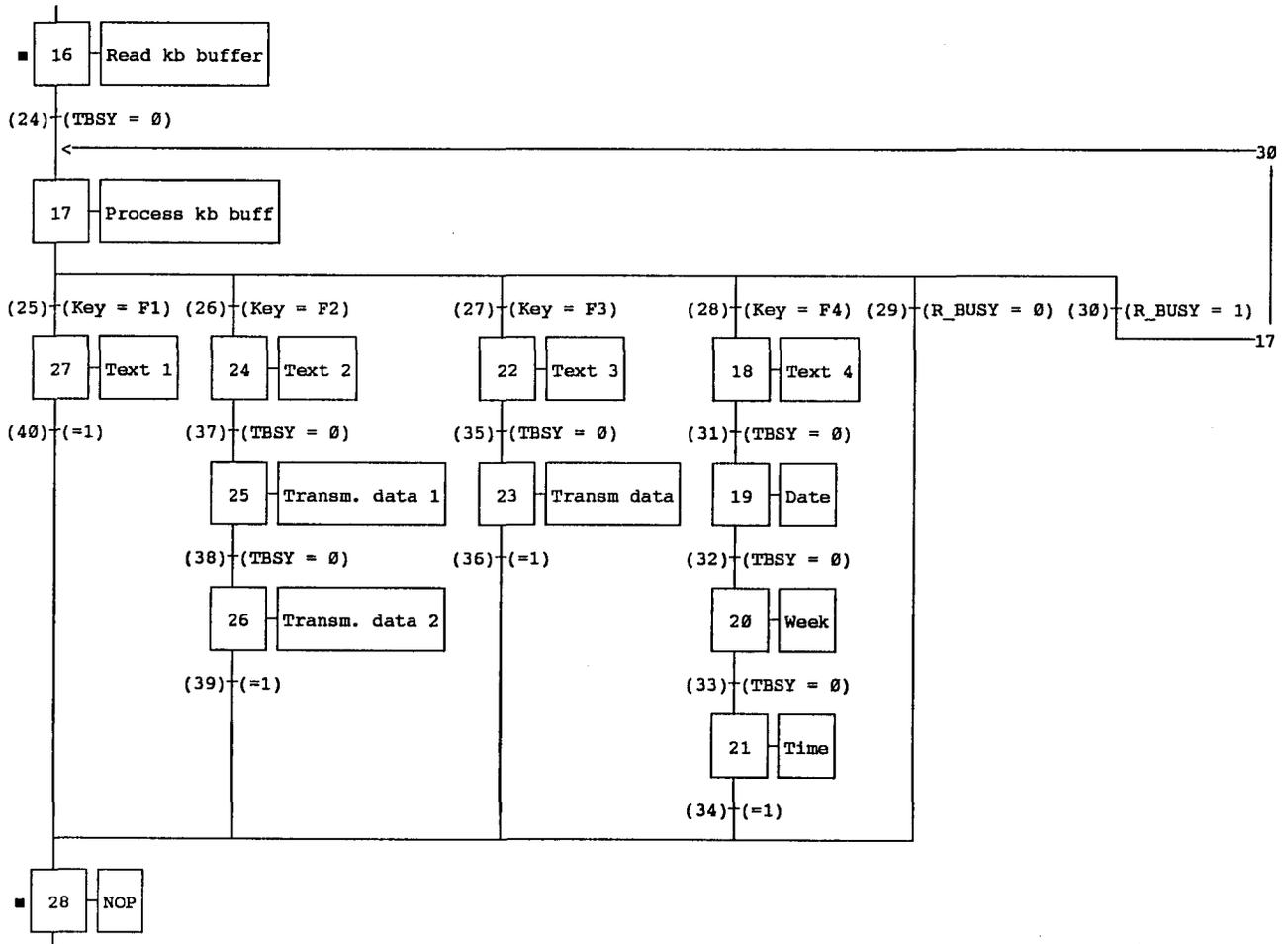
SB-NUMBER: 0

PAGE-NB: 3 Terminal 10



SB-NUMBER: 0

PAGE-NB: 5 Terminal 24



Anwender-Funktionsblöcke für den Betrieb des Terminals ..D110 mit dem S-Bus Protokoll

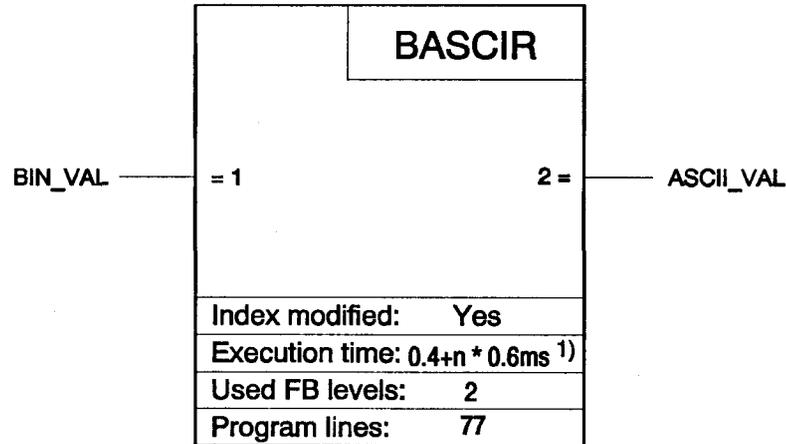
Verwendung der Funktionsblöcke im Anwenderprogramm

Um die Programmierung für das ..D110 mit S-Bus möglichst einfach zu gestalten, steht eine Reihe von Funktionsblöcken zur Verfügung. Die FB's befinden sich in der Datei D1SBUSFB.SRC. Die Datei enthält nebst den FB's auch alle Symboldefinitionen, welche zu deren Benützung notwendig sind. Die symbolischen Namen der FB's sind in der FB-Datei als PUBLIC definiert und müssen in der Anwenderdatei, von welcher sie aufgerufen werden, als EXTERNAL definiert werden. Auf diese Weise muss die FB-Datei D1SBUSFB.SRC nur einmal assembliert und anschliessend mit den anderen Anwenderdateien gelinkt werden. Mit der Assemblerdirektive \$INCLUDE kann die Datei auch in andere Anwenderdateien eingebunden werden.

Übersicht der Funktionsblöcke

BASCIR	Binär → ASCII -Umwandlung für die Ausgabe von Register, Timer oder Zähler (entspricht den Sondertexten \$R/T/Cnnnn im Modus C, Ausgabe im Normal-Format)
BASCIF	Binär → ASCII -Umwandlung für die Ausgabe von Flag, Ein- u. Ausgängen (ähnlich den Sondertexten \$I/O/Fnnnn im Modus C)
TI_DAT	Binär → ASCII -Umwandlung für die Ausgabe von Datum und Zeit (entspricht den Sondertexten \$H und \$D im Modus C)
P_CUR	Positionierung des Cursors
RES_R	Register löschen
P_CHAR	Empfangener Tastaturpuffer verarbeiten

Funktionsblock: **BASCIR** (Binary-ASCII Conversion Register)
 Binär → ASCII -Umwandlung
 für die Ausgabe von Register, Timer oder Zähler



- 1) Pos. Zahl: 0.4 + n * 0.6ms
 Neg. Zahl: 0.6 + n * 0.6ms
 wobei n = Anzahl Dezimalstellen

Funktionsbeschreibung

Mit dem FB wird der Inhalt eines Registers, Timers oder Zählers vom binären Datenformat, Digit um Digit umgewandelt und als ASCII-Zeichen in 3 Ausgaberegister gespeichert. Anschliessend kann der gewandelte Wert mit einem STXM-Befehl via S-Bus am Terminal PCD7.D110 angezeigt werden.

