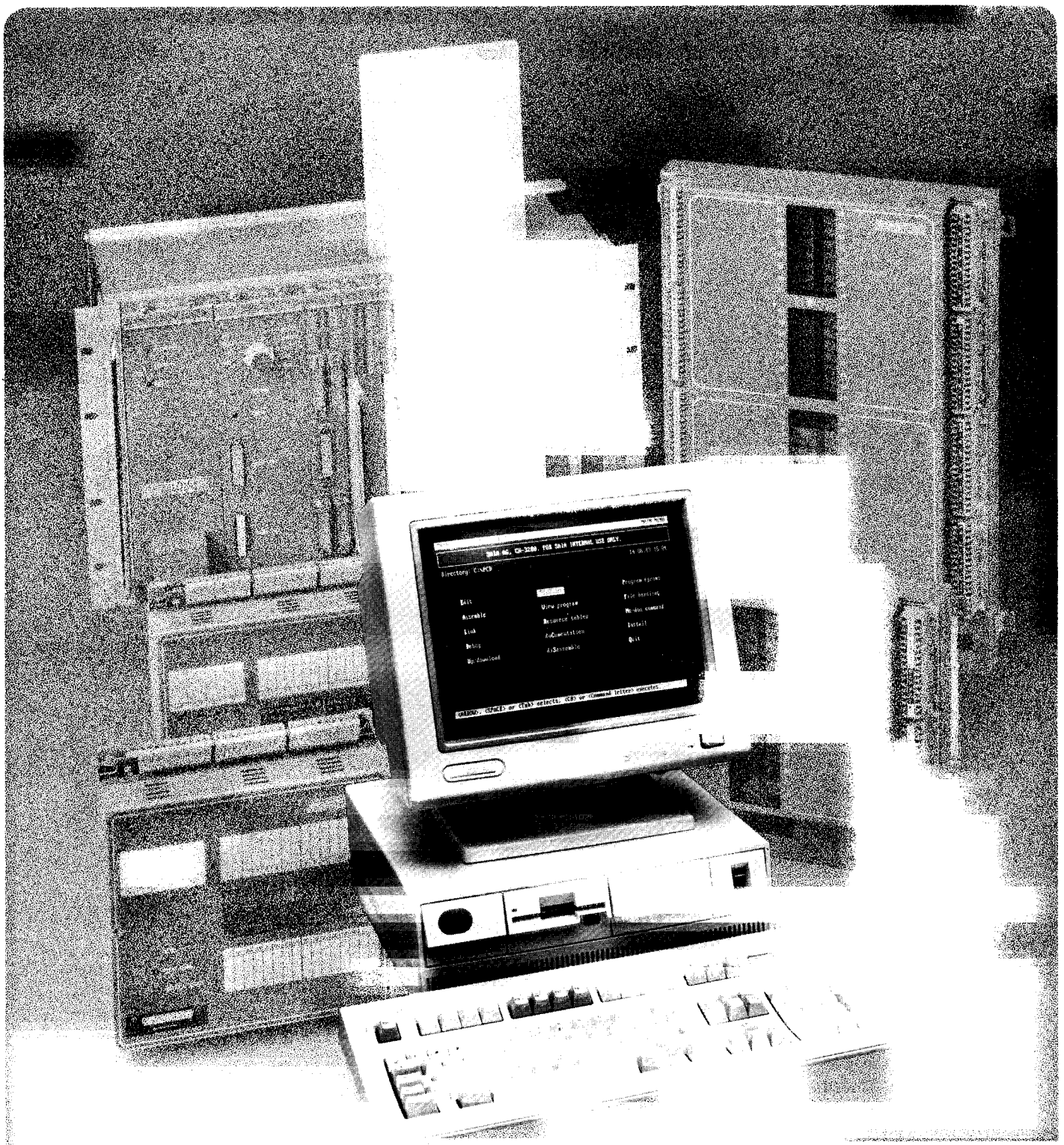
SAIA®PLC

Speicherprogrammierbare Steuerungen

Handbuch Software Stufen 2+3



Deutsche Ausgabe 26/722 D2

SOFTWARE STUFEN 2 + 3

TEIL G	UEBERSICHTEN
TEIL H	ERGAENZUNGSBEFEHLE STUFE 2
TEIL I	BEFEHLE DES WORTPROZESSORS STUFE 3
TEIL K	TEXTAUSGABE UND KOMMUNIKATION (STUFEN 2 + 3)
TEIL L	PROGRAMMIERBEISPIELE

Ausgabe 26/722 D2

Änderungen von technischen Daten vorbehalten

Verkaufspreis SFr. 60.--

INHALTSVERZEICHNIS

TEIL G UEBERSICHTEN

TEIL G UEBERSICHTEN

G 1 Kompatibilität in der PCA-Steuerungsfamilie

Die Anwendersoftware der PCA-Familie ist streng aufwärtskompatibel aufgebaut. Jede Baureihe einer höheren Stufe enthält alle Befehle der darunterliegenden Stufen. Dies hat für den Anwender den grossen Vorteil, dass "alte" Programme einer einfacheren CPU jederzeit auch auf einer leistungsfähigeren Steuerung lauffähig sind.

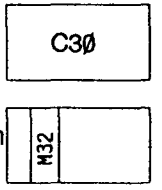
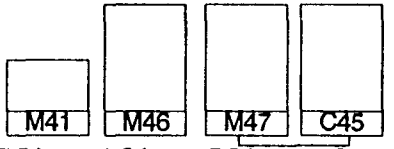
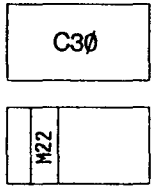
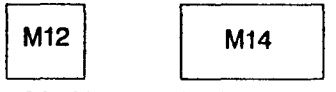
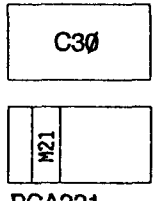
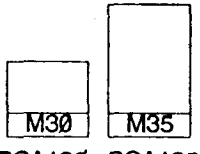
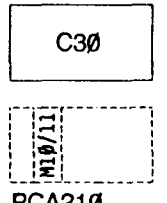
Die gleiche Kompatibilität gilt auch zwischen den verschiedenen Baureihen. Prozessoren der gleichen Software-Stufe verfügen über denselben Befehlssatz.

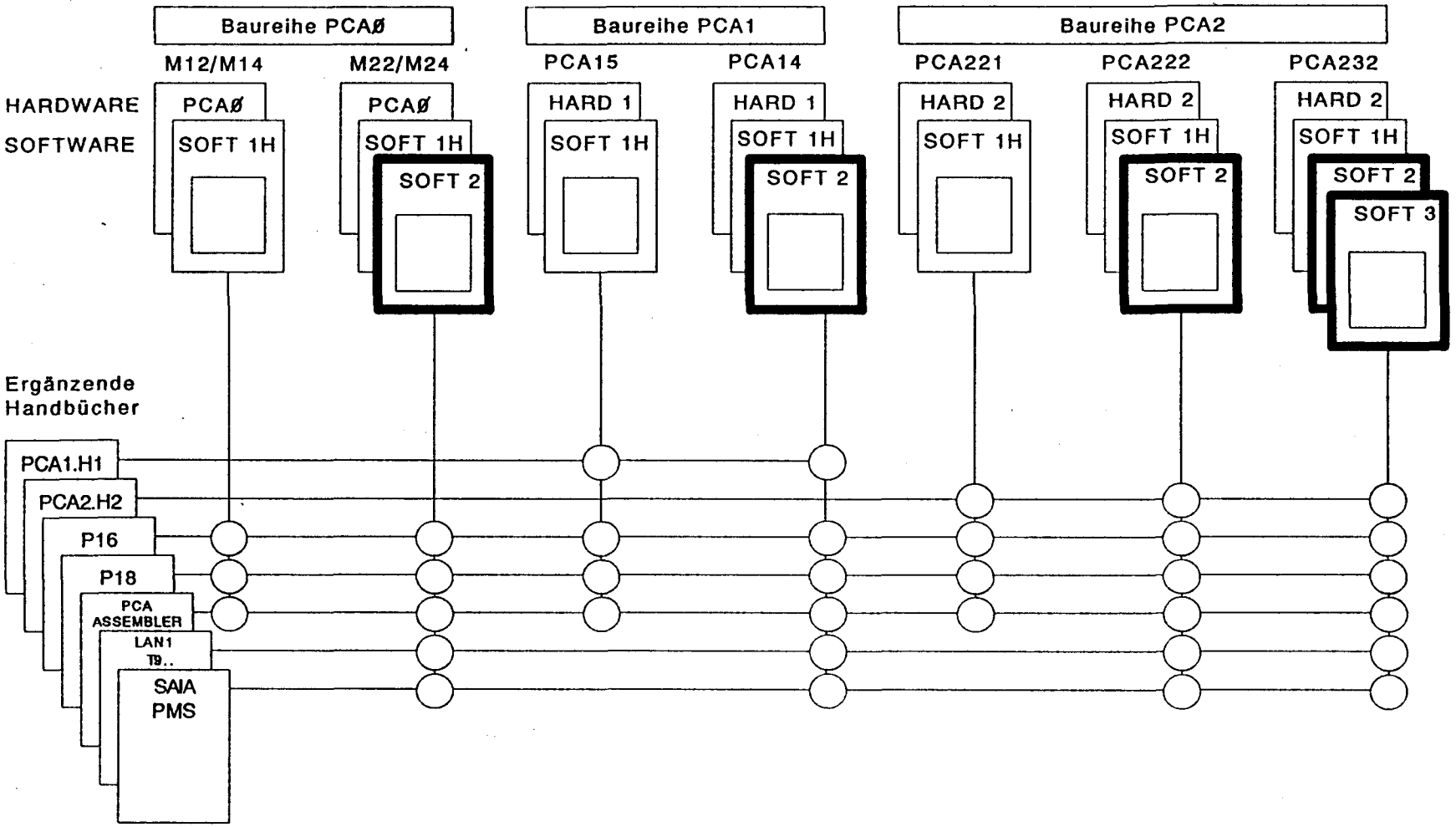
Unterschiede sind lediglich bei komplexen Modulen zu beachten, welche bezüglich Hardware verschieden aufgebaut sein können.

Bitte beachten Sie auch, dass zur Erweiterung der Leistungsfähigkeit im Laufe der Zeit neue Firmware-Versionen zu den Prozessoren geschaffen worden sind. Nähere Auskunft dazu erhalten Sie bei Ihrer Landesvertretung.

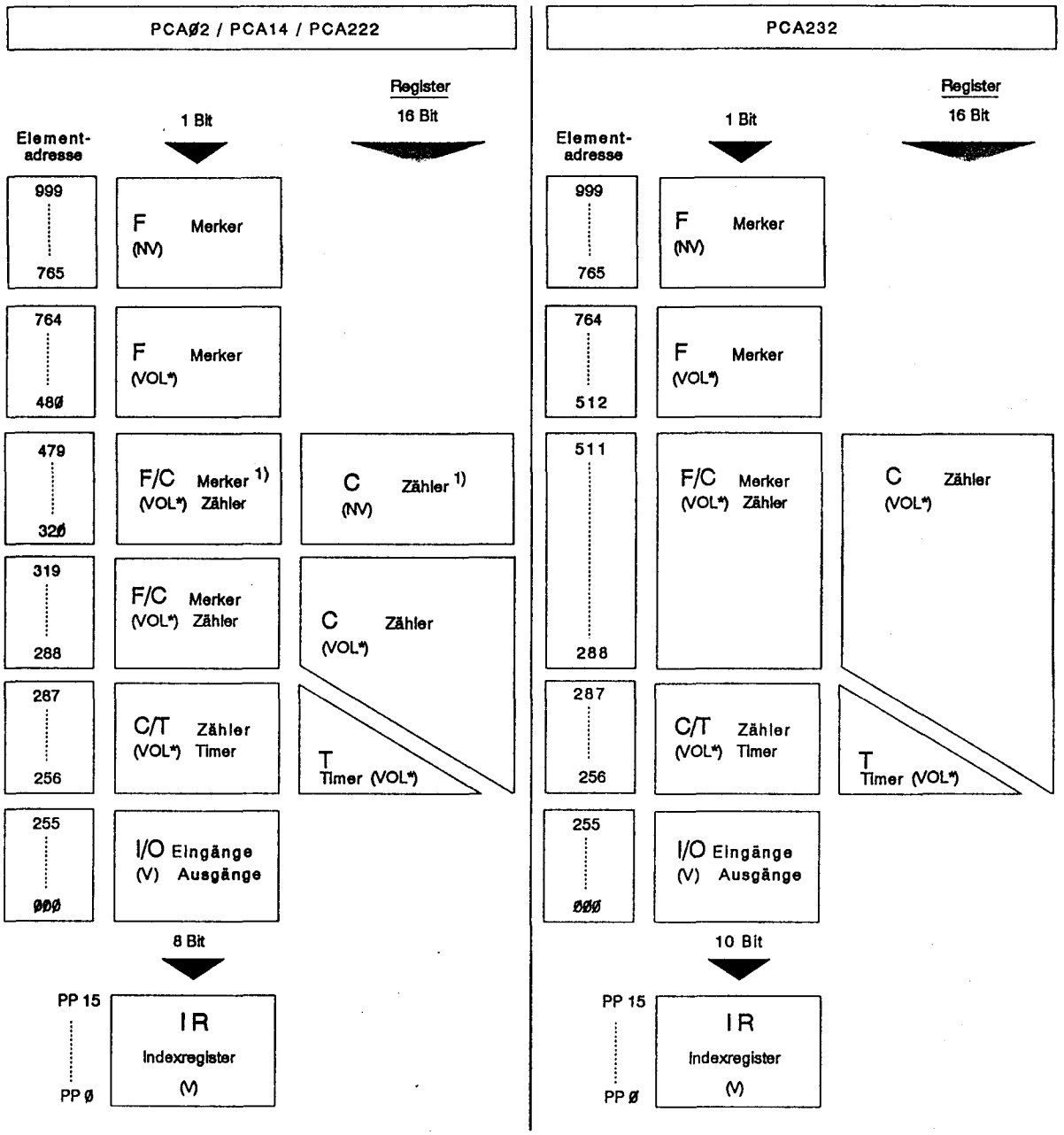
Diese Aufwärtskompatibilität kann wie folgt dargestellt werden:

Stufe	Baureihe PCAØ	Baureihe PCA1	Baureihe PCA2
(3) Wortprozessor-befehle			PCA232 (231)
(2) Erweiterungs-befehle	PCAØ2	PCA14	PCA222
(1H) Ergänzungs-befehle	PCAØ1	PCA15	PCA221
(1) Basisbefehle		(PCA13)	(PCA21Ø)

"Intelligenz"-Niveau	Baureihe PCAØ	PCA1	PCA2
Softstufe 3 Softniveau 2 + 32 Wortbefehle mit . Arithmetik ± 9 Digits . Datentransfer . Hardware-Uhr			Anwenderspeicher: 8K Programmschritte + 8K Textcharakter + 8K Byte Daten  PCA232 256 bzw 512 E/A
Softstufe 2 Softniveau 1H + serielle Schnittstelle + Datum-Uhr + Datenregister + Parameterbefehle (Soft-Interrupt, FIFO, PID)	nur OEM:  PCAØ.M12 PCAØ.M14 Anwenderspeicher: max. 4K Programmschritte max. 4K Textcharakter/Daten	 PCA141 PCA146 PCA147 + C45 32 (56) 64(112) 128(224) E/A Anwenderspeicher: max. 8K Programmschritte max. 8K Textcharakter/Daten	Anwenderspeicher: max. 8K Programmschritte max. 8K Textcharakter oder Daten  PCA222 256 bzw 512 E/A
Softstufe 1H Softniveau 1 + Arithmetik + Transferfunktionen + Check-Sum	Standard:  PCAØ.M12 PCAØ.M14 24/32 E/A 48/64 E/A Anwenderspeicher: 4K Programmschritte		Anwenderspeicher: 8K Programmschritte  PCA221 256 bzw 512 E/A
Softstufe 1 Befehlssatz mit 32 Basisbefehlen inkl. Timer Counter Indexregister Subroutinen Parallelprogrammen		 PCA13Ø PCA135 32(56) 64(112) E/A Anwenderspeicher: 2K Programmschritte	Anwenderspeicher: 2K Programmschritte  PCA21Ø 256 bzw 512 E/A


Baureihe PCAØ
Baureihe PCA1
Baureihe PCA2
M12/M14
M22/M24
PCA15
PCA14
PCA221
PCA222
PCA232
HARDWARE
SOFTWARE
PCAØ
HARD 1
HARD 2
SOFT 1H
SOFT 2
SOFT 3
Ergänzende Handbücher
PCA1.H1
PCA2.H2
P16
P18
PCA ASSEMBLER
LAN1 T9..
SAIA PMS

G 3 Register-Uebersicht der Prozessoren mit Software-Stufe (2) und (3)



(V) flüchtig
(NV) nicht flüchtig
(VOL*) flüchtig, mit Brücke auf nicht flüchtig umschaltbar

---

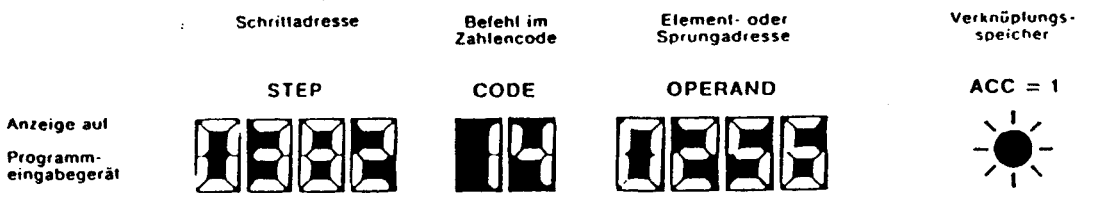
1) Zählererweiterung bis C479 ab folgenden Firmware-Versionen PCAØ2: V6.13Ø, PCA14: V6.Ø34, PCA222: V6.23Ø.

G 4 Uebersicht zum Befehlssatz Stufe (1) (Basisbefehle)

	Zahlen-code	Mnemo-code	Befehl englisch	Beschreibung
Logikbefehle	Ø1	STH	Start High	Beginn einer Verknüpfung mit Element abgefragt auf High
	Ø2	STL	Start Low	Beginn einer Verknüpfung mit Element abgefragt auf Low
	Ø3	ANH	AND High	UND-Verknüpfung zwischen ACCU und Element abgefragt auf High
	Ø4	ANL	AND Low	UND-Verknüpfung zwischen ACCU und Element abgefragt auf Low
	Ø5	ORH	OR High	ODER-Verknüpfung zwischen ACCU und nächstem Element abgefragt auf High
	Ø6	ORL	OR Low	ODER-Verknüpfung zwischen ACCU und nächstem Element abgefragt auf Low
	Ø7	XOR	Exclusive OR	Exklusiv-ODER-Verknüpfung des Elementes
	Ø8	NEG	Negate Accu	Negiere Accu (Verknüpfungsergebnis)
	Ø9	DYN	Dynamic Control	Dynamisierung der Verknüpfung (Accu wird nur im 1. Zyklus beeinflusst)
Schaltbefehle	1Ø	OUT	Set Output with Status of Accu	Setze Ausgang oder Merker mit Inhalt des Accu
	11	SEO	Set Output	Setze Ausgang oder Merker speichernd
	12	REO	Reset Output	Setze Ausgang oder Merker zurück
	13	COO	Complement Output	Frage den Ausgang oder Merker ab und setze ihn umgekehrt
Zeit- und Zählbefehle	14*	STR*	Set Timer	Setze Zeitglied auf vorgewählten Wert und starte es
	15*	SCR*	Set Counter	Setze Zähler auf vorgewählten Wert
	17	INC	Increment Counter	Erhöhe den Stand des Zählers um 1
	18	DEC	Decrement Counter	Erniedrige den Stand des Zählers um 1
Sprungbefehle	2Ø	JMP	Unconditional Jump	Unbedingter Sprung auf Schrittadresse
	21	JIO	Jump if Accu is One	Springe, wenn Accu = 1 auf Schrittadresse
	22	JIZ	Jump if Accu is Zero	Springe, wenn Accu = Ø auf Schrittadresse
	23	JMS	Jump to Subroutine	Springe in Unterprogramm
	24	RET	Return from Subrout.	Rücksprung vom Unterprogramm
Wartebefehle	25	WIH	Wait if High	Warte, solange Element High
	26	WIL	Wait if Low	Warte, solange Element Low
Hilfsbefehle	ØØ	NOP	No Operation	Keine Operation
	19	SEA	Set Accu	Setze Accu = 1
	16	SEI	Set Index	Setze Indexregister auf vorgewählten Wert
	27	INI	Increment Index	Erhöhe das Indexregister 1
	28	DEI	Decrement Index	Erniedrige das Indexregister 1
	29**	PAS**	Program Assignment	Zuweisung des Parallelprogrammes
	3Ø	DOP	Display Operand	Anzeige eines Operanden
	31	DTC	Display Timer or Counter	Anzeige des Zeit- oder Zählerstandes

* zweizeilige Befehle (zweite Zeile enthält vorgewählten Wert)
** zweizeiliger Befehl (zweite Zeile enthält Anfangsadresse des Programmes)

G 5 Uebersicht Ergänzungsbefehle Stufe (1H)



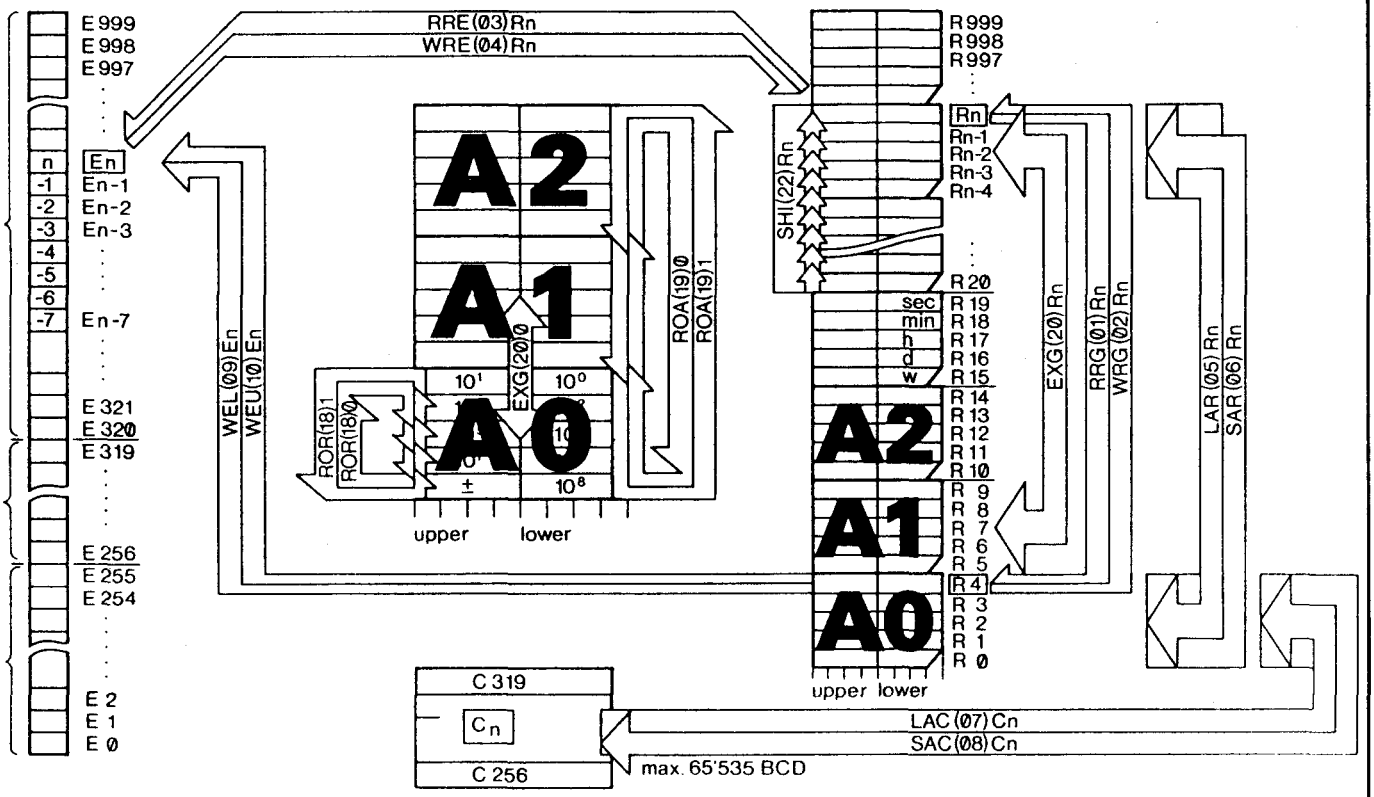
	Mnemo-Code	Zahlen-Code	Befehl englisch	Beschreibung
Transfer-Befehle	STR	14	Set Timer	
	SCR	15	Set Counter	
		19	2. Zeile	Einlesen 5 x 4 Bit BCD
		2Ø	2. Zeile	Ausgeben 5 x 4 Bit BCD
		21	2. Zeile	Ausgeben 8 Bit binär
		22	2. Zeile	Ausgeben 12 Bit binär
		23	2. Zeile	Ausgeben 16 Bit binär
		24	2. Zeile	Einlesen 8 Bit binär
		25	2. Zeile	Einlesen 12 Bit binär
		26	2. Zeile	Einlesen 16 Bit binär
		31	2. Zeile	Transfer Zähler ---> Zähler bzw. Indexregister -> Zähler
Arithmetik-Befehle	SCR	15	Set Counter	
		27	2. Zeile	Addiere +
		28	2. Zeile	Subtrahiere -
		29	2. Zeile	Multipliziere x
		3Ø	2. Zeile	Dividiere :
Indexier-Befehle (für Operanden ≥ 256)	SEI	16	Set Index	Setze Indexregister auf vorgewählten Wert
	INI	27	Increment-Index	Erhöhe das Index-register um 1
	DEI	28	Decrement-Index	Erniedrige das Index-register um 1

	Mnemo-Code	Zahlen-Code	Operand	Beschreibung
Spezial-Befehle (diese Befehle sind zweizeilig)	PAS	29	18	Veränderung der Anzahl aktiver Parallelprogramme
	PAS	29	3Ø...38	Check-Sum

G 6 Übersicht Erweiterungsbefehle Stufe (2)

Zahlen-code	Mnemo-code (2.Zeile)	Operand	Beschreibung		Beschreibung Seite
ØØ	NOP	1111	Gezielte Schnittstellenbehandlung (Empfang)		1H
29	PAS	16	PAS-Befehle Exklusiv-Betrieb eines Parallelprogrammes (mit blockierter Zeitbasis, Software Datum-Uhr und serieller Datenschnittstelle)	2-zeilige PAS-Befehle	2H
	(ØØ)	Ø	2. Zeile ist immer Ø		
29	PAS	17	Ende des Exklusiv-Betriebes		2H
	(ØØ)	Ø	2. Zeile ist immer Ø		
29	PAS	19	Ende des Interrupt-Managements		5H
	(ØØ)	Ø	2. Zeile ist immer Ø		
29	PAS	23	Befehl zur Textausgabe		5H
	(ØØ)	Ø...818	2. Zeile bestimmt die Anfangs-Textnummer Ausgabe erfolgt bis zum Zeichen <NUL> im Text		
29	PAS	24	Sofortige Ausgabe eines Telegrammes		5H
29	PAS	3Ø	Überprüfen des Systemprogrammes durch <Check-Sum>		6L
	(ØØ)	Ø			
29	PAS	31...38	Überprüfen des Anwenderprogrammes durch <Check-Sum> (31...38 = 1.K...8.K)		
	(xx)	xxxx	2. Zeile enthält die Prüfsumme		
29	PAS	5Ø	Datenaustausch zwischen Datum-Uhr und Zählerregister		6H
	(xy)	Cn	xy bestimmt die Art der Funktion: Cn bestimmt den Timer/Zähler, aus welchem der in das Register der Datum-Uhr zu schreibende Wert entnommen wird bzw. in welchem der aus dem Register der Datum-Uhr gelesene Wert abzulegen ist.		
29	PAS	54	Datenaustausch zwischen Zählerregister und Textspeicher		7H
29	PAS	56	Datenaustausch zwischen Zählerregister und Datenspeicher (nur PCA232)		7H
29	PAS	55	Datenaustausch zwischen Wortregister und Textspeicher (PCA231 und PCA232)		12H
29	PAS	57	Datenaustausch zwischen Wortregister und Datenspeicher (nur PCA232)		12H
29	PAS	58	Datenaustausch zwischen RAM-Speichermodul (PCA1.R25) und Zählerregister (nur PCA14)		16H
29	PAS	1ØØ	Befehl zum Festlegen der Übertragungsparameter der seriellen Datenschnittstelle und des Modus T, E, C, N oder P	1Ø-zeilige PAS-Befehle	57K
29	PAS	19Ø	Befehl für das Interrupt-Management (schnelle Reaktion durch Exklusiv-Betrieb gemäss Interrupt-Service-Routine)		20H
29	PAS	2ØØ : 212	Befehl für die Parameter von maximal 32 PID-Regelkreisen		28H
29	PAS	25Ø	Befehl für die Definition und Aktivierung eines Rotations- oder Schieberegisters		41H
29	PAS	251	Befehl für die Definition und Aktivierung eines Stapelregisters (FIFO)		4H

G 7 Uebersicht der Wortprozessorbefehle Stufe (3) (nur PCA23)
NOP 1248 bewirkt Wechsel in Wortprozessor-Befehlssatz

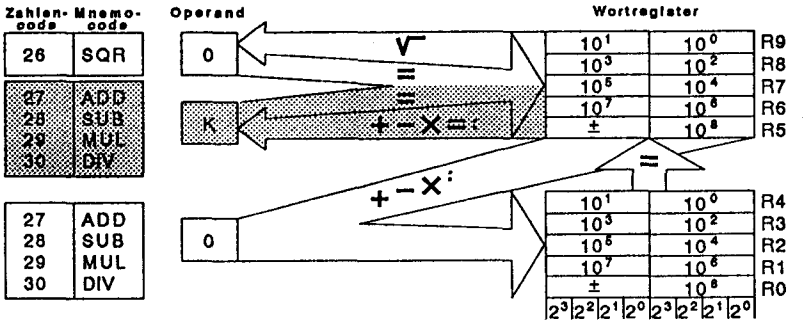


Zahlen-code	Mnemo-code	Operand	Befehl englisch	Beschreibung	Daten-format	
			Datentransfer-Befehle			
Ø1	RRG	Rn	Read Register	Lese Wort Rn und speichere es in R4	8 Bit	i
Ø2	WRG	Rn	Write into Register	Schreibe in Rn das Wort von R4	8 Bit	i
Ø3	RRE	Rn	Read Register and write in Elements	Lese Wort Rn und speichere es in den 8 Elementen En...En-7, adressiert durch A1	8 Bit	i
Ø4	WRE	Rn	Write Register with Elements	Schreibe in Rn den Inhalt der 8 Elemente En...En-7, adressiert durch A1	8 Bit	i
Ø5	LAR	Rn	Load AØ with Registers	Lade AØ mit Registerblock Rn	5x8 Bit	i
Ø6	SAR	Rn	Store AØ into Registers	Speichere AØ in den Registerblock Rn	5x8 Bit	i
Ø7	LAC	Cn	Load AØ with Counter	Lade AØ mit Zähler Cn	BCD	i
Ø8	SAC	Cn	Store AØ into Counter	Speichere AØ in den Zähler Cn	BCD	i
Ø9	WEL	En	Write Elements with {Lower} Digit	Schreibe das {R4 (10^{0})} in die Elemente	4 Bit	i
1Ø	WEU	En	Write Elements with {Upper} Digit	Digit {R4 (10^{1})} En...En-3		
			Behandlung der Arithmetik-Register			
18	ROR	Ø/1	Rotate Register	Rotiere R4...R3...R2...R1...RØ bzw. RØ...R1...R2...R3...R4		
19	ROA	Ø/1	Rotate A	Rotiere AØ...A1...A2 bzw. A2...A1...AØ		
2Ø	EXG	Ø	Exchange A1 with AØ	Tausche A1 mit AØ	5x8 Bit	
		Rn	Exchange A1 with Rn...Rn-4	Tausche A1 mit Rn...Rn-4	5x8 Bit	i

(i) = indexierbar

Arithmetik-Befehle

Zahlen-code	Mnemo-code	Operand	Befehl englisch	Beschreibung	Daten-format	
25	CMP	En	Compare AØ with A1	Vergleiche AØ mit A1 A1 > AØ ---> En = 1 A1 = AØ ---> En-1 = 1 A1 < AØ ---> En-2 = 1	BCD	i
26	SQR	Ø	Square Root of A1	$\sqrt{A1}$ ---> A1, nur ganzzahlig	BCD	
27	ADD	Ø K	Add AØ to A1 Add K to A1	A1 + AØ ---> A1, <carry> A1 + K ---> A1, <carry> (K = 1...2Ø47)	BCD BCD	
28	SUB	Ø K	Subtract AØ from A1 Subtract K from A1	A1 - AØ ---> A1, <carry> A1 - K ---> A1, <carry> (K = 1...2Ø47)	BCD BCD	
29	MUL	Ø K	Multiply A1 by AØ Multiply A1 by K	A1 . AØ ---> A1, <carry> A1 . K ---> A1, <carry> (K = 1...2Ø47)	BCD BCD	
3Ø	DIV	Ø K	Divide A1 by AØ Divide A1 by K	A1 : AØ ---> A1, Rest in AØ, <carry> A1 : K ---> A1, Rest in AØ, <carry> (K = 1...2Ø47)	BCD BCD	



Behandlung der Arithmetik-Register

14 15	CLA LAI	Ø/1/2 K	Clear A Load A immediately	Lösche Register AØ oder A1 oder A2 wenn K ≤ 99 ---> Lade R4 mit Data wenn K = 1ØØ...2Ø47 ---> Lade AØ mit Data		
16 17	DBN BND	Ø Ø	Decimal to binary Binary to decimal	Wandle AØ dezimal in AØ binär Wandle AØ binär in AØ dezimal		

Zähl- und Sprung-Befehle

11	INR	Rn	Increment Register	Incr. { BCD-Wert Rn um 1 und setze <carry> } > 99		i
12	DER	Rn	Decrement Register	Decr. { falls Resultat < Ø }		i
13	SNC	Ø En	Skip if no <carry> Skip if En = Ø	Ueberspringe nächsten Befehl, wenn { <carry> = Ø / En = Ø }		i
24	SEW	Ø En	Skip to EWP if no <carry> Skip to EWP if En = Ø	Ueberspringe bis EWP oder NOP 1248, wenn { <carry> = Ø / En = Ø }		i i

Diverse Befehle

ØØ	NOP	1248	-	Wechsel von Bit- nach Wortprozessor setzt ACCU = 1		
ØØ	NOP	Ø	No operation	Keine Operation		
21	CLK	En	Clock source	Zuweisung einer Zeitimpulsquelle		
22	SHI	Rn	Shift registers	Schiebe Wörter ab R2Ø bis Rn um 1 Adresse nach oben		i
23	TXT	Txn	Text	Start der Textausgabe		i
31	EWP	Ø	End Word Processor	Ende der Arbeit im Wortprozessor		

(i) = indexierbar

TEIL H ERGAENZUNGSBEFEHLE STUFE 2

H 1 Einzeilige Befehle NOP 1111

H 2 2-zeilige PAS-Befehle
PAS 16/17
PAS 19
PAS 23
PAS 24
PAS 50
PAS 54/56
PAS 55/57
PAS 58

H 3 10-zeilige PAS-Befehle
PAS 190 (und PAS 19)
PAS 200...212
PAS 250/251

TEIL H ERGAENZUNGSBEFEHLE STUFE 2

H 1 Einzeilige Befehle

NOP 1111 Gezielte Schnittstellenbehandlung

Der Befehl NOP 1111 kann beliebig oft in einem Programm eingesetzt werden. Er bewirkt, dass die Empfangsseite der seriellen Schnittstelle sofort bearbeitet wird.

NOP 1111 kommt vor allem dort zum Einsatz, wo durch aufeinanderfolgende Bitprozessorbefehle kein PP-Wechsel und somit keine Schnittstellenbearbeitung stattfindet (z.B. einige SCR-Befehle nacheinander).

```
SCR   26Ø
 31   262
SCR   26Ø
 28   281
NOP   1111 ; NOP 1111 bewirkt Behandlung der Schnittstelle
SCR   26Ø
 3Ø     4
 .
 .
 .
```

NOP 1111 ist unabhängig vom Zustand des ACCUs und beeinflusst diesen auch nicht.

H 2 2-zeilige PAS-Befehle

PAS 16, PAS 17 Exklusiv-Betrieb ohne Zeitbasis

Um zeitlich sehr genaue Impulse ausgeben zu können oder um einen kleinen Programmteil sehr schnell ablaufen lassen zu können, ist es oft wünschenswert, nur ein einziges Parallelprogramm aktiviert zu haben.

PAS 16 blockiert alle anderen Parallelprogramme, sowie die Timer, die serielle Datenschnittstelle und die Software Datum-Uhr.

PAS 17 hebt den Blockier-Befehl wieder auf; d.h. die Parallelprogramme, die Timer und die Datum-Uhr laufen von dort weiter, wo sie unterbrochen wurden. Die serielle Datenschnittstelle ist wieder aktiv.

Befehlsformat

PAS (29)	16	Blockier-Befehl
ØØ	Ø	2. Zeile bleibt immer Ø

PAS (29)	17	Aufhebung des Blockier-Befehls
ØØ	Ø	2. Zeile bleibt immer Ø

1. Beispiel für einen präzisen 2ms* Zeitimpuls

Das Anwenderprogramm enthält mehrere Parallelprogramme. Ein Umlaufprogramm enthält folgende Befehle:

```
        .
        .
STH     4ØØ
JIO
PAS     16
        ØØ   Ø
SCR     26Ø
        ØØ   16
SEO     4Ø
DEC     26Ø
STH     26Ø
JIO
REO     4Ø
PAS     17
        ØØ   Ø
STH     3
ANL     7
        :
```

Dauer des Interrupts, während dem alle anderen Parallelprogramme, Timer, serielle Datenschnittstelle und Software Datum-Uhr blockiert werden.

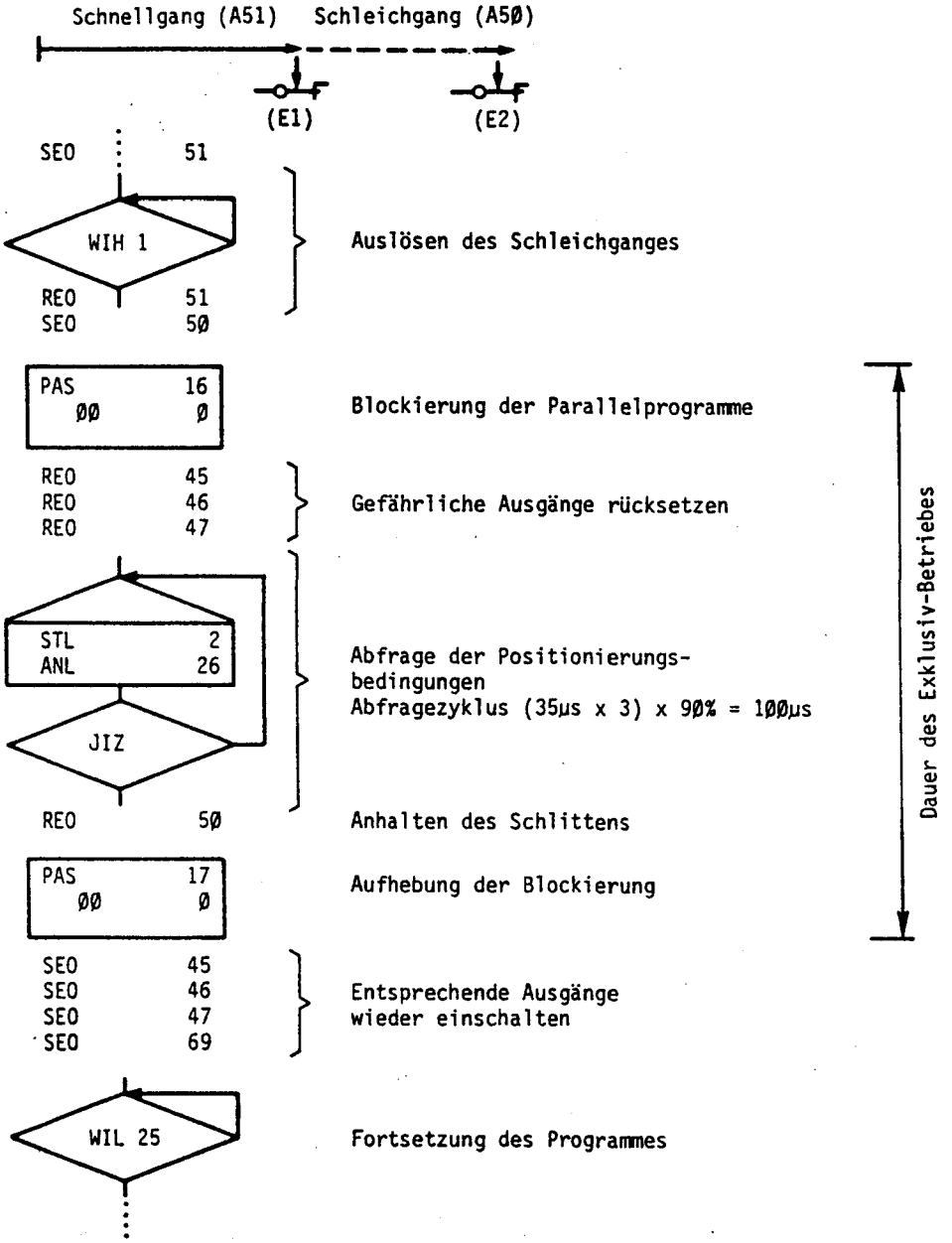
Berechnung der Impulsdauer:
(35µs x 4 x 16) x 9Ø% = 2ms*

Der Verkürzungsfaktor von 9Ø% ergibt sich durch den Wegfall der PP-Bearbeitung. Er muss durch Messung überprüft werden.