Liste der Ein- und Ausgänge

Symbol	Beschreibung	Parameter	Daten			Adresse
			Typ	Format	Wert	
BIN_VAL	R/T/C containing binary value Register/Timer/Zähler mit Binärwert	yes	R/T/ C	Binär	-2'147'483'647.. +2'147'483'647	0..4095
ASCII_VAL	Base addr. of 3 ASCII registers Basisadresse von 3 Registern, die den ASCII-Wert enthalten	yes	R	ASCII	0..255	0..4093

Aufruf des Funktionsblockes

CFB BASCIR ; Binär-ASCII Umwandlung
 BIN_VAL ; Register, Timer od. Zähler mit Binärwert
 ASCII_VAL ; Basisadr. der 3 Register mit ASCII-Wert

Funktionsprinzip

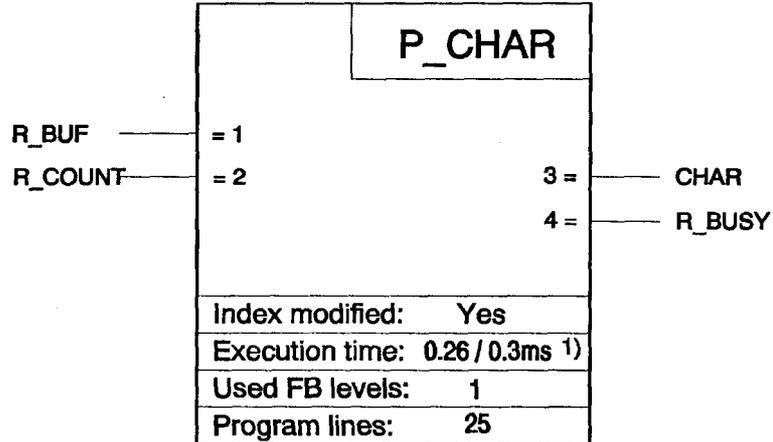
Bevor ein Register, Timer oder Zähler via S-Bus an einem Terminal D110 (Slave-Station) angezeigt werden kann, muss dessen Inhalt vom Binärformat in das ASCII-Format gewandelt werden. Ein Register kann eine Integerzahl mit max. 10 Dezimalstellen plus Vorzeichen enthalten. Um diese Integerzahl im ASCII-Format zu speichern, werden also 3 Register benötigt. Durch den Aufruf des FB's wird die Integerzahl (BIN_VAL) in das ASCII-Format gewandelt und in den 3 Ausgaberegistern (ASCII_VAL) gespeichert. Nach Abarbeitung des FB kann die Zahl angezeigt werden, indem die 3 Ausgaberegister mit einem STXM-Befehl zum D110 gesendet werden. Für die Anzeige wird nur das Normalformat unterstützt. Die nachfolgende Tabelle verdeutlicht das Funktionsprinzip.

BIN_VAL Integerzahl	ASCII_VAL			Gewandelte Integerzahl als ASCII-Wert	Mit STXM-Befehl am PCD7.D110 angezeigter Wert
	ASCII_VAL+0	ASCII_VAL+1	ASCII_VAL+2		
123456	Byte	3 2 1 0	3 2 1 0	3 2 1 0	123456
	Dez	0 0 0 0	0 49 50 0	51 52 53 54	
	ASCII	N N N N	N 1 2 N	3 4 5 6	
985	Byte	3 2 1 0	3 2 1 0	3 2 1 0	985
	Dez	0 0 0 0	0 0 0 0	0 57 56 53	
	ASCII	N N N N	N N N N	N 9 8 5	
1275944506	Byte	3 2 1 0	3 2 1 0	3 2 1 0	1275944506
	Dez	0 0 49 50	55 53 57 52	52 53 48 54	
	ASCII	N N 1 2	7 5 9 4	4 5 0 6	
-16749	Byte	3 2 1 0	3 2 1 0	3 2 1 0	-16749
	Dez	45 0 0 0	0 49 0 0	54 55 52 57	
	ASCII	- N N N	N 1 N N	6 7 4 9	
0	Byte	3 2 1 0	3 2 1 0	3 2 1 0	0
	Dez	0 0 0 0	0 0 0 0	0 0 0 48	
	ASCII	N N N N	N N N N	N N N 0	

N entspricht einem ASCII NUL, welches beim D110 keine Bedeutung hat und deshalb ignoriert wird.

Funktionsblock:

P_CHAR (Process Character)
Empfangener Tastaturpuffer verarbeiten



- 1) 0.26ms : bei leerem Tastaturpuffer
0.3ms : ein empfangener Charakter wurde verarbeitet

Funktionsbeschreibung

Mit diesem FB wird Charakter um Charakter aus den Empfangsregistern des Tastaturpuffers verarbeitet. Der Inhalt des Tasturpuffers muss dazu vorgängig mit einem SRXM-Befehl vom D110 gelesen und in die Empfangs-register gespeichert werden.

Liste der Ein- und Ausgänge

Symbol	Beschreibung	Para- meter	Daten			Adresse
			Typ	Format	Wert	
R_BUF	Base address receive buffer reg. Basisadresse von 1, 2 oder 3 Empfangsregistern des Tastaturpuffers	yes	R	ASCII	Siehe Handbuch PCD7.D1.. Tastatur-Codes	0..4095
R_COUNT	Number of receive register Anzahl Empfangsregister	yes	K	Integer	1, 2 oder 3	-
CHAR	Processed ASCII character Verarbeiteter ASCII-Charakter	yes	R	ASCII	Siehe Handbuch PCD7.D1.. Tastatur-Codes	0..4095
R_BUSY	Receive Busy Flag Empfangsbusy-Flag	yes	F/O	Binär	0/1	0..8191

Detaillierte Beschreibung der Ein/Ausgänge

Empfangsregister **R_BUF**

Der Tastaturpuffer des D110 verfügt über eine Kapazität von 12 Byte, was 12 ASCII-Charaktern entspricht. Mit einem SRXM-Befehl kann der gesamte Puffer (=3 Register), nur 8 Charakter (=2 Register) oder auch nur 4 Charakter (=1 Register) gelesen und in 1, 2 oder 3 Empfangsregister gespeichert werden. Der ASCII-Code einer gedrückten Taste belegt ein Byte im Empfangsregister.

In den Empfangsregistern kommt folgendes Format zur Anwendung:

Register	MSByte			LSByte
	3	2	1	0
R_BUF+0	1. Charakter	2. Charakter	3. Charakter	4. Charakter
R_BUF+1	5. Charakter	6. Charakter
R_BUF+2	letzter Char.	ASCII-NUL	ASCII-NUL

Charakter = ASCII-Code einer gedrückten Taste

Anzahl Empfangsregister **R_COUNT**

R_COUNT entspricht der Anzahl Register, die mit dem Befehl SRXM vom Tastaturpuffer des D110 gelesen wurden. Mit diesem Parameter wird dem FB mitgeteilt, wieviele Empfangsregister ausgewertet werden sollen.

Verarbeiteter ASCII-Charakter **R_CHAR**

Bei jedem Durchlauf des FB's wird jeweils ein ASCII-Charakter aus den Empfangsregistern ausgewertet und im LSByte dieses Registers gespeichert.

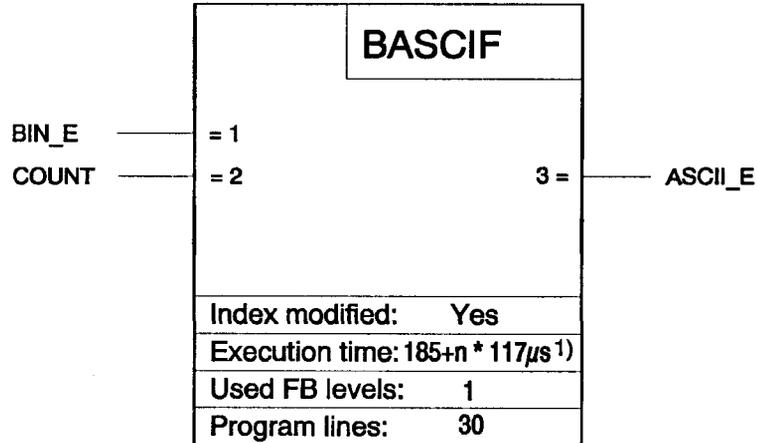
Empfangsbusy-Flag **R_BUSY**

Das Flag wird bei jedem FB-Aufruf "high" gesetzt. Es wird zurückgesetzt, sobald alle Empfangsregister ausgewertet wurden, oder wenn der letzte verarbeitete Charakter einem ASCII-NUL entspricht. Ein ASCII-NUL zeigt an, dass der Tastaturpuffer leer ist und nicht weiter verarbeitet werden muss.

Aufruf des Funktionsblockes

CFB	P_CHAR	; Verarbeite empfangene Charakter
	R_BUF	; Empfangsregister
	R_COUNT	; Anzahl Empfangsregister
	CHAR	; Ausgewerteter ASCII-Charakter
	R_BUSY	; Empfangsbusy-Flag

Funktionsblock: **BASCIF** (Binary-ASCII Conversion F/I/O)
 Binär → ASCII -Umwandlung für die Ausgabe
 von F/I/O-Status



1) n = Anzahl Elemente

Funktionsbeschreibung

Mit dem FB wird der Status von 1 bis 4 Elementen (F/I/O) vom Binärformat in das ASCII-Format gewandelt und in einem Register gespeichert. Mit einem STXM-Befehl kann anschliessend das Register zum D110 gesendet werden, um den Status der Elemente anzuzeigen.