*) Zeiten gültig für PCA232. Für PCAØ2, PCA14 und PCA222 ergeben sich doppelt so lange Zeiten.

2. Beispiel für extrem kurze Reaktionszeit

Ein Schlitten soll im Schleichgang äusserst genau durch einen Signalschalter (E2) positioniert werden.



Im obigen Beispiel besteht die Abfrageschleife für E2 aus 3 Instruktionszeilen. Je nach zeitlichem Eintreffen des Stoppsignals (Oeffnen von E2) müssen 4 bis 6 Programmzeilen bis REO 5Ø abgearbeitet werden. Dies entspricht einer inneren Reaktionszeit von ca. 125...19Øµs*.

Die Gesamtreaktionszeit hält man durch Addition der Verzögerungen an Ein- und Ausgangsinterfaces zu den o.a. Werten. Bei geglätteter Gleichstromversorgung sind diese Verzögerungen konstant und können direkt in den Vorgang hineinjustiert werden.

---

*) Zeiten gültig für PCA232. Für PCAØ2, PCA14 und PCA222 ergeben sich doppelt so lange Zeiten.

Anmerkungen

- Die Dauer der Blockierung ist abhängig vom Anwenderprogramm und kann mehrere Sekunden betragen; im o.a. Beispiel sind während dieser Dauer alle Ueberwachungsfunktionen unterbrochen. Wenn dies vermieden werden soll, können die Ueberwachungsfunktionen in die Abfrageschleife integriert werden.

- Dauert die Blockierung länger als 90ms, so gehen Software-Uhr und interne Timer um die entsprechende Zeitüberschreitung nach (bei vorgewählter Zeitbasis 1/10s).

- Während der Dauer der Blockierung ist keine Uebertragung von Informationen über die serielle Datenschnittstelle zulässig, da daraus Uebertragungsfehler resultieren könnten.

PAS 19 Ende des Interrupt-Managements.
(siehe PAS 19Ø)

PAS 23 Textausgabe vom Bitprozessor
(siehe Abschnitt K 5 - Text Ein- und Ausgabe)

PAS 24 Sofortige Telegramm-Ausgabe

PAS 24 ermöglicht die sofortige Ausgabe eines Charakters oder Telegrammes via serielle Schnittstelle unabhängig von der Zeitbasis. Mit PAS 24 lassen sich somit die Reaktionszeiten über die serielle Schnittstelle verkürzen.

Beschreibung

Durch "Hochsetzen" des Elementes TBY teilt das Anwenderprogramm dem System mit, dass ein Charakter oder ein Telegramm ausgegeben werden muss. Das Systemprogramm analysiert den logischen Zustand von TBY jedoch nur zu gewissen, genau definierten Zeitpunkten (sofern die serielle Schnittstelle nicht bereits sendet). Diese Zeitpunkte sind:

- Takt der Zeitbasis, d.h. alle 1ØØ bzw. 1Ø ms
- bei der Ausgabe oder dem Empfang der Charakter ACK, ENQ und dem letzten Charakter eines Telegrammes (ETX bzw. BCC)
- nach Beendigung einer Interrupt-Service-Routine durch den Befehl PAS 19
- nach der Ausgabe eines Textes durch den Befehl PAS 23 oder TXT (PCA232)
- durch den Befehl PAS 24

Anwendung

```
SEO    5Ø9      Transmit Busy (TBY) wird hochgesetzt
PAS     24
 ØØ      Ø      Sofortige Telegramm-Ausgabe
```

PAS 24 wird unabhängig vom ACCU ausgeführt und verändert den ACCU nicht.

PAS 5Ø Datenaustausch zwischen Datum-Uhr und Zähler

Unter Betriebsarten MAN BIT wurde der Schreib- und Lesevorgang für die Datum-Uhr im Adressbereich 4ØØØ...4ØØ7 mit dem Programmeingabegerät PØ5 beschrieben. Hier wird der Zugriff zur Datum-Uhr über die Anwendersoftware beschrieben. Dies ermöglicht das Stellen der Datum-Uhr via Anwendersoftware oder auch die Verwendung der Datum-Uhr für Echtzeit-Schaltfunktionen.

Mit einem zweizeiligen PAS-Befehl kann die Datum-Uhr gelesen oder neue Werte eingegeben werden.

	CODE	OPERAND
1. Zeile	PAS (29)	5Ø
2. Zeile	XX	ccc

Ø ≙ Schreiben in Datum-Uhr
1 ≙ Lesen von der Datum-Uhr

ccc: Zähler/Timer, der den Eingabe- bzw. gelesenen Wert aufnimmt

Adresse in Betriebsart MAN	XX	Zahlenwerte im Zähler ccc
(4ØØØ)	Ø ≙ Woche des Jahres	1...53
(4ØØ1)	1 ≙ Tag der Woche	1...7
(4ØØ2)	2 ≙ Jahr	Ø...99
(4ØØ3)	3 ≙ Monat	1...12
(4ØØ4)	4 ≙ Tag des Monats	1...28, 29*, 3Ø, 31
(4ØØ5)	5 ≙ Stunden	Ø...23
(4ØØ6)	6 ≙ Minuten	Ø...59
(4ØØ7)	7 ≙ Sekunden	Ø...59

Datum und Uhrzeit können auf verschiedene Art auch über die serielle Datenschnittstelle ausgegeben werden (siehe K 5.3 "Bedeutung der Sonderzeichen").

Beispiel:
Ueber Eingang 7 soll mit der ansteigenden Flanke eines Synchronisierimpulses jede Stunde in der 3Ø. Minute, die Minuten und Sekunden entsprechend gestellt werden.

```
  STH          7;      Synchronisiereingang
  DYN        4ØØ
┌─JIZ
  SCR        3ØØ   }   Referenzzähler auf Null stellen
  ØØ           Ø   }
  PAS          5Ø **   Sekunden auf Null stellen
  Ø7         3ØØ
  SCR        3ØØ   }   Referenzzähler auf 3Ø stellen
  ØØ           3Ø   }
  PAS          5Ø **   Minuten auf 3Ø stellen
  Ø6         3ØØ
└──────►     :
```

Teil L
Beispiel 1

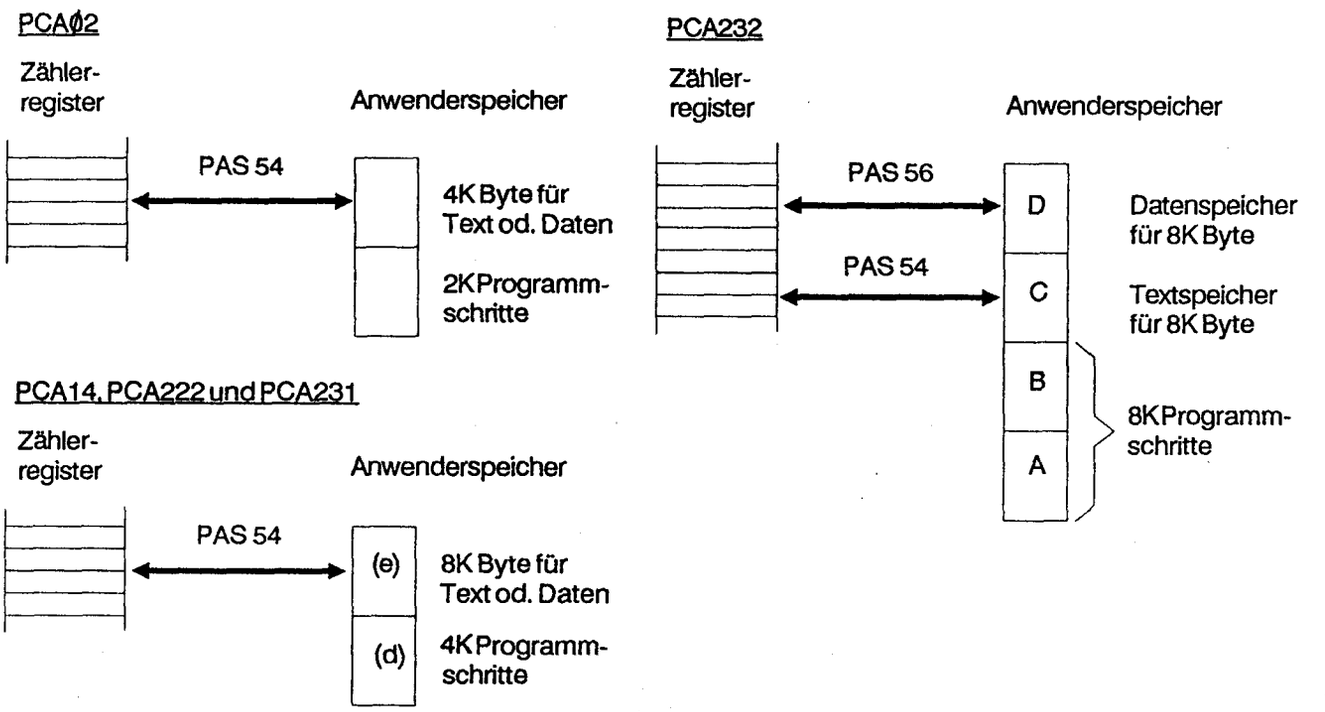
*) Das Schaltjahr wird nur von einer Hardware-Uhr berücksichtigt.
Die Software-Uhr der CPU kennt keine Schaltjahre.
**) Der PAS 5Ø-Befehl wird unabhängig vom ACCU immer ausgeführt und beeinflusst den ACCU nicht.

PAS 54 Datenaustausch Zählerregister <---> Textspeicher

PAS 56 Datenaustausch Zählerregister <---> Datenspeicher (nur PCA232)

Wie die nachstehende Uebersicht zeigt, kann mit den beiden Befehlen eine grosse Anzahl Daten in der SAIA°PCA abgelegt werden, indem dazu der Textspeicher (oder ein Teil davon) verwendet wird. Bei der PCA232 steht ein gesonderter Datenspeicher zur Verfügung, der mit PAS 56 angesprochen werden kann. Um die Aufwärtskompatibilität zu den übrigen Baureihen zu wahren, kann auch bei der PCA232 der PAS 54 zwischen Zählerregister und Textspeicher angewendet werden.

Uebersicht

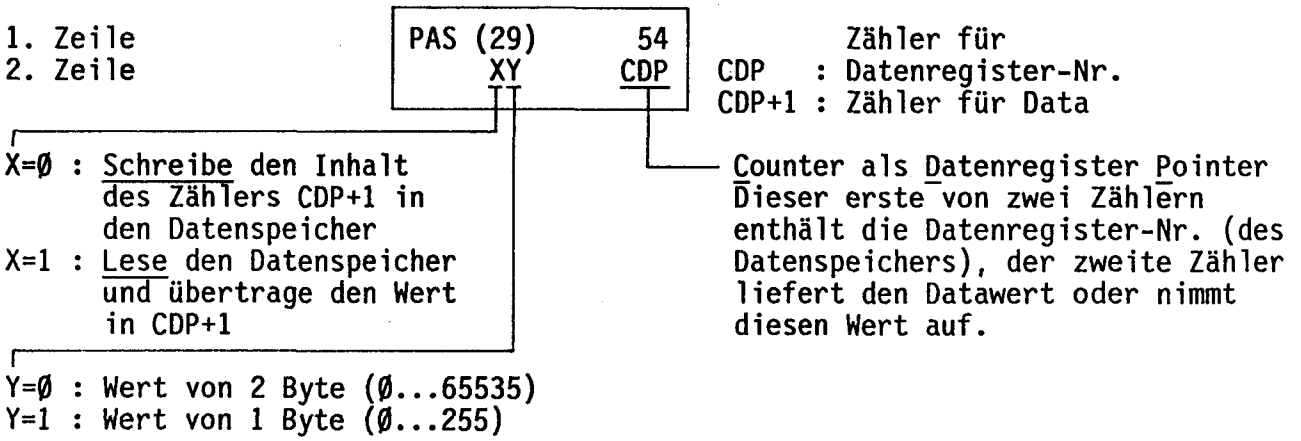


Es sind folgende Punkte zu beachten:

- Der Speicherbereich des Textspeichers bzw. Datenregisters muss als RAM ausgelegt sein, wenn dort Daten eingeschrieben werden sollen.
- Im Kapitel "Betriebsarten" ist unter "TEXT" bzw. "MAN BCD" beschrieben, wie mit Hilfe des Programmiergerätes auf "manuelle" Art der Datenspeicher gelesen bzw. überschrieben werden kann.
- Im folgenden wird für beide Speicher (Text und Daten) nur der Ausdruck "Datenspeicher" verwendet.
- Alle folgenden Beschreibungen und Beispiele beziehen sich auf PAS 54. Der Befehl PAS 56 (für PCA232) kann in genau gleicher Weise angewendet werden.

PAS 54 Transfer zwischen Datenspeicher und Zähler-Register

Befehlsformat:



PAS 54 wird unabhängig vom ACCU ausgeführt und verändert den ACCU nicht.

Der Befehl PAS 54 führt folgende Aktionen aus:

- Schreiben (X=Ø): Der Wert aus CDP+1 (1 oder 2 Byte) wird in den Datenspeicher mit Datenregister-Nr. gemäss Inhalt von CDP geschrieben.

- Lesen (X=1): Lesen der 1 oder 2 Datenregister aus dem Datenspeicher mit Datenregister-Nr. gemäss CDP und übertragen auf Zähler CDP+1.

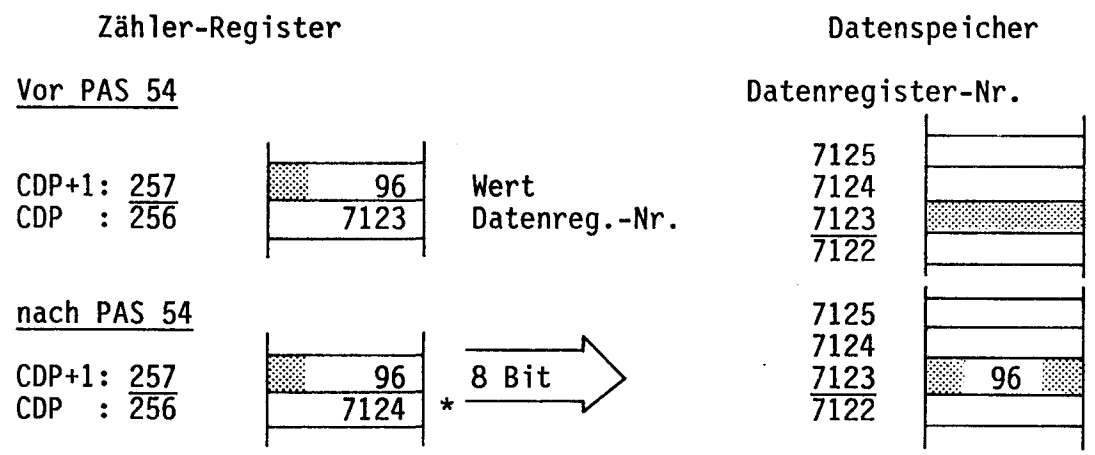
- Nach Ausführung von einer dieser Operationen wird der Inhalt des Zählers CDP automatisch um 1 bzw. 2 erhöht (entsprechend 1 oder 2 Byte). Damit steht der Pointer automatisch auf der richtigen Datenregister-Nr. für den folgenden Datentransfer.

Beispiele

- Schreiben eines Wertes von 1 Byte von einem Zähler in den Datenspeicher

Der Zähler 257 enthält den Wert 96 und soll auf Datenregister 7123 übertragen werden.

```
SCR (15)    256   } Zähler CDP mit
     ØØ    7123   } Datenregister-Nr. laden
PAS (29)     54
     Ø1     256    Schreiben von 1 Byte aus CDP+1 (C257) in den Datenspeicher
```

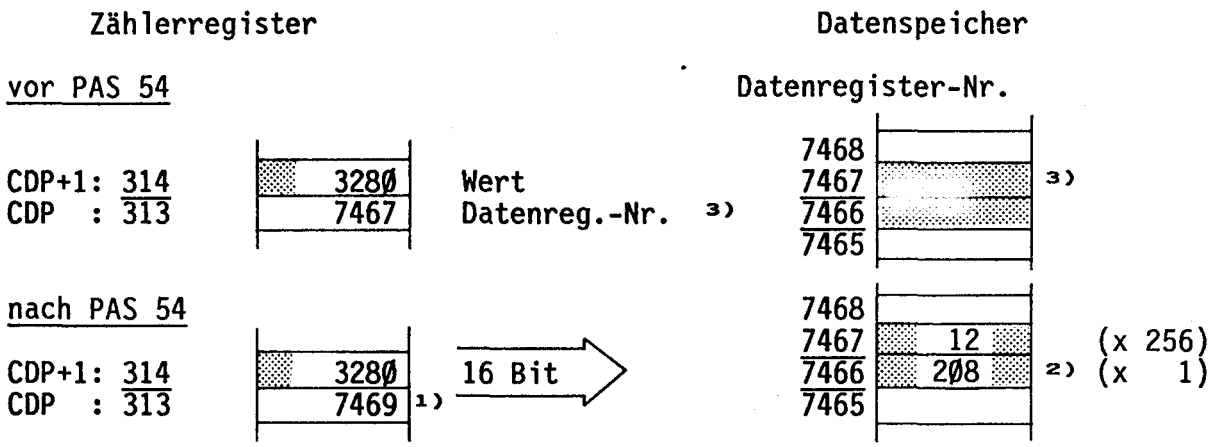


*) Der Wert von CDP ist um 1 erhöht worden.

- Schreiben eines Wertes von 2 Byte von einem Zähler in den Datenspeicher

Der Zähler 314 enthält den Wert 328Ø und soll auf die Datenregister 7466 und 7467 übertragen werden.

```
SCR (15)    313   }  Zähler CDP mit
     ØØ    7467   }  höherer Datenregister-Nr. laden
PAS (29)     54
     ØØ     313      Schreiben von 2 Byte aus CDP+1 (C314)
                     in den Datenspeicher
```



- Lesen eines Wertes von 2 Byte vom Datenspeicher und Uebertragen in einen Zähler

Der Wert 56477, abgelegt in den Datenregister-Nummern 193Ø und 1931, soll auf den Zähler 275 übertragen werden.

```
SCR (15)    274   }  Zähler CDP mit
     ØØ   1931   }  höherer Datenreg.-Nr. laden
PAS (29)     54
     1Ø     274      Schreiben von 2 Byte in CDP+1 (C275)
```

---

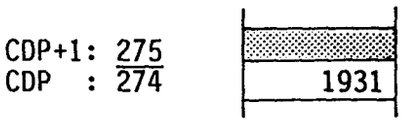
1) Der Wert von CDP ist um 2 erhöht worden.

2) Der Wert ist im Datenspeicher in binärer Form abgelegt (12 x 256 + 2Ø8 = 328Ø). Der Monitor der CPU erlaubt via P1Ø/PØ5 diese Werte als kombinierten 2-Byte Wert darzustellen (siehe "Betriebsarten").

3) Bei Werten von 2 Byte möglichst ungerade Datenreg.-Nr. wählen.

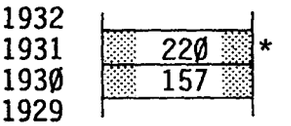
Zählerregister

vor PAS 54



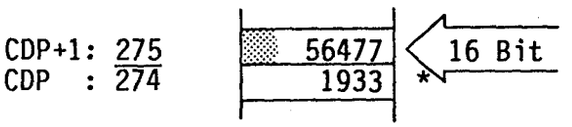
Datenspeicher

Datenregister-Nr.

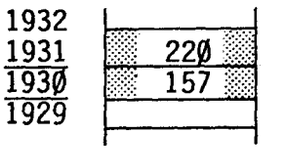


*) Wert 56'477 = 220 x 256 + 157

nach PAS 54



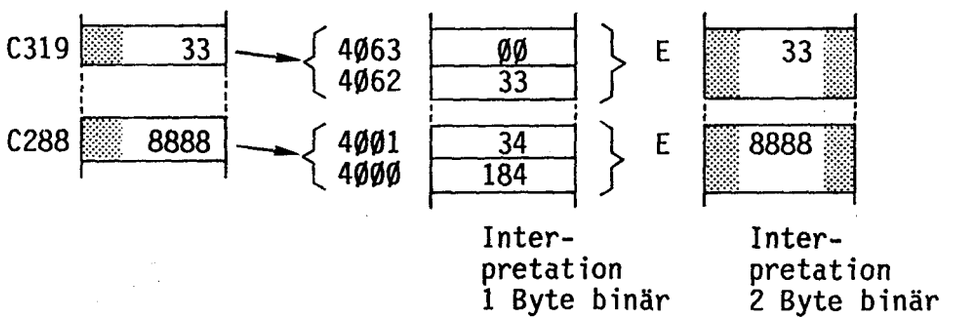
Datenregister-Nr.



*) Der Zähler CDP ist um 2 erhöht worden.

- Es sollen 32 Zähler von C288...C319 mit Werten von 0...9999 in den Datenspeicher ab Datenreg.-Nr. 4001 übertragen werden.

```
SEI      0

SCR    260  } Zähler CDP mit tiefster
 00   4001  } Datenreg.-Nr. laden

SCR    261  } Uebertragzähler (CDP+1) für
 31   1288  } C288...C319

PAS     54
 00    260   Schreiben von 2 Byte aus CDP+1 (C261) in den Datenspeicher

INI     31
JIO
```



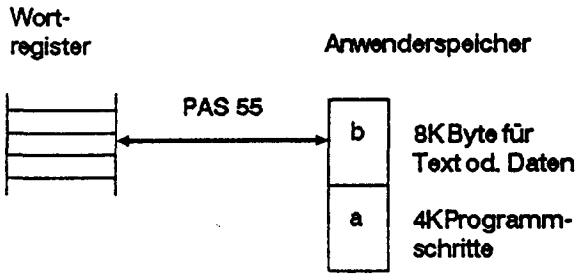
PAS 55 Datenaustausch Wortregister <---> Textspeicher (PCA231 und 232)

PAS 57 Datenaustausch Wortregister <---> Datenspeicher (nur PCA232)

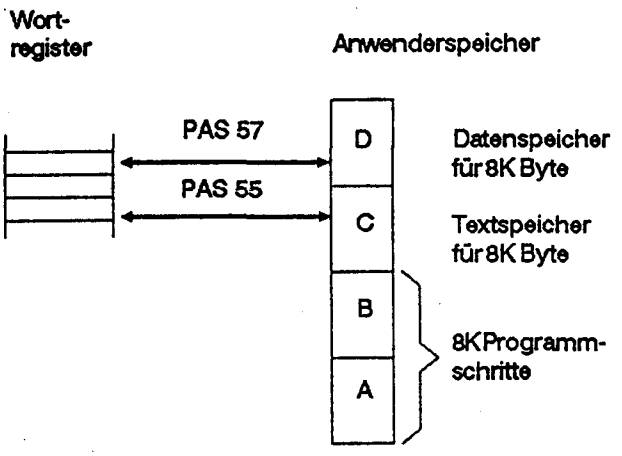
Analog zu PAS 54/56 kann bei den PCA23 mit den Befehlen PAS 55/57 Datenaustausch zu den Wortregistern bewirkt werden.

Im folgenden wird nur der Befehl PAS 57 beschrieben. PAS 55 hat die entsprechende Funktion zum Textspeicher wie aus den beiden Uebersichten hervorgeht.

PCA231



PCA232



Befehlsformat:

1. Zeile
2. Zeile

PAS	(29)	57
	XØ	CDP

CDP : Zähler für Datenregister
CDP+1 : Zähler für Wort-register-Adresse

X=Ø : Schreibe den Inhalt des Wortregister-Blockes (Pointer CDP+1) in den Datenspeicher (Pointer CDP)

X=1 : Lese den Inhalt des Datenspeichers (Pointer CDP) und übertrage den Wert in das Wortregister (Pointer CDP+1)

Counter als Datenregister Pointer
Dieser erste von zwei Zählern (CDP) enthält die Register-Nr. (des Datenspeichers), der zweite Zähler (CDP+1) enthält die Adresse des Wortregister-Blockes.

Der Befehl PAS 57 führt folgende Aktionen aus:

- Schreiben (X=Ø): Der Inhalt des Wortregisterblockes mit Adresse CDP+1 (5 Wortregister = total 9 BCD-Werte + Vorzeichen) wird als 5x8 Bit-Wert in den Datenspeicher ab Datenregister gemäss CDP geschrieben.

- Lesen (X=1): Die 5 Werte ab Nummer gemäss CDP werden im Textspeicher gelesen und in den Wortregisterblock gemäss CDP+1 übertragen.

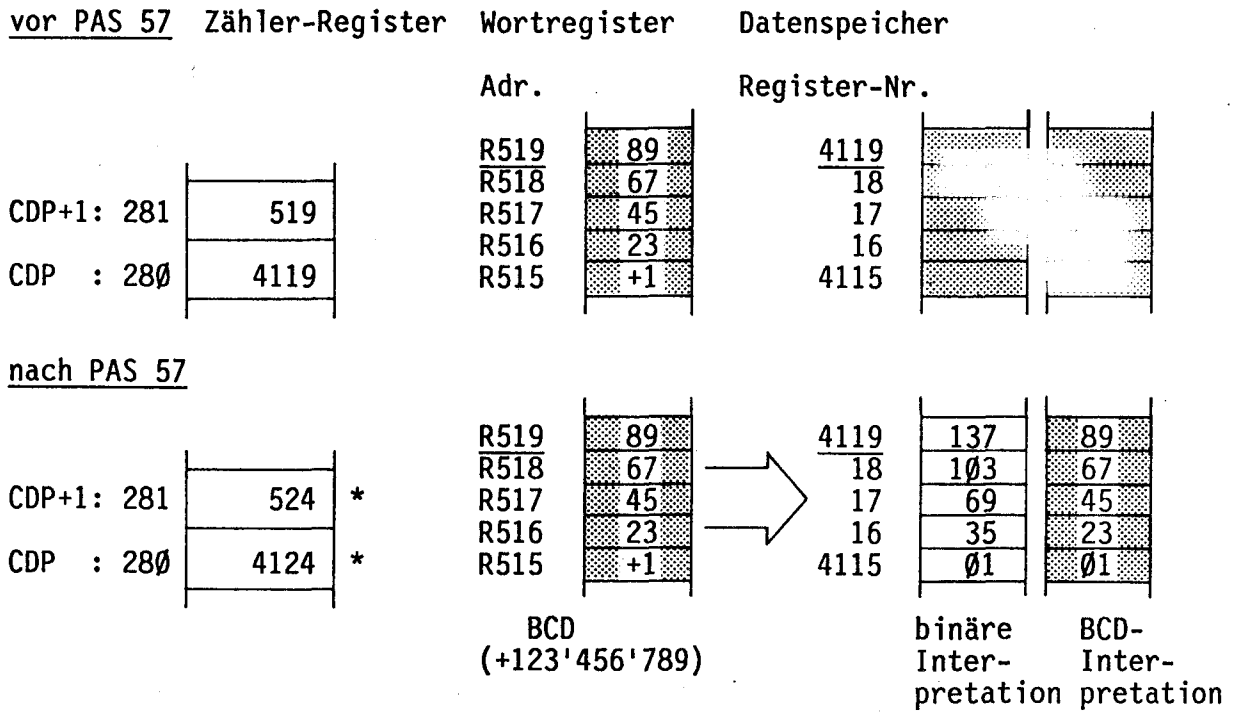
- Anschliessend werden die Inhalte der Zähler CDP und CDP+1 je um 5 erhöht. Damit stehen die Pointer für den folgenden Datentransfer bereits an der richtigen Stelle.

Beispiele

- Schreiben eines Wortregisterblockes in den Datenspeicher

Der Registerblock R515...R519 soll auf die Datenregister-Nummern 4115...4119 des Datenspeichers geschrieben werden.

```
SCR (15)    28Ø  }  CDP laden mit
     ØØ   4119   }  Register-Nr. (MSD)

SCR (15)    281  }  CDP+1 laden mit
            519  }  Wortregister-Adresse (MSD)

PAS (29)     57     Wert von Registerblock in
     ØØ     28Ø    Datenspeicher schreiben
```



Das Vorzeichen + wird als Ø übernommen. Die übrigen Werte werden bit-richtig zu je 8 Bit abgelegt, jedoch bei einer manuellen Abfrage binär interpretiert. Manuelle Umwandlung in BCD-Format siehe Kapitel "Betriebsarten". Bei einer späteren Rückübertragung ins Wortregister werden diese Daten dort wieder im BCD-Format mit den ursprünglichen Werten gelesen.

---

*) Die Zähler für CDP und CDP+1 sind um 5 erhöht worden.
Zum Transferieren von Registerblöcken ist es vorteilhaft, Adressen mit den Endziffern 4 oder 9 zu wählen.

- Lesen von 5 Charaktern im Datenspeicher und Uebertragen ins Wortregister

Es sollen die Datenregister 422Ø...4224 gelesen und auf den Wortregisterblock R62Ø...R624 ** übertragen werden.

```
SCR (15)    28Ø  }
     ØØ    4224  }   CDP laden mit
                     höchster Datenregister-Nr. (MSD)

SCR (15)    281  }
     ØØ     624  }   CDP+1 laden mit höchster
                     Wortregister-Adresse (MSD)

PAS (29)     57      Wert in Datenspeicher lesen und
     1Ø     28Ø      nach Wortregister übertragen
```

vor PAS 57

	Zähler-Register	Wortregister	Datenspeicher	binäre Interpretation	BCD-Interpretation
		R624	4224	33	21
		R623	23	67	43
CDP+1: 281	624	R622	22	1Ø1	65
		R621	21	135	87
CDP : 28Ø	4224	R62Ø	422Ø	153	99

nach PAS 57

	Zähler-Register		Wortregister	BCD (-987'654'321)	Datenspeicher	binäre Interpretation	BCD-Interpretation
			R624	2 1	4224	33	21
			R623	4 3	23	67	43
CDP+1: 281	629	*	R622	6 5	22	1Ø1	65
			R621	8 7	21	135	87
CDP : 28Ø	4229	*	R62Ø	- 9	422Ø	153	99

*) Die Zähler für CDP und CDP+1 sind automatisch um 5 erhöht worden.

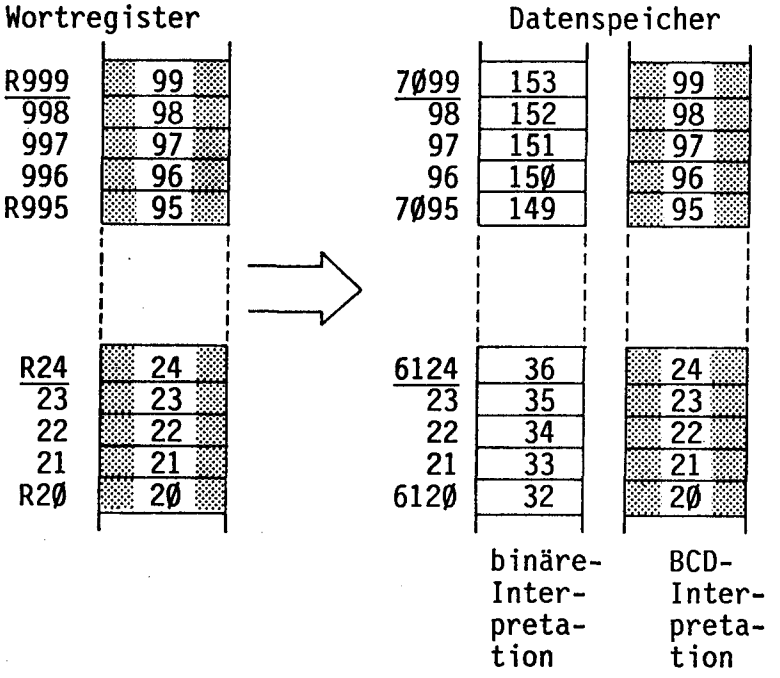
**) Zur Behandlung von Registerblöcken möglichst Adressen mit Endziffern 4 oder 9 wählen.

- Das gesamte Wortregister R2Ø...R999 soll auf den Datenspeicher, beginnend mit Datenregister-Nr. 612Ø kopiert werden.

  R2Ø...R999 sind 196 Registerblöcke.

```
SCR     289   }  Schlaufenzähler laden mit
 ØØ     196   }  Anzahl Registerblöcken

SCR     29Ø   }  CDP laden mit
 ØØ    6124   }  tiefster Datenregister-Nr. + 4

SCR     291   }  CDP+1 laden mit tiefster
 ØØ     24    }  Wortregister-Adresse + 4

PAS      57
 ØØ     29Ø      Schreiben auf Datenspeicher

DEC     289
STH     289
JIO
```

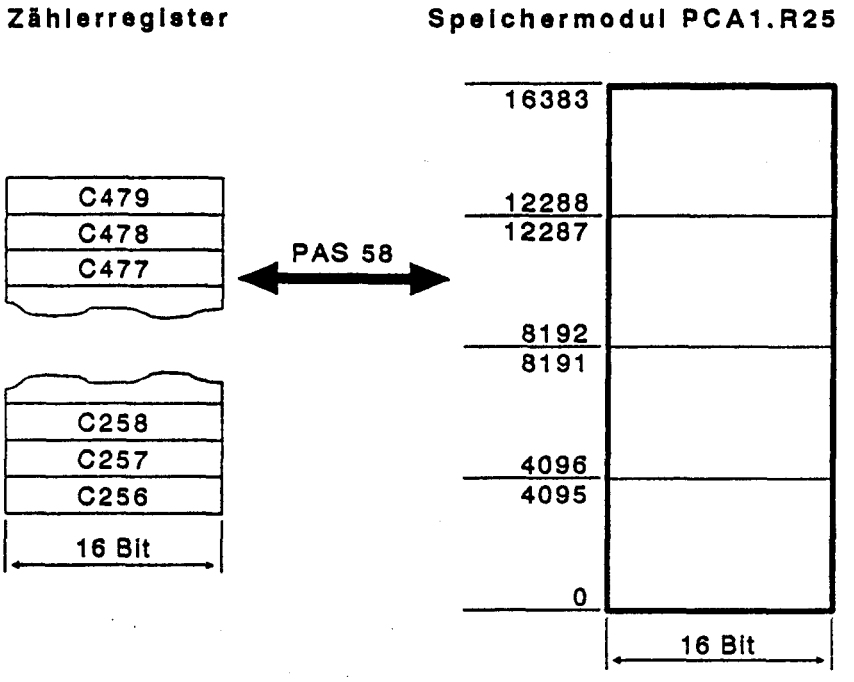
nach PAS 57



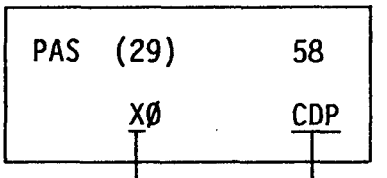
PAS 58 Datenaustausch zwischen RAM-Speichermodul PCA1.R25 und Zählerregister (nur PCA14 ab V6.Ø36)

Mit dem RAM-Speichermodul PCA1.R25 verfügt die PCA14 über einen grossen, gepufferten Datenspeicher von 16 K Wörter (32 K Byte). Der Befehl PAS 58 ermöglicht den Datentransfer zwischen diesem Datenspeicher und den Zähler-Registern.

Uebersicht



Befehlsformat:



CDP : Zähler für Speicheradresse R25
CDP+1 : Zähler für Data

X=Ø : Schreibe den Inhalt des Zählers CDP+1 in das Speichermodul R25.

X=1 : Lese im Speichermodul R25 und übertrage den Wert in CDP+1.

Counter als Datenregister Pointer
Dieser erste von zwei Zählern (CDP) enthält die Speicheradresse des Moduls R25, der zweite Zähler liefert den Datawert oder nimmt diesen Wert auf.

PAS 58 wird unabhängig vom ACCU ausgeführt und verändert den ACCU nicht.

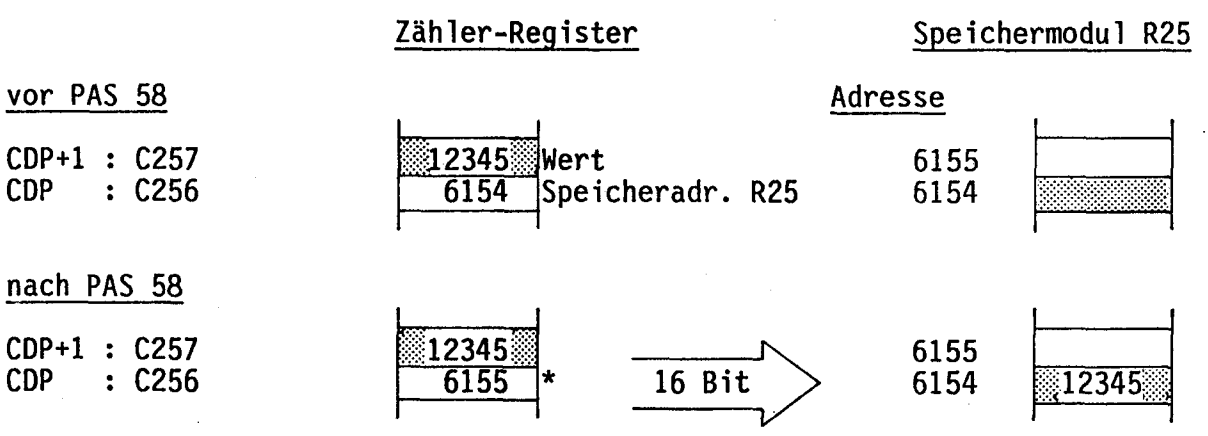
Der Befehl PAS 58 führt folgende Aktionen aus:

- Schreiben (X=Ø): Der Wert aus CDP+1 (1 Wort = 2 Byte) wird in das Speichermodul PCA1.R25, Speicheradresse gemäss Inhalt von CDP, geschrieben.
- Lesen (X=1): Lesen des Wertes (1 Wort = 2 Byte) im Speichermodul PCA1.R25, Speicheradresse gemäss Inhalt von CDP und übertragen auf Zähler CDP+1.
- Nach Ausführung von einer dieser Operationen wird der Inhalt des Zählers CDP automatisch um 1 erhöht. Damit steht der Pointer auf der richtigen Speicheradresse des R25 für den nächstfolgenden Datentransfer.

Beispiele

- Schreiben des Inhaltes von C257 (z.B. Wert 12345) in den Datenspeicher des Moduls PCA1.R25, Adresse 6154.

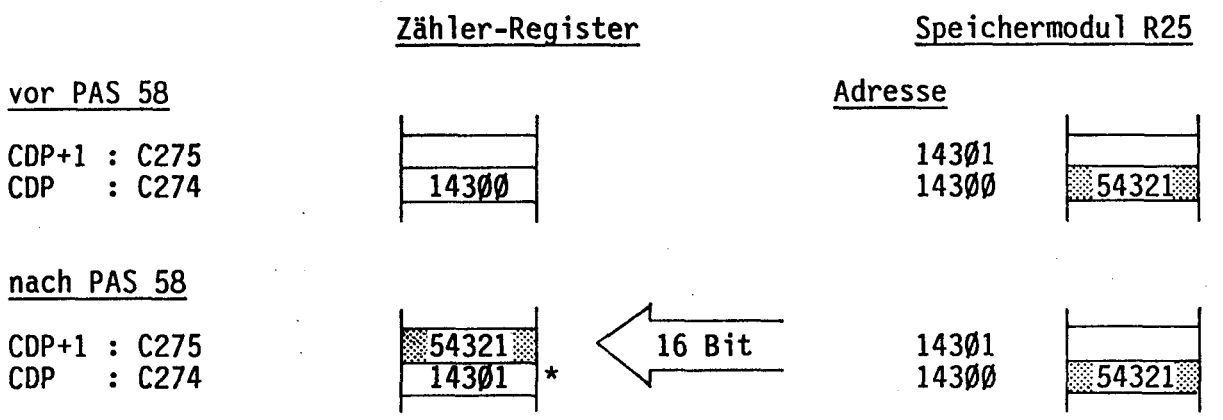
```
SCR (15)   256   ; Zähler CDP mit Speicheradresse des R25 laden
     ØØ   6154
PAS (29)    58
     ØØ   256    ; Schreiben des Wertes aus CDP+1 (C257) in das
                   Speichermodul R25
```



*) Der Wert von CDP ist um 1 erhöht worden.

- Lesen des Wertes (z.B. 54321) vom Speichermodul R25, Adresse 143ØØ, und in den Zähler C275 übertragen.

```
SCR (15)  274    ; Zähler CDP mit Speicheradresse des R25 laden
     Ø6  2Ø12      (6 x 2Ø48 + 2Ø12 = 143ØØ)

PAS (29)   58
     1Ø   274    ; Lesen des Wertes auf R25 und in CDP+1 (C275) übertragen
```



*) Der Wert von CDP ist um 1 erhöht worden.

- Es sollen die 16Ø Zählerinhalte von C32Ø...C479 in das Speichermodul PCA1.R25 ab Adresse 4ØØ1 übertragen werden.

```
      SEI     Ø
      SCR   26Ø  ;  Zähler CDP mit Speicheradresse des R25 laden
       ØØ  4ØØ1

┌──►   SCR   261  ;  Uebertragungszähler (CDP+1) für C32Ø...C479
│     31  132Ø
│
│    ┌──────────┐
│    │PAS    58 │
│    │ ØØ   26Ø │
│    └──────────┘
│
│     INI   159
└──   JIO
```

- Es sollen die 16Ø Werte des Speichermoduls R25, Adressen 4ØØ1...416Ø, in die Zähler C32Ø...C479 übertragen werden.

```
      SEI     Ø
      SCR   26Ø  ;  Zähler CDP mit Speicheradresse des R25 laden
       ØØ  4ØØ1

┌──►  ┌──────────┐
│     │PAS    58 │
│     │ 1Ø   26Ø │
│     └──────────┘
│
│     SCR  132Ø  ;  Wert von Uebertragungszähler (CDP+1) auf C32Ø...C479
│     31   261      übertragen
│
│     INI   159
└──   JIO
```

H 3 1Ø-zeilige PAS-Befehle

PAS 19Ø und PAS 19 Interrupt-Management (IM)

Funktionsbeschreibung

Durch ein Element, normalerweise einen Eingang, kann exklusiv eine Interrupt-Routine in Funktion gesetzt werden. Diese Interrupt-Service-Routine (ISR) ist anstelle des Parallel-Programmes PP15 mit PAS 19Ø zu assignieren.

Das Element INTRQ (Interrupt Request) kann entweder auf aktiv "H" oder aktiv "L" definiert werden.

Ist das Element INTRQ aktiv und das Element INTA (Interrupt Acknowledge, normalerweise ein Ausgang) auf Niveau "L", so wird die Interrupt-Service-Routine beim nächsten Wechsel eines Parallel-Programmes exklusiv gestartet. Gleichzeitig werden die übrigen Parallelprogramme in ihrer Arbeit unterbrochen.

Im Interrupt-Service-Programm kann nach Durchführung der erforderlichen Massnahmen mit PAS 19 die normale Parallel-Programm-Abarbeitung wieder freigegeben werden.

Die Assignierung durch PAS 19Ø erfolgt anstelle des 15. PP durch einen 1Ø-zeiligen Befehl (normalerweise am Programmanfang im Assignierungsteil):

```
1   PAS   (29)     19Ø     ;    Interrupt-Assignierung
2         ØØ       ssss  *  ;    Beginn ISR (Interrupt-Service-Routine)
3         ØØ         En     ;    Interrupt Request (Eingang INTRQ)
4         ØØ       ØØØx     ;    Level INTRQ ("H" = 1, "L" = Ø)
5         ØØ         En     ;    Interrupt Acknowledge (Ausgang INTA)
6         ØØ          Ø     )
7          .          .     |
8          .          .     }    Null
9          .          .     |
1Ø        ØØ          Ø     )
```

Die Aufhebung der Exklusiv-Abarbeitung erfolgt am Ende des Interrupt mit dem zweizeiligen Befehl PAS 19:

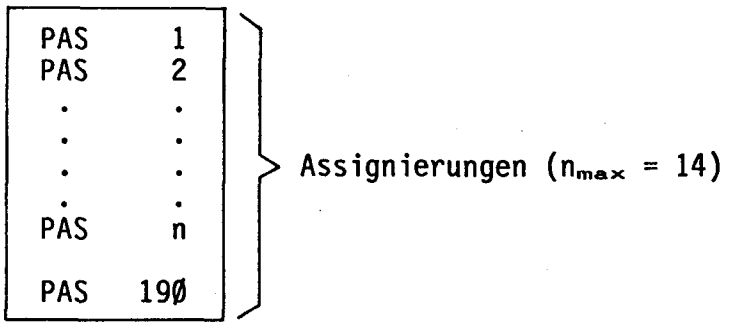
```
1   PAS   (29)      19     ;    Ende des Exklusiv-Laufes der Interrupt-
                             ;    Service-Routine (End ISR)
2         ØØ         Ø     ;    Null
```

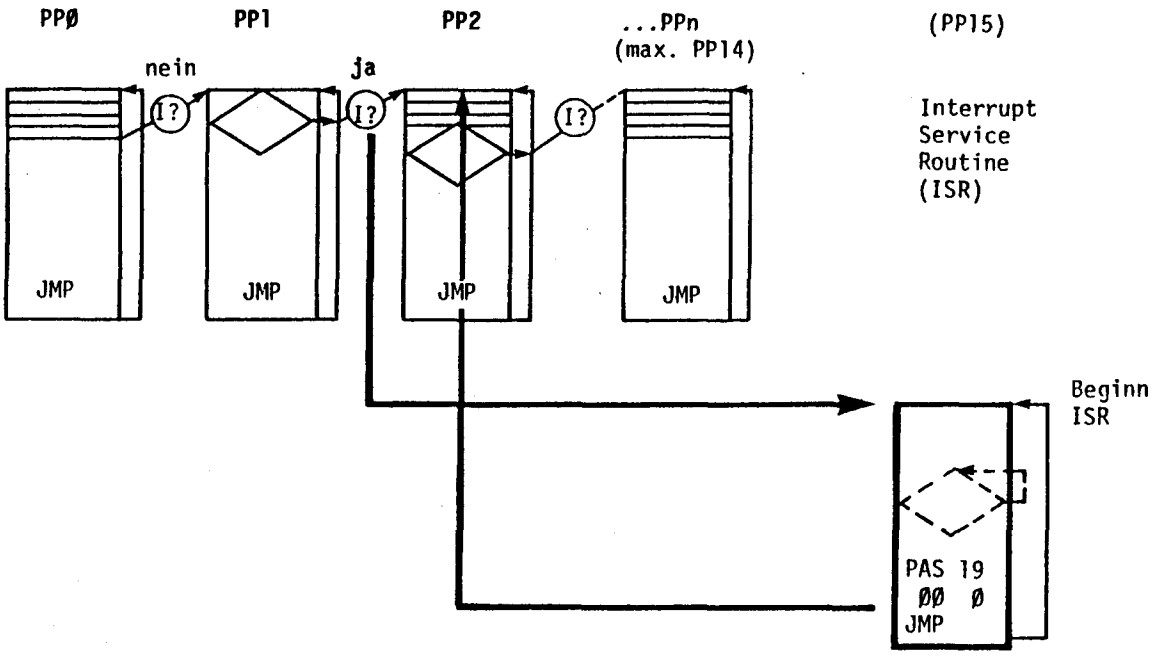
---

*) Bei Verwendung von IAD/KAD (INSERT und KILL) durch das Programm "CI" des PCA-Assemblers wird diese Schrittadresse nicht automatisch korrigiert.

Detailaspekte

a) Programmstruktur





Die Interrupt-Bedingungen werden bei jedem Wechsel von einem PP zum nächsten abgefragt, wobei im positiven Fall die Interrupt-Service-Routine exklusiv abgearbeitet wird.

Sind die Bedingungen für die Rückkehr zum Normal-Programm gegeben, so wird dies in der ISR mit PAS 19 eingeleitet:

- Die PP werden dort fortgesetzt, wo sie zum Zeitpunkt des Interrupts standen.
- Mit Neuassignierung der PP kann ein definierter Neustart bewirkt werden.

b) Reaktionszeit

Da die Abfrage auf die Interrupt-Bedingungen bei jedem PP-Programmwechsel erfolgt, ist die Reaktionszeit zwischen Schliessen des "Interrupt-Kontaktes" und Start des ISR-Programmes praktisch nur von der Eingangsverzögerung des "Interrupt-Einganges" abhängig. Zur Standard-Eingangsverzögerung von 8ms kommt auch bei ungünstiger Programmstruktur max. 1ms (PCA232) bzw. 2ms (PCAØ2/14/222) durch die interne Abarbeitung hinzu.

c) Timer-Abarbeitung und serielle Datenschnittstelle

Während der Exklusiv-ISR ist die Zeitbasis aktiv, d.h. sowohl die Timer als auch die serielle Datenschnittstelle sind in vollem Umfang benützbar. Dadurch können in der ISR nebst Zeitfunktionen auch Ausgabe von Texten oder Meldungen an ein übergeordnetes System enthalten sein.

d) Abhängigkeiten zwischen INTRQ, INTA und der Interrupt-Service-Routine (ISR)

Für den Start der Interrupt-Service-Routine müssen folgende Kriterien erfüllt sein:

- INTRQ (normalerweise ein Eingang) auf "H" bzw. auf "L", je nach Definition der 4. Zeile in der Interrupt-Assignierung.

- INTA (normalerweise ein Ausgang) auf "L".

INTA wird durch das Systemprogramm selbständig auf "H" gesetzt. Nach ca. 1ms beginnt der Exklusivlauf der ISR. Die Ablaufzeit der ISR sollte kürzer sein als die Zeit zwischen zwei Wechseln von INTRQ nach "H" bzw. nach "L".

Bezüglich REO INTA können zwei Fälle unterschieden werden, welche ganz unterschiedliche Wirkungen ergeben:

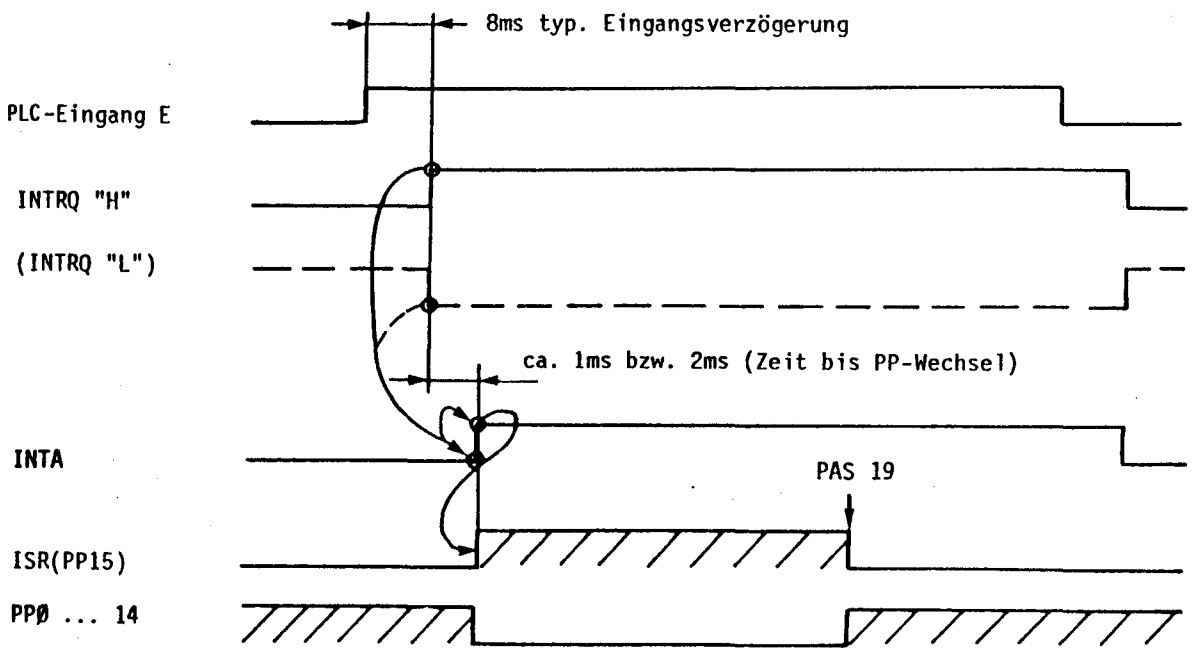
REO INTA befindet sich
- 1) ausserhalb der ISR
- 2) innerhalb der ISR

1) REO INTA ausserhalb der ISR (allg. Fall) in einem Umlaufprogramm

a) INTRQ aktiv "H" :
```
STL INTRQ
REO INTA
```

b) INTRQ aktiv "L" :
```
STH INTRQ
REO INTA
```

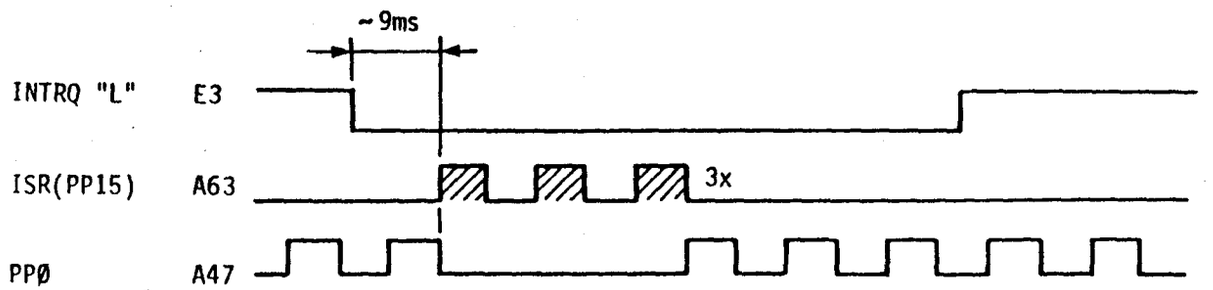
In diesem Fall wird die ISR nur einmal abgearbeitet, wenn INTRQ nach "H" bzw. nach "L" geht.



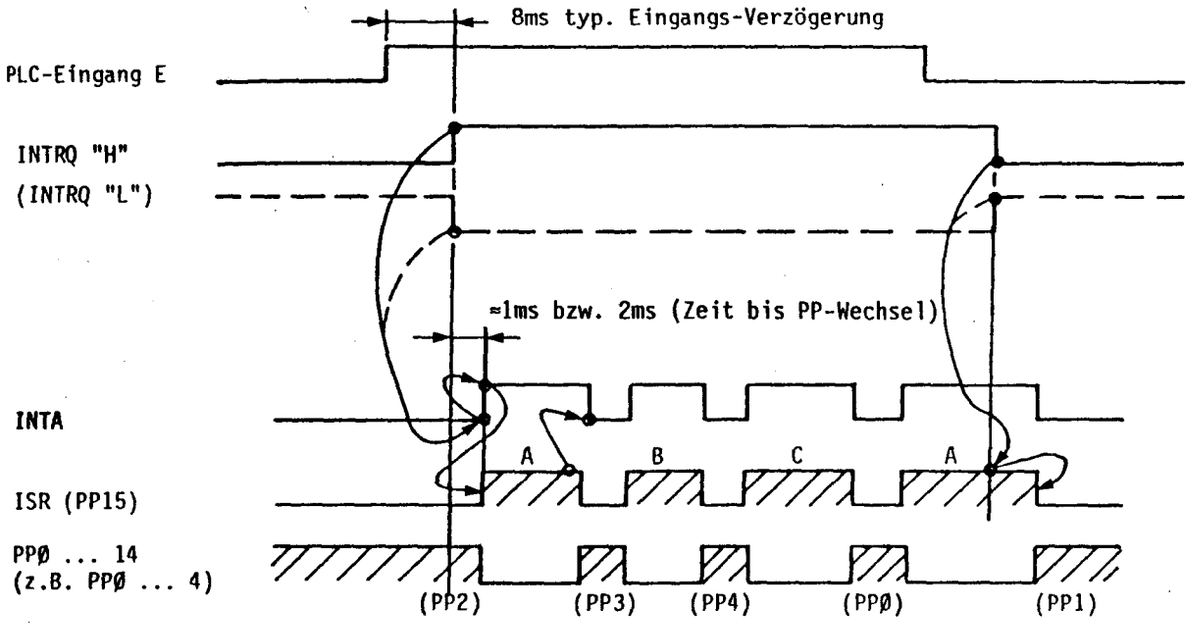
Beispiele für einmalige, exklusive ISR-Abarbeitung (REO INTA ausserhalb ISR, INTRQ aktiv "L")

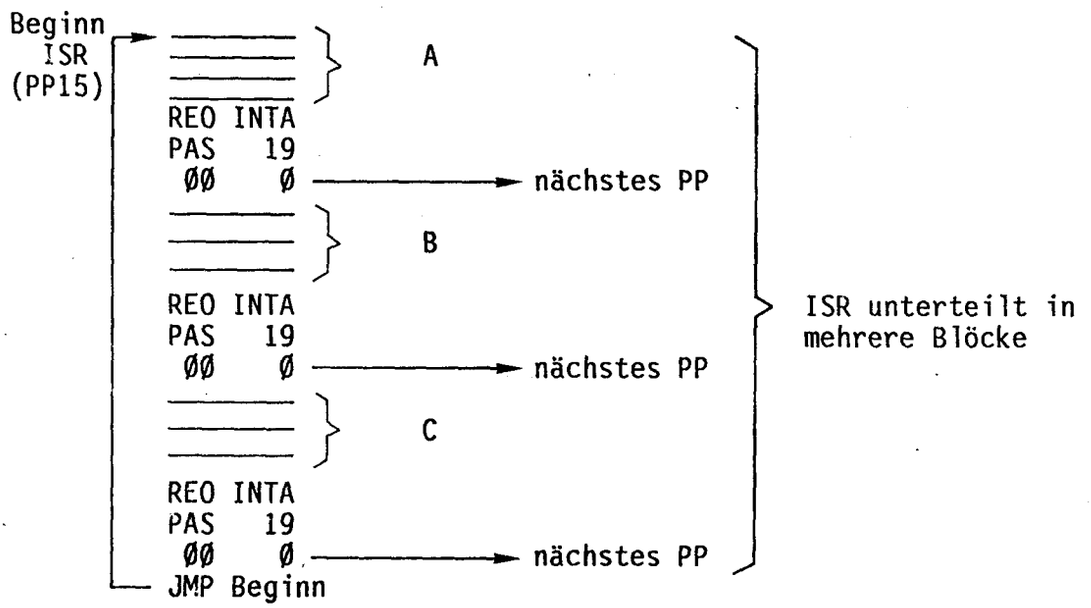
Didaktisches Beispiel:

```
851   PAS   190
  2    ØØ   873     Beginn ISR
  3    ØØ     3     INTRQ
  4    ØØ     Ø     LEVEL INTRQ aktiv "L"
  5    ØØ    55     INTA
  6    ØØ     Ø
  7    ØØ     Ø
  8    ØØ     Ø
  9    ØØ     Ø
86Ø    ØØ     Ø                                  } Assignierung
-------------------------------------------------
863   STR   256
  4    ØØ     5
  5   WIH   256     (PP-Wechsel --->)
  6   COO    47     ┌─┐ ┌─┐ ┌─┐
  7   STH     3     INTRQ                        } PPØ
  8   REO    55     INTA
869   JMP   863     (PP-Wechsel --->)
-------------------------------------------------
873   SCR   26Ø     Beginn ISR
  4    ØØ     6
  5   STR   257
  6    ØØ     2
  7   WIH   257
  8   COO    63     ┌┐┌┐┌┐
  9   DEC   26Ø                                  } ISR (PP15)
88Ø   STH   26Ø
  1   JIO   875
  2   PAS    19     Ende ISR
  3    ØØ     Ø
884   JMP   873
-------------------------------------------------
```



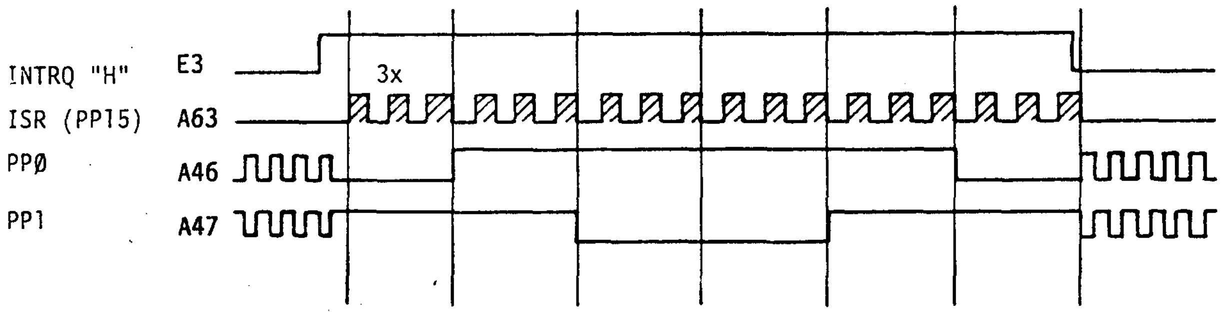
2) Erfolgt REQ INTA innerhalb der ISR (PP15), so wird die ISR so lange bearbeitet als INTRQ "H" bzw. "L" ist. Der Befehl PAS 19 bezeichnet dabei jeweils das Ende des Exklusivlaufes der ISR und gibt dem nächsten PP die Möglichkeit, eine Verknüpfung abzuarbeiten. Anschliessend wird bereits wieder die ISR behandelt. Die ISR kann auch aus mehreren Blöcken bestehen, wobei diese mit wesentlich höherer Priorität behandelt werden (z.B. zeitkritische Aktionen) als die restlichen PP.





Didaktisches Beispiel für mehrfache ISR-Abarbeitung mit hoher Priorität (REO INTA innerhalb ISR, INTRQ aktiv "H")

```
899   PAS    1                                    } Assignierung PP1
900   00     920

901   PAS    190                                  }
  2   00     930     Beginn ISR                   }
  3   00     3       INTRQ                        }
  4   00     1       LEVEL INTRQ aktiv "H"        }
  5   00     55      INTA                         }
  6   00     0                                    } Assignierung ISR (PP15)
  7   00     0                                    }
  8   00     0                                    }
  9   00     0                                    }
910   00     0                                    }

913   STL    256                                  }
  4   STR    256     Blinker                      }
  5   00     1       A46                          }
  6   COO    46                                   } PP0
  7   JMP    918     PP-Wechsel*                  }
918   JMP    913     PP-Wechsel                   }

920   STL    257                                  }
  1   STR    257     Blinker                      }
  2   00     1       A47                          } PP1
  3   COO    47                                   }
924   JMP    920     PP-Wechsel                   }

930   SCR    260     Beginn ISR                   }
  1   00     6                                    }
932   STR    258                                  }
  3   00     5                                    }
  4   WIH    258     Blinker                      }
  5   COO    63      A63                          }
  6   DEC    260     3x                           } ISR (PP15)
  7   STH    260                                  }
938   JIO    932                                  }
  9   REO    55                                   }
940   PAS    19      INTA                         }
  1   00     0       Ende ISR                     }
942   JMP    930                                  }
```

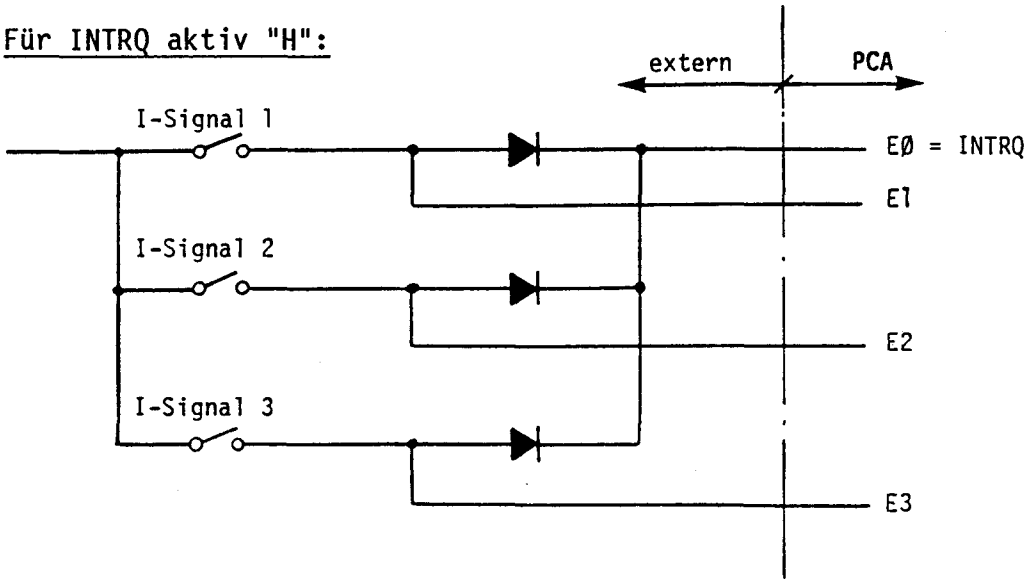


*) Durch den künstlich eingefügten JMP wird ein zusätzlicher PP-Wechsel bewirkt.

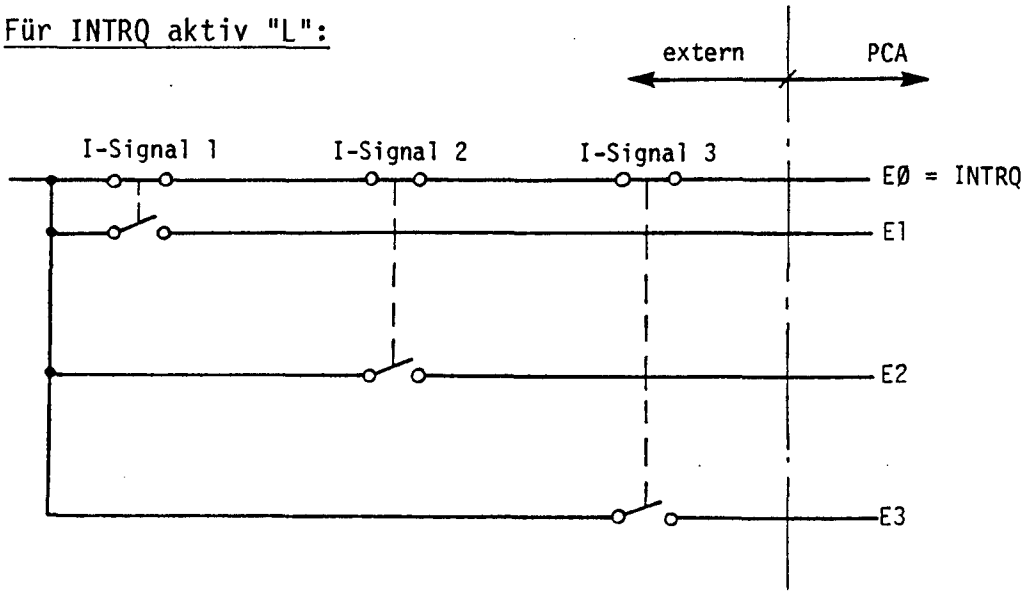
e) Verschiedene Interrupt-Prioritäten

Sollen verschiedene Prioritäts-Niveaux oder differenzierte ISR berücksichtigt werden, so kann dies wie folgt erreicht werden:

- Für INTRQ aktiv "H":



- Für INTRQ aktiv "L":



Die verschiedenen Interrupt-Signale werden je auf einen Eingang geführt.

Bei INTRQ "H" wird über eine Diodenmatrix ein gemeinsamer Interrupt-Eingang gebildet (z.B. EØ).

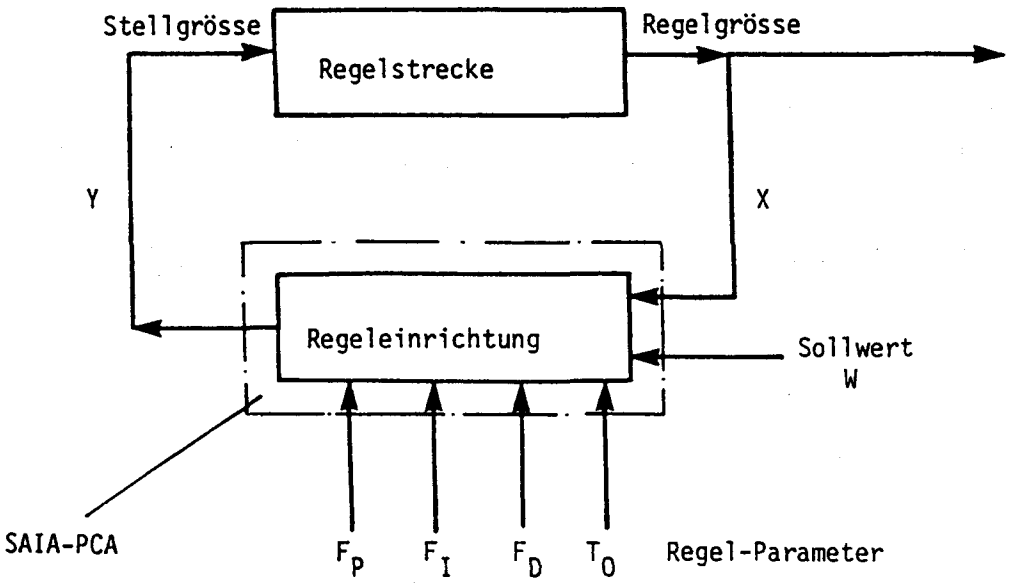
Bei INTRQ "L" benötigt man für jedes I-Signal 2 Kontakte. Ueber eine Serienschaltung des 2. Kontaktes wird der gemeinsame Interrupt-Eingang gebildet. Dieser muss im PAS 19Ø als INTRQ assigniert werden.

Innerhalb der ISR kann dann den einzelnen I-Signalen die entsprechende Priorität mit dem entsprechenden Programmteil zugewiesen werden.

PAS 2ØØ bis 212 PID-Regelung

Allgemeines

Der praktische Einsatz von PID-Regelungen setzt regelungstechnische Kenntnisse voraus. Im folgenden kann lediglich das Prinzip eines Regelkreises und die softwareseitige Behandlung mit den Befehlen PAS 2ØØ...212 erläutert werden.



Obige Figur zeigt einen geschlossenen Regelkreis. Am Ausgang der Regelstrecke erscheint die Regelgrösse X. Dieser Wert X wird der Regeleinrichtung zugeführt und mit dem Sollwert W verglichen. Das Resultat dieses Vergleiches, beeinflusst durch die Regelparameter F_P, F_I und F_D, wird im Takt der Abtastzeit $T_Ø$ als Stellgrösse Y der Regelstrecke wiederum zugeführt.

Die digitale Regeleinrichtung, wie sie eine SAIA°PCA darstellt, besteht aus folgenden Teilen:

- Analoger Eingang für die Regelgrösse X (mit A/D-Wandler)
- Recheneinheit (CPU)
- Analoger Ausgang (mit D/A-Wandler)

Jede CPU der Reihe PCA stellt Register (sog. Datablocks) für max. 32 PID-Regelkreise zur Verfügung. Genügend analoge Ein- und Ausgänge vorausgesetzt, bedeutet das, dass mit einer einzigen SAIA°PCA bis 32 voneinander unabhängige Regelkreise in der sogenannten DDC-Technik (Direct Digital Control) bearbeitet werden können.

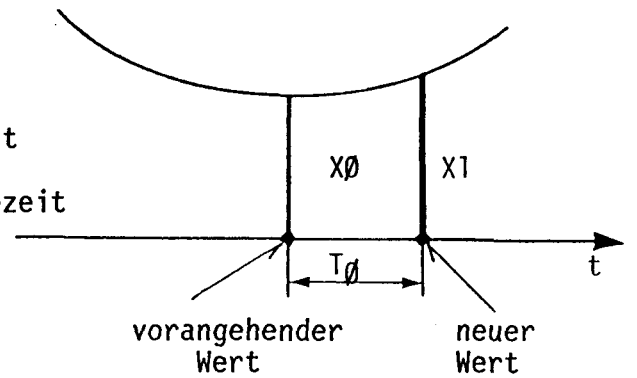
Die Recheneinheit (CPU der PCA) benützt folgende Formel zur Berechnung der Stellgrösse Y:

$$Y_1 = F_P [(W-X_1) + F_I S_1 + F_D (X_0 - X_1)]$$

wobei $S_1 = S_\emptyset + (W - X_1)$

Darin bedeuten:

W = Sollwert
X = Regelgrösse (geregelter Wert)
Y = Stellgrösse
F_P = Proportional-Faktor = 1 / X_P
X_P = Proportionalbereich
F_I = Integral-Faktor = $T_\emptyset$ / T_I
$T_\emptyset$ = Abtastzeit
T_I = Integralzeit = T_N = Nachstellzeit
F_D = Differential-Faktor = T_D / $T_\emptyset$
T_D = Differentialzeit = T_V = Vorhaltezeit



Die Indizes Ø und 1 bedeuten:

Ø = vorangegangener Wert
1 = neuer Wert

Bei der PCA232 (ab Firmware Version V7.12Ø) kann alternativ mit der folgenden erweiterten Formel gearbeitet werden, nach welcher auch Sollwertsprünge rasch berücksichtigt werden:

$$Y_{1W} = F_P \{ (W_1 - X_1) + F_I S_1 + F_D [(X_0 - X_1) - (W_0 - W_1)] \}$$

Programmaufbau

Durch Aufruf des Parameterbefehls PAS 2ØØ...212 berechnet das Systemprogramm der PCA die neue Stellgrösse Y_1 aufgrund der vorangegangenen Werte $X_ø$, $Y_ø$, $S_ø$ und der im PAS-Befehl enthaltenen Regelparameter.

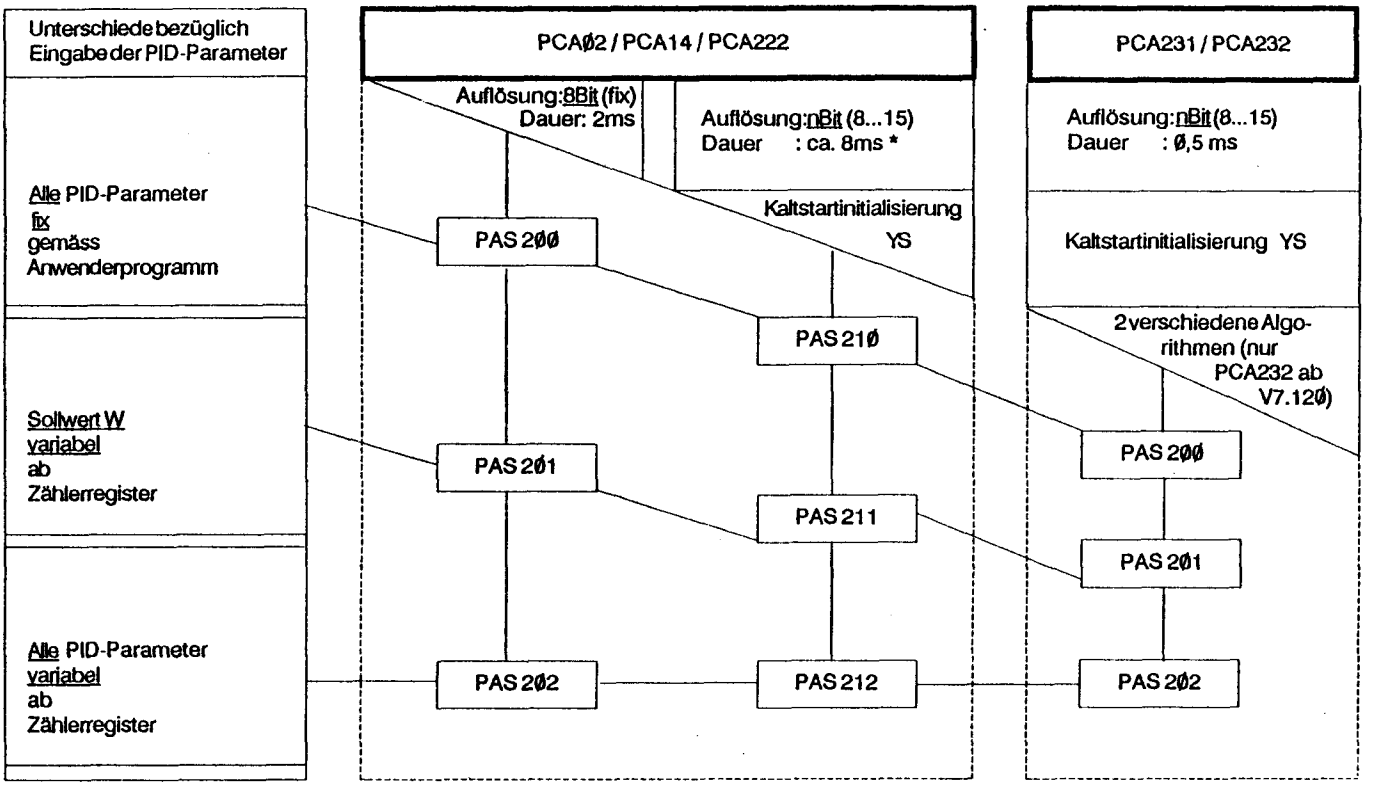
Der PAS 2ØØ...212-Befehl ist also kein Assignierungsbefehl, sondern er muss in einem Umlaufprogramm gemäss nachstehendem Muster zur periodischen Abarbeitung eingebaut werden.

Beispiel:

```
        .
        .
        .
    STH  256        } Abfrage, ob Timer für Tø (Abtastzeit) abgelaufen ist.
    JIO             }
    STR  256        } Wenn ja, dann:
     ØØ    4        } - Neu setzen mit Abtastzeit von z.B. Ø,4 sec.

    SEO    3        }
    SEO    8        } - Regelgrösse X von Analog-Eingangskanal 3 einlesen,
    SCR  26Ø        }   d.h. Wert von 12 Bit auf Hilfszähler 26Ø transferieren
     25   11        }   (gilt für Modul PCA2.W14)

    PAS  2ØØ...212  }
    1Ø-zeiliger     } - Stellgrösse Y berechnen
    Befehl          }

    SCR  26Ø        }
     22   11        } - Stellgrösse Y auf Analog-Ausgangskanal 15 ausgeben
    SEO   15        }   und somit der Regelstrecke zuführen
                    }   (gilt für Modul PCA2.W14)
        .
        .
        .
```

Beschreibung der Parameterbefehle PAS 2ØØ...212

Die Bedeutung der Befehle PAS 2ØØ...212 ist zwischen den CPUs mit 8 Bit-µP (PCAØ2, PCA14 und PCA222) und denjenigen mit 16 Bit-µP (PCA231 und PCA232) leicht unterschiedlich, wie dies aus der nachfolgenden Tabelle hervorgeht. Ebenfalls wird daraus ersichtlich, dass für jede Befehlsgruppe die PID-Parameter zwischen "fix gemäss Anwenderprogramm" bis "alles variabel ab Zählerregister" gewählt werden kann. Dies ermöglicht es dem Anwender die Regel-Parameter vor jeder Veränderung zu schützen oder sie z.B. über BCD-Schalter veränderbar zu wählen.



PAS 2ØØ, 2Ø1, 2Ø2 — Für Baureihen PCAØ2 / PCA14 / PCA222
(Schneller PID-Algorithmus für 8 Bit Analogmodule)

Detailbeschreibung der Parameterbefehle PAS 2ØØ und 2Ø1

Uebersicht:				Kommentar		Zahlenbereich
1	PAS(29)	2ØØ	;	PID-Regelbefehl		2ØØ od. 2Ø1
2	ØØ	DBLK	;	Data Block	Nr.	31...Ø
3	ØØ	W	;	Sollwert	Wert	Ø...255*
4	ØØ	X, Y	;	Zähler-Adr. für Regel- bzw. Stellgrösse	ADR	256...319
5	XX	F_I	;	Integrationsfaktor ($T_Ø/T_I$)	Wert	Ø...32767
6	XX	F_D	;	Differentialfaktor ($T_D/T_Ø$)	Wert	Ø...32767
7	XX	F_P	;	Proportionalfaktor ($1/X_P$)	Wert	Ø...32767
8	ØØ	DR	;	Totzone	Wert	Ø...255
9	ØØ	Ø	}	Null		
1Ø	ØØ	Ø	}			

*) für PAS 2Ø1: Zähler-Adr. 256...319

- 1. Zeile

PAS 2ØØ: Der Sollwert W von Zeile 3 ist direkt zu programmieren.
PAS 2Ø1: Der Sollwert W von Zeile 3 wird dem entsprechenden Zähler entnommen.

- 2. Zeile Data Block

Es können max. 32 unabhängige PID-Regelkreise definiert werden. Damit die für die folgende Y-Berechnung notwendigen Werte $X_Ø$, $Y_Ø$ und $S_Ø$ in separaten Registern hinterlegt werden können, ist hier für jeden Regelkreis die Nr. des Data Blockes 31...Ø aufzuführen. Um die Zähler-Register (bis C479) nicht zu tangieren, wird mit Vorteil mit Data Block 31 begonnen (siehe "Organisation der Register" im Hardware-Manual).