Liste der Ein- und Ausgänge

Symbol	Beschreibung	Para- meter	Daten			Adresse
			Typ	Format	Wert	
BIN_E	Base address binary elements Basisadresse der binären Elemente	yes	I/O/F	Binär	0/1	0..8191
COUNT	Number of elements Anzahl Elemente	yes	K	Integer	1..4	-
ASCII_E	ASCII Elements Register mit ASCII-Werten	yes	R	ASCII	48/49 pro Byte	0..4095

Aufruf des Funktionsblockes

CFB BASCIF ; Binär-ASCII Umwandlung für I/O/F
 BIN_E ; Basisadresse der binären Elemente
 COUNT ; Anzahl Elemente
 ASCII_E ; Register mit ASCII-Werten

Beispiel

Der Status der Eingänge 0 bis 3 soll an Terminal 5 angezeigt werden.

```

...
CFB          BASCIF
              I 0          ; Eingang 0
              K 4          ; 4 Elemente
              R 10         ; ASCII-Werte in R 10
LD           R 0          ; Register für Stationsnummer
              5            ; Station 5

```

...
TBSY-Flag abfragen

```

...
STXM        1            ; Schnittstelle 1
              1            ; 1 Register
              R 10         ; ASCII-Werte
              R 10         ; Standard-Zielregister im D110

```

...

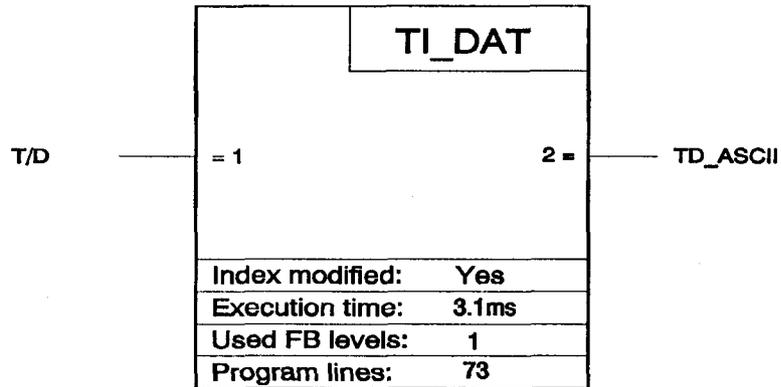
Nach der Ausführung dieser Befehle wird Status der 4 Eingänge an der aktuellen Cursorposition wie folgt angezeigt:

Eingang	0	1	2	3	
	0	0	0	0	(alle Eingänge low)
	1	1	1	1	(alle Eingänge high)

Funktionsblock:

TI_DAT (Time/Date binary-ASCII conversion)

Binär → ASCII -Umwandlung für die Ausgabe von Zeit oder Datum



Funktionsbeschreibung

Der FB liest die Zeit oder das Datum der PCD-internen Uhr und wandelt die Werte vom Binär- in das ASCII-Format. Nach der Ausführung des FB's sind die gewandelten ASCII-Werte des Datums oder der Zeit in drei aufeinander folgenden Registern gespeichert. Mit dem STXM-Befehl können anschliessend die Register zum D110 gesendet werden, um das Datum oder die Zeit anzuzeigen.

Liste der Ein- und Ausgänge

Symbol	Beschreibung	Para- meter	Daten			Adresse
			Typ	Format	Wert	
T/D	Time/Date flag Zeit/Datum Flag	yes	I/F	Binär	1/0	0..8191
TD_ASCII	Time or Date in ASCII Zeit oder Datum in ASCII	yes	R	ASCII	siehe detaillierte Beschreibung	0..4093

Detaillierte Beschreibung der Ein/Ausgänge

Zeit/Datum-Flag T/D

Definiert ob die Zeit oder das Datum gewandelt werden soll.

T/D = 0 → Datum wird gewandelt

T/D = 1 → Zeit wird gewandelt

Zeit/Datum in ASCII TD_ASCII

Basisadresse von drei aufeinander folgenden Registern mit den ASCII-Werten für die Zeit oder das Datum.

Aufruf des Funktionsblockes

CFB TI_DAT ; Zeit/Datum-Umwandlung ASCII → Binär
 T/D ; Zeit/Datum-Flag
 TD_ASCII ; Zeit/Datum ASCII

Beispiel 1

An Terminal 12 soll die Uhrzeit angezeigt werden.

...

ACC H
 SET F 100 ; Zeit umwandeln
 CFB TI_DAT ; Zeit/Datum-Umwandlung
 F 100 ; Zeit/Datum-Flag
 R 120 ; Basisreg. Zeit in ASCII
 LD R 0 ; Register für Stationsnummer
 12 ; Station 12
 TBSY-Flag abfragen
 STXM 1 ; Schnittstelle 1
 3 ; 3 Register
 R 120 ; Basisreg. Zeit in ASCII
 R 10 ; Standard-Zielregister im D110

...

Angenommen diese Befehlsfolge wird um 14.55 Uhr abgearbeitet, so wird die Uhrzeit an der aktuellen Cursorposition wie folgt angezeigt:

14:55:00

Beispiel 2

An Terminal 7 soll das Datum angezeigt werden.

...

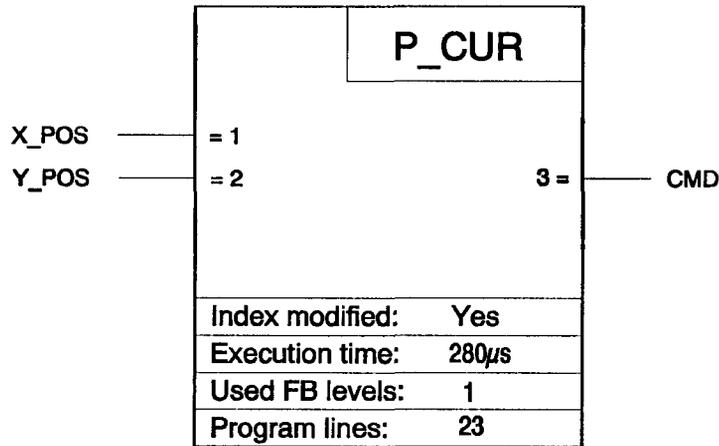
ACC H
 RES F 100 ; Datum umwandeln
 CFB TI_DAT ; Zeit/Datum-Umwandlung
 F 100 ; Zeit/Datum-Flag
 R 120 ; Basisreg. Datum in ASCII
 LD R 0 ; Register für Stationsnummer
 7 ; Station 7
 TBSY-Flag abfragen
 STXM 1 ; Schnittstelle 1
 3 ; 3 Register
 R 120 ; Basisreg. Datum in ASCII
 R 10 ; Standard-Zielregister im D110

...

Angenommen diese Befehlsfolge wird am 15. Dezember 1992 abgearbeitet, so wird das Datum an der aktuellen Cursorposition wie folgt angezeigt:

92/12/15

Funktionsblock: **P_CUR** (Position the Cursor)
Positionierung des Cursors



Funktionsbeschreibung

Der FB erzeugt die Steuersequenz um den Cursor des D110 zu positionieren. Die Steuersequenz wird im Register CMD gespeichert. Die Positionierung wird ausgelöst, indem das Register mit einem STXM-Befehl zum D110 gesendet wird.

Liste der Ein- und Ausgänge

Symbol	Beschreibung	Parameter	Daten			Adresse
			Typ	Format	Wert	
X_POS	Cursor X-Position	yes	K	Integer	32..51	-
Y_POS	Cursor Y-Position	yes	K	Integer	32..35	-
CMD	Command code Steuersequenz Cursor- Positionierung	yes	R	Steuer- zeichen	Siehe Handbuch PCD7.D1..	0..4095

Aufruf des Funktionsblockes

```
CFB      P_CUR      ; Cursor-Positionierung
         X_POS      ; X-Position
         Y_POS      ; Y-Position
         CMD        ; Register für Steuersequenz
```

Beispiel

Der Cursor von Terminal 2 soll an folgende Position

X: 45

Y: 34

bewegt werden.

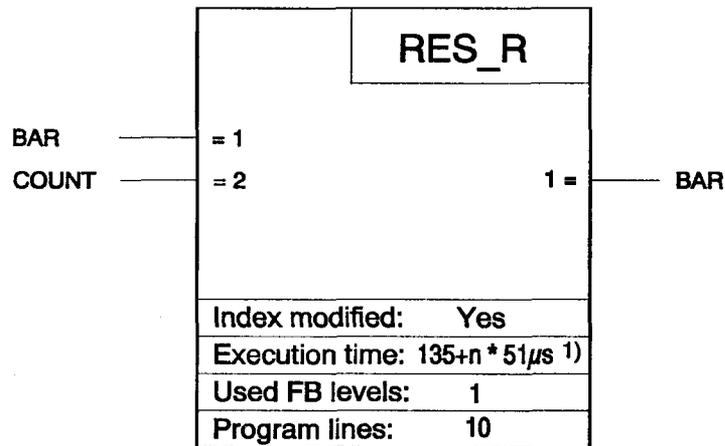
```
...
CFB      P_CUR      ; Cursor-Positionierung
          K 45      ; X-Position
          K 34      ; Y-Position
          R 50      ; Steuersequenz
LD       R 0        ; Register für Stationsnummer
          2         ; Station 2

...
TBSY-Flag abfragen

...
STXM     1          ; Schnittstelle 1
          1          ; 1 Register
          R 50      ; Steuersequenz
          R 10      ; Standard-Zielregister im D110

...
```

Funktionsblock: RES_R (Reset registers)
 Register löschen



1) n = Anzahl Register

Funktionsbeschreibung

Mit dem FB wird eine vorgebene Anzahl Register gelöscht. Löschen heisst, die Register werden mit dem Wert Null geladen.