- 3. Zeile Sollwert W

. Für PAS 2ØØ ist der 8-bitige Wert Ø...255 direkt einzusetzen (immediat).
. Für PAS 2Ø1 ist die entsprechende Zähler-Adr. (256...319) einzusetzen, von welcher der Wert (Ø...255) entnommen werden soll. PAS 2Ø1 eignet sich demnach für externe Sollwerteingabe.

- 4. Zeile Hilfszähler für X bzw. Y

Für das Einlesen der aktuellen Regelgrösse X1 und das Ausgeben der aktuellen Stellgrösse Y1 wird ein Hilfszähler (Adr. 256...319) benötigt. Siehe C26Ø im Beispiel unter "Programmaufbau". Dieser muss mit der Adresse unter Zeile 4 übereinstimmen.

- Zeilen 5...7 F_I, F_D, F_P

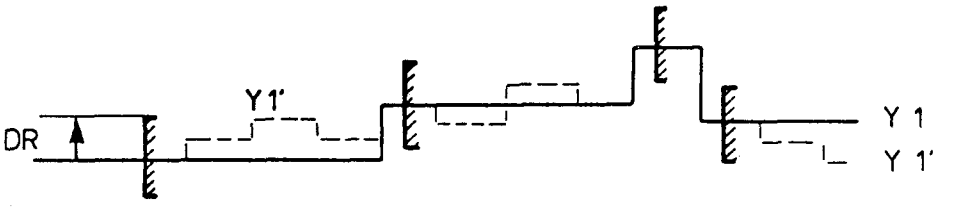
  In der Praxis bewegen sich diese Werte allgemein im Bereich von ca. Ø,1 bis 7,99. Um eine genügende Auflösung zu erreichen, muss der 256 fache Wert eingegeben werden.

Praxis-Wert	Eingabe-Wert im Operand	
Ø,1	25	
1,Ø	256	
7,99	2Ø45	
24,25	62Ø8	[C] ---> Ø 3 / ØØ64

  Auch Operanden, welche grösser sind als 2Ø47, können direkt eingegeben werden. Die PLC speichert das Modulo im Operand und das Vielfache von 2Ø48 im Code ab. Um den Wert in der einen oder anderen Art darzustellen, dient die Taste [C] (convert).

- Zeile 8 Totzone DR (Dead Range)

  Durch Festlegen einer Totzone werden kleine Schwankungen in der Stellgrösse Y ausgeglichen, wodurch Schwingungen verhindert werden können.



Bleibt der errechnete Wert Y1'innerhalb der Totzone DR, so erfolgt keine Veränderung der ausgegebenen Stellgrösse Y1.

DR wird als Wert zwischen Ø...255 in der gleichen Einheit wie der Sollwert W eingegeben. DR wirkt sowohl für Schwankungen nach oben wie unten.

Wichtig:

PAS 2ØØ, 2Ø1 und 2Ø2 werden unabhängig vom ACCU immer ausgeführt. Die Abarbeitungszeit des 1Ø-zeiligen Parameterbefehles beträgt ca. 2 ms.

Detailbeschreibung des Parameterbefehls PAS 2Ø2

Für das Austesten und Inbetriebnehmen eines Regelkreises kann es von Vorteil sein, wenn alle Parameter von extern eingegeben und während des Betriebes nach Wunsch verändert werden können. Diesem Bedürfnis entspricht PAS 2Ø2.

Uebersicht:

```
 1   PAS(29)    2Ø2   ;  PID-Regelbefehl
 2      ØØ     DBLK   ;  Data Block
 3      ØØ      FIC   ;  erste Adresse von 6 Zählern
                         für die Werte W, X/Y, FI, FD, FP, DR
 4      ØØ        Ø   }
 5      ØØ        Ø   }
 6      ØØ        Ø   }
 7      ØØ        Ø   }  Null
 8      ØØ        Ø   }
 9      ØØ        Ø   }
1Ø      ØØ        Ø   }
```

- Zeile 3 FIC (first counter)

  Steht für FIC z.B. die Adresse 26Ø, so erfolgt automatisch die folgende Zuordnung:

		Zahlenwerte für C	
C 26Ø	Sollwert W	Ø...255	
C 261	Hilfszähler für X1 bzw. Y1	Ø...255	
C 262	F_I Integrations-Faktor	Ø...32767	} *
C 263	F_D Differential-Faktor	Ø...32767	} *
C 264	F_P Proportional-Faktor	Ø...32767	} *
C 265	DR Totzone	Ø...255	

Die Zahlenwerte, welche vorgängig in die entsprechenden Zähler-Register einzulesen sind, haben die gleiche Bedeutung wie bei PAS 2ØØ und 2Ø1.

*) Um die Zahlenwerte zu erhalten, welche in die Zähler einzugeben sind, müssen die Praxiswerte für F_I, F_D und F_P mit 256 multipliziert werden.

Teil L
Beispiel 6

PAS 21Ø, 211, 212	Für Baureihen PCAØ2 / PCA14 / PCA222
PAS 2ØØ, 2Ø1, 2Ø2	Für Baureihen PCA231 / PCA232

Auflösung wählbar 8...15 Bit

Aus "historischen" Gründen wird für die gleiche Funktion bei den unterschiedlichen Baureihen nicht der gleiche Befehl verwendet. Das nach der PCA14 entwickelte Prozessormodul PCA2.M31 bzw. M32 wies von Anfang an eine wählbare Auflösung zur Berechnung der Stellgrösse Y auf. Mit der Einführung von Analogmodulen mit 12 Bit-Auflösung zur PCA1-Reihe musste auch bei der PCA14 ein entsprechender PAS-Befehl geschaffen werden. Um die Kompatibilität zu den bestehenden Programmen beizubehalten, musste der neue Befehl unterschiedlich zum bestehenden 8 Bit-Befehl benannt werden, nämlich PAS 21Ø etc.

Im folgenden werden nur die Befehle PAS 21Ø, 211, 212 für die Baureihen PCAØ2 / PCA14 / PCA222 erwähnt. Alle Ausführungen gelten aber ebenso für die Befehle PAS 2ØØ, 2Ø1, 2Ø2 für die Baureihen PCA231 / 232 (siehe auch Tabelle Seite 31H).

Unterschiede bestehen lediglich in 2 Punkten:

- Abarbeitungszeit für PAS 2ØØ...212 ← PCAØ2 / 14 / 222 : ca. 8 ms
  PCA231 / 232 : Ø,5 ms

  Trotz der 8 ms wird die serielle Schnittstelle jede 1ms bedient, sodass die Kommunikation bis 96ØØ Baud bezüglich diesem Befehl nicht beeinträchtigt wird.

- Die PCA232 stellt ab Firmware V7.12Ø einen erweiterten Algorithmus zur Verfügung, in welchem auch Sollwertsprünge rasch berücksichtigt werden (siehe Zeile 9 des PAS-Befehls).

Detailbeschreibung der Parameterbefehle PAS 21Ø und 211 für PCAØ2 / 14 / 222
(bzw. 2ØØ und 2Ø1 für PCA231 / PCA232)

Uebersicht				Kommentar	Zahlenbereich	
1	PAS(29)	21Ø	;	PID-Regelbefehl	21Ø od. 211	
2	n	DBLK	;	Auflösung n Bit und Data Block	Nr.	31... Ø
3	ØØ	W	1);	Sollwert	Wert	Ø... 4Ø95 2)3)
4	ØØ	X, Y	;	Zähler-Adr. für Regel- bzw. Stellgrösse	ADR	256... 319 2)
5	XX	F_I	;	Integrationsfaktor ($T_Ø/T_I$)	Wert	Ø...32767
6	XX	F_D	;	Differentialfaktor ($T_D/T_Ø$)	Wert	Ø...32767
7	XX	F_P	;	Proportionalfaktor ($1/X_P$)	Wert	Ø...32767
8	ØØ	DR	;	Totzone	Wert	Ø... 4Ø95 2)
9	ØX	YS	;	Zähler-Adr. für Kalt-startwert YS	ADR	256... 319 2)
1Ø	ØØ	Ø	;	Ø		

- Zeile 1:
  PAS 21Ø: Der Sollwert W von Zeile 3 ist direkt zu programmieren.
  PAS 211: Der Sollwert W von Zeile 3 wird dem entsprechenden Zähler entnommen.

- Zeile 2: Auflösung n Bit und Data Block
  Im Code wird mit n die Anzahl der zu behandelnden Bit für die Werte W, X, Y, YS und DR angegeben:

  n = Ø8 4) ≙ 8 Bit ---> Ø... 255
  12 5) ≙ 12 Bit ---> Ø... 4Ø95
  15 ≙ 15 Bit ---> Ø...32767

Es können max. 32 unabhängige PID-Regelkreise definiert werden. Damit die für die folgende Y-Berechnung notwendigen Werte $X_Ø$, $Y_Ø$ und $S_Ø$ in separaten Registern hinterlegt werden können, ist hier für jeden Regelkreis im Operand die Nr. des Data Blockes 31...Ø aufzuführen. Um die Zähler-Register bei der PCA14 (bis C479) nicht zu tangieren, wird mit Vorteil mit Data Block 31 begonnen (siehe "Organisation der Register" im Hardware-Manual).

- Zeile 3: Sollwert W
  . Für PAS 21Ø ist der 12-bitige Wert Ø...4Ø95 3) direkt einzusetzen (immediate).
  . Für PAS 211 ist die entsprechende Zähler-Adresse (256...319) einzusetzen, von welcher der Wert (Ø...4Ø95) 3) entnommen werden soll. PAS 211 eignet sich demnach für eine externe Sollwerteingabe.

- Zeile 4: Hilfszähler für X bzw. Y
  Für das Einlesen der aktuellen Regelgrösse X_1 und das Ausgeben der aktuellen Stellgrösse Y_1 wird ein Hilfszähler (Adresse 256...319) benötigt. Siehe C26Ø im Beispiel unter "Programmaufbau". Dieser muss mit der Adresse unter Zeile 4 übereinstimmen.

---

1) Für PAS 211: Zähler-Adr. 256...319
2) Die Werte im Operand bzw. im Zähler hängen von n in der Zeile 2 ab.
3) Gilt für n = 12 der Zeile 2.
4) Bei der PCA14 kann anstelle Ø8 auch ØØ eingegeben werden für 8 Bit.
5) Bei der PCA232 kann anstelle 12 auch ØØ eingegeben werden für 12 Bit.

- Zeile 5...7: F_I, F_D, F_P

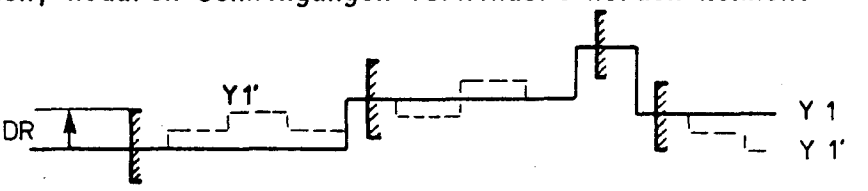
In der Praxis bewegen sich diese Werte allgemein im Bereich von ca. Ø,1 bis 7,99. Um eine genügende Auflösung zu erreichen, wird der 256-fache Wert eingegeben.

Praxis-Wert	Eingabe-Wert im Operand
Ø,1	25
1,Ø	256
7,99	2Ø45
24,25	62Ø8 [C] ---> Ø 3 / ØØ64

Auch Operanden, welche grösser sind als 2Ø47 können direkt eingegeben werden. Die PLC speichert das Modulo im Operand und das Vielfache von 2Ø48 im Code ab. Um den Wert in der einen oder anderen Art darzustellen, dient die Taste [C] (convert) des P1Ø/PØ5-Programmiergerätes.

- Zeile 8: Totzone DR (Dead Range)

Durch Festlegen einer Totzone werden kleine Schwankungen in der Stellgrösse Y ausgeglichen, wodurch Schwingungen verhindert werden können.



Bleibt der errechnete Wert Y_1' innerhalb der Totzone DR, so verfolgt keine Veränderung der ausgegebenen Stellgrösse Y1.

DR wird als Wert zwischen Ø...4Ø95 in der gleichen Einheit wie der Sollwert W eingegeben. DR wirkt sowohl für Schwankungen nach oben wie nach unten.

- Zeile 9: Kaltstart-Wert YS (Anfangs-Stellgrösse)

Da die Datablöcke 31...Ø nullspannungssicher sind, müssen die entsprechenden Werte für die erste Benützung (oder für eine andere Regelstrecke) vorerst auf einen gewissen Anfangswert gesetzt werden.

Wird der im Operand der Zeile 9 adressierte Zähler CYS mit dem Wert YS geladen, so werden für den Kaltstart folgende Werte verwendet (und im Data Block abgelegt):

Data Block: $X_Ø = X$; $Y_Ø = YS$; $SØ = \dfrac{YS/F_P - (W - X)}{F_I}$

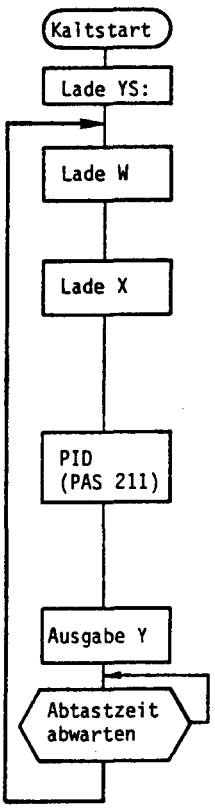
Stellgrösse: $Y = YS$

Nach dem ersten Abarbeiten des PAS wird der Zähler CYS automatisch auf Ø gesetzt und beim nächsten Durchlauf nicht mehr berücksichtigt. Ausser dem Kaltstart können mit dem Zähler CYS auch sogenannte stossfreie Uebergänge von Hand- auf Automatik-Betrieb oder bei Aenderung der Regelparameter erzielt werden (siehe Programmbeispiel 7).

Im Code der Zeile 9 wird mit ØØ der normale Algorithmus zur Berechnung der Stellgrösse Y aufgerufen. Wird dort Ø1 gesetzt, so verwendet die PCA232 (ab V7.12Ø) die erweiterte Formel mit den Sollwertsprüngen (gemäss Seite 29H).

Da die Behandlung des Analogmoduls der Reihe PCA2 einfacher ist als das entsprechende Modul der PCA1-Reihe, wird hier mit CPU PCA2.M22 und Analogmodul PCA2.W14 auf Basisadresse 176 gearbeitet.

Ein Kaltstart-Regelungsprogramm ist demnach wie folgt aufgebaut:



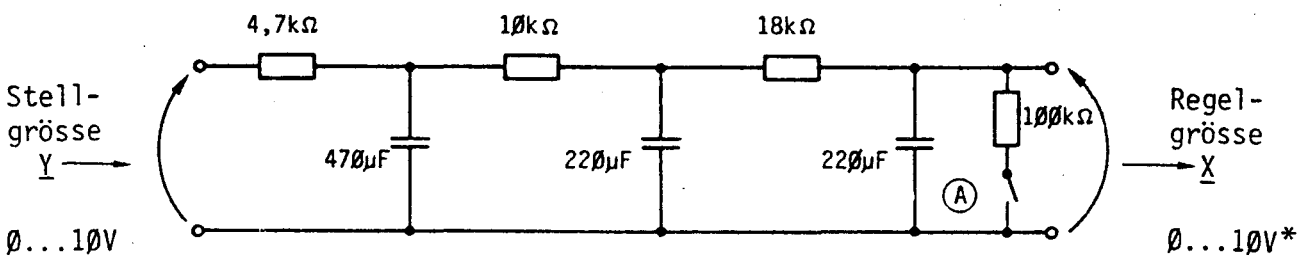

```
100  WIL   0
101  SCR   282        Kaltstart-Wert YS:
102   00   1024       z.B. 1/4 von Y max.
103  SEO   176
104  SEO   184        E-Kanal 176 (0): W
105  SCR   276        (als Analogwert)
106   25   187
107  SEO   177
108  SEO   184        E- Kanal 177 (1): X
109  SCR   277
110   25   187
111  PAS   211
112   12    31
113   00   276
114   00   277        PAS 211
115    FD              C276: W
116    FI              C277: X/Y
117    FP              C282: YS
118    DR
119   00    282
120   00   0000
121  SCR   277
122   22   187        A-Kanal 191 (15): Y
123  SEO   191
124  STR   256
125   00    02        T0 = 0,2s
126  WIH   256
127  JMP   103
```

Basisadresse		0		176	
<-- A	15	o		191	(Y) <-- C277
	.	o		.	
	.	o		.	
	.	o		.	
A	12	o		188	
E	7	o		183	
	.	o		.	
	.	o		.	
	.	o		.	
	.	o		.	
	.	o		.	
--> E	1	o		177	(X) --> C277
--> E	0	o		176	(W) --> C276
					YS --> C282

Adressierung des Analog-Moduls PCA2.W14

Regelungs-Beispiel: Mit obigem Programm soll die nachfolgend beschriebene Regelstrecke (ein Simulations-Schaltkreis) PID geregelt werden.

Simulierte Regelstrecke:



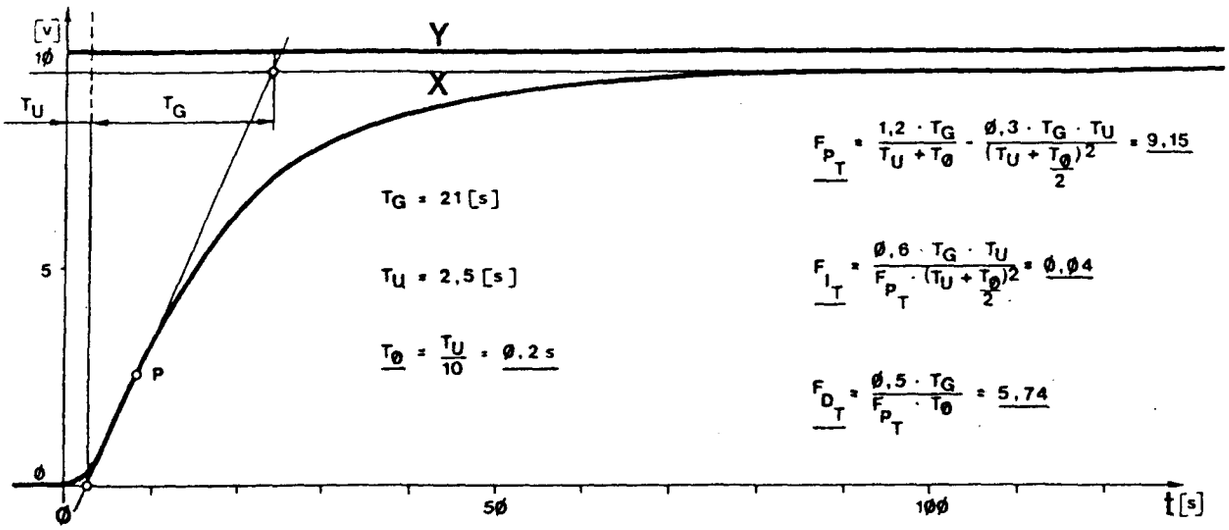
Schliessen des Schalters (A) bewirkt eine Störgrösse von ca. 20%.

Mit Hilfe der Sprungantwort (Y wechselt von 0 ---> 10V) können die Parameter der Regelstrecke F_P , F_I , F_D und T_0 gemäss den Formeln von Takahashi bestimmt werden.

---

*) Das Messinstrument für X muss Eingangswiderstand von min. 1MΩ aufweisen.

Auswertung der Sprungantwort einer Regelstrecke



Für die Eingabe in PAS 211 müssen die oben berechneten Werte mit 256 multipliziert werden:

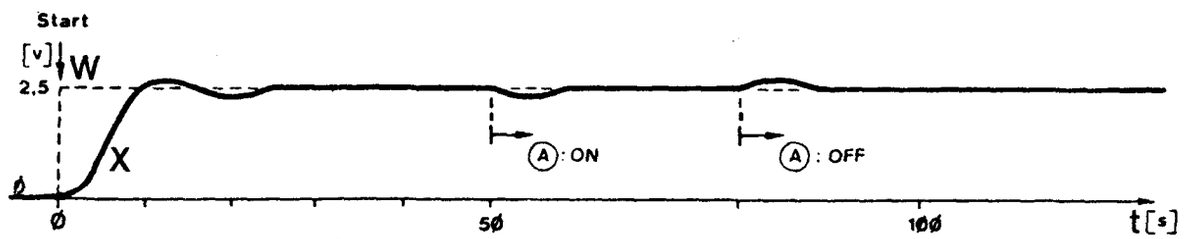
$F_I = F_{I_T} \times 256 = 10$

$F_D = F_{D_T} \times 256 = 1469$

$F_P = F_{P_T} \times 256 = 2341$

```
PAS     211
12       31   ;  12 Bit, Data Block 31
00      276   ;  Zähler für W
00      277   ;  Zähler für X/Y
00       10   ;  FI
00     1469   ;  FD
01      293   ;  FP ( C = E E 2341)
00       00   ;  DR
00      282   ;  Zähler für YS
00       00   ;  --
```

Verhalten der Regelstrecke mit obigen Parametern für Sollwert W = 1024 (2,5V):



(A) : Simulierte Störgrösse (≈20%).

Detailbeschreibung des Parameterbefehls PAS 212

Für das Austesten und Inbetriebnehmen eines Regelkreises kann es von Vorteil sein, wenn alle Parameter von extern eingegeben und während des Betriebes nach Wunsch verändert werden können. Diesem Bedürfnis entspricht PAS 212 (PAS 2Ø2 für PCA231/232).

Uebersicht:

1	PAS(29)	212	;	PID-Regelbefehl
2	n	DBLK	;	Auflösung n Bit und Data Block
3	ØØ	FIC	;	erste Adresse von 7 Zählern für die Werte W, X/Y, F_I, F_D, F_P, DR, YS
4	ØØ	Ø		Null
5	ØØ	Ø		
6	ØØ	Ø		
7	ØØ	Ø		
8	ØØ	Ø		
9	ØØ	Ø		
1Ø	ØØ	Ø		

- Zeile 2: siehe PAS 21Ø

- Zeile 3: FIC (First Counter)

  Steht für FIC z.B. die Adresse 276 so erfolgt automatisch die folgende Zuordnung:

		Zahlenwerte für C	
C 276	Sollwert W	Ø... 4Ø95	2)
C 277	Hilfszähler für X1 bzw. Y1	Ø... 4Ø95	2)
C 278	F_I Integrations-Faktor	Ø...32767	1)
C 279	F_D Differential-Faktor	Ø...32767	1)
C 28Ø	F_P Proportional-Faktor	Ø...32767	1)
C 281	DR Totzone	Ø... 4Ø95	2)
C 282	YS Kaltstart-Stellgrösse	Ø... 4Ø95	2)

  Die Zahlenwerte, welche vorgängig in die entsprechenden Zähler-Register einzulesen sind, haben die gleiche Bedeutung wie bei PAS 21Ø und 211.

  Wichtig:

  Alle PAS 2ØØ...212 werden unabhängig vom ACCU immer ausgeführt.

---

1) Um die Zahlenwerte zu erhalten, welche in die Zähler einzugeben sind, müssen die Praxiswerte für F_I, F_D und F_P mit 256 multipliziert werden.

2) Werte für Analog-Module mit 12 Bit Auflösung (n).

PAS 25Ø/251 Rotations-, Schiebe- und Stapelregister FIFO

PAS 25Ø Rotations- und Schieberegister

Mit dem zehnzeiligen Befehl PAS 25Ø kann ein Rotations- oder Schieberegister im Bit-Prozessor definiert und betrieben werden, das vorzugsweise dem Merkerbereich - selbstverständlich kann ein Register auch mit lesbaren Ausgängen definiert und betrieben werden - zugeordnet wird. Der PAS-Befehl ist wie folgt aufgebaut:

```
 1   PAS   (29)    25Ø   ;  Aktivierung der Parameterfunktion
 2         ØØ      SADD  ;  Anfangsadresse des Registers SADD   1)
 3         ØØ      DADD  ;  Endadresse des Registers DADD       1)
 4         ØØ      NNN   ;  Breite des Registers N Bit          2)
 5         ØØ      Ø     }
 6         ØØ      Ø     }
 7         ØØ      Ø     }  Null
 8         ØØ      Ø     }
 9         ØØ      Ø     }
1Ø         ØØ      Ø     }
```

Je nachdem, ob SADD oder DADD auf der tieferen Adresse steht, wird das Rotations- oder Schieberegister vorwärts oder rückwärts geschoben. Die Parameterfunktion PAS 25Ø wird unabhängig vom ACCU-Zustand ausgeführt und beeinflusst den ACCU auch nicht.

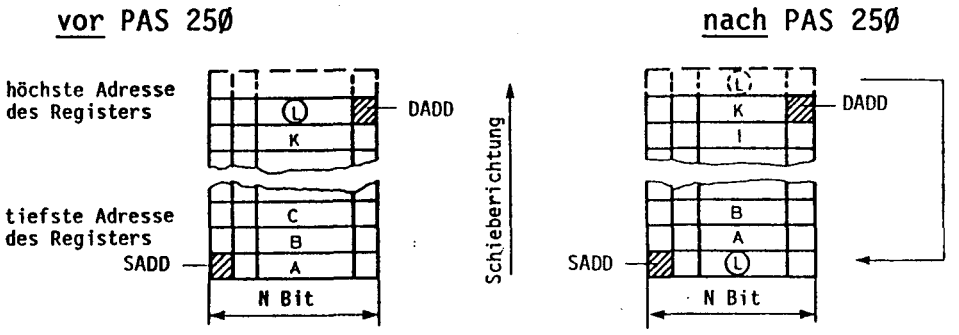
---

1) Sowohl SADD (Source Address) als auch DADD (Destination Address) sind indexierbar.

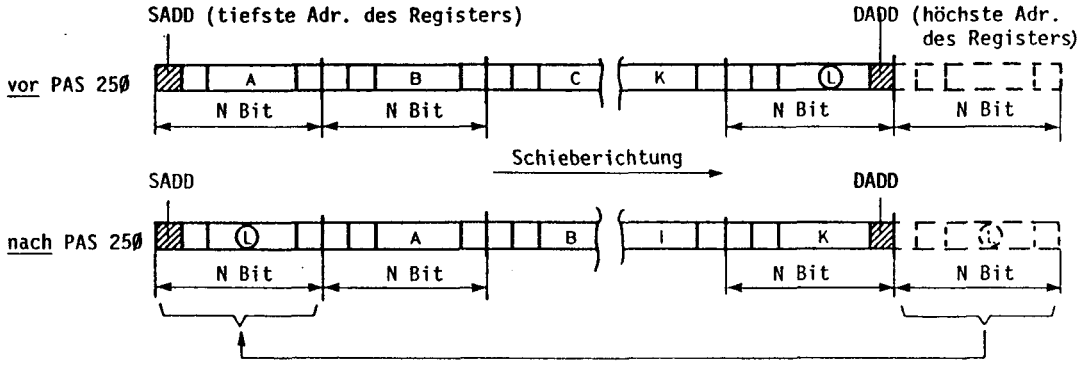
2) Die Breite des Registers kann 1...255 Bit betragen.
NNN = 1...255: Fester Wert
NNN = 256...511: Die Information N Bit befindet sich im entsprechenden Zähler.

Funktionsweise

a) SADD < DADD ≙ Schieben vorwärts

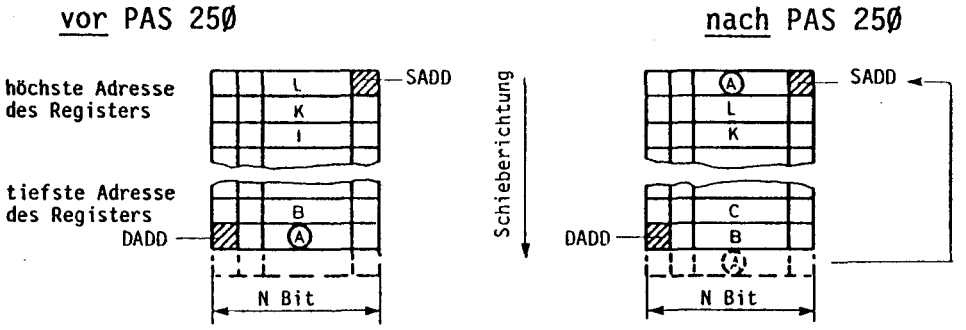


Dasselbe Register anders dargestellt:



Die Darstellungen zeigen, dass ab der Adresse DADD die nächstfolgenden Adressen von N Bit reserviert werden müssen, da eine zusätzliche Registerebene als Umlaufpuffer verwendet wird.

b) SADD > DADD ≙ Schieben rückwärts



Die Darstellungen zeigen, dass vor der Adresse DADD die unmittelbar davorliegenden Adressen von N Bit reserviert werden müssen.

Wie aus den Figuren ersichtlich ist, behandelt der Befehl PAS 25Ø ein Rotationsregister. Soll daraus ein lineares Schieberegister gemacht werden, so muss im Fall a) (schieben vorwärts) dafür gesorgt werden, dass die Informationen auf der Ebene DADD entnommen werden. Nach dem Schieben ist die neu einzugebende Information auf die Ebene SADD zu übertragen.

Teil L
Beispiel 8 und 9

PAS 251 Eingabe in das FIFO-Register (First in - First out)

Mit dem zehnzeiligen Befehl PAS 251 können Informationen von N Bit in ein FIFO-Register geschrieben und bis zum letzten freien Platz geschoben werden.

Die Grösse des FIFOs wird mit der Anfangsadresse SADD und der Endadresse DADD definiert. Der PAS-Befehl ist wie folgt aufgebaut:

```
 1   PAS    (29)    251   ;  Aktivierung der Parameterfunktion
 2          ØØ      SADD  ;  Anfangsadresse des Register SADD   1)
 3          ØØ      DADD  ;  Endadresse des Registers DADD      1)
 4          ØØ      NNN   ;  Breite des Registers N Bit         2)
 5          ØØ      Ø     }
 6          ØØ      Ø     }
 7          ØØ      Ø     }  Null
 8          ØØ      Ø     }
 9          ØØ      Ø     }
10          ØØ      Ø     }
```

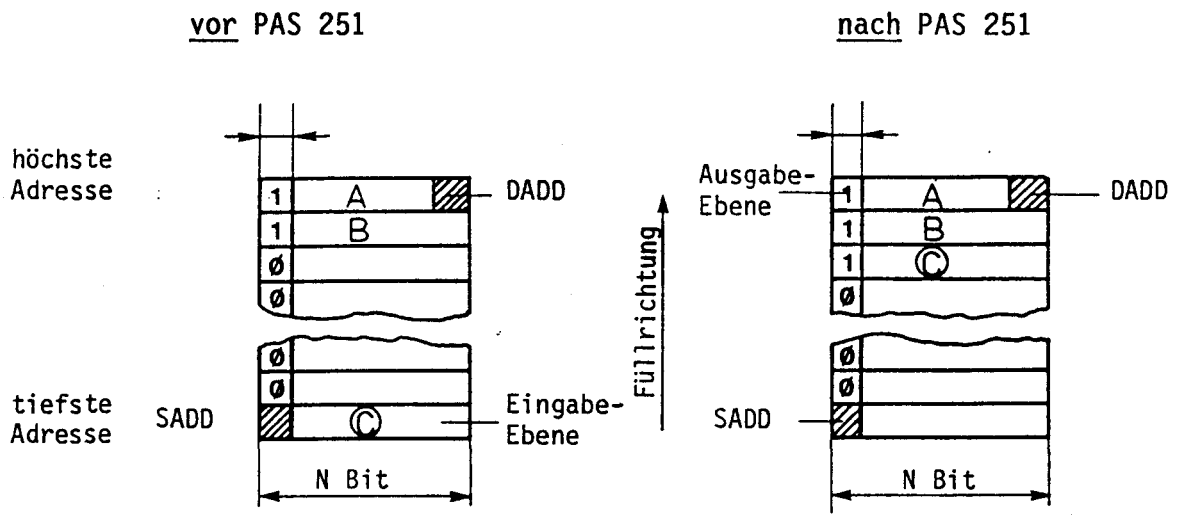
SADD muss immer auf der tieferen Adresse stehen als DADD. Die Parameterfunktion wird unabhängig vom ACCU-Zustand ausgeführt. PAS 251 beeinflusst jedoch den ACCU (siehe übernächste Seite).

---

1) Sowohl SADD (Source Address) als auch DADD (Destination Address) sind indexierbar.

2) Die Breite des Registers kann 1...255 Bit betragen.

Funktionsweise

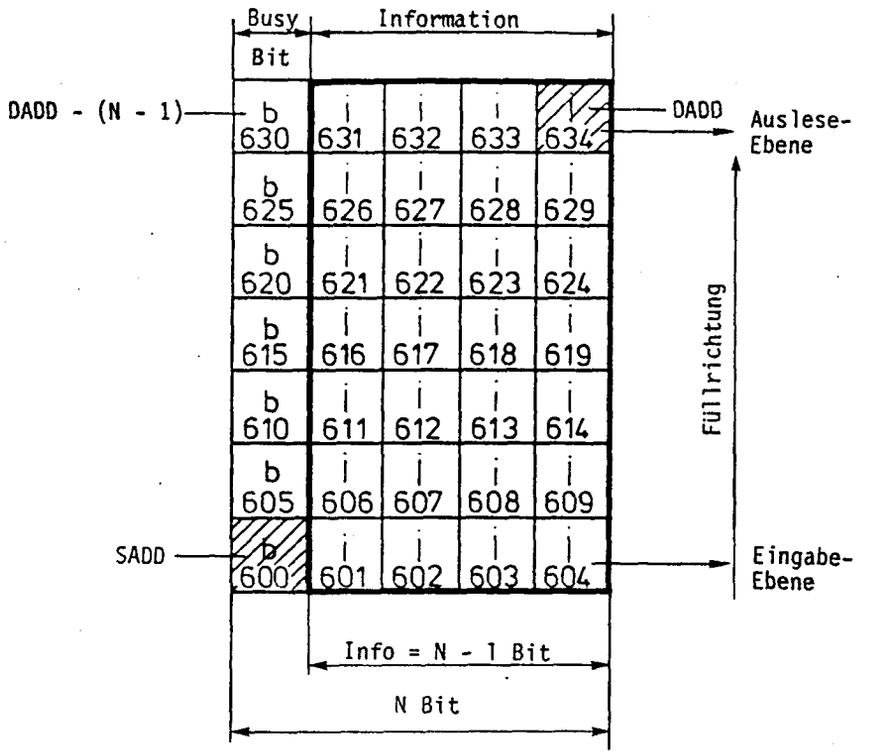


Obige Darstellungen zeigen, dass im FIFO-Register ein neuer Wert von N Bit auf den letzten freien Platz in der Richtung SADD ---> DADD abgelegt wird.

Damit innerhalb des FIFOs nichts verloren geht, muss man sich vor jeder Eingabe eines neuen Wertes versichern, dass das FIFO noch nicht voll ist. Wird dies missachtet, so wird der zuletzt eingegebene Wert mit dem neuen Wert überschrieben. Um diese Prüfung durchführen zu können, wird vor jeder Registerebene ein "Busy"-Bit reserviert. Dieses Bit ist "H", wenn die entsprechende Ebene besetzt ist.

Es genügt nun, vor jeder neuen Eingabe durch PAS 251 das "Busy"-Bit der Eingabe-Ebene zu testen und dann, abhängig vom Resultat, die richtige Entscheidung zu treffen.

Das FIFO-Register sieht demnach als Beispiel wie folgt aus:



SADD = 6ØØ
DADD = 634
N = 5 Bit
Information = 4 Bit

Vorgehen:

a) Wenn der zuletzt eingegebene Wert nicht gelöscht werden soll, ist vorgängig zu prüfen, ob SADD (Busy-Bit) "H" ist.

   SADD = H : FIFO voll
   SADD = L : freie Plätze im FIFO

b) Der Anwender gibt nun auf die Eingabe-Ebene eine Information von (N-1) Bit ein und setzt das Bit "SADD" (Busy-Bit) auf "H".

c) PAS 251 abarbeiten.
   Nach dem PAS 251 wird SADD automatisch auf "L" gesetzt, sofern das FIFO noch nicht voll ist. Es bleibt auf "H", wenn alle Registerebenen besetzt sind.

d) Behandlung des Resultates nach PAS 251:

   ACCU = 1: Keine Fehler festgestellt.

   ACCU = Ø: Fehler in der Definierung des FIFO, PAS 251 wurde nicht ausgeführt.

   SADD = H: FIFO voll. Durch die Eingabe eines weiteren Wertes wird der zuletzt eingegebene Wert überschrieben.

   SADD = L: FIFO frei, d.h. freie Plätze für weitere Information(en).

   DADD - (N-1) = L: FIFO leer.

Notizen:

TEIL I BEFEHLE DES WORTPROZESSORS STUFE 3 (nur PCA232 oder PCA231)

TEIL I BEFEHLE DES WORTPROZESSORS STUFE 3 (nur PCA232 oder PCA231)

I 1 Wechsel zum Wort-Prozessor und Befehlssatz des Wortprozessors

NOP 1248 Wechsel von Bit- nach Wort-Prozessor

Mit den 5 Bit, welche für den Befehlscode zur Verfügung stehen, können lediglich 32 Instruktionen definiert werden. Mit dem Umschalt-Kennwort NOP 1248 kann auf weitere 32 Befehle des Wortprozessors umgeschaltet werden.

Beispiel:

```
        :
STH      3
ANH      4
ORH     1Ø
JIZ
```
Programmteil im Bit-Prozessor

Umschaltbefehle für Wechsel nach Wortprozessor

```
NOP  W 1248
```

```
LAC  W  256
ADD  W  13Ø
EXG  W   Ø
SAC  W  256
```
Programmteil im Wortprozessor

```
EWP  W   Ø
```
Ende Wortprozessor

```
SCR     256
 22     311
        :
```
Programmteil im Bit-Prozessor

NOP 1248 wird unabhängig vom ACCU immer ausgeführt.

NOP 1248 setzt den ACCU = 1

Gewisse Wortprozessor-Befehle setzen den ACCU = Ø, wenn z.B. Kapazitätsgrenzen überschritten werden oder bei Division durch Null.
Durch Kontrolle des ACCU-Zustandes nach EWP kann festgestellt werden, ob die zulässigen Grenzen eingehalten wurden.

```
NOP  W 1248    - Umschaltbefehl nach Wortprozessor, ACCU=1

LAC  W  256      C256 in AØ
EXG  W   Ø       C256 in A1
CLA  W   Ø       AØ = Ø
DIV  W   Ø       A1 : Ø ---> ACCU = Ø

EWP  W   Ø     - Ende Wortprozessor

JIZ              Sprung in Fehlermelde-Routine
```

Hinweis: Von NOP 1248 bis JIZ erfolgt kein Wechsel des Parallelprogrammes. Dies muss bei zeitkritischen PP berücksichtigt werden, indem die Wortprozessor-Programme dann in kürzere Teilstücke aufgeteilt werden.

Befehlssatz des Wortprozessors

Zahlen-code	Mnemo-code	Operand	Befehl Englisch	Beschreibung	Daten-format	
ØØ	NOP	1248	---	Wechsel von Bit- nach Wortprozessor setzt ACCU = 1		

Datentransfer-Befehle

Ø1	RRG	Rn	Read Register	Lese Wort Rn und speichere es in R4	8 Bit	i
Ø2	WRG	Rn	Write into Register	Schreibe in Rn das Wort von R4	8 Bit	i
Ø3	RRE	Rn	Read Register and write in Elements	Lese Wort Rn und speichere es in den 8 Elementen En...En-7 adressiert durch A1	8 Bit	i
Ø4	WRE	Rn	Write Elements into Register	Schreibe in Rn den Inhalt der 8 Elemente En...En-7 adressiert durch A1	8 Bit	i
Ø5	LAR	Rn	Load AØ with Registers	Lade AØ mit Registerblock Rn	5x8 Bit	i
Ø6	SAR	Rn	Store AØ in Registers	Speichere AØ in den Registerblock Rn	5x8 Bit	i
Ø7	LAC	Cn	Load AØ with Counter	Lade AØ mit Zähler Cn	BCD	i
Ø8	SAC	Cn	Store AØ in Counter	Speichere AØ in den Zähler Cn	BCD	i
Ø9	WEL	En	Write into elements { Lower / Upper } digit	Schreibe { R4 (10^0) / das Digit R4 (10^1) } in die Elemente En...En-3	4 Bit	i
1Ø	WEU	En				

Zähl- und Sprungbefehle

11	INR	Rn	Increment Register	Incr. / Decr. } BCD-Wert Rn um 1 und setze "carry" { > 99 / < Ø falls Resultat		i
12	DER	Rn	Decrement Register			i
13	SNC	Ø En	Skip if no "carry" Skip if En = Ø	Ueberspringe nächsten Befehl, wenn { "carry" = Ø / En = Ø		i
24	SEW	Ø En	Skip to EWP if no "carry" Skip to EWP if En = Ø	Ueberspringe bis EWP oder NOP 1248, wenn { "carry" = Ø / En = Ø		i

(i) = indexierbar

Behandlung der Arithmetik-Register

14	CLA	Ø,1,2	Clear A	Lösche Register AØ oder A1 oder A2		
15	LAI	K	Load A immediately	wenn K ≤ 99 ---> Lade R4 mit Data wenn K = 1ØØ...2Ø47 ---> Lade AØ mit Data		
16	DBN	Ø	Decimal to Binary	Wandle AØ dezimal in AØ binär		
17	BND	Ø	Binary to Decimal	Wandle AØ binär in AØ dezimal		
18	ROR	Ø, 1	Rotate Register	Rotiere R4...RØ, bzw. RØ...R4		
19	ROA	Ø, 1	Rotate A	Rotiere AØ...A2, bzw. A2...AØ		
2Ø	EXG	Ø	Exchange A1 with AØ	Tausche A1 mit AØ		
		Rn	Exchange A1 with Rn...Rn-4	Tasuche A1 mit Rn...Rn-4		i

Artihmetik-Befehle

25	CMP	En	Compare AØ with A1	Vergleiche AØ mit A1 A1 > AØ ---> En = 1 A1 = AØ ---> En-1 = 1 A1 < AØ ---> En-2 = 1	BCD	i
26	SQR	Ø	Square Root from A1	$\sqrt{A1}$ ---> A1, nur ganzzahlig	BCD	
27	ADD	Ø	Add AØ to A1	A1 + AØ ---> A1, "carry"	BCD	
		K	Add K to A1	A1 + K ---> A1, "carry" (K=1...2Ø47)	BCD	
28	SUB	Ø	Subtract AØ from A1	A1 - AØ ---> A1, "carry"	BCD	
		K	Subtract K from A1	A1 - K ---> A1, "carry" (K=1...2Ø47)	BCD	
29	MUL	Ø	Multiply A1 by AØ	A1 . AØ ---> A1, "carry"	BCD	
		K	Multiply A1 by K	A1 . K ---> A1, "carry" (K=1...2Ø47)	BCD	
3Ø	DIV	Ø	Divide A1 by AØ	A1 : AØ ---> A1, Rest in AØ, "carry"	BCD	
		K	Divide A1 by K	A1 : K ---> A1, Rest in AØ, "carry" (K=1...2Ø47)	BCD	

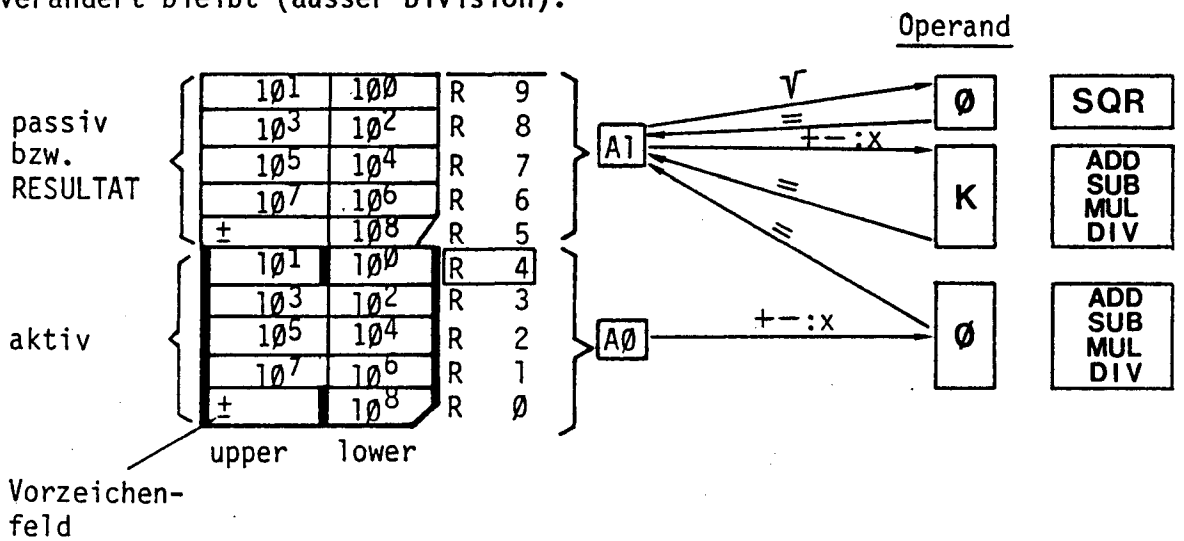
Diverse Befehle

ØØ	NOP	Ø	No operation	Keine Operation		
21	CLK	En	Clock source	Zuweisung einer Zeitimpulsquelle		
22	SHI	Rn	Shift registers	Schiebe Wörter ab R2Ø bis Rn um 1 nach oben		i
23	TXT	Txn	Text	Start der Textausgabe		i
31	EWP	Ø	End word processor	Ende der Arbeit im Wortprozessor		

(i) = indexierbar

I 2 Arithmetik-Befehle

Die Arithmetik-Operationen werden ausschliesslich zwischen den A-Registern A1 und AØ bzw. einer Konstanten K im Operand ausgeführt. A1 enthält dabei den passiven Operator. In A1 wird das Resultat der Operation abgelegt, während AØ unverändert bleibt (ausser Division).



Für die arithmetischen Operationen müssen die Werte im BCD-Format bereit stehen. Es werden ganzzahlige positive und negative BCD-Werte verarbeitet bis zur Kapazität ±1Ø⁹-1 (±999'999'999). Das Vorzeichen ist dabei vor der Dezimale 1Ø⁸ ins Vorzeichenfeld plaziert mit folgender Bedeutung:

Ø = positiv +
9 = negativ -

Wird ein Registerblock gelesen, so werden alle von Ø abweichenden Bitmuster im Vorzeichenfeld als negativ bewertet.

Für den Datentransfer können beliebige Bitmuster von 1Øx4 Bit verwendet werden.

ADD Addition

ADD: Add AØ (or K) to A1

Mnemo-code	Zahlen-code	Operand	Operation
ADD	27	Ø	A1 + AØ ---> A1
		K = 1...2Ø47	A1 + K ---> A1

Rechenkapazität: ±999'999'999 bzw. ±(1Ø⁹-1)
Bei "overflow" wird "carry" = 1 und ACCU = Ø. In A1 steht dann lediglich der Rest.

SUB Subtraktion

SUB: Subtract AØ (or K) from A1

Mnemo-code	Zahlen-code	Operand	Operation
SUB	28	Ø	A1 - AØ ---> A1
		K = 1...2Ø47	A1 - K ---> A1

Rechenkapazität: ±999'999'999 bzw. ±(1Ø⁹-1)

Bei "overflow" wird "carry" = 1 und ACCU = Ø. In A1 steht dann lediglich der Rest.

MUL Multiplikation

MUL: Multiply A1 by AØ (or by K)

Mnemo-code	Zahlen-code	Operand	Operation
MUL	29	Ø	A1 x AØ ---> A1
		K = 1...2Ø47	A1 x K ---> A1

Kapazitätsgrenze für das Resultat: ±999'999'999 bzw. ±(1Ø⁹-1)

Bei "overflow" wird "carry" = 1 und ACCU = Ø. In A1 steht der Maximalwert ±999'999'999.

DIV Division

DIV: Divide A1 by AØ (or by K)

Mnemo-code	Zahlen-code	Operand	Operation
DIV	3Ø	Ø	A1 : AØ ---> A1
		K = 1...2Ø47	A1 : K ---> A1

Ein Divisions-Rest wird in AØ abgelegt.
Der "carry" wird = 1 und der ACCU = Ø gesetzt bei Division durch Ø. Die Operation wird dann nicht ausgeführt und alle Daten bleiben unverändert.
Kapazitätsgrenze für das Resultat: ±999'999'999.

SQR Quadratwurzel

SQR: Square Root from A1

Mnemo-code	Zahlen-code	Operand	Operation
SQR	26	Ø	$\sqrt{A1}$ ---> A1

Kapazitätsgrenze $\sqrt{+999'999'999}$.
Das Resultat wird nur ganzzahlig ausgegeben (z.B. $\sqrt{3}$ = 1). Der Rest geht dabei verloren.
Höhere Genauigkeit wird erreicht, indem der Radient zuerst um ein Vielfaches von 1ØØ multipliziert wird (z.B. $\sqrt{3'ØØ'ØØ}$ = 173).

Wird die Quadratwurzel aus einer negativen Zahl gezogen, so wird der "carry" = 1 und der ACCU = Ø. In A1 steht dann der unveränderte negative Radient.

CMP Vergleiche die Zahlenwerte A1 mit AØ

CMP: Compare A1 with AØ (A1 - AØ)

Mnemo-code	Zahlen-code	Operand	Operation
CMP	25	Elementadresse En = 2...999(i)	A1 > AØ ---> En =1, carry=Ø A1 = AØ ---> En-1=1, carry=Ø A1 < AØ ---> En-2=1, carry=1

(i) = indexierbar

Das Ergebnis der Vergleichsoperation wird direkt auf 3 Elementadressen (Merker oder Ausgänge) ausgegeben, sodass die entsprechende Information dem Prozess direkt zur Verfügung steht. Durch die Operation werden die Register AØ und A1 nicht verändert.

Der Vergleich erfolgt unter Berücksichtigung des Vorzeichens bis ±999'999'999 bzw. ±($1Ø^9$-1). Dabei ist z.B. +3 grösser als -5ØØ.

I 3 Befehle zur Behandlung der A-Register

CLA Lösche Register AØ, A1 oder A2

CLA: Clear A

Mnemo-code	Zahlen-code	Operand	Beschreibung
CLA	14	Ø oder 1 oder 2	Lösche AØ, A1 oder A2

LAI Lade Register AØ mit Konstante K

LAI: Load AØ immediately

Mnemo-code	Zahlen-code	Operand	Beschreibung
LAI	15	K (Ø...2Ø47)	K = Ø...99 ---> nur R4 wird geladen K = 1ØØ...2Ø47 ---> ganzer AØ wird geladen

Beträgt die Konstante z.B. 57, so wird nur R4 geladen, RØ bis R3 bleiben unverändert.
Beträgt die Konstante aber z.B. 225, so wird das ganze AØ geladen, d.h. die 7 Stellen vor der Zahl 225 werden mit Nullen aufgefüllt.

DBN, BND Umwandlung dezimal ---> binär bzw. binär ---> dezimal

DBN, BND: Decimal to binary, binary to decimal

Mnemo-code	Zahlen-code	Operand	Beschreibung
DBN	16	Ø	Wandle AØ dezimal in AØ binär (Kapazitätsgrenze ±999'999'999)
BND	17	Ø	Wandle AØ binär in AØ dezimal (Kapazitätsgrenze R4 bis R1)

Alle arithmetischen Operationen werden nur mit Zahlen im BCD-Format korrekt ausgeführt. Werden z.B. ab Analog-Eingängen Werte in reiner Binärform geladen, so müssen diese vor der weiteren Verarbeitung zuerst in BCD-Form umgewandelt werden.
Umgekehrt steht der Fall z.B. für die Daten-Ausgabe auf Analog-Ausgänge.

ROR Rotiere Register

ROR: Rotate Registers

Mnemo-code	Zahlen-code	Operand	Beschreibung
ROR	18	Ø	Rotiere R4->R3->R2->R1->RØ->R4
		1	Rotiere RØ->R1->R2->R3->R4->RØ

Die Wirkungsweise dieser Befehle ist ersichtlich aus der Uebersicht zu Beginn des folgenden Kapitels.

ROA Rotiere A

ROA: Rotate A

Mnemo-code	Zahlen-code	Operand	Beschreibung
ROA	19	Ø	Rotiere AØ -> A1 -> A2 -> AØ
		1	Rotiere A2 -> A1 -> AØ -> A2

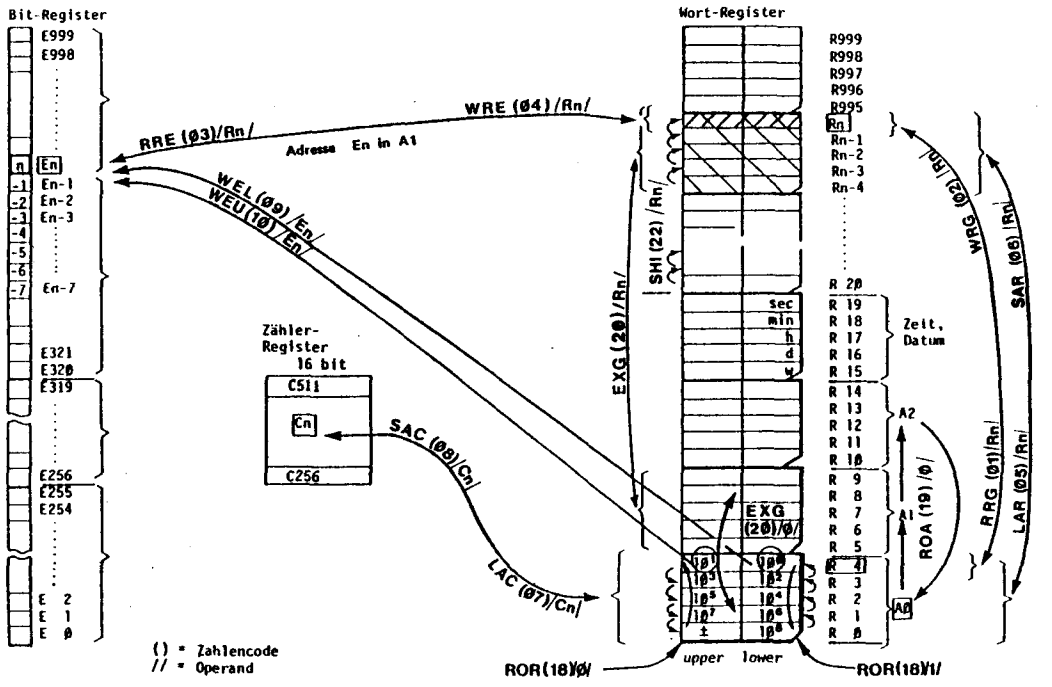
EXG Tausche A1 mit AØ oder mit Registerblock Rn-4...Rn

EXG: Exchange A

Mnemo-code	Zahlen-code	Operand	Beschreibung
EXG	2Ø	Ø	Tausche A1 mit AØ
		Rn von 14...999 (i)	Tausche A1 mit Rn-4...Rn R9 <---> Rn R8 <---> Rn-1 R7 <---> Rn-2 R6 <---> Rn-3 R5 <---> Rn-4

(i) = indexierbar

I 4 Uebersicht der Transfer-Befehle



Zahlen-code	Mnemo-code	Operand	Befehl Englisch	Beschreibung	Datenformat	

Datentransfer-Befehle

Ø1	RRG	Rn	Read Register	Lese Wort Rn und speichere es in R4	8 Bit	i
Ø2	WRG	Rn	Write into Register	Schreibe in Rn das Wort von R4	8 Bit	i
Ø3	RRE	Rn	Read Register and write in Elements	Lese Wort Rn und speichere es in den 8 Elementen En...En-7 adressiert durch A1	8 Bit	i
Ø4	WRE	Rn	Write Elements into Register	Schreibe in Rn den Inhalt der 8 Elemente En...En-7 adressiert durch A1	8 Bit	i
Ø5	LAR	Rn	Load AØ with Registers	Lade AØ mit Registerblock Rn	5x8 Bit	i
Ø6	SAR	Rn	Store AØ in Registers	Speichere AØ in den Registerblock Rn	5x8 Bit	i
Ø7	LAC	Cn	Load AØ with Counter	Lade AØ mit Zähler Cn	BCD	i
Ø8	SAC	Cn	Store AØ in Counter	Speichere AØ in den Zähler Cn	BCD	i
Ø9	WEL	En	Write into {Lower} digit	Schreibe {R4 ($1Ø^Ø$)} in die Ele-		
1Ø	WEU	En	elements {Upper}	das Digit {R4 ($1Ø^1$)} mente En...En-3	4 Bit	i

Behandlung der Arithmetik-Register

18	ROR	Ø, 1	Rotate Register	Rotiere R4...RØ, bzw. RØ...R4		
19	ROA	Ø, 1	Rotate A	Rotiere AØ...A2, bzw. A2...AØ		
2Ø	EXG	Ø	Exchange A1 with AØ	Tausche A1 mit AØ		
		Rn	Exchange A1 with Rn...Rn-4	Tausche A1 mit Rn...Rn-4		i

(i) = indexierbar

Auf der vorhergehenden Uebersicht sind die Wirkungsweisen all dieser Befehle anschaulich dargestellt. Datentransfer ist demnach nicht nur innerhalb des Wort-Registers, sondern auch zum Zähler- und Bit-Register möglich.
Es stehen verschiedene Befehle mit Wortlängen von 4, 8 oder 4Ø Bit zur Verfügung. Im allgemeinen wird es sich dabei um Daten im BCD-Format handeln. Rein binäre Daten können aber, mit Ausnahme des Zähler-Registers, zwischen allen übrigen Registern ausgetauscht werden. Zur arithmetischen Verarbeitung müssen die zu behandelnden Daten jedoch BCD-Format aufweisen (siehe DBN bzw. BND).

RRG, WRG Datentransfer zwischen Rn und R4

RRG: Read Register (Rn)

WRG: Write into Register (Rn)

Mnemo-code	Zahlen-code	Operand	Beschreibung
RRG	Ø1	Rn von	Lese Wort Rn (8 Bit) und speichere es in R4
WRG	Ø2	Ø...999(i)	Schreibe in Rn das Wort von R4 (8 Bit)

Das Datenformat kann BCD oder binär sein.

LAR, SAR Datentransfer zwischen Registerblock Rn...Rn-4 und AØ

LAR: Load AØ with Registers

SAR: Store AØ into Registers

Mnemo-code	Zahlen-code	Operand	Beschreibung
LAR	Ø5	Rn von	Lade AØ mit Registerblock Rn...Rn-4 (5 x 8 Bit)
SAR	Ø6	9...999(i)	Transferiere AØ in den Registerblock Rn...Rn-4 (5 x 8 Bit)

Im Unterschied zu den Befehlen RRG und WRG werden hier ganze Registerblöcke 5 x 8 Bit oder 9 BCD-Ziffern plus Vorzeichen transferiert. Die Daten können aber auch rein binäres Format haben.

Mit Angabe des Operanden Rn wird, wie gesagt, der ganze Registerblock von Rn...Rn-4 erfasst. Der Uebersichtlichkeit halber wird empfohlen, das Wort-Register in 5er-Blöcke aufzuteilen und für diese Befehle nur Adressen mit den Endziffern 4 oder 9 zu verwenden.

EXG: Tausche A1 mit Registerblock Rn...Rn-4: siehe I 3

(i) = indexierbar

LAC, SAC Datentransfer zwischen Zähler-Register Cn und AØ

LAC: Load AØ with Counter (Cn)

SAC: Store AØ into Counter (Cn)

Mnemo-code	Zahlen-code	Operand	Beschreibung
LAC	Ø7	Cn von	Lade AØ mit Zähler Cn
SAC	Ø8	256...511(i)	Transferiere AØ in den Zähler Cn

Die von AØ nach Cn zu transferierenden Daten müssen BCD-Format haben. Die in AØ von Cn abgelegten Daten haben ebenfalls BCD-Format.
Die Kapazitätsgrenze von Cn = 65535 muss beachtet werden. Das Vorzeichen von AØ kann dabei nicht berücksichtigt werden.

Das Zähler-Register dient grundsätzlich den im Bitprozessor aktivierbaren Zählern und Timern. Mit dem Befehl LAC können Zähler- und aktuelle Timerwerte ausgelesen und transferiert werden. Soll hingegen ein Timerwert aus AØ übernommen werden, so muss das Programm wie folgt aufgebaut werden:

```
SAC        256    Transfer AØ ---> C256 (ins Zähler-Register!)
 .          .
 .          .
EWP         Ø
 .          .
 .          .
STR (14)   256 ▌  Uebertragen des Wertes und gleichzeitig
     31    256 ▌  Aktivierung als Timer
```

WEL, WEU Transfer von 1 BCD-Digit nach Elementen (En)

WEL: Write Elements with lower Digit

WEU: Write Elements with upper Digit

Mnemo-code	Zahlen-code	Operand	Beschreibung	
WEL	Ø9	En von	Schreibe in den Elementen	R4(1Ø⁰)
WEU	1Ø	3...999(i)	En...En-3 das Digit von	R4(1Ø¹)

Dieser Befehl ermöglicht die digitweise Ausgabe von Zahlen (4 Bit) direkt auf Merker oder Ausgänge, wie sie z.B. für die Multiplexausgabe auf externe Displays benötigt wird.

Die 4 Digits können sowohl BCD-Format als auch ein beliebiges anderes Bitmuster aufweisen.

(i) = indexierbar

RRE, WRE Transfer von Registerwörtern Rn nach Elementen En

RRE: Read Register and write in Elements

WRE: Write Register with Elements

Mnemo-Code	Zahlen-Code	Operand	Beschreibung
RRE	Ø3	Rn von Ø...999(i)	Lese Wort Rn (8 Bit) und speichere es in den Elementen En...En-7. En wird dabei adressiert durch A1.
WRE	Ø4	Rn von Ø...999(i)	Schreibe in Rn den Inhalt der Elemente En...En-7 (8 Bit). En wird dabei adressiert durch A1.

Vor der Anwendung dieses Befehls muss in A1 die Elementadresse En stehen. Für diese Elementadresse sind nur die Ziffern der Register R9 und R8 signifikant.

I 5 Zähl- und Skip-Befehle

INR, DER Veränderung des Registerwertes um 1

INR: Increment Register by 1

DER: Decrement Register by 1

Mnemo-code	Zahlen-code	Operand	Beschreibung
INR	11	Rn von Ø...999(i)	Erhöhe BCD-Wert Rn um 1 und setze "carry" falls Resultat > 99
DER	12		Erniedrige BCD-Wert Rn um 1 und setze "carry" falls Resultat < Ø

Diese Befehle, zusammen mit SNC, dienen dem Aufbau beliebig grosser BCD-Zähler innerhalb des Registers Rn.

(i) = indexierbar

SNC "Ueberspringe" bedingt den nächsten Befehl

SNC: Skip if no carry

Mnemo-code	Zahlen-code	Operand	Beschreibung
SNC	13	Ø	"Ueberspringe" den nächsten Befehl, falls "carry" = Ø
		En (i)	"Ueberspringe" den nächsten Befehl, falls En = L

Zusammen mit INR und DER lassen sich damit Zählketten aufbauen.

SEW "Ueberspringe" alle Befehle bis EWP oder bis NOP 1248

SEW: Skip to EWP or to NOP 1248

Mnemo-code	Zahlen-code	Operand	Beschreibung
SEW	24	Ø	"Springe" bis EWP oder NOP 1248, falls "carry" = Ø
		En (i)	"Springe" bis EWP oder NOP 1248, falls En = L

Ist "carry" = 1 oder En = H, so hat der Befehl SEW keinen Einfluss auf den Programmablauf. Ist aber der "carry" = Ø oder En = L, so werden alle Befehle bis EWP bzw. NOP 1248 ignoriert. EWP schaltet dann um in Bitprozessor, während bei NOP 1248 das Programm weiter im Wortprozessor bleibt.

(i) = indexierbar

I 6 Diverse Befehle

CLK Zuweisung einer Zeitimpulsquelle

CLK: Clock Source

Mnemo-code	Zahlen-code	Operand	Beschreibung
CLK	21	En	Zuweisung der Zeitimpulsquelle zum Uhr-Register
		Ø...999	wirkt auf sec-Register
		1ØØØ...1999*	wirkt auf min-Register
		2ØØØ	Rückanweisung auf intern

*) Wird zur Elementadresse 1ØØØ addiert (nicht zu verwechseln mit Indexierung), so wird das Minuten-Register angesprochen.

Mit CLK kann die Uhr via einen PLC-Eingang von einer externen Impulsquelle her (z.B. Mutteruhr) geführt werden. Je nach Operand wirken diese Impulse auf das sec- bzw. min-Register.
Der Impuls selber soll eine Impuls- bzw. Pausendauer von je min 125ms (max. 4Hz) aufweisen.
Falls die Zuweisung durch CLK fehlt, erfolgt automatisch die Fortschaltung der Uhr ab internem "clock" bzw. ab Modul R27. Mit dem Operand 2ØØØ wird dieser Zustand wieder hergestellt.

SHI Schiebe-Register

SHI: Shift Register

Mnemo-code	Zahlen-code	Operand	Beschreibung
SHI	22	Rn 21...999(i)	Schiebe Register-Wörter (8 Bit) ab R2Ø...Rn um 1 Adresse nach oben

Auf der Uebersicht zu Beginn des Kapitels I 4 ist die Wirkungsweise dargestellt. Damit lassen sich ein- oder mehrbitige Schieberegister aufbauen.

Nach der Schiebeoperation bleibt die Information auf R2Ø unverändert erhalten, die Information auf Rn geht verloren.

(i) = indexierbar

TXT Start der Textausgabe

TXT: Text

Mnemo-code	Zahlen-code	Operand	Beschreibung
TXT	23	TXn von Ø...818 (i)	Start der Textausgabe

Der Text mit der Textnummer TXn wird über die serielle Datenschnittstelle ausgegeben bis der Charakter NULL (im Text) vorgefunden wird.
Details siehe K 5.

NOP Keine Operation

NOP: No Operation

Mnemo-code	Zahlen-code	Operand	Beschreibung
NOP	ØØ	Ø	Keine Operation

Gleich wie im Bitprozessor-Befehlssatz ergibt NOP Ø Leerzeilen im Wort-Programmteil. Diese werden abgearbeitet, haben aber keine Wirkung auf das Programm.

EWP Ende der Arbeit im Wortprozessor

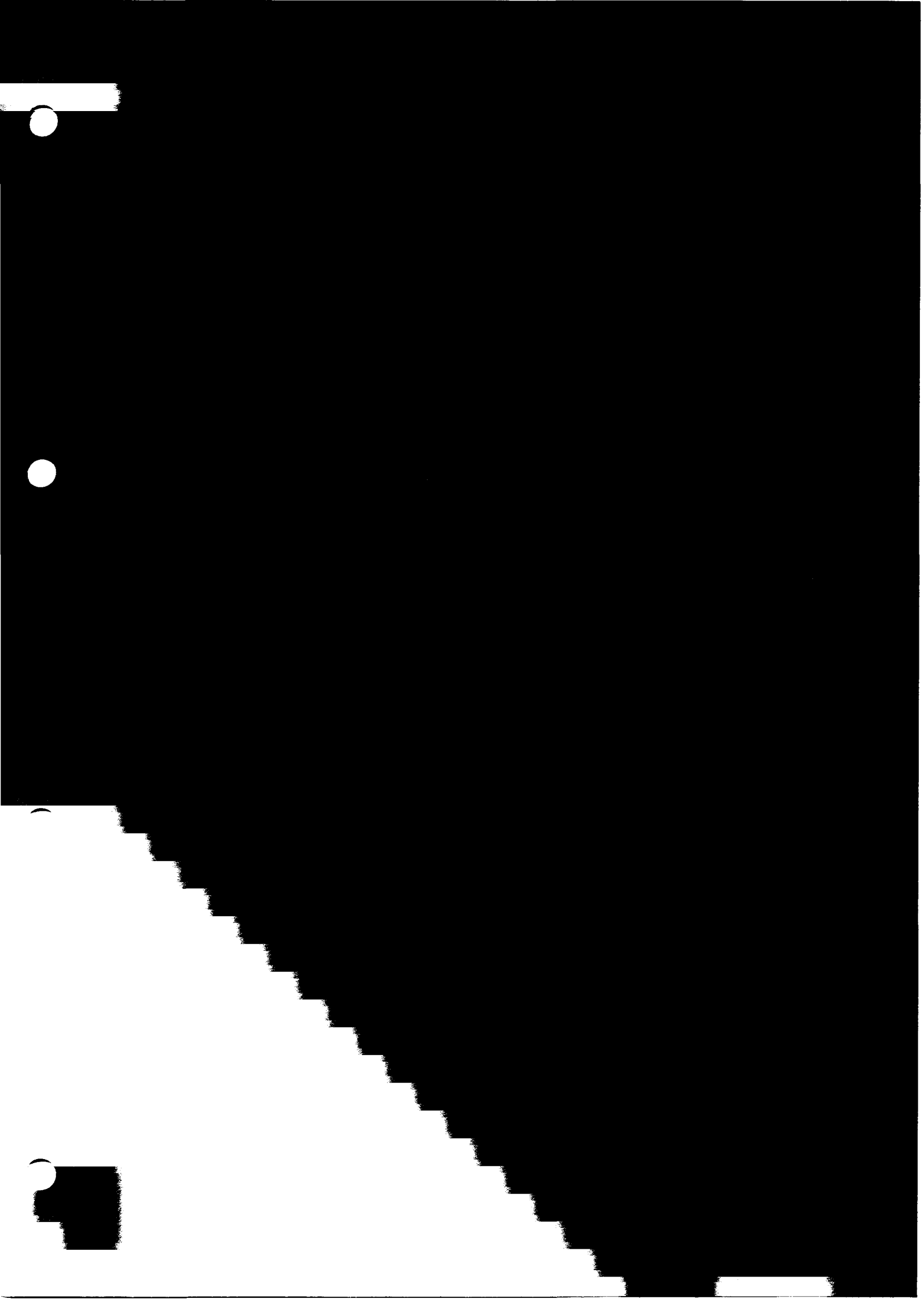
EWP: End Word Processor

Mnemo-code	Zahlen-code	Operand	Beschreibung
EWP	31	Ø	Ende der Arbeit im Wortprozessor

Jeder Programmteil im Wortprozessor wird eröffnet durch den Bit-Befehl NOP 1248 und endet mit EWP Ø. An diesen Stellen erfolgt jeweils die Umschaltung auf den entsprechenden Befehlssatz.

(i) = indexierbar

Notizen:



TEIL K TEXTAUSGABE UND KOMMUNIKATION

TEIL K TEXTAUSGABE UND KOMMUNIKATION

K 1 Die serielle Datenschnittstelle

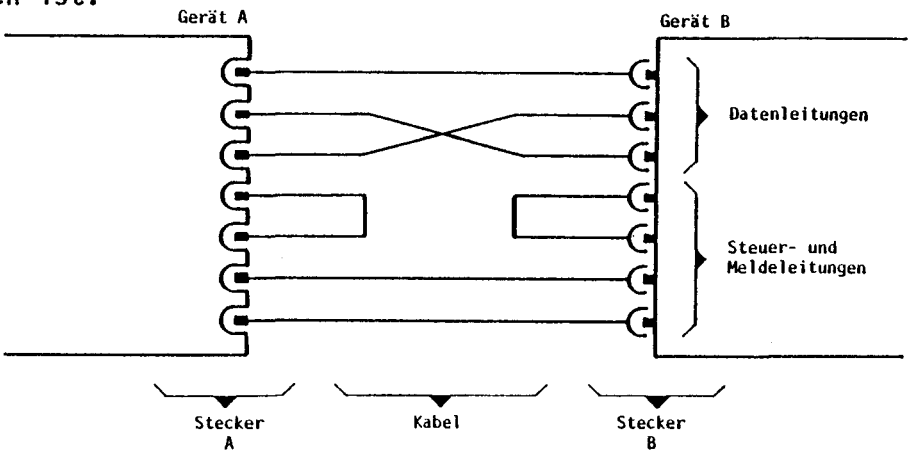
Die serielle Datenschnittstelle der PCA kann für folgende Aufgaben eingesetzt werden:

- Ausgabe von Texten ab Textspeicher (Modus TEXT)
- Eingabe von Texten mittels eines Peripheriegerätes in den Textspeicher (Modus EDITOR)
- Austausch einzelner ASCII-Charakter mit einem anderen System (Modus C)
- Austausch numerischer Daten gemäss DIN 66Ø19 (Modus N)
- Direktzugriff zu PC-Daten wie Elemente, Zähler, Register, Programmzeilen gemäss DIN 66Ø19 (Modus P)

K 1.1 Was ist eine serielle Datenschnittstelle?

Wenn Ihnen die Bedeutung von ASCII-Code, Baud-Rate, Parity-Bit, RS232c und 2ØmA Current Loop wohlvertraute Begriffe sind, dann können Sie weiterblättern zum Kapitel K 2; wenn nicht, dann lohnt es sich, diesen Abschnitt besonders gut zu lesen.

Obschon heute die meisten Peripheriegeräte und Computer Schnittstellen aufweisen, welche einschlägigen Normen wie z.B. EIA RS232c bzw. CCITT V.24 oder DIN 66Ø2Ø bzw. 66259 entsprechen, ergeben sich beim Anschliessen dieser Geräte immer wieder Probleme. Der Grund liegt darin, dass alle diese "Normen" noch einen erheblichen Spielraum offen lassen, der von Fall zu Fall im einzelnen abzuklären ist.



Wie die Abbildung zeigt, stellen sich rein von der Hardware her folgende Fragen:

- Was für ein Stecker (Polzahl, männlich/weiblich) wird verwendet?
- Wie lautet die Steckerbelegung?
- Werden neben Datenleitungen auch Steuer- und Meldeleitungen benötigt?
- Wie ist das Kabel auszulegen?

Von der Charakteristik der beiden Geräte her sind unter anderem folgende Fragen zu klären:

- Baudrate?
- Anzahl Stopbits?
- Parität?
- Voll- oder Halbduplex?
- Welches Gerät ist aktiv (liefert den Strom von 2ØmA)?

usw.

K 1.1.1 Was ist der ASCII-Code?

In der Informatik stehen als elektrische Signale nur die "Ø" und die "1" zur Verfügung. Sollen Ziffern oder andere Zeichen elektrisch dargestellt werden so ist die Kombination mehrerer Bits erforderlich. Mit 7 Bit lassen sich $2^7 = 128$ Zeichen darstellen. Der ASCII-Code ordnet diese 128 Zeichen in einer Tabelle und gliedert sie in sogenannte "Cases". Im "Control Case" finden wir 32 Steuerbefehle wie z.B. "CR" = Carriage Return (Wagenrücklauf) oder "LF" = Line Feed (Linien-Vorschub) etc.

Im "Symbols Case" sind 32 graphische Zeichen und die Dezimalnummern untergebracht. Im "Upper Case" finden wir die Grossbuchstaben und im "Lower Case" die Kleinbuchstaben.

In der abgebildeten Tabelle sind die Zeichen dezimal von Ø bis 127 durchnumeriert. Diese Dezimalnummern brauchen wir für die Texteingabe mit dem Programmeingabegerät PØ5. In anderen Darstellungen erfolgt die Numerierung auch hexadezimal von Ø bis 7F.

Die englische Bedeutung der "Control Cases" ist im folgenden aufgelistet:

(Ø)	NUL	: Null (leer)	(8)	BS	: Back Space
(1)	SOH	: Start of Heading	(9)	HT	: Horizontal Tabulation
(2)	STX	: Start of Text	(1Ø)	LF	: Line Feed
(3)	ETX	: End of Text	(11)	VT	: Vertical Tabulation
(4)	EOT	: End of Transmission	(12)	FF	: Form Feed
(5)	ENQ	: Enquiry	(13)	CR	: Carriage Return
(6)	ACK	: Acknowledgement	(14)	SO	: Shift-Out
(7)	BEL	: Bell	(15)	SI	: Shift-In
(16)	DLE	: Data Link Escape	(24)	CAN	: Cancel
(17)	DC1	: Device Control 1	(25)	EM	: End of Medium
(18)	DC2	: Device Control 2	(26)	SUB	: Substitute
(19)	DC3	: Device Control 3	(27)	ESC	: Escape
(2Ø)	DC4	: Device Control 4	(28)	FS	: File Separator
(21)	NAK	: Negative Acknowledgement	(29)	GS	: Group Separator
(22)	SYN	: Synchronous Idle	(3Ø)	RS	: Record Separator
(23)	ETB	: End of Transmission Block	(31)	US	: Unit Separator

ASCII CODE CHART

B7 B6 B5 / BITS B4 B3 B2 B1	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
	CONTROL		SYMBOLS		UPPERCASE		LOWERCASE	
0 0 0 0	NUL 0	DLE 16	SP 32	0 48	@ 64	P 80	\ 96	p 112
0 0 0 1	SOH 1	DC1 17	! 33	1 49	A 65	Q 81	a 97	q 113
0 0 1 0	STX 2	DC2 18	" 34	2 50	B 66	R 82	b 98	r 114
0 0 1 1	ETX 3	DC3 19	# 35	3 51	C 67	S 83	c 99	s 115
0 1 0 0	EOT 4	DC4 20	$ 36	4 52	D 68	T 84	d 100	t 116
0 1 0 1	ENQ 5	NAK 21	% 37	5 53	E 69	U 85	e 101	u 117
0 1 1 0	ACK 6	SYN 22	& 38	6 54	F 70	V 86	f 102	v 118
0 1 1 1	BEL 7	ETB 23	' 39	7 55	G 71	W 87	g 103	w 119
1 0 0 0	BS 8	CAN 24	(40	8 56	H 72	X 88	h 104	x 120
1 0 0 1	HT 9	EM 25	) 41	9 57	I 73	Y 89	i 105	y 121
1 0 1 0	LF 10	SUB 26	* 42	: 58	J 74	Z 90	j 106	z 122
1 0 1 1	VT 11	ESC 27	+ 43	; 59	K 75	[91	k 107	{ 123
1 1 0 0	FF 12	FS 28	, 44	< 60	L 76	\ 92	l 108	¦ 124
1 1 0 1	CR 13	GS 29	- 45	= 61	M 77	] 93	m 109	} 125
1 1 1 0	SO 14	RS 30	. 46	> 62	N 78	^ 94	n 110	~ 126
1 1 1 1	SI 15	US 31	/ 47	? 63	O 79	_ 95	o 111	RUBOUT (DEL) 127

Die beiden Zeichen aus den "Character Cases":

SP (Nr. 32) = Space (Leerschlag)
DEL (Nr. 127) = Delete bzw. Rubout (löschen)

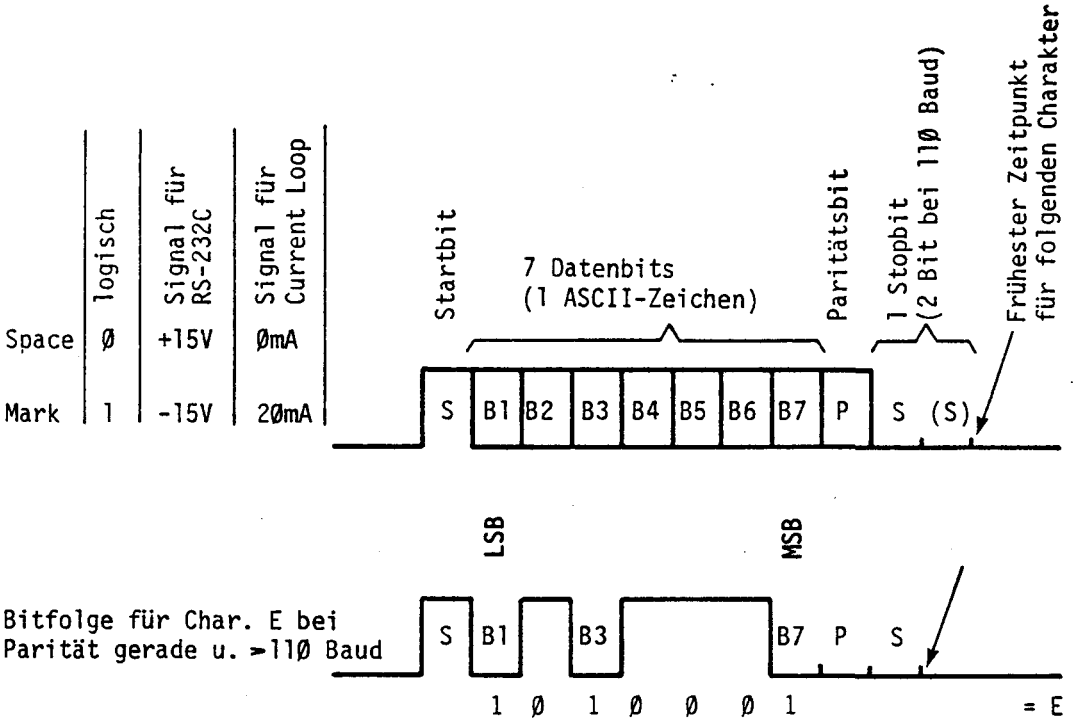
---

*) Bei den meisten Peripheriegeräten können die Charakter des "Control Case" dadurch erzeugt werden, dass gleichzeitig zur Taste CTRL der entsprechende Charakter aus dem "Upper Case" gedrückt wird. Z.B. "FF" = Form Feed (Nr. 12) erhält man durch gleichzeitiges Betätigen von CTRL und L ---> CTRL/L.

K 1.1.2 Serielle Datenübertragung, Baudrate, Parity

Wie der ASCII-Tabelle entnommen werden kann, entspricht dem Charakter E der binäre Wert 1ØØØ1Ø1 (Dezimalwert 69). Der binäre Wert lässt sich als serielles Signal über eine elektrische Leitung senden. Nach RS 232c sind das Format und die elektrischen Pegel für diese Uebertragung festgelegt. Damit der Anfang und das Ende eines Charakters umgrenzt ist, wird zu Beginn ein Startbit (als Space) und am Ende ein Stopbit (als Mark) gesendet. Einzig bei 11Ø Baud sind 2 Stopbit nötig.

Die "Norm" bzw. der Usus für 2ØmA-Stromschleife übernimmt die Regeln von RS232c.