Liste der Ein- und Ausgänge

Symbol	Beschreibung	Parameter	Daten			Adresse
			Typ	Format	Wert	
BAR	Basisadresse Register	yes	R	Integer	0	0..4095
COUNT	Anzahl Register	yes	K	Integer	1..4096	-

Aufruf des Funktionsblockes

```
CFB      RES_R      ; Register löschen
          BAR        ; Basisregister
          COUNT      ; Anzahl Register
```

Beispiel

Die Register 50 bis 69 sollen gelöscht werden.

```
CFB      RES_R      ; Register löschen
          R 50       ; Basisregister
          K 20       ; 20 Register
```

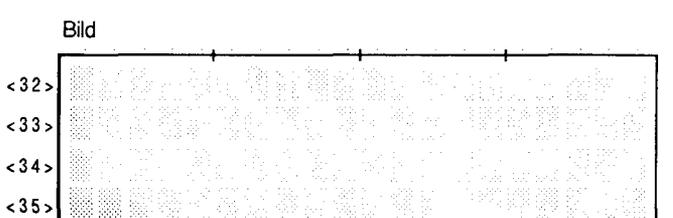
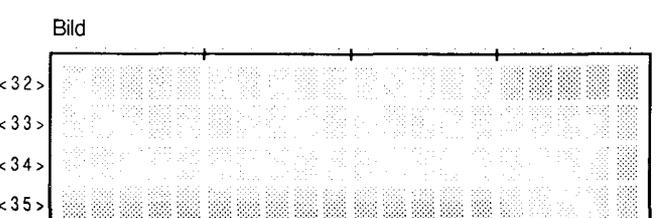
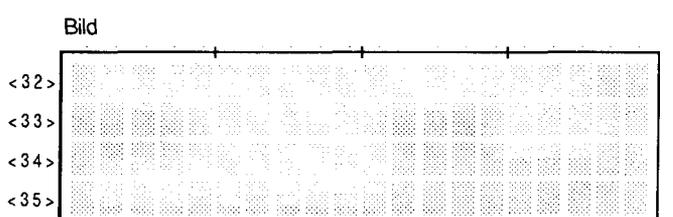
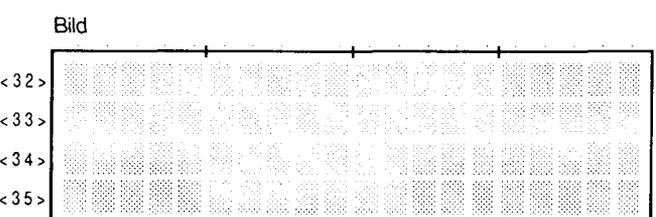
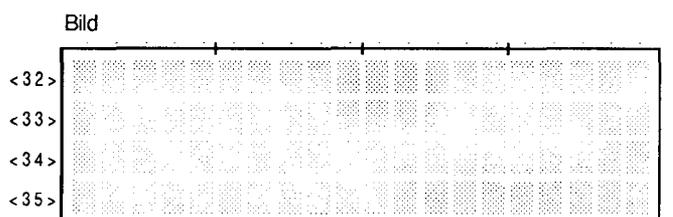
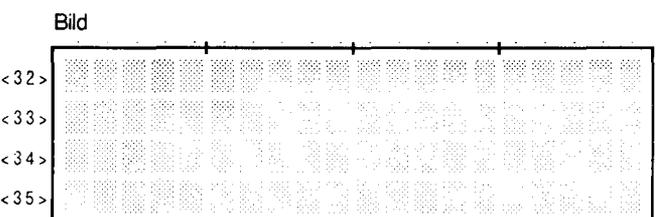
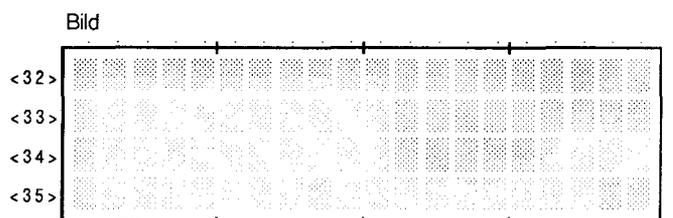
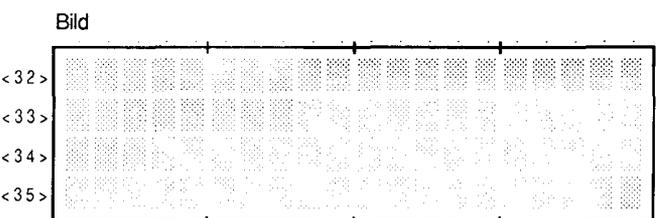
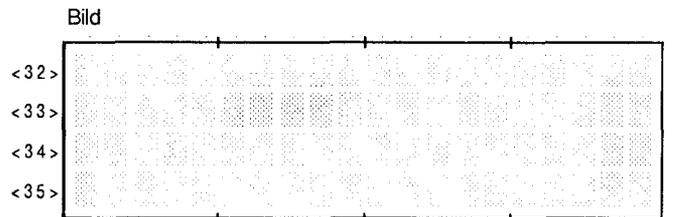
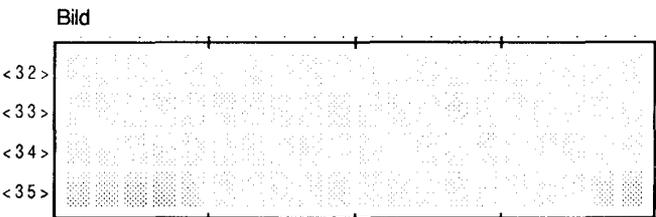
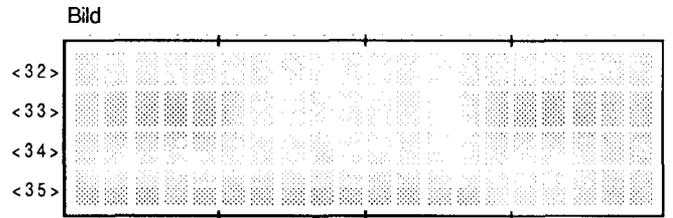
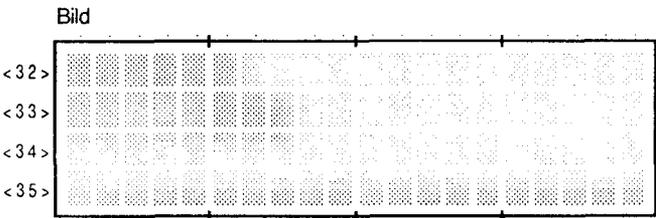
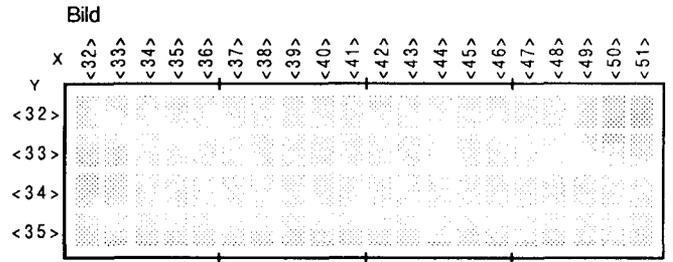
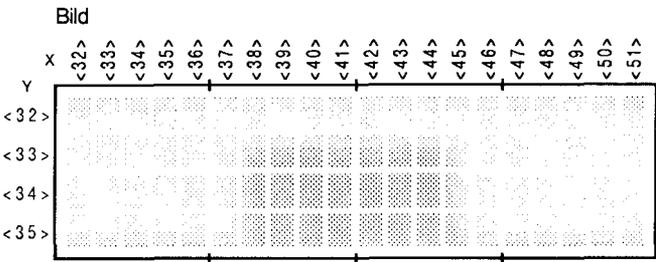
Intern verwendete und reservierte Elemente mit symbolischem Namen:

Von den FB's werden intern insgesamt 6 Arbeitsregister und 2 Arbeitsflag als Zwischenspeicher verwendet. Die FB's benutzen selber 2 weitere Sub-FB's. Alle Symboldefinitionen befinden sich in der FB-Datei und können vom Anwender angepasst werden.