Die Geschwindigkeit bzw. die Anzahl Bit, welche pro Sekunde übertragen werden, wird als Baudrate bezeichnet. Als Standards haben sich folgende Werte eingebürgert:

11Ø, 15Ø, 3ØØ, 6ØØ, 12ØØ, 24ØØ*, 48ØØ*, 96ØØ* Baud.

Bei 11 Bit für das ganze Bitformat ergibt sich bei 11Ø Baud eine Uebertragungsgeschwindigkeit von max. 1Ø ASCII-Charakter pro Sekunde, bei 96ØØ Baud von ca. 1ØØØ Charakter/s.

Zur Kontrolle der übertragenen Zeichen kann den 7 Bit ein Paritätsbit angefügt werden. Es wird normalerweise so gewählt, dass die Anzahl der Einsen innerhalb der Folge von 8 Bit bei asynchroner Uebertragung gradzahlig ist (s. Figur). Stellt der Empfänger eine ungerade Zahl fest, so liegt ein Uebertragungsfehler vor.

---

*) Zur Erreichung von höheren Baudraten muss das Anwenderprogramm eine entsprechende Struktur aufweisen (Prüfung mit dem Programm "RTA" des PCA-ASSEMBLERS wird empfohlen).

K 2 Die 2ØmA-Stromschleife
(auch TTY-, Linienstrom- oder Current-Loop-Schnittstelle genannt)

Als elektrisches Signal zur Darstellung der logischen "Ø" oder "1" kann eine Spannung oder ein Strom verwendet werden. Nach RS232c bzw. V.24 wird eine ± Spannung von 15V vorgeschrieben. Obschon diese Schnittstelle am weitesten verbreitet ist, hat sie im industriellen Einsatz entscheidende Nachteile:

- Es ist keine galvanische Trennung zwischen Sender und Empfänger möglich.
- Wegen leichter Störmöglichkeit sind Verbindungen von mehr als 15m Länge zu unterlassen.

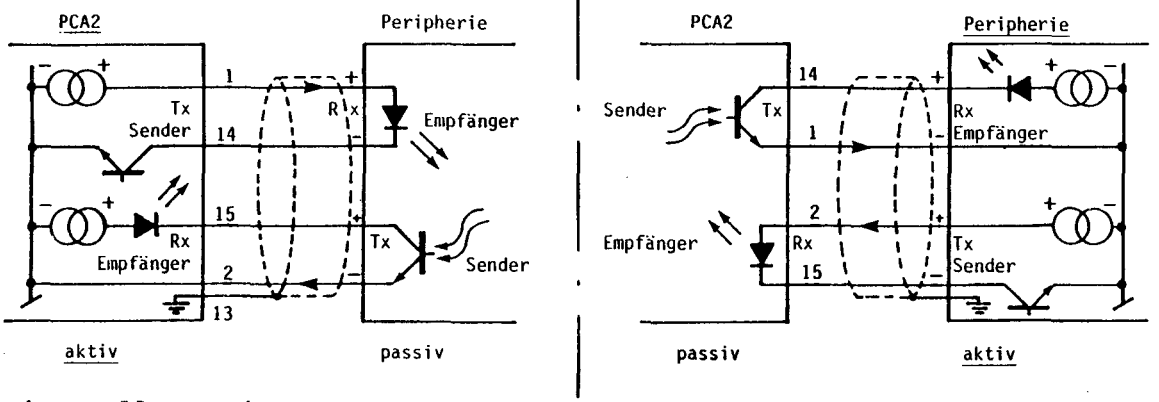
Die Stromschleife bietet andererseits:

- Möglichkeit der Uebertragung von 2ØmA direkt auf Optokoppler und damit galvanische Trennung vom Sender zum Empfänger.
- Gute Störsicherheit für Leitungslängen bis 1ØØØm.
- Begrenzte Uebertragungsgeschwindigkeit von max. 96ØØ Baud (was bei SPS nicht ins Gewicht fällt).

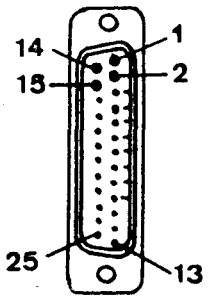
Bei einer Stromschleifen-Schnittstelle muss jeweils festgelegt werden, welche Seite den Strom liefert. Diese sog. aktive Seite ist natürlich mit der Leitung galvanisch verbunden, während die passive Seite über Optokoppler getrennt wird.

K 2.1 PCA2.M22 und M32

Durch Umstecken eines Vielfachsteckers (siehe PCA2 Hardware) kann die Schnittstelle "aktiv" oder "passiv" geschaltet und damit dem Peripheriegerät angepasst werden. Das Verbindungskabel PCA2-Peripherie ist entsprechend dem nachfolgenden Schema auszulegen.



Schnittstellenstecker:



Bezeichnung:
DATA LINES, Stecker männlich
25-poliger D-Sub Stecker mit Möglichkeit zur Schraubverriegelung.

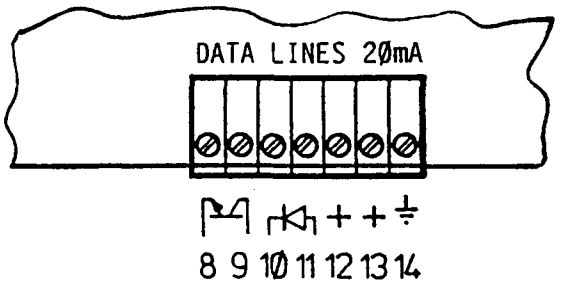
K 2.2 PCA14 und PCAØ2

Bei diesen beiden Typen kann die serielle Schnittstelle (2ØmA Stromschleife) über steckbare Schraubklemmen angeschlossen werden. Durch entsprechende Verdrahtungen kann die PCA aktiv oder passiv geschaltet werden. Hier ist es auch möglich nur den Sender aktiv, den Empfänger aber passiv zu schalten.

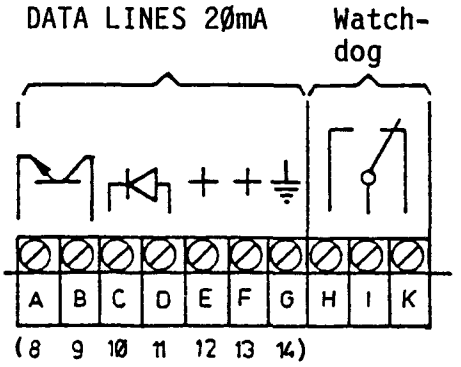
PCA14 und PCAØ2 sind abgesehen von der unterschiedlichen Klemmenbezeichnung sowohl bezüglich Hardware als auch bezüglich Software vollständig kompatibel.

Klemmenanordnung

PCA14

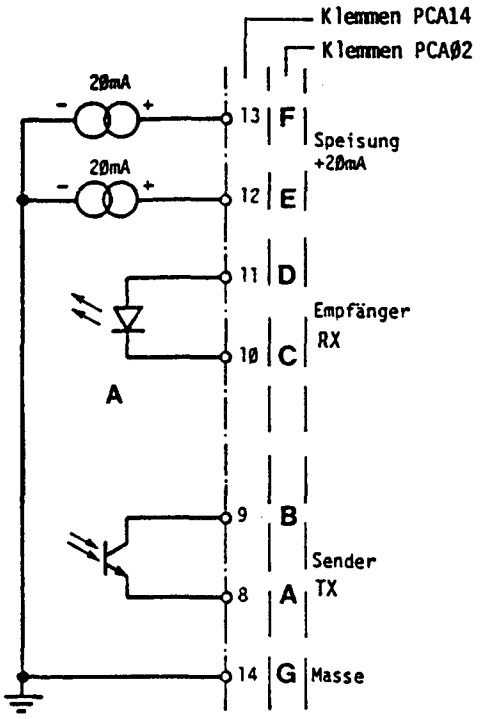


PCAØ2



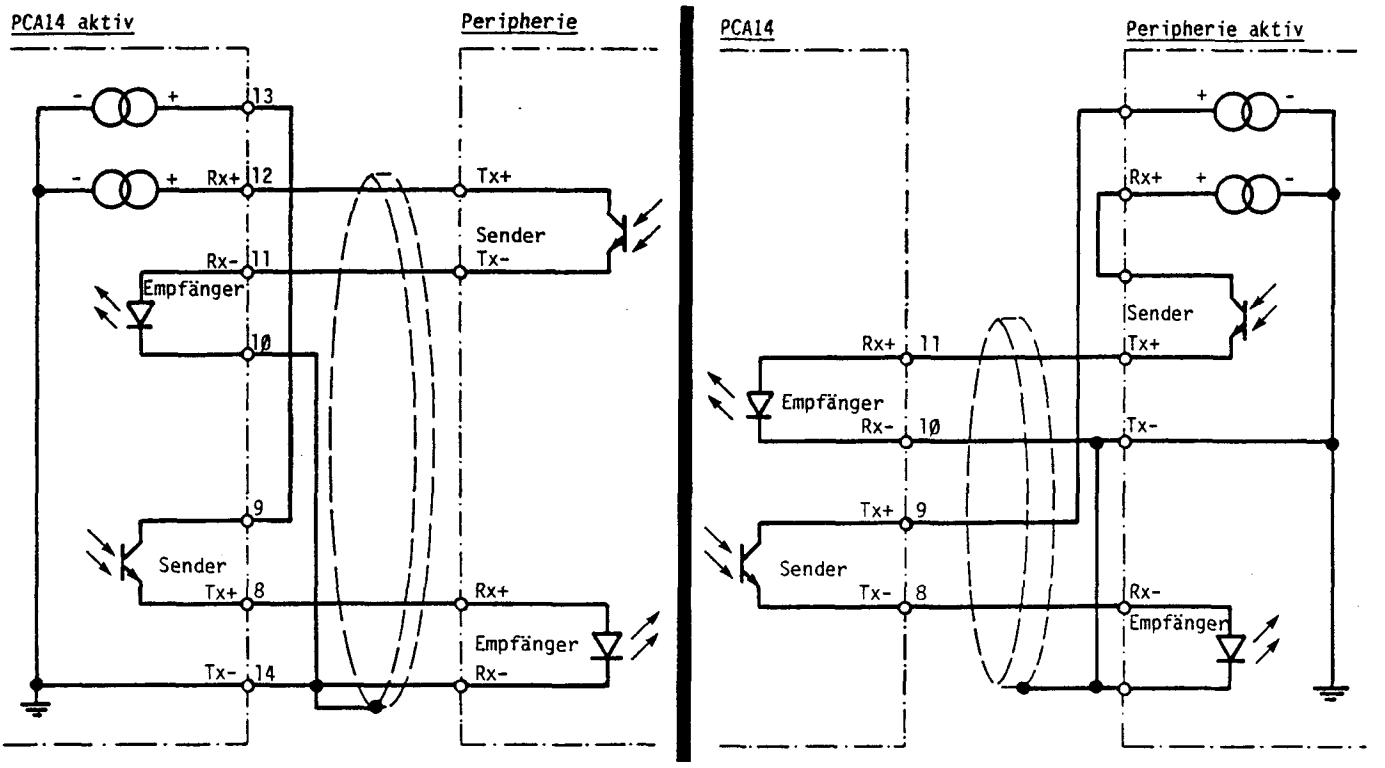
entsprechende Klemmen auf PCA14

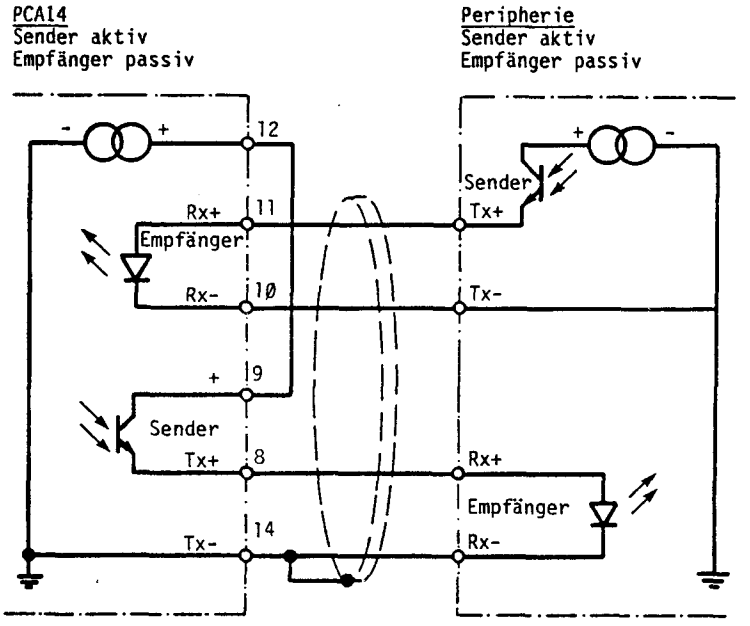
Anschluss-Schema



Verbindungs-Schemata (PCA14 und PCAØ2)

Im folgenden sind nur die Schemata für die PCA14 dargestellt. Aufgrund der Klemmenanordnung für die PCAØ2 auf der vorangehenden Seite ist leicht ersichtlich, wie auch die PCAØ2 in allen drei Varianten angeschlossen werden kann.





K 3 Funktion der Error-Lampe im Bedienerfeld

Die Error-Lampe wird aktiviert wenn:

a) der Charakter falsch ist, d.h. Parity falsch, Framing oder Overrun

b) das Telegramm falsch ist, d.h.
- STX nicht erster Charakter ist
- das Telegramm nicht erlaubte oder zuviele Charakter enthält
- der BCC falsch ist

c) der Stromkreis unterbrochen ist.

Die Errorlampe wird gelöscht bei jedem Sekundentakt (bzw. bei jedem 1/10s-Takt, wenn die Zeitbasis 1/100s gewählt wurde).

K 4 Die Assignierung der Schnittstelle auf der CPU der PCA

Wie aus der Einleitung hervorgeht, müssen für einen seriellen Datenaustausch nicht nur das Verbindungskabel und die Stecker stimmen, sondern die verschiedenen Uebertragungsparameter der beiden zu korrespondierenden Geräte müssen festgelegt werden und übereinstimmen. Die Festlegung der Uebertragungsparameter erfolgt bei der PCA durch den Befehl PAS 1ØØ.

Mit PAS 1ØØ wird ein 1Ø-zeiliger Befehl eröffnet, wobei die einzelnen Zeilen folgende Funktionen haben:

```
Zeile 1:  PAS   (29)     1ØØ   ;  Aktivierung der Schnittstelle
      2:        ØØ       xxxx   ;  Baudrate, Parity und Stopbit
      3:        Øx       xxx    ;  Text-Busy Flag (TXB)
      4:        n        xxx    ;  Receive Buffer Busy Flag (RBY)
      5:        n        xxx    ;  Transmit Buffer Busy Flag (TBY)
      6:        Øx         d    ;  Block Check Character
      7:        ØØ         Ø    }
      8:        ØØ         Ø    }  leer
      9:        ØØ         Ø    }
     1Ø:        ØØ         Ø    }
```

PAS 1ØØ wird im allgemeinen am Anfang eines Programmes im Assignierungsteil stehen und somit nur einmal abgearbeitet werden.
Muss ein assignierter Modus (C, N, P) gewechselt werden, so kann im Anwenderprogramm jederzeit eine entsprechende Neuassignierung gemacht werden.

Programmzeile 1: PAS 1ØØ

Zeile 2: Baudrate, Parity und Stopbit

Im Operand der zweiten Zeile steht die Summe der nachstehenden parameter-Werte. Der Code ist immer ØØ.

Parameter		Zu addierende Konstante
Baudrate:	11Ø Baud	Ø
	15Ø Baud	1
	3ØØ Baud	2
	6ØØ Baud	3
	12ØØ Baud	4
	24ØØ Baud*	5
	48ØØ Baud*	6
	96ØØ Baud*	7
Anzahl Databit pro Charakter:	7	Ø
	8	32
Paritäts-Generierung und -Kontrolle:	inaktiv	Ø
	aktiv	128
Falls Parität aktiv:	ungerade	Ø
	gerade	256
Anzahl Stopbit:	1 Bit	512
	1 1/2 Bit	1Ø24
	2 Bit	1536

*) Die höheren Baudraten bedingen eine entsprechende Programmstruktur mit kurzen PP-Wechselzeiten. Eine Prüfung mit dem Programm "RTA" des PCA-ASSEMBLERS wird empfohlen.

Beispiel: 12ØØ Baud	4	
7 Databit	Ø	
Parität aktiv	128	
Parität gerade	256	
1 Stopbit	512	
Summe	9ØØ	≙ Operand der 2. Zeile

Zeile 3: Text Busy Flag (TXB)

ØØ im Code gilt für allgemeine Anwendung der Datenschnittstelle.
Ø1 im Code aktiviert den Editor für die Texteingabe über die serielle Datenschnittstelle (siehe K 5.3).

Im Operand wird die Adresse eines Merkers oder Ausganges angegeben, der jeweils "H" ist, solange Text ausgegeben wird. Nach Beendigung der Text-Ausgabe geht dieses Busy-Element wieder auf "L".

Damit wird ermöglicht, einen Text zuerst vollständig auszugeben, bevor von anderer Seite ein weiterer Text gestartet wird.

Soll jedoch ein Text Priorität haben (z.B. eine Alarmmeldung), so kann ohne Abfrage des BUSY-Elementes jederzeit der laufende Text unterbrochen werden, um dem Prioritätstext Raum zu geben.

Zeile 4 und 5: Empfangs- bzw. Sende-Buffer

Im Code wird die Grösse dieses Buffers festgelegt und im Operand die Adresse des Busy-Merkers angegeben. Nähere Einzelheiten siehe Abschnitt K 6.

Zeile 6: Prüfsumme (BCC)

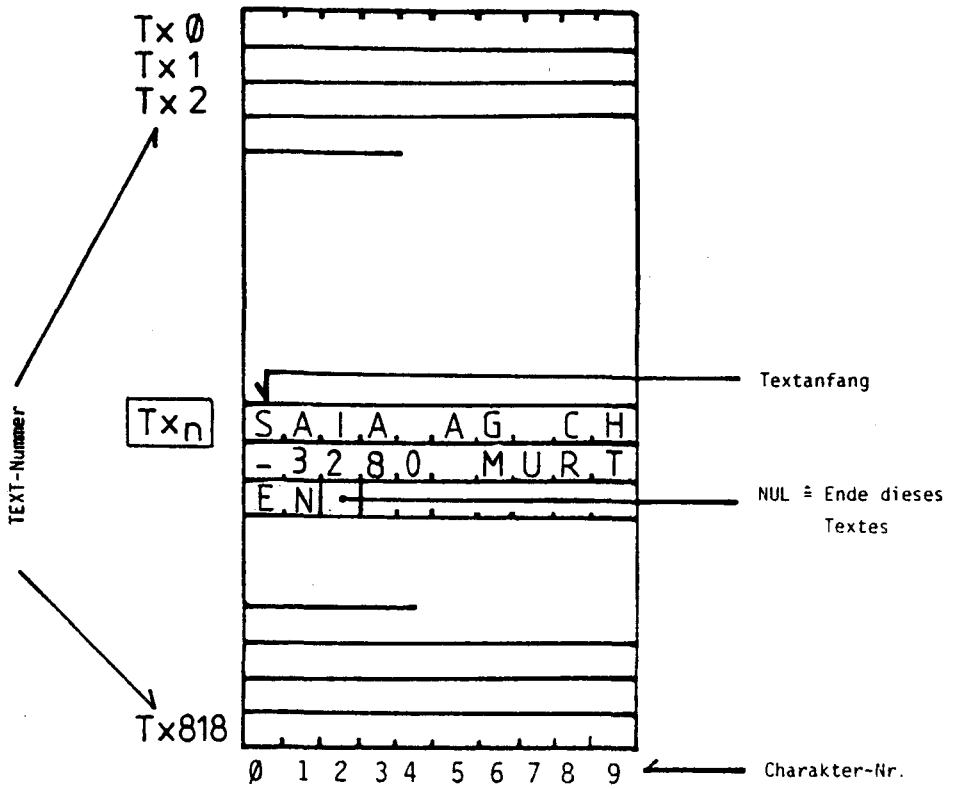
Durch Festlegen einer Prüfsumme kann die Uebertragungssicherheit bei Uebertragungen gemäss *DIN 66Ø19* erhöht werden. *Einzelheiten siehe Abschnitt* K 6.3.

K 5 Text Ein-/Ausgabe

K 5.1 Organisation des Textspeichers

Gemäss der Hardwarebeschreibung bildet der Textspeicher einen Teil des Anwenderspeichers. 4K Anwenderspeicher entsprechen 818 Texten zu 1Ø Charakter, d.h. 8K Charakter.

Jeder Textanfang ist adressierbar zwischen TxØ...Tx818. Der erste Charakter steht auf der Charakter-Nr. Ø. Das Textende muss immer mit der Dezimalnummer Ø (ASCII-Zeichen NUL) markiert sein. Ein längerer Text kann aus beliebig vielen Text-Nummern zusammengesetzt werden.



K 5.2 Ausgabe eines Textes

Alle Prozessoren der Stufe 2 benutzen den Befehl PAS 23. In der Stufe 3 (PCA231 und 232) steht im Wortprozessor der Befehl TXT zur Verfügung. Die Wirkung ist in beiden Fällen genau gleich. Beide Befehle werden unabhängig vom ACCU-Zustand immer ausgeführt und beeinflussen diesen nicht.

Sowohl der Bitprozessor-Befehl

```
PAS  (29)   23
     ØØ    375   (i)
```

als auch der Wortprozessor-Befehl (nur PCA231/232)

```
TXT  (23)   375   (i)
```

geben den Text ab Text-Nummer 375 bis zum Zeichen "NUL" aus. Die Ausgabe erfolgt über die serielle Schnittstelle (Stromschleife 2ØmA). Um die Uebertragung sicherzustellen, müssen die entsprechenden Parameter vorgängig festgelegt und programmiert werden.

- Stromschleife aktiv/passiv wird hardwaremässig mit dem Wahlstecker auf dem CPU-Print festgelegt (siehe Handbuch Hardware).
- Die Uebertragungsparameter, wie Baudrate, Parity etc. sind durch die 2. Zeile der Instruktion PAS 1ØØ im Anwenderprogramm softwaremässig einzugeben (siehe vorangehenden Abschnitt).
- Durch die Instruktion PAS 1ØØ wird mit der 3. Zeile ein Element (Merker oder Ausgang) definiert, welches solange "H" ist als Text ausgegeben wird. Nach Beendigung der Text-Ausgabe geht das BUSY-Element TXB wieder auf "L".
- In der 4. und 5. Zeile des PAS 1ØØ-Befehles sollen Elemente adressiert werden, die normalerweise dauernd auf "L" sind.

Beispiel:

Der Text "SAIA AG CH-328Ø Murten" stehe auf der Text-Nummer 375. Er soll durch Schliessen des Kontaktes E7 auf ein Terminal ausgegeben werden.

Das zugehörige Programm heisst wie folgt:

```
 1   PAS   (29)    1ØØ   Schnittstellen-Assignierung
 2         ØØ      9Ø2   Parameter für Terminal*
 3         ØØ       32   Busy-Anzeige (TXB)
 4         ØØ      254   } die höhere Adresse von 2 Elementen,
 5         ØØ      254   } welche dauernd auf "L" bleiben (254 und 253)
 6         ØØ        Ø   }
 7         ØØ        Ø   }
 8         ØØ        Ø   } Null
 9         ØØ        Ø   }
1Ø         ØØ        Ø   }
11   STH   (Ø1)      7   Kontakt 7 abwarten
12   DYN   (Ø9)    4ØØ
13   JIZ   (22)     11
14   WIH   (25)     32   warten bis Text ausgegeben
15   PAS   (29)     23
16         ØØ     375   Text ab Nr. 375 ausgeben
17   JMP   (2Ø)     11
```

Textausgabe im Bitprozessor

```
15   NOP   (ØØ) W  1248
16   TXT   (23) W   375
17   EWP   (31) W     Ø
18   JMP    2Ø       11
```

Textausgabe im Wortprozessor

*) 48ØØ Baud, 7 Bit, Parität aktiv und gerade, 1 Stopbit
(i) = indexierbar

K 5.3 Eingabe von Texten

Texte werden auf den RAM-Textspeicher im Anwenderprogramm eingegeben.

Nach Eingabe und Austesten der Texte kann der RAM-Inhalt in gleicher Weise kopiert werden, wie dies bei der Kopie von Anwenderprogrammen erfolgt.

Soll ein bestimmter Textbereich kopiert werden, so müssen die Text-Nummern wie folgt in Schrittadressen umgerechnet werden:

```
Anfangs-Text-Nr.  x5      ≙ Anfangs-Schrittadresse
Ziel   -Text-Nr.  x5 (+4) ≙ Ziel-Schrittadresse
```

Beispiel:

```
Anfangs-Text-Nr. 15Ø ≙ Anfangs-Schrittadresse  75Ø (15Ø x 5)
Ziel   -Text-Nr. 2ØØ ≙ Ziel-Schrittadresse    1ØØ4 (2ØØ x 5 + 4)
```

Sonderzeichen

Die Textspeicher können neben fixen Texten auch Sonderzeichen enthalten, mit deren Hilfe Register ausgelesen und in Kombination mit Klartext zu sprechenden Protokollen oder graphischen Darstellungen gebracht werden können.

Mit Ausnahme des Charakters "NUL" (Ø)*, welcher das Textende markiert, können alle Zeichen der ASCII-Tabelle von 1...127 (1...7F hex) verwendet werden. Sie werden im allgemeinen ohne Modifikation übertragen, mit Ausnahme des Sonderzeichens $, gefolgt von 1 bis 4 Zeichen.

Wird mit 8 Bit pro Charakter gearbeitet, so können auch die ASCII-Zeichen 128...255 ausgegeben werden, deren Wirkung vom angeschlossenen Terminal abhängig ist.

---

*) Das ASCII-Zeichen "NUL" kann auf den meisten Terminals durch CTRL/@ erzeugt werden.

Bedeutung der Sonderzeichen

Zeichenfolge im Textspeicher		Zeichenfolge in der Form, wie sie ausgegeben wird
Alphanum. Eingabe	Eingabe mit PØ5 (Dez.-Nr.)	
$$	36, 36	$
$C Cn *	36, 67,...	Inhalt des Zähler-Registers Cn, 5 Dezimal-Ziffern ohne Vornullenunterdrückung
$c Cn *	36, 99,...	Inhalt des Zähler-Registers Cn, 5 Dezimal-Ziffern mit Vornullenunterdrückung (Spaces)
$A Rn *	36, 65,...	Inhalt des Registerblockes Rn, als 1Ø Dezimal-Ziffern ohne Vornullenunterdrückung **
$a Rn *	36, 97,... ***	Inhalt des Registerblockes Rn, 9 Dezimal-Ziffern (mit Vorzeichen, falls negativ) mit Vornullen-unterdrückung (Spaces)**
$R Rn *	36, 82,...	Inhalt des Registerwortes Rn, 2 Dezimal-Ziffern**
$r Rny	36, 114,...	Inhalt des LOWER bzw. UPPER-Teiles des Register-wortes, 1 Dezimalziffer, y=Ø --> LOWER, y=1 --> UPPER**
$E En *	36, 69,...	Status der 8 Elemente En...En-7 8 Charakter 1 oder Ø
$e En *	36, 1Ø1,...	Ausgabe des ASCII-Charakters, der aus den 7 bzw. 8 Elementen En...En-7 gebildet wird
$S En *	36, 83,...	Ist Element En = "H", so wird Character "-" ausgegeben Ist Element En = "L", so wird Character "+" (PCA14) bzw. "space" (PCA232) ausgegeben
		Inhalt der Datum-Uhr:
$T	36, 84	: 22 : Ø6 / 13 : 3Ø : 42 Woche Tag Stunde Min. Sec. (22.) (SAM)
$H	36, 72	83 - Ø6 - Ø4 / 13 : 3Ø : 42 Jahr Mt. Tag Stunde Min. Sec.
$D	36, 68	83 - Ø6 - Ø4 Jahr Mt. Tag

*) siehe nächste Seite
**) siehe nächste Seite
***) nur möglich mit PCA231/232

$ xxx [1]	36, _, ...	Repetierte Ausgabe desjenigen Charakters, der unmittelbar nach dem Zeichen $ folgt, und zwar so oft, wie in xx festgelegt ist: - xxx ≤ 127 Repetition entsprechend xxx - xxx = 256...319 bzw. 511 Repetition entsprechend dem Inhalt des entsprechenden Zählers (max. 127) → Zu repetierender Charakter. Es dürfen alle ASCII-Zeichen (auch NUL) verwendet werden, ausser den folgenden, welche Funktionen auslösen: $, C, c, D, E, e, H, L, T, U, A, a, R, r
$L xxx [2]	36, 76,...	Sprung auf Text-Nr. xxx mit Abspeicherung der Absprungadresse (Textsubroutine analog JMS)
$U	36, 85	Rücksprung in den Ausgangstext (analog RET, wobei nur 1 Ebene zulässig ist). [3]

Teil L
Beispiele
1,2,3,4 und 11

[1] Für Cn, Rn, En bzw. xxx müssen immer 3 Ziffern eingegeben werden, z.B. Ø27.
[2] Erreichen binär formatierte 4-Bit-Wörter Werte zwischen 1Ø und 15, so werden folgende Charakter ausgegeben:
1Ø ---> : 11 ---> ; 12 ---> < 13 ---> = 14 ---> > 15 ---> ?
[3] Ab gewissen Firmware-Versionen sind 3 Subroutinen-Ebenen zulässig:
PCAØ2: V6.132; PCA14: V6.Ø34; PCA222: V6.23Ø

Texteingabe:

Zur eigentlichen Eingabe von Texten in den Textspeicher stehen auf der Geräte-Seite zwei Möglichkeiten zur Verfügung:

a) mit den Eingabegeräten P1Ø/PØ5 oder P21, eingesteckt am 25-poligen PGU-Stecker

b) mit einem Peripheriegerät mit Stromschleifen-Schnittstelle, eingesteckt am Stecker "DATA LINES".

a) Texteingabe mit Programmeingabegerät PCA2.P1Ø/PØ5

Das Programmeingabegerät wird am PGU-Stecker eingesteckt. Betriebswahlschalter auf TEXT stellen. Die Eingabe erfolgt ähnlich wie unter PROG.

Z.B. den Text "SAIA AG CH-328Ø MURTEN" auf Textanfang Nr. 375 eingeben:

	Text-Nummer / Char.Nr.		Nr. des ASCII-Codes in Dezimalform		Text
[A]	375Ø *	[E]	83	[E]	S
	(3751)**		65	[E]	A
	(3752)		73	[E]	I
	(3753)		65	[E]	A
	(3754)		32	[E]	
	(3755)		65	[E]	A
	(3756)		71	[E]	G
	(3757)		32	[E]	
	.		.	.	.
	.		.	.	.
	.		.	.	.
	(3776)		13	[E]	"CR"
	(3777)		1Ø	[E]	"LF"
	(3778)		Ø	[+]	"NUL" = Textende

*) Bitte beachten, dass die Charakter-Nr. eingegeben wird, d.h. für Textanfang 375 ---> 375Ø.

**) Die nachfolgenden Charakter-Nr. müssen nicht mehr eingegeben werden. Sie erscheinen automatisch im Display.

b) Texteingabe mit einem Peripheriegerät mit Stromschleifen-Schnittstelle

Nach vorheriger Kabelauslegung und Festlegung der Uebertragungsparameter wird das Peripheriegerät am Stecker DATA LINES angeschlossen. Gleichzeitig soll das Programmeingabegerät PØ5 eingesteckt sein, wobei die PCA weder auf RUN noch auf BREAK stehen darf.

Software-Bedingungen:

Der Benützer muss dafür besorgt sein, dass die CPU die 1Ø-zeilige PAS 1ØØ-Instruktion abgearbeitet hat, in welcher die 3. Zeile den Operations-Code Ø1 enthält. Damit wird die CPU alle erhaltenen Charakter durch ihren Text-Editor verarbeiten lassen (siehe folgendes Anwendungsbeispiel).

Bedeutung von Sondercharaktern des Text-Editors

Der Text-Editor kann in zwei verschiedenen Betriebsmodi sein:

- Textdisplay-Modus
- Texteingabe-Modus

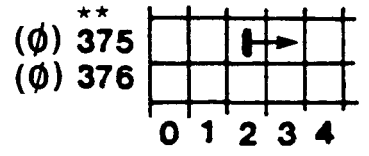
- Textdisplay-Modus

Mit Hilfe einiger Sonderbefehle kann in diesem Modus eingegebener Text auf einem Bildschirm angezeigt bzw. mit dem "Pointer" an verschiedene Stellen eines Textes gesprungen werden.

Txn [CR] **

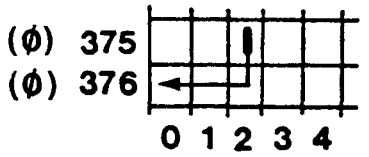
Eingabe einer Textnummer Txn gefolgt von "CR" (nur die drei letzten Ziffern sind signifikant). Dadurch wird der "Pointer" an den Anfang des Textes Nr. Txn gesetzt und die betreffende Textnummer mit Inhalt (1Ø Charakter) angezeigt.

[>]



Der Pointer wird um eine Stelle in der Charakter-Nr. nach rechts verschoben. Bei längerem Drücken der Taste wird dieser Vorgang mehrmals repetiert*. Dabei wird der gespeicherte Text fortlaufend und ohne Zwischenzeile ausgegeben.

[+]



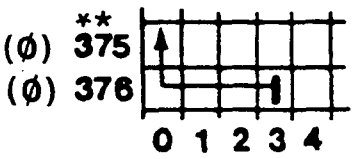
Der Pointer wird bis zum Anfang der nächsthöheren Text-Adresse nach unten gerückt. Bei längerem Drücken der Taste wird dieser Vorgang mehrmals repetiert*. Der gespeicherte Text wird dabei fortlaufend und mit je 1 Zeile Zwischenraum ausgegeben.

---

*) Die Repetier-Funktion ist vom Peripheriegerät abhängig.

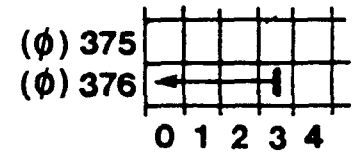
**) Die PCA232 verlangt eine 4-stellige Eingabe, sodass für Text 375 ---> Ø375 eingegeben werden muss.

[-]



Der Pointer wird bis zum Anfang der nächsten tieferen Text-Adresse nach oben gerückt. Bei längerem Drücken der Taste wird dieser Vorgang mehrmals repetiert*. Der gespeicherte Text wird dabei rückwärts und mit je 1 Zeile Zwischenraum angezeigt.

[L]



Der Pointer wird zum Anfang der aktuellen Text-Nummer gebracht, wobei sein Inhalt ausgegeben wird.

[CTRL/T] Wechsel in den Texteingabe-Modus.

- Texteingabe-Modus

Mit CTRL/T wird der Texteingabe-Modus aktiviert. Nach vorangehender Wahl einer Textanfangs-Nummer kann in diesem Modus der Lauftext mit der Schreibmaschinentastatur eingegeben werden.

[^] Wird dieses Zeichen vor "CR" eingegeben, so wird "CR" als Wagenrücklauf-Anweisung in den Text übernommen und nicht in den Textdisplay-Modus gewechselt.

[BS] Die PCA232 erlaubt mit "BS" (Back Space) falsche Eingaben zu korrigieren, d.h. der sich beim Cursor befindliche Charakter wird gelöscht.

[CR] oder [CTRL/M] Beide Zeichen, "CR" und CTRL/M, sind identisch und bewirken den Wechsel in den Textdisplay-Modus.

Hinweis: Nach Benützung des Editors ist dieser Modus unbedingt mit "CR" wieder zu verlassen, bevor eine andere Tätigkeit aufgenommen wird.

Beispiel: Eingabe eines Textes mit Peripheriegerät ab Textnummer 3Ø.

Normalerweise werden am Programm-Anfang die Assignierungen der Paralleprogramme und PAS 1ØØ mit der betriebsmässigen Assignierung der Datenschnittstelle stehen.

Einen zweiten Block mit PAS 1ØØ zum Gebrauch für die Texteingabe wird man am Ende des Programmspeichers plazieren, z.B. ab Schritt 4ØØ1 gefolgt von der zweizeiligen Instruktion PAS (29) 23
ØØ 3Ø.

In der Betriebsart STEP wird man anschliessend:

- zur Schrittadresse 4ØØ1 springen, um den Texteditor zu aktivieren.
- den eingegebenen Text zur Kontrolle ausgeben lassen.

---

*) Die Repetierfunktion ist vom Peripheriegerät abhängig.
**) Die PCA232 verlangt eine 4-stellige Eingabe, sodass für Text 375 ---> Ø375 eingegeben werden muss.

Aktivierung der Datenschnittstelle für Texteingabe mittels Editor:

```
4001  PAS  (29)    100 ;  Parameter für 4800 Baud, 7 Bit, Parität gerade,
4002        00      902    1 Stopbit
4003        01       63 ;  (01) für Texteditor; 63 für Text Busy (TXB)
4004        00      254 }  die höhere Adresse von 2 Elementen, welche
4005        00      254 }  dauernd auf "L" bleiben (254 und 253)
4006        00        0 ┐
 .          .         . │
 .          .         . │ Null
 .          .         . │
4010        00        0 ┘
4011  PAS  (29)     23 ←┐
4012        00       30  │ Textnummer 30 ausgeben
4013  JMP  (20)      0  │
4014        00     4011 ─┘
```

Unter der Textnummer 30 wollen wir folgenden Text eingeben:

SAIA-PLC GEKROENT MIT PCA232

Vorgehen:

1) P05-Programmeingabegerät bei PROG. UNIT einstecken und Betriebswahlschalter auf "STEP" stellen.

2) Durch [A] 4001 [+] auf P05-Programmeingabegerät auf Anfang der Assignierung springen.

3) Mit [+] Assignierung abarbeiten bis Schrittadresse 4011.

4) Auf Peripheriegerät 030 "CR" eingeben.**

Display auf Bildschirm (Tasteneingaben sind unterstrichen)	Beschreibung
030 "CR" ** 030:.A.B. . . . ^@^@^@^@ 030:	Der angezeigte Text ist bei einem ungelöschten Textspeicher zufällig. Ein Punkt vor einem Charakter bedeutet, dass dieser Charakter aus dem sichtbaren Teil der ASCII-Tabelle stammt (32...127). Für Steuerbefehle aus dem "Control-Case" steht ein Pfeil oder ein ^ davor.

**) Die PCA232 verlangt eine 4-stellige Eingabe für die Text-Nr., d.h. die letzten 4 eingegebenen Ziffern sind signifikant.

Display auf Bildschirm
(Tasteneingaben sind unterstrichen)

CTRL/T ← 5) Um in den Texteingabe-Modus zu gelangen, wird die Taste [CTRL] und gleichzeitig die Taste [T] (CTRL/T) gedrückt.

6) Anschliessend werden mit dem Sonderzeichen [$] 6 x Space eingegeben, gefolgt vom Namenszug SAIA°PLC.

7) Man beachte, dass nach Eingabe von 1Ø Charaktern der Bildschirm automatisch auf die folgende Textnummer weiterschaltet.

```
30  .$. .0.0.6.S.A.I.A.-
31  .P.L.C^M^J.G.E.K.R.O
32  .E.N.T. .M.I.T. .P.C
33  .A.2.3.2^M^J^@^@^@^@
```

8) Um "CR" eingeben zu können, ohne dass es den Text-Modus ändert, muss davor [^] gedrückt werden. Die CPU antwortet in beiden Fällen mit ^ M (CTRL/M).

9) Um "LF" = Line Feed einzugeben, wird [CTRL/J] gedrückt, was dem "LF" entspricht. Es erscheint ^ J.

1Ø) Bei der PCA232 kann man im Texteingabe-Modus falsche Eingaben mit der Taste "BS" (Back Space) korrigieren. D.h. der sich beim Cursor befindliche Charakter wird durch "BS" gelöscht.

11) Um den Text abzuschliessen, ist NUL einzugeben, was via [CTRL/@] möglich ist (je nach Peripheriegerät evtl. anders).

12) Mit Eingabe von ["CR"] wird der Editor-Mode verlassen.

13) Sollen die eingegebenen Charakter nun im Klartext erscheinen, so ist am Programmeingabegerät PØ5 mehrmals die [+] Taste zu betätigen. Damit wird die Programmschleife 4Ø11...4Ø14 abgearbeitet und die Textausgabe über den Befehl PAS 23 aktiviert.

SAIA-PLC
GEKROENT MIT PCA232

Text-Editor in Kurzform

a) Programmierung der Datenschnittstelle als Editor

```
4001  PAS  (29)    100
4002       00      902    Uebertragungs-Parameter
4003       (01)     63    TXB        (01) ---> Editor aktivieren
4004       00      254    } die höhere Adresse von 2 Elementen, welche
4005       00      254    } dauernd auf "L" bleiben (254 und 253)
4006       00        0    }
  .         .        .    } Null
  .         .        .    }
4010       00        0    }
4011  PAS  (29)     23    } Textausgabe
4012       00      Txn    } im STEP-Betrieb
4013  JMP  (20)      0    }
4014       00     4011    }
```

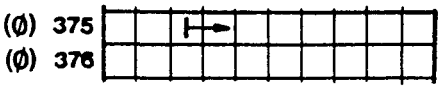
b) Aktivierung des Editors

Im STEP-Mode mit Programmeingabegerät P05 die Schrittadresse 4001 anspringen und bis 4011 abarbeiten.

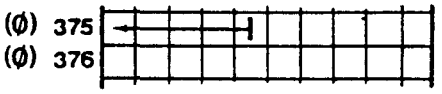
c) Display-Modus (Display des Textspeichers)

Txn [CR] Textzeile mit Nr. Txn wird angezeigt

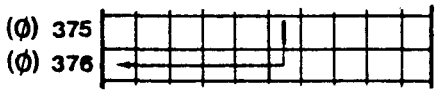
[>]



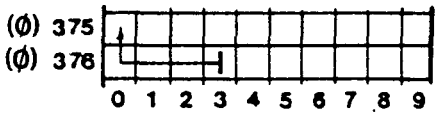
[L]



[+]



[-]



d) Texteingabe-Modus

[CTRL/T] Texteingabe erfolgt ab Charakter-Nr. Txn, welche im Display-Modus mit dem "Pointer" vorgewählt wurde.

[BS] Löschen eines eingegebenen Charakters (nur PCA232)

[^] [CR] Wird als "CR" in den Textspeicher übernommen

[CR] Wechsel in Display-Modus

e) Textausgabe

Im STEP-Mode mit Programmeingabegerät P05 [+] -Taste 2mal betätigen, wobei jedesmal der Text zur Kontrolle ausgegeben wird.

K 5.4 Text Ein-/Ausgabe mit 8 Bit

Die Anzahl Databit pro Charakter kann 7 (entsprechend dem ASCII-Code Ø...127) oder 8 Bit (Ø...255) betragen, d.h. man kann wahlweise mit 7 oder 8 Databit pro Charakter arbeiten. Die Vorwahl erfolgt in Zeile 2 des Befehls PAS 1ØØ. Die Bedeutung der zusätzlichen Charakter 128...255 ist von der angeschlossenen Peripherie (Computer, Graphik-Terminal) abhängig und von Gerät zu Gerät verschieden.

Texteingabe

Die Texteingabe ist mit dem Programmiergerät PØ5/P1Ø/P21 oder mit einem Peripheriegerät mit Stromschleifen-Schnittstelle möglich:

a) Texteingabe mit dem Programmeingabegerät PØ5/P1Ø via PGU-Stecker

Die Eingabe erfolgt wie bisher in Dezimalform, wobei zu den ASCII-Zeichen, entsprechend Ø...127 zusätzlich die Werte 128...255 möglich sind.

b) Texteingabe mit Peripheriegerät über den Texteditor der PCA

Für diese Art der Texteingabe muss ein Peripheriegerät mit Stromschleifen-Schnittstelle zur Verfügung stehen, das tatsächlich 8 Databit sendet.
Auf Seite der PCA erfolgt die Festlegung 7 oder 8 Databit pro Charakter in der 2. Zeile des PAS 1ØØ.

Textausgabe

a) Textdisplay des Textspeichers über den Texteditor der PCA

Unabhängig davon ob ein Peripheriegerät mit 7 oder 8 Databit pro Charakter arbeitet, werden die 256 Zeichen durch den Texteditor wie folgt ausgegeben:

Charakter-Nr.	Editor-Ausgabe	Textausgabe via Anwenderprogramm
Ø...31	^ ⌊‾‾⌋	ASCII-Code (wie hisher)
32...127	. ⌊‾‾⌋	
128...159	! ⌊‾‾⌋ 1)	nicht definiert, je nach Peripherie verschieden
16Ø...255	: ⌊‾‾⌋ 2)	

⌊‾‾⌋Charakter

1) Als Charakter erscheint ein ASCII-Zeichen 64...95
2) Als Charakter erscheint ein ASCII-Zeichen 32...127

b) Textausgabe via Anwenderprogramm

Es können Zeichen mit 7 oder 8 Databit pro Charakter ausgegeben werden, je nachdem, wie es die Kommunikation verlangt. Ferner können durch die zusätzlichen 128 Charakter zum Beispiel Anweisungen für Graphik-Displays generiert werden.

K 6 Datenaustausch über die serielle Schnittstelle

K 6.1 Einleitung, Kommunikationsmodi

Im vorangehenden Kapitel wurde die Text-Ein-/Ausgabe behandelt. Während die Text-Eingabe eine reine "Off-Line"-Funktion ist (ähnlich wie das Eingeben des Anwendungsprogrammes in den Speicher), erfolgt die Textausgabe "On-Line". Die Texte werden vom Anwenderprogramm, z.B. aufgrund eines Eingangssignals ausgelöst und über die serielle Schnittstelle ausgegeben. D.h., es handelt sich um Einweg-Informationsbetrieb von der SPS auf ein Peripheriegerät (typische Anwendung: protokollieren). Ein Dialog zwischen Peripheriegerät und SPS ist mit der Text-Ausgabe allein noch nicht möglich.

Im vorliegenden Kapitel werden nun Betriebsarten der seriellen Schnittstelle behandelt, die einen echten Dialog (= Informationsaustausch in beiden Richtungen) zwischen folgenden Geräten ermöglichen:

- PCA14 *) mit "nicht-intelligentem" Peripheriegerät wie TTY oder Video-Terminal
- PCA14 *) mit weiterer SPS, z.B. PCAØ, PCA14, PCA23
- PCA14 *) mit anderem "intelligenten" System (z.B. Computer).

Dieser Informationsaustausch wird in folgende Betriebsarten unterteilt:

Modus C Austausch einzelner ASCII-Charakter

Der Modus C eignet sich vor allem für Dialoge zwischen der PCA und einem personenbedienten Peripherie-Gerät (TTY oder Video-Terminal). Ueber die serielle Schnittstelle lassen sich 256 verschiedene Zeichen (davon 12Ø ASCII-Zeichen) austauschen, denen verschiedene Anwender-Funktionen zugeordnet werden können. In der PCA werden diese Charakter in einem definierten Merker-Feld bereitgestellt bzw. abgeholt.

Modus N Austausch numerischer Daten in Telegrammform nach DIN 66Ø19

Im Modus N vollzieht sich der Informationsaustausch gemäss DIN 66Ø19. Die Uebertragungssicherheit ist dank der Stromschleifentechnik und der Summenkontrolle (BCC) ausserordentlich hoch. Das Anwendungsgebiet erweitert sich beträchtlich, da in einem Telegramm bis zu 31 numerische Charakter übermittelt werden können. So können z.B. von einer Lesestation codierte Informationen übernommen, gespeichert und ausgewertet werden. Auch kann mit einem anderen "intelligenten" System korrespondiert werden.

Eine Variante, welche den numerischen Datenaustausch auch mit einem personenbedienten Peripheriegerät ohne "Intelligenz" erlaubt, wurde für einfachere Installationen geschaffen. In der Programmier- und Inbetriebsetzungsphase ist diese Variante ebenfalls sehr nützlich. Im Modus N gibt es drei Varianten: N1, N2 und N3. Nähere Erläuterungen folgen in der Beschreibung.

---

*) oder andere PCA mit serieller Schnittstelle, wie PCAØ2, PCA222, PCA232.

Modus P Austausch von PCA-bezogenen Daten in Telegrammform nach DIN 66Ø19

In den Betriebsmodi C und N besteht indirekt via Anwenderprogramm bereits Zugriff zu Eingängen, Ausgängen, Merkern, Timern und Zählern. Mit dem Datenaustausch nach Modus P erfolgt dieser Zugriff jedoch direkt und wird zudem auf Schrittadressen und Operanden ausgedehnt. Es ergeben sich damit Eingriffsmöglichkeiten in die PCA durch ein übergeordnetes intelligentes System. Die PCA ist dabei immer passiv. Es werden die zwei Varianten: P1 und P2 unterschieden.

Kombination verschiedener Betriebsarten

Interessante Aspekte ergeben sich durch die Kombination der verschiedenen Betriebsmodi unter sich sowie kombiniert mit der reinen Textausgabe.

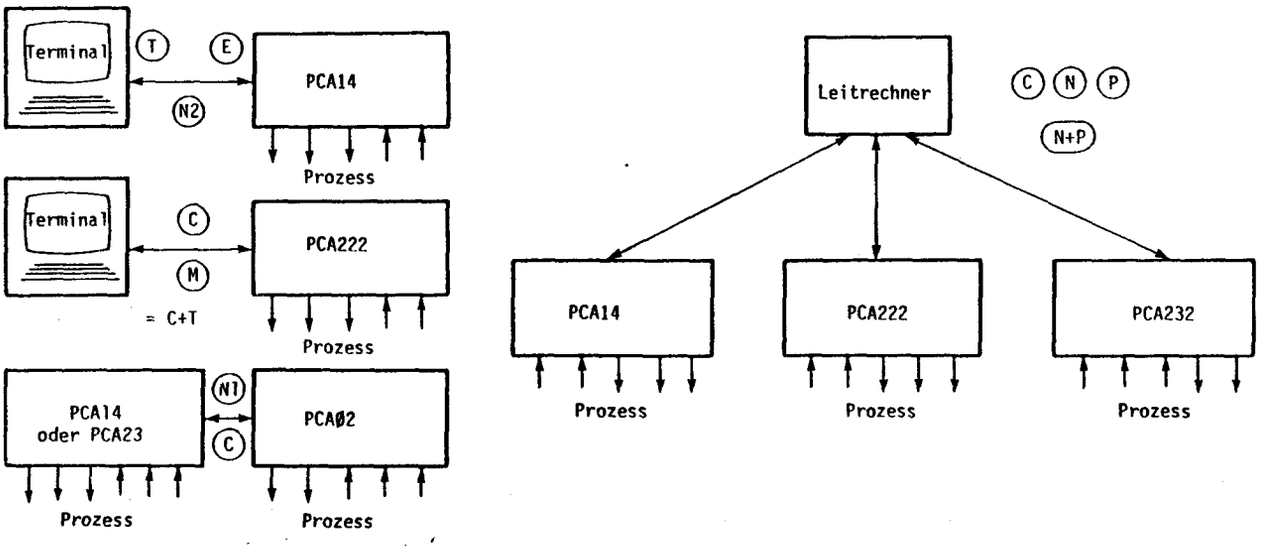
Menü-Modus = Kombination von Textausgabe mit Modus C (Einzelcharakter)

Mit dieser Mischung kann ein lebendiger Dialog mit Klartextausgaben zwischen einem bedienten Peripheriegerät und der PCA aufgebaut werden (Bedienerführung). Mittels Text wird ein Menü ausgegeben. Durch Betätigen von Tasten werden Einzelcharakter eingelesen und entsprechende Funktionen in der PCA ausgelöst.

Computer-Modus = Kombination des Modus N mit Modus P

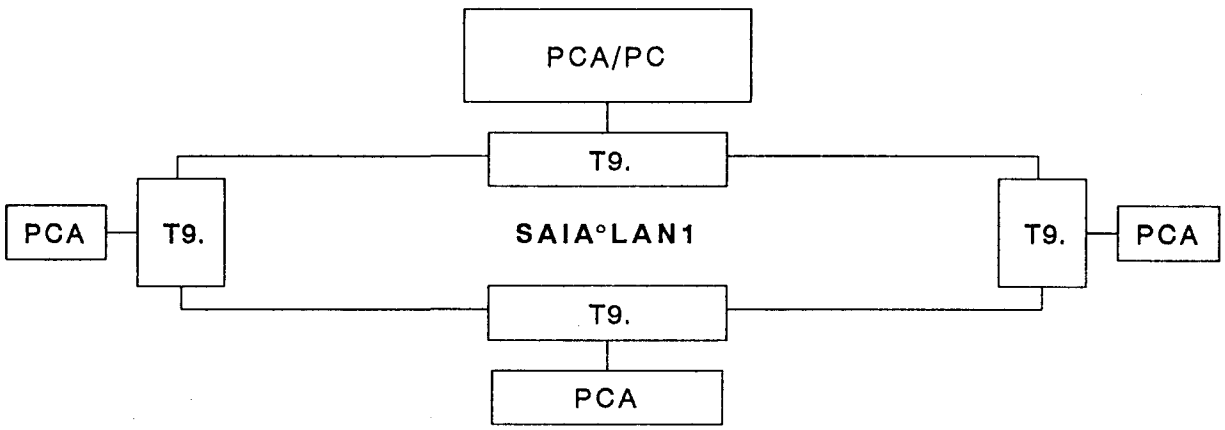
Der Austausch numerischer und PCA-bezogener Daten ergibt kombiniert viele Möglichkeiten der Einflussnahme eines übergeordneten Systems auf die PCA und andererseits von Rückmeldungen an das hierarchisch höhere System. Die Steuerungsstruktur der "verteilten Intelligenz" lässt sich auf diese Weise leicht verwirklichen.

Gerätekombination für Punkt-Punkt-Verbindungen



SAIA°LAN 1

Alle Kommunikationsmodi inkl. Textausgabe lassen sich auch im lokalen Netzwerk SAIA°LAN 1 anwenden. Nach der Durchwahl von einer Ausgangsstation zu einer Zielstation kann die Kommunikation vollkommen transparent wie bei direkter Punkt-Punkt-Verbindung durchgeführt werden. Bitte verlangen Sie dazu die speziellen Unterlagen zu SAIA°LAN 1.



Zusammenfassung der Kommunikationsmodi für die serielle Schnittstelle

Textein- und ausgabe in/von Textspeicher

Modus EDITOR	(E)	Texteingabe off-line mittels Editor	→	Text-Speicher
Modus TEXT	(T)	Textausgabe gemäss Anwenderprogramm	←	Text-Speicher

---

Datenaustausch mit Merkerfeld (Flag-Buffer)

Modus C	(C)	Ein/Ausgabe einzelner Charakter	→ ←	Merker-Buffer
Modus N (1/2/3)	(N)	Ein/Ausgabe numerischer Daten in Telegramm-Format nach DIN 66Ø19	→ ←	Merker-Buffer

C und N sind vom Anwenderprogramm abhängig (Merker-Buffer)

Direktverkehr mit PCA

Modus P (1/2)	(P)	Direktein-/ausgabe von PCA-Daten in Telegramm-Format nach DIN 66Ø19	→ ←	PCA PCA

P ist vom Anwenderprogramm unabhängig (Master-Slave-Betrieb)

Bedienerführung

Menü-Modus (T+C)	(M)	Eingabe von Einzelcharaktern	→ (C)	Merker-Buffer
		Ausgabe von Einzelcharaktern und/oder Texten	← (C)	Text-Speicher
			← (T)	Text-Speicher

Kommunikation mit übergeordnetem System

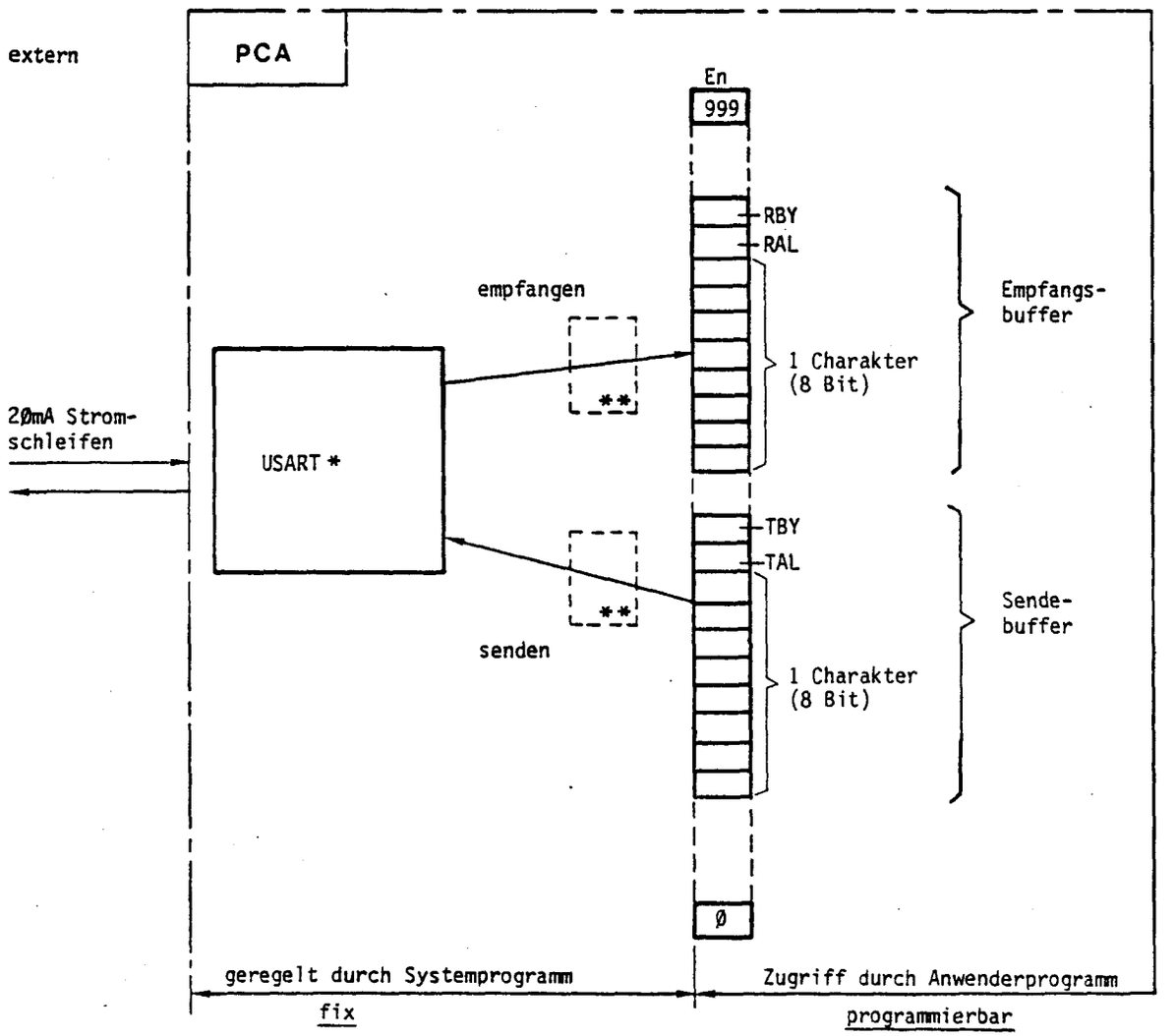
Computer-Modus (N+P)		Ein-/Ausgabe von Daten mit Direktzugriff	→ (P) ←	PCA PCA
		Ein-/Ausgabe von numerischen Daten via Merkerfeld	→ (N) ←	Merker-Buffer

K 6.2 Definition des Modus C

Beim Modus C handelt es sich um die Ein- und Ausgabe einzelner ASCII-Charakter. Um die serielle Schnittstelle der PCA für diese Art der Uebertragung zu aktivieren, sind im Systemprogramm entsprechende Programmpakete enthalten. Diese regeln die Interpretation des ankommenden Charakters und die Formatierung des zu sendenden Charakters. Durch das Systemprogramm wird ein Charakter auf Element-Adressen (meist Merker) übertragen, zu welchen über das Anwenderprogramm zugegriffen werden kann.

Wie die nachfolgende Figur zeigt, muss der Anwender im Merker-Bereich Adressen für den Empfangs-Buffer (Charakter-Einleseplatz) und für den Sende-Buffer (Charakter-Ausgabeplatz) definieren.

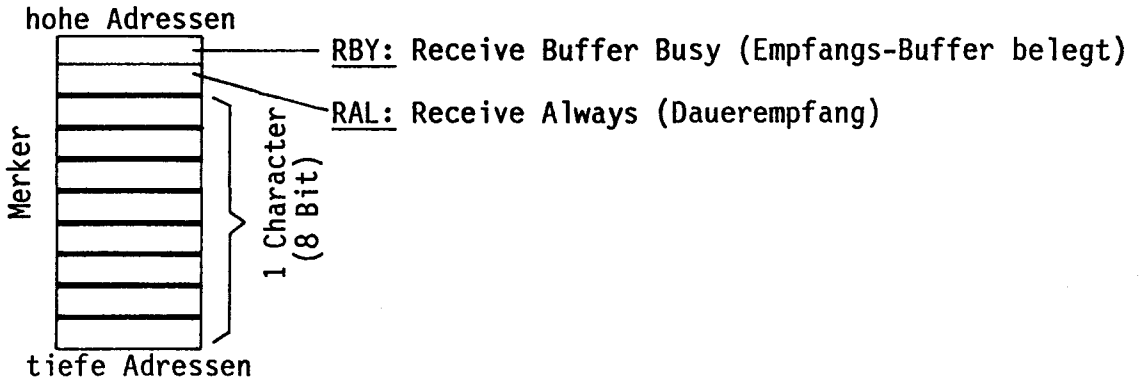
Untenstehendes Schema verdeutlicht diese Zusammenhänge:



*) USART = Universal Synchronous/Asynchronous Receiver Transmitter.

**) Genau genommen gelangt das Telegramm zuerst in einen Zwischenspeicher. Dort wird es auf Richtigkeit geprüft und bei positivem Ergebnis durch das Systemprogramm auf die entsprechenden Merkerbuffer übertragen.

Empfangs-Buffer für den Modus C



Wenn beim Empfang keine Uebertragungsfehler (wie Parität, Framing, etc.) festgestellt wurden, so erfolgt die Abspeicherung des Charakters im Empfangsbuffer in Abhängigkeit des logischen Zustandes von RBY und RAL:

	RAL	RBY	1. Folge	2. Folge
Dauer-Empfang	H	L	Zwischenspeicherung im Buffer	→ RBY wird H
	H	H		
Einzel-Emfpang	L	(L)	Zwischenspeicherung im Buffer	→ RBY wird H
	L	H	kein Effekt	--

Dauerempfang:
Wird durch das Anwenderprogramm RAL = H gesetzt, so werden (bei fehlerfreiem Empfang) die Charakter jederzeit im Empfangs-Buffer abgelegt.

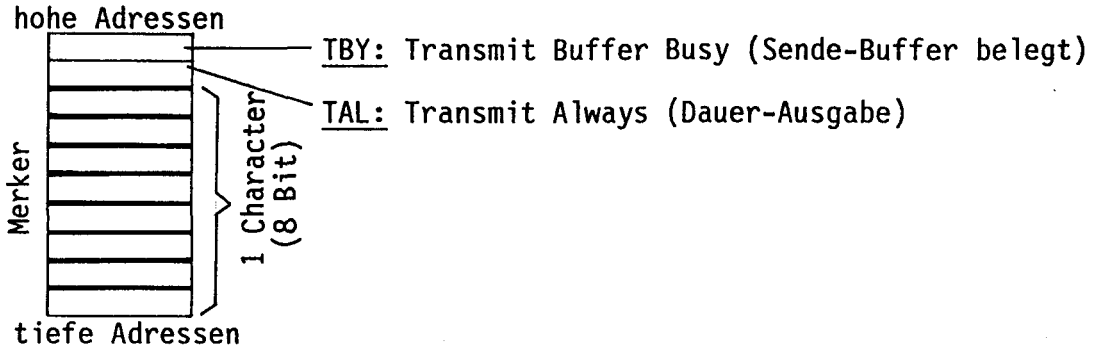
Einzelempfang:
Wird jedoch RAL = L gesetzt, so wird (bei fehlerfreiem Empfang) der Charakter nur dann im Buffer abgelegt, wenn auch RBY = L ist. Anschliessend wird automatisch RBY = H gesetzt. Dadurch wird verhindert, dass ein neuer Charakter den Empfangs-Buffer überschreibt. Das Anwenderprogramm kann somit zuerst die Buffer-Bits verarbeiten, bis durch REO RBY der nächste Charakter aufgenommen werden kann.

N.B.: Durch das Systemprogramm selber wird kein Echo erzeugt. Wird ein Echo verlangt, so muss diese Rückantwort vom Anwenderprogramm erzeugt werden.

Programmstruktur für Einzelempfang: (RAL = L)

```
        WIL   RBY   ; Warten, wenn neuer Charakter noch nicht eingetroffen ist.

      | Lesen des |
      | Empfangs- | ; Der eingetroffene Character wird durch das Anwender-
      | Buffers   |   programm gelesen (ev. in Zähler transferiert).

        REO   RBY   ; E-Buffer ist frei für den Empfang eines neuen Charakters.
```

Sende-Buffer für den Modus C



Jede 1/1Øs bzw. 1/1ØØs (abhängig von der gewählten Zeitbasis) kann dem USART die Ausgabe eines Charakters befohlen werden. Dies erfolgt in Abhängigkeit der Merker-Zustände TBY und TAL:

	TAL	TBY	1. Folge	2. Folge
Dauer-Ausgabe	H	L	Charakter-Ausgabe ab Buffer	
	H	H	→	TBY wird L
Einzel-Ausgabe	L	L	kein Effekt	--
	L	(H)	Charakter-Ausgabe →	TBY wird L

Dauer-Ausgabe:

Wird durch das Anwenderprogramm TAL = H gesetzt, so wird jede 1/1Øs bzw. 1/1ØØs der durch den Buffer-Inhalt gebildete Charakter ausgegeben.

Einzelausgabe:

Wird jedoch TAL = L gesetzt, so wird der im Sende-Buffer anstehende Charakter nur dann gesendet, wenn das Anwenderprogramm TBY = H setzt. Das Systemprogramm setzt dabei automatisch TBY = L, um dem Anwenderprogramm das Ende der Uebertragung zu melden (sog. Handshake).

Das Systemprogramm erwartet keine Rückantwort auf die Charakter-Ausgabe.

Programmstruktur für Einzel-Ausgabe: (TAL = L)

```
WIH   TBY  ; Warten, wenn vorangehende Sendung noch nicht beendet ist.

Füllen des
Sende-
Buffers    ; Durch das Anwenderprogramm wird ein Character (8 Bit)
           ; auf den Sende-Buffer übertragen.

SEO   TBY  ; Senden des Characters ab Sende-Buffer
```

K 6.2.1 Assignierung für Modus C

```
 1:  PAS (29)    1ØØ ; Aktivierung der Schnittstelle
 2:       ØØ    xxxx ; Baudrate, Parity und Stopbits
 3:       ØØ     xxx ; TXB (Text-Busy Flag)        } Kennzeichen des Modus C
 4:       ØØ     xxx ; RBY (Empfangsbuffer)        }
 5:       ØØ     xxx ; TBY (Sendebuffer)           }
 6:       ØØ      Ø  ; kein BCC möglich
 7:       ØØ      Ø  }
 8:       ØØ      Ø  } leer
 9:       ØØ      Ø  }
1Ø:       ØØ      Ø  }
```

- Zeile 3: TXB (Text-Busy Flag)

  Für xxx ist die Adresse eines freien Merkers (oder Ausgangs) einzusetzen, der beim Einschalten der PCA "L" ist.

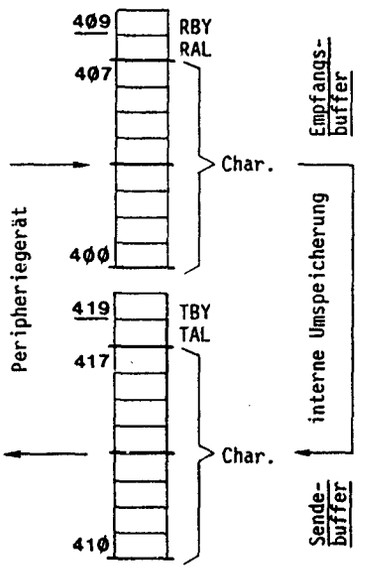
- Zeile 4: RBY (Empfangsbuffer)

  Für den Empfangsbuffer müssen 1Ø zusammenhängende freie Merker (evtl. Ausgänge) reserviert werden, wobei xxx die höchste Adresse ist.

- Zeile 5: TBY (Sendebuffer) siehe RBY, Zeile 4

  Es ist immer ein Empfangs- und ein Sendebuffer zu definieren, auch wenn nur einer davon gebraucht werden sollte.

Beispiel: Erzeugung eines "Echo" durch das Anwenderprogramm. Der am Empfangsbuffer eintreffende Charakter wird auf den Sendebuffer übertragen und erscheint dadurch auf dem Bildschirm.



Programm (für Bildschirm VC 44Ø4)

```
 Ø  NOP    Ø  ;                          }
 1  PAS   1ØØ  ;                         }
 2   ØØ   9Ø3  ; (96ØØ Baud)              }
 3   ØØ   399  ; TXB                      }
 4   ØØ   4Ø9  ; RBY                      }
 5   ØØ   419  ; TBY                      }
 6   ØØ    Ø   ;                          }
 .    .    .                              }
 .    .    .                              }
1Ø   ØØ    Ø  ;                          }
11  WIL   4Ø9  ; Empfang frei?            }
12  SCR   256  ;   E-Buffer               }
13   24   4Ø7      lesen                  }
14  REO   4Ø9  ; E-Buffer frei!           }
15  WIH   419  ; Sender frei?             }
16  SCR   256      S-Buffer               }
17   21   417      einschreiben           }
18  SEO   419  ; senden                   }
19  JMP    11  ;                          }
```

Assignierung (nur Initialisierung beim Einschalten)

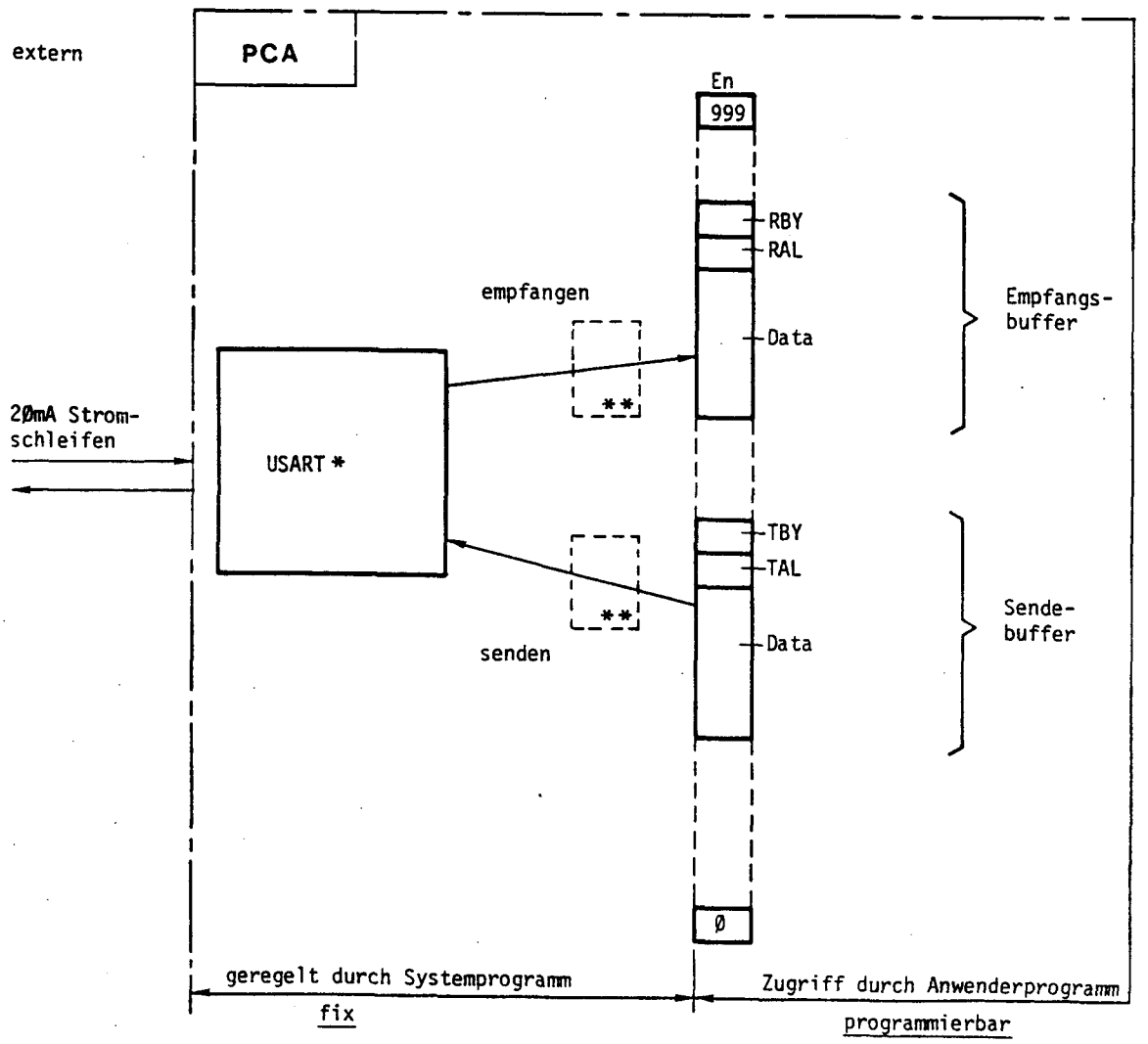
Programm

K 6.3 Definition des Modus N

Beim Modus N handelt es sich um den Austausch numerischer Daten in Telegrammform nach DIN 66Ø19. Um die serielle Schnittstelle der PCA für diese Art der Uebertragung zu aktivieren, sind im Systemprogramm entsprechende Programmpakete enthalten. Diese regeln die Interpretation der ankommenden Informationen und die Formatierung der gesendeten Informationen. Durch das Systemprogramm werden die Informationen auf Element-Adressen (meist Merker) übertragen, zu welchen über das Anwenderprogramm zugegriffen werden kann.

Wie die nachfolgende Figur zeigt, muss der Anwender im Merker-Bereich Adressen für den Empfangs-Buffer (Daten-Einleseplatz) und für den Sende-Buffer (Daten-Ausgabeplatz) definieren.

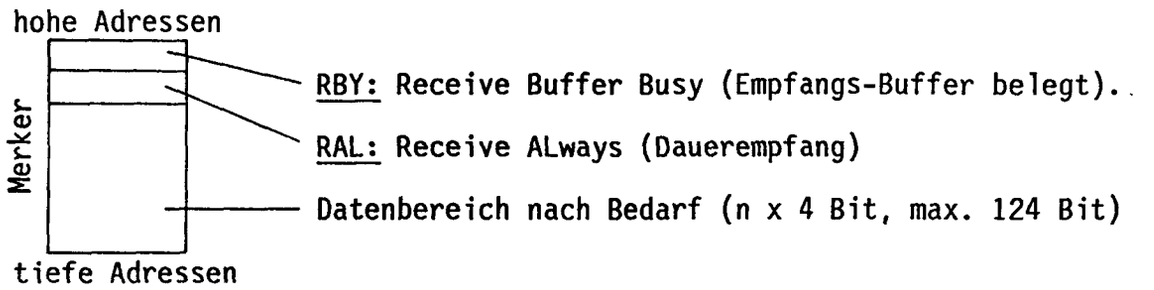
Die Grösse des Empfangs- und des Sende-Buffers ist variabel und je nach Anwendung zu definieren.



*) USART = Universal Synchronous/Asynchronous Receiver Transmitter.

**) Genau genommen gelangt das Telegramm zuerst in einen Zwischenspeicher. Dort wird es auf Richtigkeit geprüft und bei positivem Ergebnis durch das Systemprogramm auf die entsprechenden Merkerbuffer übertragen.

Empfangs-Buffer für den Modus N



Wenn beim Empfang keine Uebertragungsfehler (wie Parität, Framing, etc.) festgestellt wurden, so erfolgt die Abspeicherung der Daten im Empfangsbuffer in Abhängigkeit des logischen Zustandes von RBY und RAL:

	RAL	RBY	1. Folge	2. Folge
Dauer-Empfang	H	L	Zwischenspeicherung im Buffer →	RBY wird H
	H	H		
Einzel-Emfpang	L	(L)	Zwischenspeicherung im Buffer →	RBY wird H
	L	H	kein Effekt	--

Dauerempfang:

Wird durch das Anwenderprogramm RAL = H gesetzt, so werden (bei fehlerfreiem Empfang) die Daten jederzeit im Empfangs-Buffer abgelegt.

Einzelempfang:

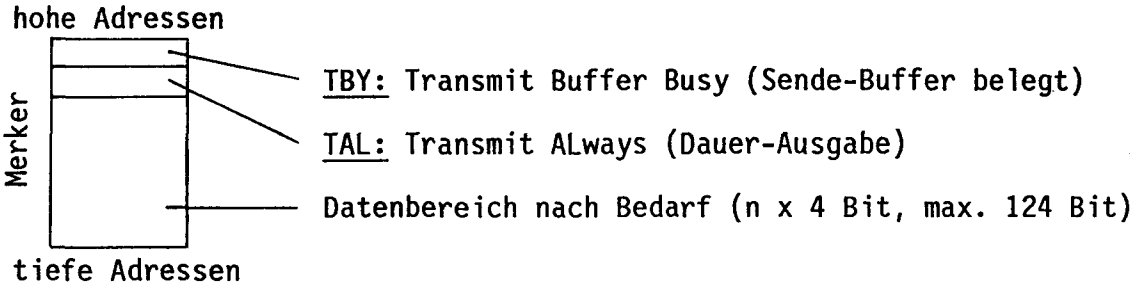
Wird jedoch RAL = L gesetzt, so werden (bei fehlerfreiem Empfang) die Daten nur dann im Buffer abgelegt, wenn auch RBY = L ist. Anschliessend wird automatisch RBY = H gesetzt. Dadurch wird verhindert, dass neue Daten den Empfangs-Buffer überschreiben. Das Anwenderprogramm kann somit zuerst die Buffer-Bits verarbeiten, bis durch REO RBY die nächsten Daten aufgenommen werden können.

Programmstruktur für Einzelempfang: (RAL = L)

```
        WIL   RBY ; Warten, wenn neue Daten noch nicht eingetroffen sind.

  Lesen des
  Empfangs-   ; Die eingetroffenen Daten werden durch das Anwender-
  Buffers       programm gelesen (und ev. in Zähler-Register transferiert).

        REO   RBY ; E-Buffer ist frei für den Empfang von neuen Daten.
```

N.B.: RBY wird H mit dem ersten empfangenen Zeichen; daher muss nach dem WIL sichergestellt werden, dass alle Zeichen empfangen wurden. Die Wartezeit ist abhängig von der Uebertragungsrate.

Sende-Buffer für den Modus N



Jede 1/10s bzw. 1/100s (abhängig von der gewählten Zeitbasis) kann dem USART die Ausgabe von Daten befohlen werden. Dies erfolgt in Abhängigkeit der Merker-Zustände TBY und TAL:

	TAL	TBY	1. Folge	2. Folge
Dauer-Ausgabe	H	L	Daten-Ausgabe ab Buffer →	TBY wird L
	H	H		
Einzel-Ausgabe	L	L	Kein Effekt	
	L	(H)	Daten-Ausgabe ab Buffer →	TBY wird L

Dauer-Ausgabe:

Wird durch das Anwenderprogramm TAL = H gesetzt, so werden jede 1/10s bzw. 1/100s die durch den Buffer-Inhalt gebildeten Daten ausgegeben.

Einzelausgabe:

Wird jedoch TAL = L gesetzt, so werden die im Sende-Buffer anstehenden Daten nur dann gesendet, wenn das Anwenderprogramm TBY = H setzt. Das Systemprogramm setzt dabei automatisch TBY = L, um dem Anwenderprogramm das Ende der Uebertragung zu melden (sog. Handshake).

Programmstruktur für Einzel-Ausgabe: (TAL = L)

```
WIH   TBY   ; Warten, wenn vorangehende Sendung noch nicht beendet ist.

Füllen des
Sende-
Buffers     ; Durch das Anwenderprogramm werden die zu sendenden
              Daten auf den Sende-Buffer übertragen.

SEO   TBY   ; Senden des Bufferinhaltes
```

N.B.: TBY wird L beim ersten gesendeten Zeichen; daher muss nach dem WIH sichergestellt werden, das alle Zeichen gesendet wurden. Die Wartezeit ist abhängig von der Uebertragungsrate.

K 6.3.1 Assignierung für Modus N

```
 1:  PAS (29)   1ØØ ; Aktivierung der Schnittstelle

 2:       ØØ   xxxx ; Baudrate, Parity und Stopbits

 3:       ØØ    xxx ; TXB (Text-Busy Flag)
 4:        n*   xxx ; RBY (Empfangsbuffer)
 5:        n*   xxx ; TBY (Sendebuffer)
 6:       ØØ      d ; ohne BCC (Var. N2)
          Ø1      d ; mit  BCC (Var. N1)

 7:       ØØ      Ø
 8:       ØØ      Ø
 9:       ØØ      Ø
1Ø:       ØØ      Ø
```

Kennzeichen des Modus N

leer

- Zeile 3: TXB (Text-Busy Flag)

  Für xxx ist die Adresse eines freien Merkers (oder Ausgangs) einzusetzen, der beim Einschalten der PCA "L" ist.

- Zeile 4: RBY (Empfangsbuffer)*

  n = Ø1 bis 31 entsprechend einem Datenbuffer von 4 x n Bit (d.h. 4 bis 124 Bit).
  Total müssen 2 + 4 x n zusammenhängende freie Merker (evtl. Ausgänge) reserviert werden, wobei xxx die höchste Adresse ist (RBY).