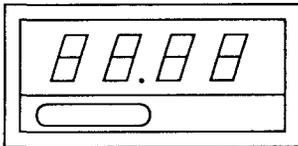
Symbol	Beschreibung	Daten		Adresse
		Typ	Format	
WORK_R	Base address of 6 work registers Basisadresse von 6 Arbeitsregistern	R	Integer	0..4090 (+5)
WORK_F	Base address of 2 work flags Basisadresse von 2 Arbeitsflag	F/O	Binär	0..8190 (+1)
SHIFT	FB shift register by 1 byte FB Register um 1 Byte schieben	FB	-	0..999
ABS	FB built absolute value FB Absolutewert bilden	FB	-	0..999

Bildschirm-Vorlagen

Pos.: <16> <X> <Y>



2. PCA2.D12 Display-Modul mit 4 Ziffern



Allgemeines

Das Modul PCA2.D12 ist ein Ferndisplay, welches über die Ausgänge der SAIA[®]PLC angesteuert wird. Es besitzt eine 4-stellige Anzeige sowie die Möglichkeit, einen Dezimalpunkt darzustellen. Das Display kann in einem grösseren Abstand zur SPS irgendwo eingebaut werden, zum Beispiel in einer Schaltschranktür oder einem Bedienfeld. Dadurch, dass die Datenübertragung über Ausgänge erfolgt, können von einer SPS aus mehrere Displays angesteuert werden.

Aufbau und Wirkungsweise

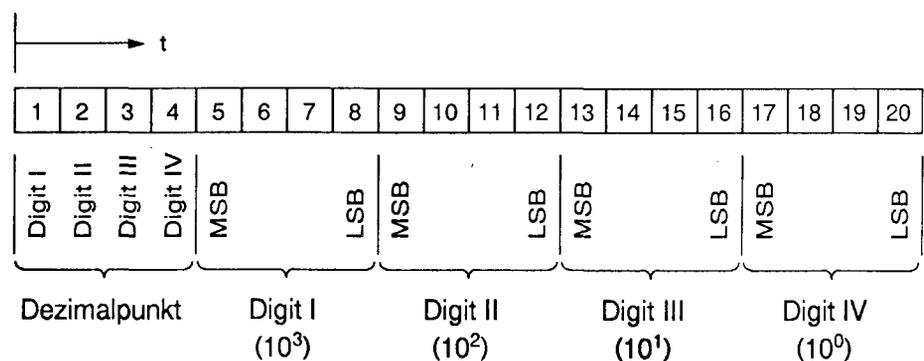
Das Modul ist im selben Gehäuse untergebracht wie der elektronische Summenzähler Typ CKG. Es besteht im wesentlichen aus den Teilen:

- Speisung 24 VDC
- 3 Eingänge für 24 VDC
- Decoder/Treiber
- 4-stelliges 7-Segment-LED-Display mit Dezimalpunkt

Die 3 Ausgänge der SPS resp. die 3 Eingänge des Displays tragen die Bezeichnung Enable, Data und Clock. Mit dem **Enable-Signal** wird das Display angesprochen, d. h. Enable = "L" → Display kann Daten empfangen, Enable = "H" → Display ist inaktiv (kann keine neuen Daten empfangen). Über die Leitung "DATA" werden die Daten im BCD-Format seriell, d.h. Bit für Bit aus der SPS an das Display gesendet. Jedes Bit wird mit der negativen Flanke des Signals "Clock" vom Display übernommen.

Für eine vollständige Anzeige (4 Digit, mit oder ohne Dezimalpunkt) müssen immer 20 Clocksignale erzeugt und 20 Datenbits gesendet werden (4 BCD-Werte + 4 Bit für den Dezimalpunkt).

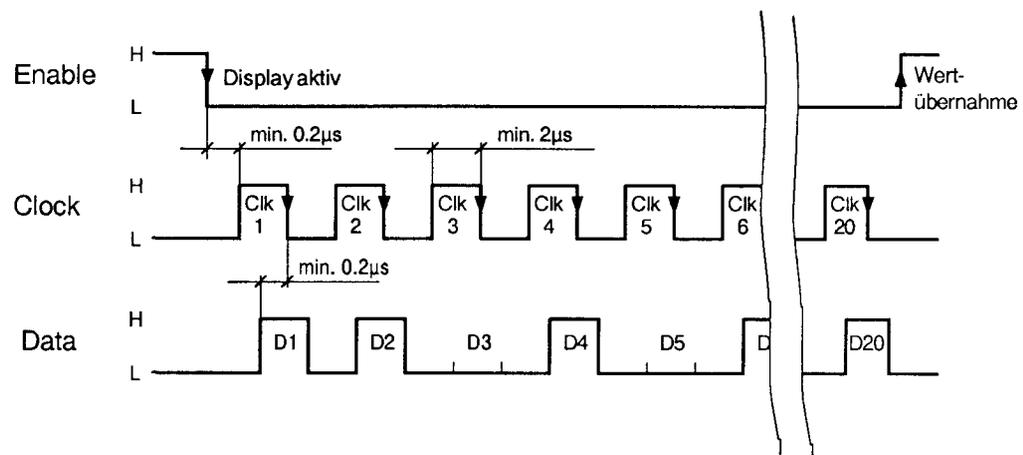
Für diese 20 Datenbits muss folgende Reihenfolge eingehalten werden:



Es können folgende 16 Zeichen pro Segment dargestellt werden:

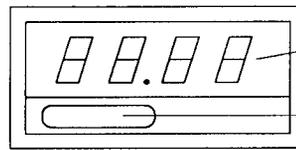
Zeichen	Code	Zeichen	Code
0	0000	R	1010
1	0001	l	1011
2	0010	ll	1100
3	0011	U	1101
4	0100	-	1110
5	0101	"blank"	1111
6	0110		
7	0111		
8	1000		
9	1001		

Den Zusammenhang zwischen Enable, Clock und Data verdeutlicht das Zeitdiagramm:



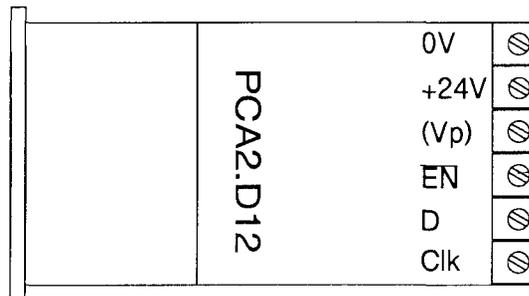
Für die Erzeugung des "Clocks" sowie für die Übergabe der Daten ist vom Anwender ein kurzes Programm zu erstellen. In diesem Programm sind genau die Funktionen zu realisieren, welche im obigen Zeitdiagramm dargestellt sind (siehe PB 10 im folgenden Beispiel).

Präsentation und Klemmenanordnung



4-stelliges Display

Knopf hat beim PCA2.D12
keine Funktion

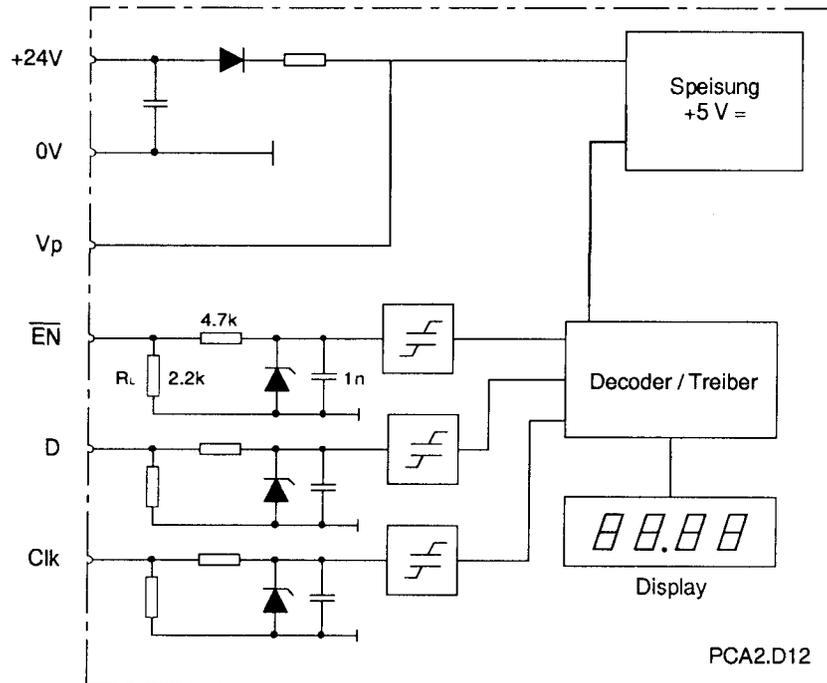


Schraubklemmen M3

Technische Daten

- | | |
|--|--|
| - Anzeige | 4 Digit mit Dezimalpunkt,
7 Segment LED |
| - Ziffernhöhe | 10mm |
| - Speisespannung | 24 VDC $\pm 20\%$,
zweiweggleichgerichtet genügt |
| - Stromaufnahme
ab Speisung 24 V | 60 mA |
| - Eingangsspannung
für EN, D, CLK | 24 VDC, geglättet |
| - Eingangsstrom bei
24 VDC | 10 mA |
| - Definition der
Eingangsspannungen | "H": 19V... 32V
"L": 0V... 4V |
| - Eingangsverzögerung | kleiner 1ms |
| - Verwendbare
SAIA [®] SPS-Aus-
gangsmodule | PCD2.A400
PCD4.A400, B900
PCD6.A400 |
| - Ansteuerung | seriell über 3 SPS-Ausgänge |

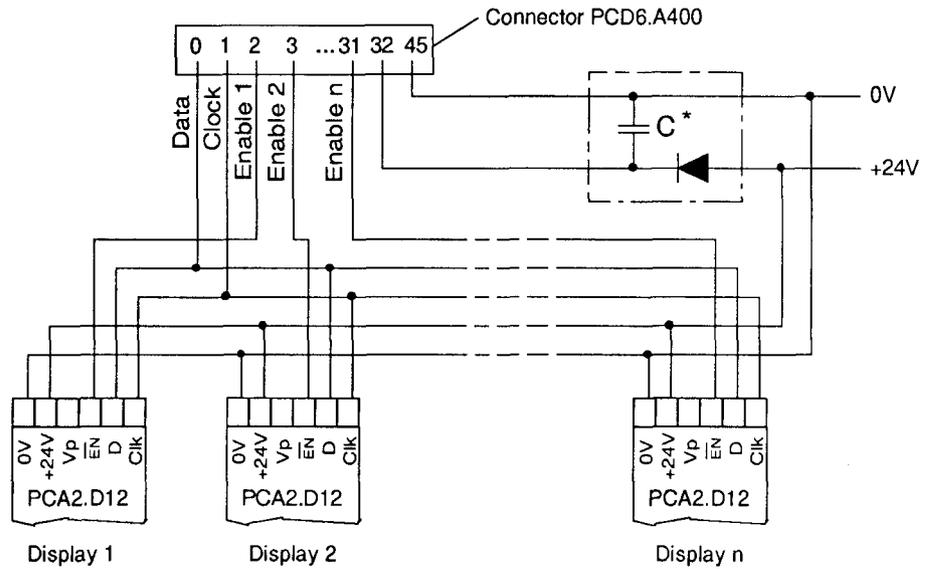
Eingangsschaltung und Blockschema



Anschluss von mehreren Displays

Da das Modul PCA2.D12 über ein Enable-Signal verfügt, d.h. aktiv oder inaktiv geschaltet werden kann, können für **mehrere Displays dieselben Signale Clock und Data** verwendet werden. Diese werden parallel an jedes Display gelegt. Das Enable-Signal entscheidet, welches Display angesprochen wird. Dies bedeutet, dass für **jedes Display ein** Enable-Signal notwendig ist (1 Ausgang pro Display), dass jedoch für **beliebig viele Displays nur ein Data-** und nur **ein Clock-Ausgang** vorgesehen werden muss.

Anschlussbeispiel: PCD6.A400



- *) Bei pulsierender Gleichspannung müssen folgende Punkte beachtet werden:
- Glättung durch Kondensator C von 100 μ F, 40V (ausreichend für 5 Displays)
 - Keine grossen Lasten an der durch C geglätteten Spannung anschliessen

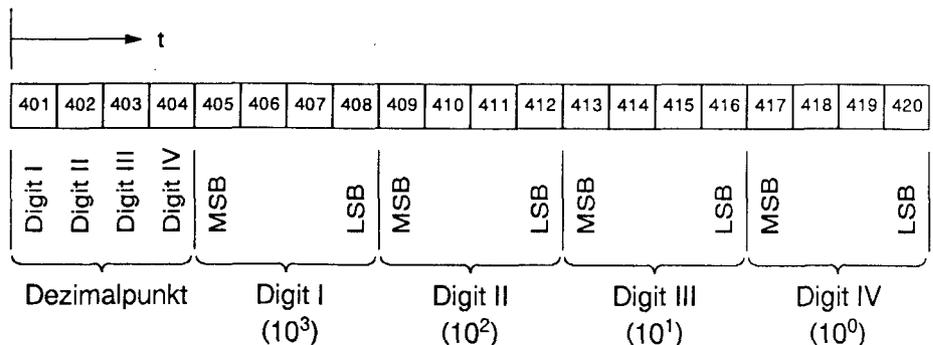
Programmbeispiel PCA2.D12

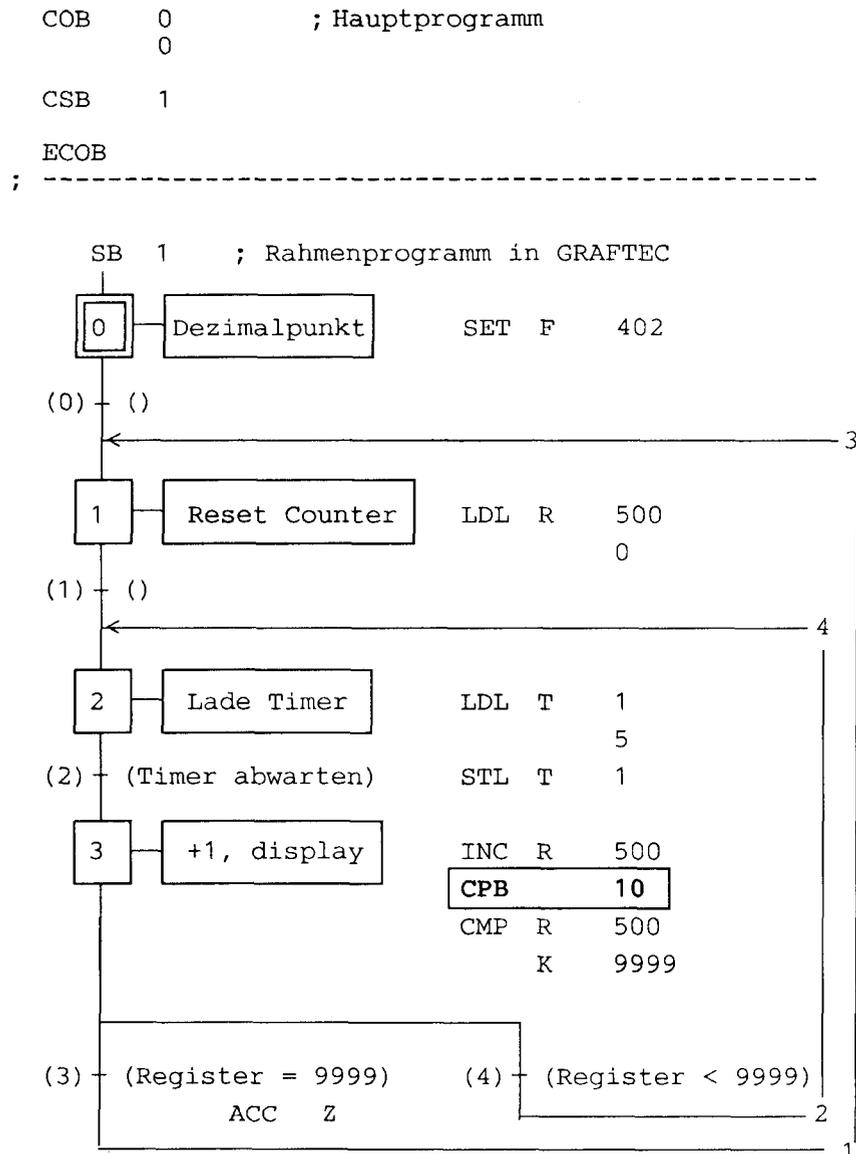
Aufgabe: Es soll ein Register R500 jede halbe Sekunde bis zum Wert 9999 inkrementiert und anschliessend auf Null zurückgestellt werden. Der Inhalt dieses Registers ist auf dem Display-Modul PCA2.D12, mit einem Dezimalpunkt an der 2.Stelle versehen, anzuzeigen.

Für Clock, Data und Enable sind folgende Ausgänge zu belegen:

- CLOCK: Ausgang O45
- DATA: Ausgang O46
- ENABLE: Ausgang O47

Verwendete Merker: 401 .. 420



Programm:

PB 10 ; Ansteueroutine für PCA2.D12

```

DIGOR 4
      R 500
      F 405

RES O 47 ; ENABLE
SEI K 0

LOOP: SET 0 45 ; CLOCK
      STHX F 401
      OUT 0 46 ; DATA
      ACC H
      RES O 45 ; CLOCK
      INI K 19
      JR H LOOP
      ACC H
      SET 0 47 ; ENABLE

EPB

```

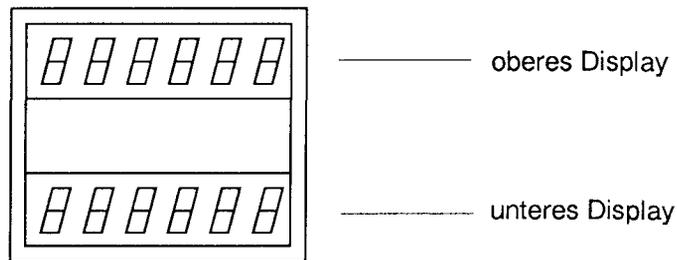
Masse und Montage

Das Display-Modul kann irgendwo, z.B. in einer Schaltschranktür oder einem Bedienpult eingebaut werden, wobei die Einbaulage beliebig ist.

Für die Befestigung stehen 3 Varianten zur Verfügung:

<p>Abmessungen</p>		
<p>Öffnung für den Einbau für beide Befestigungsarten gültig</p>		
<p>Befestigung mit Klemmfeder</p>		<p>Standard Version</p>
<p>Befestigung mit Frontrahmen und 2 Senkschrauben M3/90°</p> <p>mit hohem Frontrahmen</p> <p>mit breitem Frontrahmen</p>	<p>Montageart vertikal</p> <p>Montageart horizontal</p>	<p>Zubehör Bestell-Nr. 4'108'3671'0</p> <p>Bestell-Nr. 4'108'3672'0</p>

3. PCA2.D14 Display-Modul mit 2x6 Ziffern



Allgemeines

Das Modul PCA2.D14 ist ein Ferndisplay, das über 3 Ausgänge der SAIA°SPS angesteuert wird. Das Modul hat zwei 6-stellige rote LED-Anzeigen. Für mehr als zwei Anzeigen können mehrere PCA2.D14 hintereinander geschaltet werden.

Anwendung, Ansteuerung

Das Modul dient vor allem zur Anzeige von Zählerständen bei Verwendung der H-Module. Drei digitale Ausgänge genügen jedoch um beliebige Prozessdaten zur Anzeige zu bringen.

Wird das PCA2.D14 im Zusammenhang mit einer PCD eingesetzt, so wird die anzuzeigende Information am einfachsten mit einer Standard-Programmroutine ab einem Merkerfeld über 3 SAIA°SPS-Ausgänge seriell übermittelt.

Es können folgende 16 Zeichen pro Segment dargestellt werden:

Zeichen	Code	Zeichen	Code
0	0000	R	1010
1	0001	l	1011
2	0010	ll	1100
3	0011	U	1101
4	0100	-	1110
5	0101	"blank"	1111
6	0110		
7	0111		
8	1000		
9	1001		

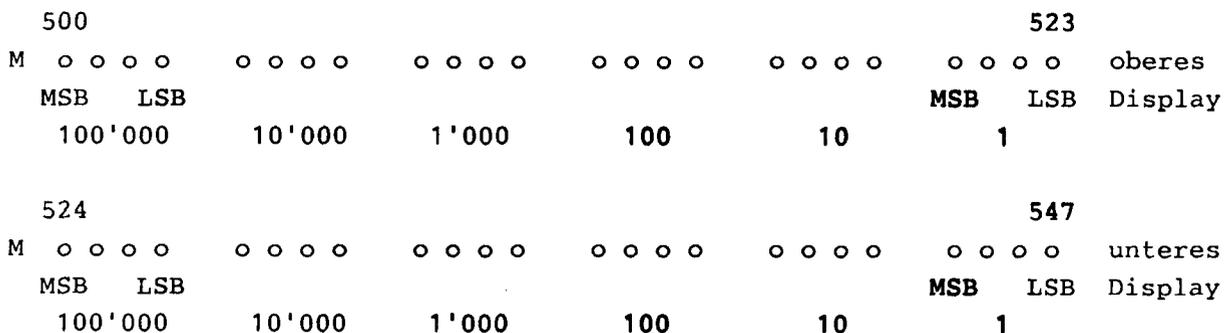
Aufbau, Wirkungsweise

Das Modul ist im gleichen Gehäuse untergebracht wie der elektronische Summenzähler CKG/AC.

Klemmen: Schraubklemmen kombiniert mit Anschlusszungen (2,8x0,8mm) für Flachsteckhülsen oder zum Löten.

SPS-Ausgang	Clock	---->	Clk	PCA2.D14
SPS-Ausgang	Data-In	---->	D-IN	
SPS-Ausgang	Enable	---->	EN	
Übertrag	Data-Out	<----	D-OUT	
Speisung	+24V	---->	+24V	
Speisung	0V	---->	0V	

Die Daten für eine Anzeige von 2x6 Stellen werden am einfachsten in einem zusammenhängenden Merkerfeld, z.B. M500.. 547 wie folgt im BCD-Format dargestellt. Befinden sich die Werte in Registern, so müssen diese vorgängig in die Merkerfelder übertragen werden.



Technische Daten

- Anzeige 2x6 Digit, 7 Segment LED
- Ziffernhöhe 10mm
- Speisespannung 24 VDC $\pm 20\%$,
zweiweggleichgerichtet genügt
- Stromaufnahme
ab Speisung 24 V 100 mA
- Eingangsspannung
für EN, D, CLK 24 VDC, geglättet
- Eingangsstrom bei
24 VDC 10 mA

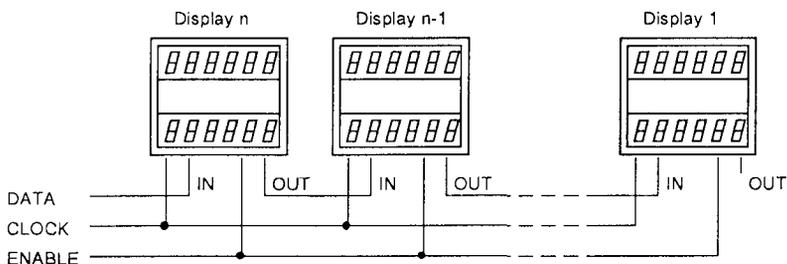
- | | |
|---------------------------------------|--|
| - Definition der Eingangsspannungen | "H": 19V... 32V
"L": 0V... 4V |
| - Eingangsverzögerung | kleiner 1ms |
| - Verwendbare SAIA°SPS-Ausgangsmodule | PCD2.A400
PCD4.A400, B900
PCD6.A400 |
| - Ansteuerung | seriell über 3 SPS-Ausgänge
unabhängig von der Anzahl D14 |

Anschluss von mehr als 1 D14-Modul

Mehrere D14-Module können in Serie geschaltet werden, wobei folgende Vor- und Nachteile beachtet werden müssen:

- + Nur 3 SPS-Ausgänge für alle D14-Module
- + Nur 1 Software-Routine in welcher lediglich 1 Stelle geändert werden muss
- n mal längere Abarbeitungszeit der Routine !

Das Hintereinanderschalten mehrerer PCA2.D14 zeigt die folgende Skizze. Jedes D14 zeigt seine individuellen Daten an.



Das Merkerfeld, ab welchem die anzuzeigende Information geholt wird, muss entsprechend erweitert werden:

- für 1 PCA2.D14 = 1x48 Merker
- für 2 PCA2.D14 = 2x48 Merker (96 Merker)
- für 3 PCA2.D14 = 3x48 Merker (144 Merker) usw.

Die Routine im Anwenderprogramm bleibt jedoch gleich, einzig der Befehl "INI" muss angepasst werden:

- für 1 PCA2.D14 = INI 47
- für 2 PCA2.D14 = INI 95
- für 3 PCA2.D14 = INI 143 usw.
- für n PCA2.D14 = INI $n \cdot 48 - 1$

Programmbeispiel PCA2.D14

Aufgabe: Es sind an einem Displaymodul PCA2.D14 im oberen Display die Uhrzeit und im unteren Display das Datum anzuzeigen. Die Daten werden der Hardwareuhr der PCD entnommen.

Verwendete Elemente:

CLOCK	Ausgang O 45
DATA	Ausgang O 46
ENABLE	Ausgang O 47
FLAGS	F 500-547
COUNTER	C 999

```

COB      0      ; Hauptprogramm
          0
RTIME   R   200 ; Uhrzeit auf R 200, Datum auf R 201
CPB     20     ; Ansteuerroutine für PCA2.D14
ECOB
; -----
PB      20     ; Ansteuerroutine für PCA2.D14
; -----
DIGOR   6
          R   200 ; Wert für oberen Display (6Digits)
          F   500 ;   auf Flags 500-523
DIGOR   6
          R   201 ; Wert für unteren Display (6Digits)
          F   524 ;   auf Flags 524-547
ACC     H
RES     O   47  ; ENABLE
SEI     K   0
L1:    ACC     H
        SET     O   46  ; DATA
        LDL     C   999
          4
L2:    SET     O   45  ; CLOCK
        RES     O   45  ; CLOCK
        DEC     C   999
        STH     C   999
        JR      H   L2
        ACC     H
        LDL     C   999
          16
L3:    STHX    F   500
        OUT     O   46  ; DATA
        ACC     H
        SET     O   45  ; CLOCK
        RES     O   45  ; CLOCK
        INI     K   47
        JR      L   L4
        DEC     C   999
        STH     C   999
        JR      H   L3
        JR      L   L1
L4:    ACC     H
        SET     O   47  ; ENABLE
EPB

```

Masse und Montage

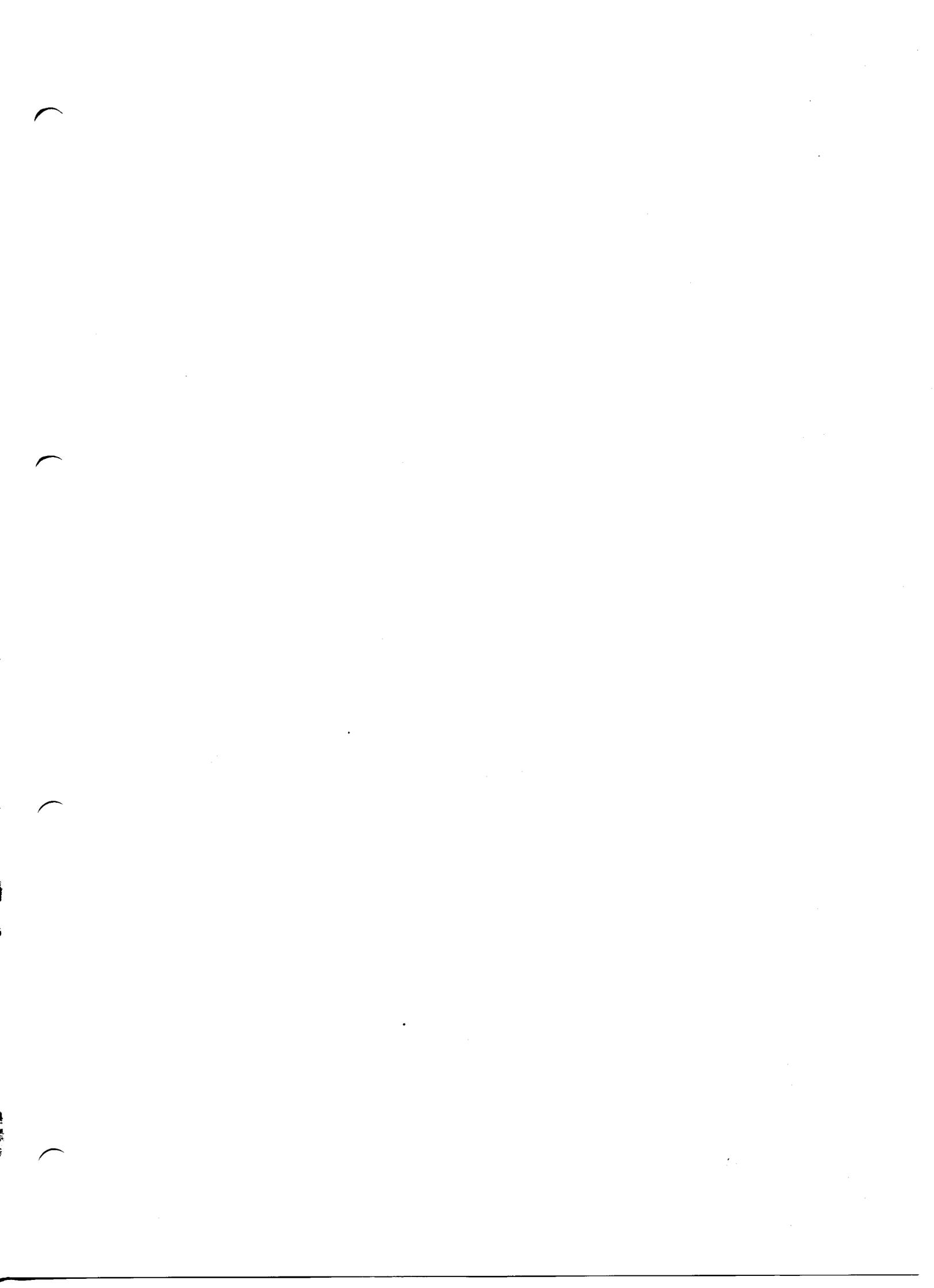
Die Einbaulage ist beliebig.

Für die Befestigung stehen 2 Varianten zur Verfügung:

- mittels Klemmfeder
- mittels Frontrahmen und Schrauben

(Die Klemmfeder und der Frontrahmen werden mit jedem PCA2.D14 mitgeliefert).

<p>Abmessungen</p>	
<p>Öffnung für den Einbau (für beide Befestigungsarten gültig)</p>	
<p>Befestigung mit Klemmfeder</p>	
<p>Befestigung mit Frontrahmen und 2 Senkschrauben M3/90° (für Montageart vertikal oder horizontal)</p>	



SAIA AG

Industrie-Elektronik und Komponenten
CH-3280 Murten/Schweiz

Zentrale	Telefon	037 727 111
	Telefax	037 714 443
	Telex	942 127
Verkauf Schweiz	Telefon	037 727 727
	Telefax	037 711 983

Vertretungen

- Belgique** Landis & Gyr Belge SA, Dépt. Industrie
Avenue des Anciens Combattants 190, B-1140 Bruxelles
☎ 02 729 02 11, Fax 02 726 23 31
- Danmark** Kemp & Lauritzen A/S
Roskildevej 12, DK-2620 Albertslund
☎ 045 43 62 48 22, Fax 045 43 62 10 46
- Deutschland** SAIA GmbH
Daimlerstrasse 1 K, Postfach 10 21 33, D-63267 Dreieich
☎ 06103 8906-0, Fax 06103 8906 66
- España** Landis & Gyr BC SA
Batalla del Salado 25, Apartado 575, E-28080 Madrid
☎ 91 467 19 00, Fax 91 539 44 79
- France** SAIA-Burgess
10, Bd Louise Michel, F-92238 Gennevilliers cedex
☎ 1 4086 03 45, Tx 613 189, Fax 1 4791 40 13
- Great Britain** Burgess Ltd.
Dukes Way, Team Valley, Gateshead, Tyne & Wear NE 11 0UB
☎ 091 487 7171, Tx 53229, Fax 091 487 1610
- Italia** SAIA S.r.l.
Via Cadamosto 3, 20094 Corsico MI
☎ 02 48 69 21, Fax 02 48 60 06 92
- Nederland** Landis & Gyr BV, Industrie
Kampenringweg 45, Postbus 444, NL-2800 AK-Gouda
☎ 01820 65 683, Tx 20 657, Fax 01820 32 437
- Norge** Malthe Winje AS
Haukelivn. 22, Postboks 531, N-1411 Kolbotn
☎ 66 99 25 00, Fax 66 99 25 05
- Österreich** SAIA-Burgess Ges.m.b.H
Schallmooser Hauptstrasse 38, A-5020 Salzburg
☎ 0662 88 49 10, Fax 0662 88 49 10-11
- Portugal** Infocontrol Electronica e Automatismo LDA
Rua Alvares Botelho-Lote 154, Alfragide sul, P-2700 Amadora
☎ 1 47 10123, Fax 1 47 10676
- Suomi
Finnland** Landis & Gyr Suomi OY
SF-02430 Masala
☎ 90-29 731, Tx 121 039, Fax 90-29 755 31
- Sverige** Beving Compotech AB
St. Eriksgatan 113, Box 21 029, S-10031 Stockholm
☎ 08 31 47 80, Fax 08 34 31 42
- USA** After sales services; Maxmar Controls Inc.
99 Castleton Street, Pleasantville, New York 10570-3403
☎ 914 747 3540, Fax 914 747 3567
- Australia** Landis & Gyr Building Control a Division of L&G (AUS) Pty Ltd
15, Nyadale Drive, AU-Scoresby Victoria 3179
☎ 03 763 7211, Tx 32 224, Fax 03 763 2437
- Argentina** Murten S.r.l.
Av. del Libertador 184, 4° «A», RA-1001 Buenos Aires
☎ 0541 312 0172, Fax 0541 312 0172