- Zeile 5: TBY (Sendebuffer)* siehe RBY, Zeile 4

  Die Sende- und Empfangsbuffer können auch verschieden gross sein.

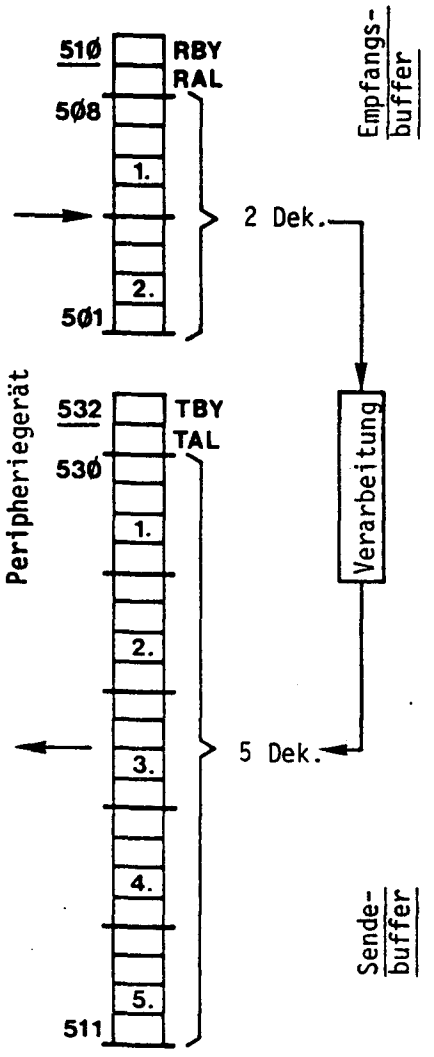
- Zeile 6:

  Die Zeile 6 hat eine mehrfache Bedeutung: Mit dem Code wird festgelegt, ob mit oder ohne BCC gearbeitet wird:
  Code Ø1: mit BCC (bedingt, dass das Peripheriegerät BCC verarbeiten kann)
  Modus-Variante N1
  Code ØØ: ohne BCC (Betrieb mit einem "nicht intelligenten" Peripheriegerät)
  Modus-Variante N2

  Bei Operand d = Ø wird jede 3Ø. Zeiteinheit (alle 3s bei 1/1Øs) das nicht quittierte Telegramm wiederholt.
  Bei Operand d = 1 wird das Telegramm nur einmal gesendet, unabhängig davon ob quittiert oder nicht quittiert. Fehlt die Quittierung, dann bleibt TBY auf "H". Time-out-Ueberwachung erforderlich.
  Bei Operand d = 3 wird das Telegramm ebenfalls nur einmal gesendet, wobei die PCA anschliessend keine Rückantwort erwartet. Modus-Variante N3.

*) Wird nur einer der beiden Buffer verwendet (nur Empfang oder nur Senden), so muss auch der nicht benützte Buffer dennoch definiert sein, z.B. Ø1 399, d.h. mit der Minimalgrösse von 2 + 4 Bit von 394...399.

Beispiel: (Modus N2) Empfangen eines 2-stelligen Wertes, Verarbeitung desselben und Senden des 5-stelligen Resultates. Da die Korrespondenz mit einem "nicht intelligenten", bedienten Gerät erfolgen soll, wird gemäss Modus N2 auf den BCC verzichtet. Vor jeder Dateneingabe muss CTRL/B (für STX) und nach den Daten CTRL/C (für ETX) getippt werden, um das Telegramm zu senden.



Programm (für SILENT 743)

```
Ø   NOP   Ø
1   PAS   1ØØ
2    ØØ   898  ; (3ØØ Baud)
3    ØØ    32  ; TXB
4    Ø2   51Ø  ; RBY
5    Ø5   532  ; TBY
6    ØØ    Ø  ; kein BCC
7    ØØ    Ø
8    ØØ    Ø
9    ØØ    Ø
1Ø   ØØ    Ø
11  WIL   51Ø  ; Empfang frei?
12  SCR   256  }  E-Buffer
13   16   5Ø8  }  übertragen
14  REO   51Ø  ; E-Buffer frei!
15  WIH   532  ; Sender frei?
16  SCR   256  }  x 64
17   29    64  }
18  SCR   256  }  Resultat auf
19   2Ø   53Ø  }  S-Buffer
2Ø  SCR   256  }  Resultat auf
21   2Ø    63  }  Ausgänge
22  SEO   532  ; senden
23  JMP    11
```

Assignierung

Programm

K 6.3.2 Die Telegramme des Modus N

a) Steuercharakter

In einem Telegramm werden verschiedene Steuercharakter aus dem "Control Case" der ASCII-Tabelle verwendet. Die nachfolgende Zusammenstellung gibt einen Ueberblick:

Kurz-zeichen	Benennung e	Benennung d	Platz bzw. Dez.-Nr.	CTRL/..
STX	Start of Text	Anfang des Textes	2	B
ETX	End of Text	Ende des Textes	3	C
EOT 1)	End of Trans-mission	Ende der Uebertragung	4	D
ENQ 2)	Enquiry	Stationsauf-forderung	5	E
ACK	Acknowledge	Positive Rückmeldung	6	F
NAK	Negative Acknowledge	Negative Rückmeldung	21	U
#	Hash	Nummern-Zeichen	35	

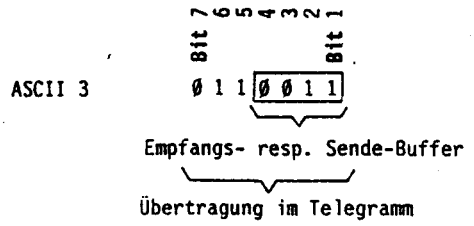
1) EOT bewirkt, dass die sendende PCA in den Ruhestand zurückkehrt.
2) ENQ verlangt von der PCA eine Rückantwort (Telegramm oder NAK) oder bewirkt die Annulierung eines begonnenen Telegrammes.

b) Das übertragene Telegramm

Im Modus N werden ausschliesslich numerische Daten ausgetauscht. Dies sind aus der ASCII-Tabelle die Charakter der Spalte 3, Dezimal Nm 48 bis 63. Die Bit-Nummer 5, 6 und 7 sind für diese Charakter immer dieselben (siehe ASCII-Tabelle).
Die Empfangs- und Sendebuffer werden deshalb mit Gruppen von 4 Bit gebildet, eine Gruppe entspricht somit einem numerischen Charakter.

Beim Senden fügt das System die konstanten Bit 5, 6 und 7 dazu und bildet die Summenkontrolle BCC über alle Datenbytes incl. ETX. Stimmt diese mit derjenigen der empfangenden PCA überein, werden die Daten wiederum ohne die konstanten Bit 5, 6 und 7 auf den Empfangsbuffer übertragen.

z.B.:



c) Was heisst BCC ?

BCC = Block Check Character (Summenkontrolle)

Von einem Telegramm, das mit STX beginnt und mit ETX endet, werden vom Teil, bestehend aus Daten und ETX, die Längssumme gemäss nachstehendem Beispiel gebildet und mit dem gesendeten Wert verglichen. Damit wird eine sehr hohe Uebertragungszuverlässigkeit erreicht.

Telegramm:

STX	Daten					ETX	BCC	
CTRL/B	3	6	9	=	8	CTRL/C	Längssumme aus Daten plus ETX	:
B1 ØØ1Ø	ØØ11	Ø11Ø	1ØØ1	11Ø1	1ØØØ	ØØ11		
B7 ØØØ	Ø11	Ø11	Ø11	Ø11	Ø11	ØØØ	Ø11 1Ø1Ø	

d) Struktur eines Telegrammes

Ein Telegramm besteht aus den folgenden Teilen:

STX = Start des Telegrammes

DATA = Telegramm-Inhalt

ETX = Ende des Telegrammes

BCC = Summenkontrolle des Telegrammes

Telegramm: STX DATA ETX BCC →

Normalerweise muss auf jedes Telegramm eine Rückantwort folgen.
Diese lautet:

ACK = Telegramm korrekt empfangen

NAK = Telegramm nicht korrekt empfangen

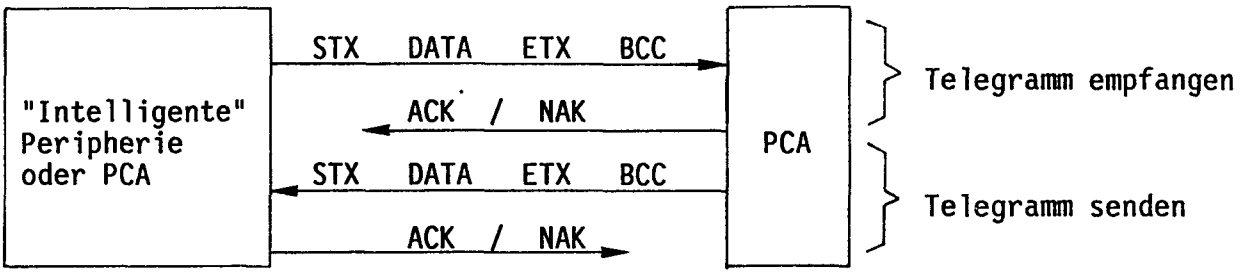
Antwort: ACK bzw. NAK ←

e) Telegrammverkehr mit der PCA im Modus N

Um den verschiedenen Anforderungen gerecht zu werden, stehen dem Anwender neben dem nach DIN 66Ø19 festgelegten Vorgehen verschiedene andere Varianten zur Verfügung.

e1) Modus N1 - Vorgehen gemäss DIN 66Ø19

Die Norm legt den Datenaustausch zwischen 2 "Punkten" fest. Sie setzt voraus, dass die zwei zu korrespondierenden Systeme in der Lage sind, die Charakter STX, ETX sowie die Summenkontrolle BCC, sowie die Verständigungscharakter ACK, NAK, ENQ und EOT zu interpretieren.

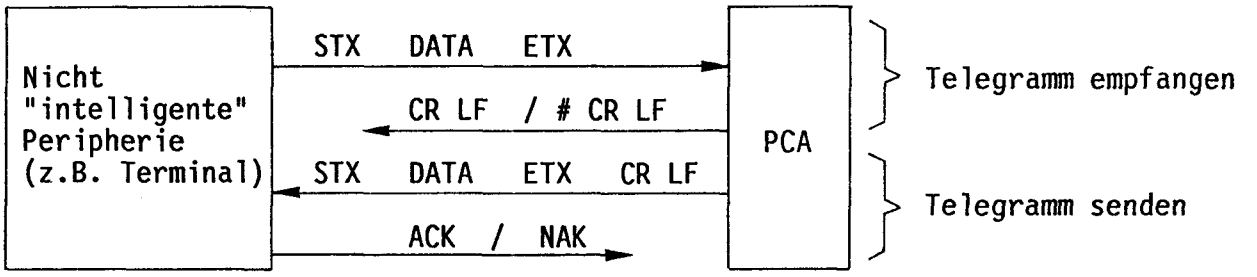


e2) Modus N2

Wird mit einem personenbedienten Terminal korrespondiert, so ist die Bildung des BCC kaum möglich. Um dennoch einen Datenaustausch mit einem einfachen Peripheriegerät zu ermöglichen (z.B. bei der Programm-Erstellung oder zur Inbetriebnahme) wurde eine Variante geschaffen, bei welcher auf die BCC-Kontrolle verzichtet wird und bei welcher die nicht sichtbaren Charakter ACK und NAK ersetzt werden durch

ACK ---> durch "CR LF" (Carriage Return mit Line Feed)

NAK ---> durch # "CR LF".



Festlegen der Variante im PAS 1ØØ-Befehl

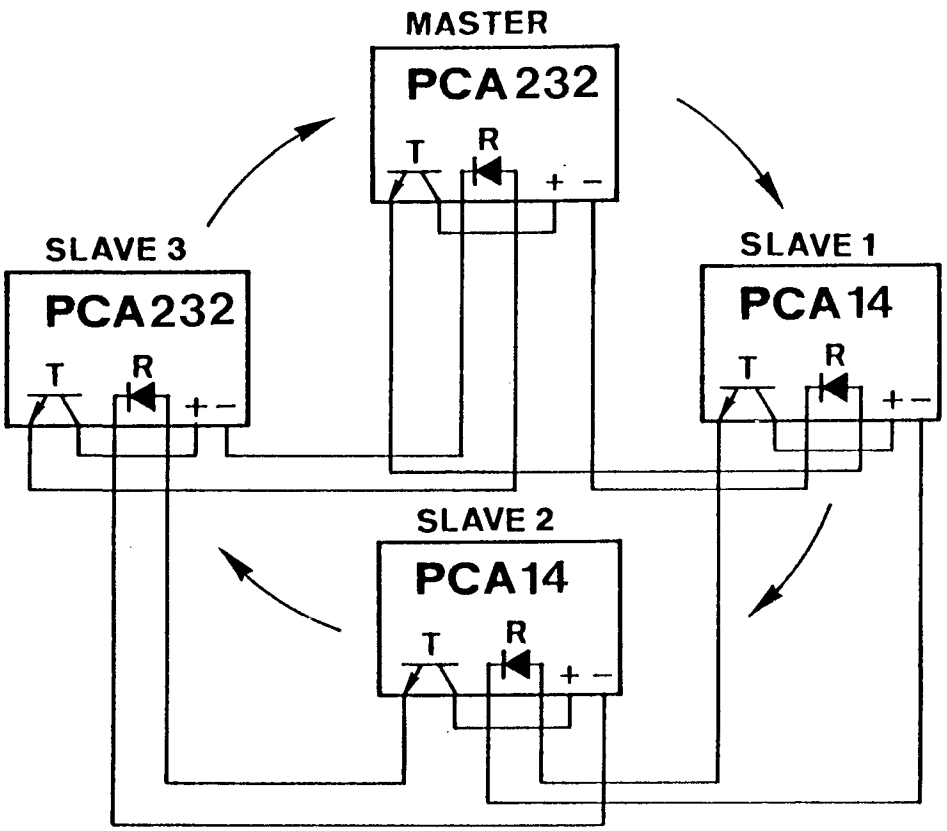
Dies erfolgt in der Zeile 6:

CODE = Ø1 ---> mit BCC, Betrieb mit "intelligentem" Gerät gemäss DIN 66Ø19
---> Modus N1 (Rückanwort nicht sichtbar)

CODE = ØØ ---> kein BCC, Betrieb mit bedientem Peripheriegerät
---> Modus N2 (Rückantwort sichtbar)

e3) Modus N3

Der Modus N3 wurde geschaffen, um in einer Ringstruktur die Datenübertragung über mehrere PLC zu ermöglichen. Es ist dabei von Vorteil, eine PCA als Master zu definieren.



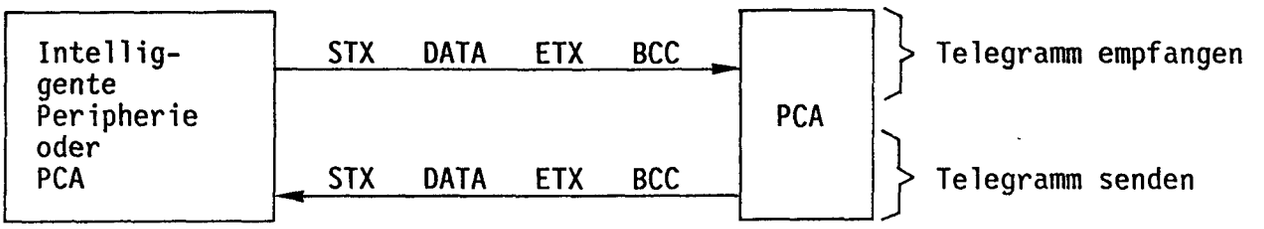
Während bei den Modi N1 und N2 ein durch die PCA empfangenes Telegramm quittiert werden muss (ACK/NAK), ist beim Modus N3 diese Rückmeldung nicht nötig. Dies bedeutet, dass die PCA, welche ein Telegramm gesendet hat, keine Antwort erwartet von dem System, welches das Telegramm empfängt.

Das durch den Ring zurückkehrende Telegramm bestätigt dem Master, dass die Information von allen PCA übertragen wurde.

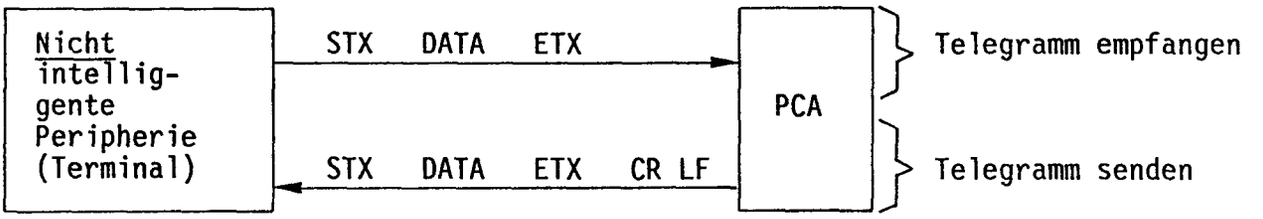
Durch die Summenkontrolle BCC ist eine hohe Uebertragungszuverlässigkeit gewährleistet. Wird ein Uebertragungsfehler festgestellt, so wird das angekommene Telegramm nicht auf den Merkerbuffer übertragen.

Das Telegramm Modus N3

- Mit BCC:



- Ohne BCC:



Die Variante N3 wird im PAS 1ØØ in Zeile 6 festgelegt (siehe Kapitel K 6.3.1).

K 6.4 Definition des Modus P

Beim Modus P handelt es sich um den Direktzugriff in Telegrammform auf PCA-Daten. In den bisher besprochenen Modi C und N erfolgte der Datenaustausch über einen Merkerbuffer. Im Modus P wird durch ein "intelligentes" System direkt auf die PCA-Daten zugegriffen. Der Zugriff erfolgt somit ausserhalb des Wirkungsbereiches des Anwenderprogramms (und somit auch ohne Benützung eines Empfangs- bzw. Sende-Buffers), wobei die PCA die Befehle unbedingt auszuführen hat und nur "ACK" antworten kann (NAK, falls der Befehl nicht ausgeführt werden konnte).

Unter PCA-Daten werden verstanden:

Elemente	(Adr. Ø.... 999)	---> Zeichen E
8 Elemente	(Adr. 7.... 999)	---> Zeichen e
Timer-Register	(Adr. 256.... 287)	---> Zeichen T
5 Timer-Register	(Adr. 26Ø.... 287)	---> Zeichen t [1]
Zähler-Register	(Adr. 256.... 511)	---> Zeichen C [2]
5 Zähler-Register	(Adr. 26Ø.... 511)	---> Zeichen c [3]
Operanden-Register		---> Zeichen O
Schrittadressen	(Adr. Ø... 8191)	---> Zeichen S
2-Byte Textspeicher	(Adr. Ø...15999)	---> Zeichen M [4]
5 mal 2-Byte Textspeicher	(Adr. 4...15999)	---> Zeichen m [4]
1 Registerwort (8 Bit)	(Adr. Ø.... 999)	---> Zeichen R [5]
1 Registerblock (5 x 8 Bit)	(Adr. 4.... 999)	---> Zeichen A [5]
3 Registerblöcke (15 x 8 Bit)	(Adr. 14.... 999)	---> Zeichen B [5]
1Ø Digits Datenspeicher	(Adr. 4...15999)	---> Zeichen N [6]
3Ø Digits Datenspeicher	(Adr. 14...15999)	---> Zeichen n [6]
Zum Schreiben wird verwendet		----------> Zeichen W (write)
Zum Lesen wird verwendet		----------> Zeichen D (display)

[1] PCAØ2 ab Version V6.13Ø
PCA14 ab Version V6.Ø31
PCA222 ab Version V6.23Ø
PCA232 ab Version V7.131

[2] PCAØ2 : Adr. 256...319 (ab Version V6.132 Adr. 256...479)
PCA14 : Adr. 256...319 (ab Version V6.Ø34 Adr. 256...479)
PCA222: Adr. 256...319 (ab Version V6.23Ø Adr. 256...479)
PCA232: Adr. 256...511

[3] PCAØ2 ab Version V6.13Ø (Adr. 26Ø...319, ab Version V6.132 Adr. 26Ø...479)
PCA14 ab Version V6.Ø31 (Adr. 26Ø...319, ab Version V6.Ø34 Adr. 26Ø...479)
PCA222 ab Version V6.23Ø (Adr. 26Ø...479)
PCA232 ab Version V7.131 (Adr. 26Ø...511)

[4] PCAØ2 ab Version V6.13Ø (höchste Adresse 4Ø95)
PCA14 ab Version V6.Ø31 (höchste Adresse 8191)
PCA222 ab Version V6.23Ø (höchste Adresse 8191)
PCA232 ab Version V7.131 (höchste Adresse 15999)

[5] Nur für PCA231, 232.

[6] Die Telegramme N und n existieren nur bei der PCA232, ab Version V7.131. Diese beiden Telegramme sind nur sinnvoll, wenn vorgängig Werte durch den Befehl PAS 57 vom Wortregister in den Datenspeicher transferiert wurden.

K 6.4.1 Assignierung für Modus P

a) PAS 1ØØ-Befehl
Der PAS 1ØØ-Befehl im Modus P ist identisch mit demjenigen im Modus N. Unterschiedlich ist jedoch das Telegramm.

```
 1:  PAS (29)   1ØØ ; Aktivierung der Schnittstelle

 2:       ØØ   xxxx ; Baudrate, Parity und Stopbits

 3:       ØØ    xxx ; Element, das sicher "L" ist

 4:       Ø1    xxx ; Höchste Adresse von 6 Merkern

 5:       Ø1    xxx ; Höchste Adresse von 6 weiteren Merkern

 6:       ØØ      Ø ; Ohne BCC (Var. P2)
          Ø1      Ø ; Mit  BCC (Var. P1)

 7:       ØØ      Ø
 8:       ØØ      Ø   } leer
 9:       ØØ      Ø
1Ø:       ØØ      Ø
```

- Zeile 3: TXB (Text-Busy Flag)
  Für xxx ist die Adresse eines freien Merkers (oder E/A) einzusetzen, der beim Einschalten der PCA "L" ist.

- Zeile 4: RBY (Empfangs-Buffer)
  Wird nur im P-Modus gearbeitet, so müssen 2x6 Merker reserviert werden. Für Computer-Modus (Modus N+P) können die Zeilen 4 und 5 gemäss dem verwendeten N-Modus assigniert werden.

- Zeile 5: TBY (Sende-Buffer)
  Siehe Zeile 4

- Zeile 6: BCC
  Code = ØØ ---> kein BCC, Betrieb mit bedientem Peripheriegerät (Rückantwort sichtbar) ---> Modus P2
  Code = Ø1 ---> mit BCC, Betrieb mit "intelligentem" Gerät (Rückantwort nicht sichtbar) ---> Modus P1

K 6.4.2 Die Telegramme des Modus P

b) Steuercharakter
In einem Telegramm werden verschiedene Steuercharakter aus dem "Control Case" der ASCII-Tabelle verwendet. Die nachfolgende Zusammenstellung gibt einen Ueberblick:

Kurz-zeichen	Benennung e	Benennung d	Platz bzw. Dez.-Nr.	CTRL/..
STX	Start of Text	Anfang des Textes	2	B
ETX	End of Text	Ende des Textes	3	C
EOT 1)	End of Trans-mission	Ende der Uebertragung	4	D
ENQ 2)	Enquiry	Stationsaufford.	5	E
ACK	Acknowledge	Positive Rückmeldung	6	F
NAK	Negative Acknowledge	Negative Rückmeldung	21	U
#	Hash	Nummern-Zeichen	35	-

1) EOT bewirkt, dass die sendende PCA in den Ruhestand zurückkehrt.
2) ENQ verlangt von der PCA eine Rückantwort (Telegramm oder NAK) oder bewirkt die Annulierung eines begonnenen Telegrammes.

c) Was heisst BCC ?

BCC = Block Check Character (Summenkontrolle)

Von einem Telegramm, das mit STX beginnt und mit ETX endet, werden vom Teil, bestehend aus Daten und ETX, die Längssumme gemäss nachstehendem Beispiel gebildet und mit dem gesendeten Wert verglichen. Damit wird eine sehr hohe Uebertragungszuverlässigkeit erreicht.

Telegramm:

STX	Daten					ETX	BCC	
CTRL/B	W	E	Ø	3	2	1	CTRL/C	CTRL/Q
B1 ØØ1Ø	Ø111	Ø1Ø1	ØØØØ	ØØ11	ØØ1Ø	ØØØ1	ØØ11	ØØØ1
B7 ØØØ	1Ø1	1ØØ	Ø11	Ø11	Ø11	Ø11	ØØØ	ØØ1

Längssumme aus Daten plus ETX

c) Struktur eines Telegrammes

Ein Telegramm besteht aus den folgenden Teilen:

STX = Start des Telegrammes

DATA = Telegramm-Inhalt

ETX = Ende des Telegrammes

BCC = Summenkontrolle des Telegrammes

Telegramm: STX DATA ETX BCC →

Normalerweise muss auf jedes Telegramm eine Rückantwort folgen.
Diese lautet:

ACK = Telegramm korrekt empfangen

NAK = Telegramm nicht korrekt empfangen

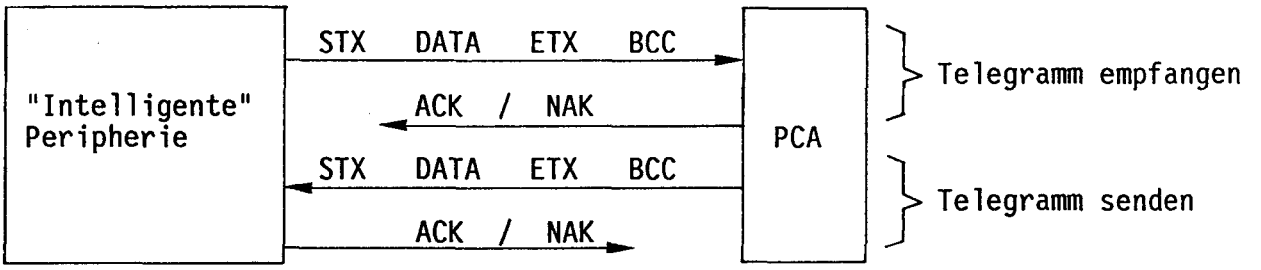
Antwort: ← ACK bzw. NAK

d) Telegrammverkehr mit der PCA im Modus P

Um den verschiedenen Anforderungen gerecht zu werden, stehen dem Anwender neben dem nach DIN 66Ø19 festgelegten Vorgehen verschiedene andere Varianten zur Verfügung.

d1) Modus P1 - Vorgehen gemäss DIN 66Ø19

Die Norm legt den Datenaustausch zwischen 2 "Punkten" fest. Sie setzt voraus, dass die zwei zu korrespondierenden Systeme in der Lage sind, die Charakter STX, ETX sowie die Summenkontrolle BCC, sowie die Verständigungscharakter ACK, NAK, ENQ und EOT zu interpretieren.

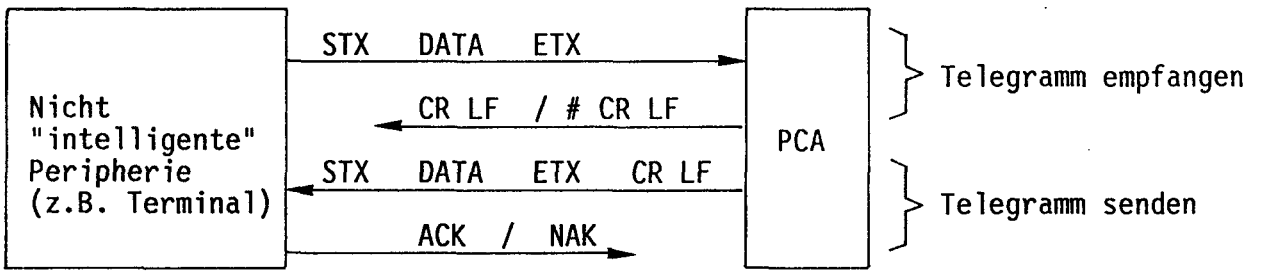


d2) Modus P2

Wird mit einem personenbedienten Terminal korrespondiert, so ist die Bildung des BCC kaum möglich. Um dennoch einen Datenaustausch mit einem einfachen Peripheriegerät zu ermöglichen (z.B. bei der Programm-Erstellung oder zur Inbetriebnahme) wurde eine Variante geschaffen, bei welcher auf die BCC-Kontrolle verzichtet wird und bei welcher die nicht sichtbaren Charakter ACK und NAK ersetzt werden durch

ACK ---> durch "CR LF" (Carriage Return mit Line Feed)

NAK ---> durch # "CR LF".



Festlegen der Variante im PAS 1ØØ-Befehl

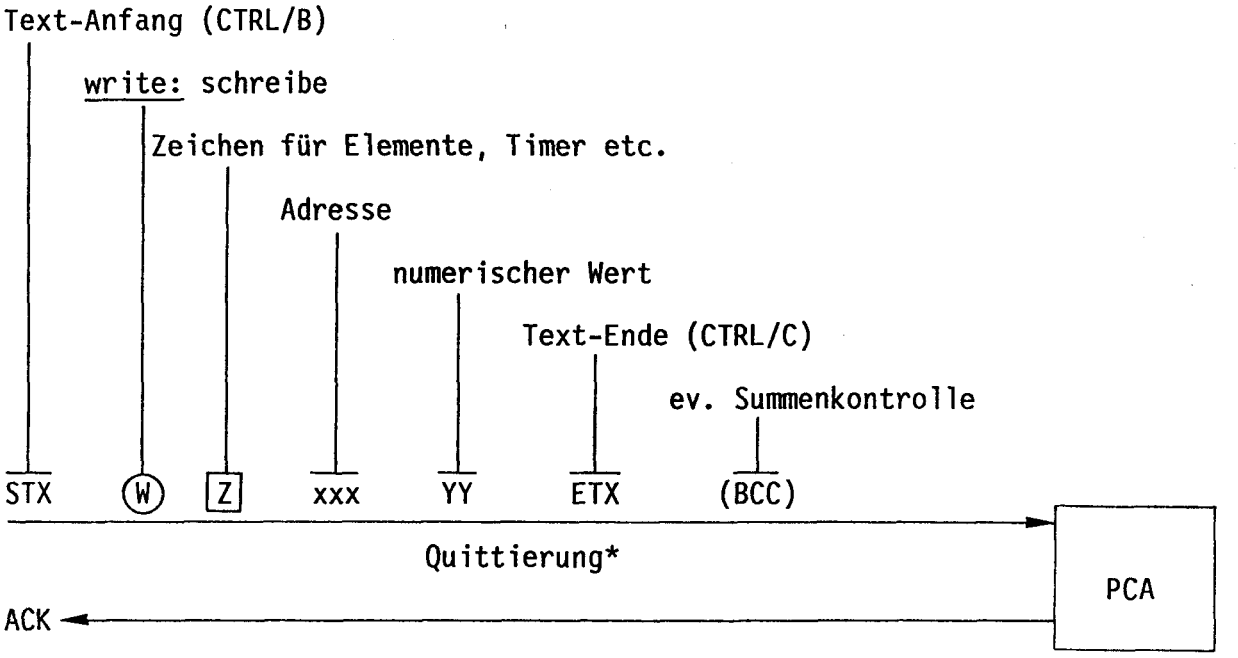
Dies erfolgt in der Zeile 6:

CODE = Ø1 ---> mit BCC, Betrieb mit "intelligentem" Gerät gemäss DIN 66Ø19
---> Modus P1 (Rückanwort nicht sichtbar)

CODE = ØØ ---> kein BCC, Betrieb mit bedientem Peripheriegerät
---> Modus P2 (Rückantwort sichtbar)

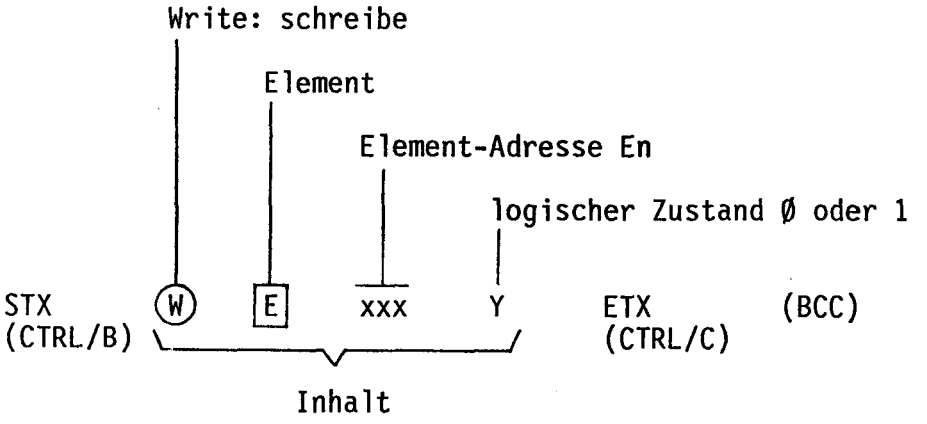
K 6.4.3 Schreiben von Daten in die PCA

Das Telegramm, welches der CPU zu senden ist, lautet allgemein:



positive Anwort der CPU, dass Befehl ausgeführt werden konnte (auch CR LF).
NAK (bzw. # CR LF), falls der Befehl nicht ausgeführt werden konnte.

- Setzen einzelner Elemente En (E, A, T, C, M) bezüglich ihres logischen Zustandes (Ø oder 1)

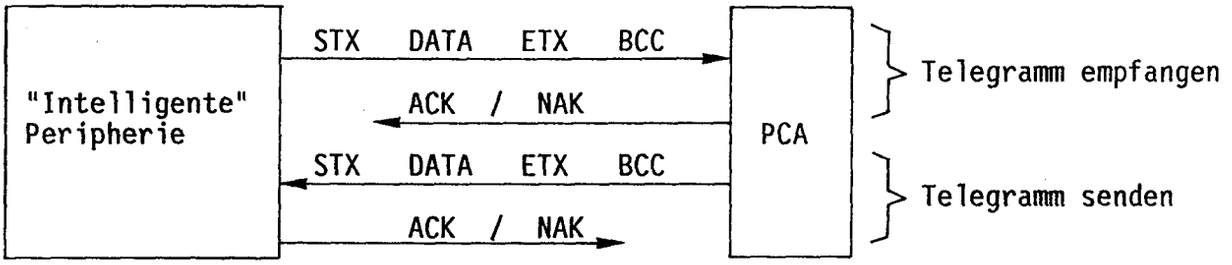


---> setze Element xxx (3 Ziffern) in den logischen Zustand Y (1 Ziffer Ø oder 1)

*) Erfolgt sofort (ohne Abwarten des Zeittaktes).

d1) Modus P1 - Vorgehen gemäss DIN 66Ø19

Die Norm legt den Datenaustausch zwischen 2 "Punkten" fest. Sie setzt voraus, dass die zwei zu korrespondierenden Systeme in der Lage sind, die Charakter STX, ETX sowie die Summenkontrolle BCC, sowie die Verständigungscharakter ACK, NAK, ENQ und EOT zu interpretieren.

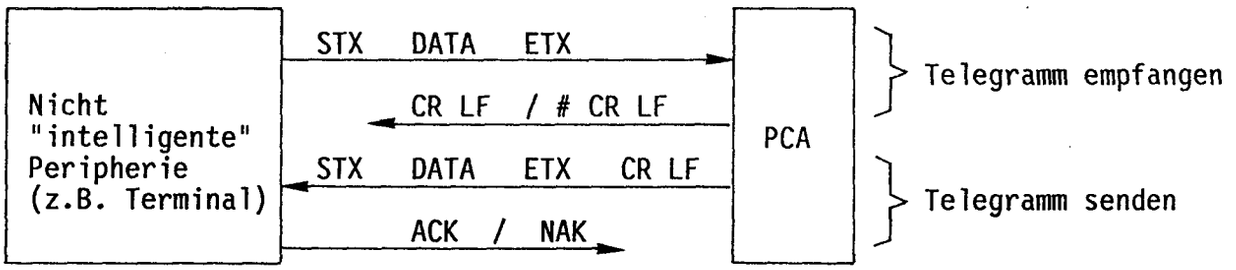


d2) Modus P2

Wird mit einem personenbedienten Terminal korrespondiert, so ist die Bildung des BCC kaum möglich. Um dennoch einen Datenaustausch mit einem einfachen Peripheriegerät zu ermöglichen (z.B. bei der Programm-Erstellung oder zur Inbetriebnahme) wurde eine Variante geschaffen, bei welcher auf die BCC-Kontrolle verzichtet wird und bei welcher die nicht sichtbaren Charakter ACK und NAK ersetzt werden durch

ACK ---> durch "CR LF" (Carriage Return mit Line Feed)

NAK ---> durch # "CR LF".



Festlegen der Variante im PAS 1ØØ-Befehl

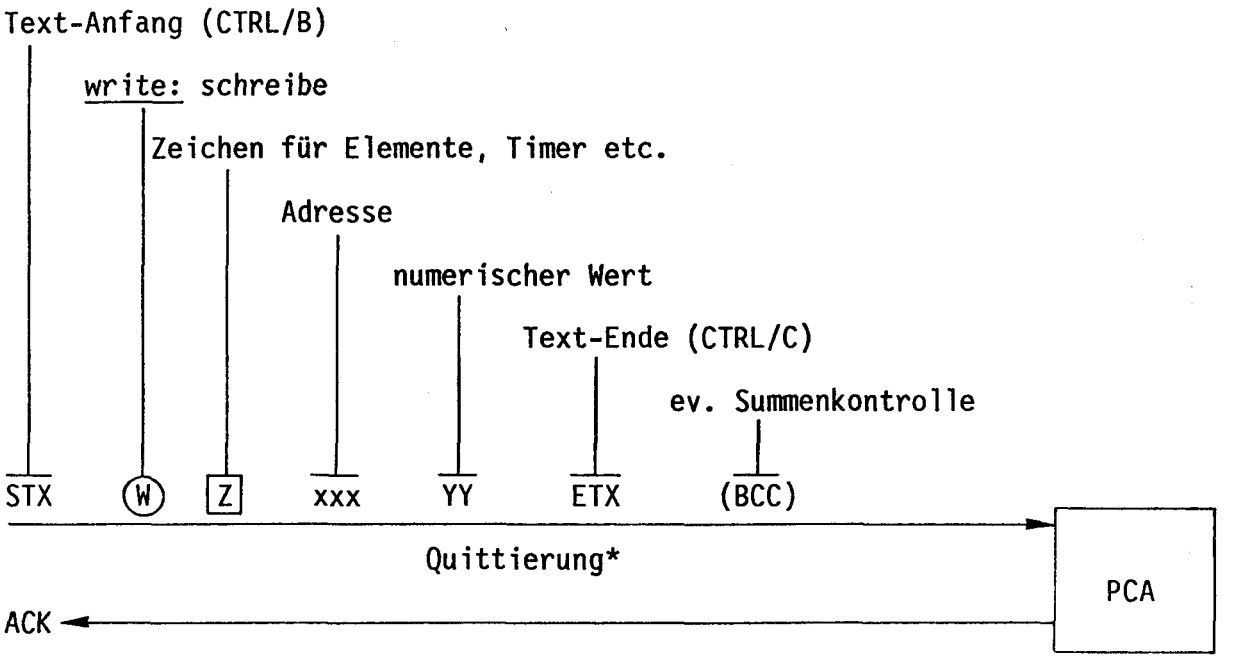
Dies erfolgt in der Zeile 6:

CODE = Ø1 ---> mit BCC, Betrieb mit "intelligentem" Gerät gemäss DIN 66Ø19
---> Modus P1 (Rückanwort nicht sichtbar)

CODE = ØØ ---> kein BCC, Betrieb mit bedientem Peripheriegerät
---> Modus P2 (Rückantwort sichtbar)

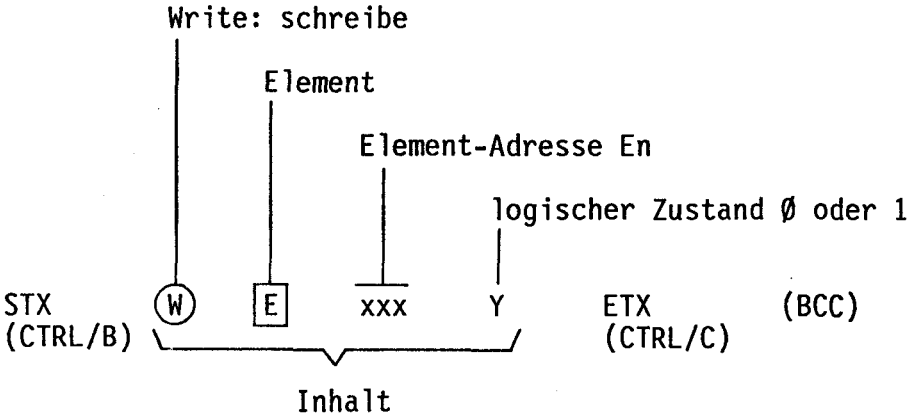
K 6.4.3 Schreiben von Daten in die PCA

Das Telegramm, welches der CPU zu senden ist, lautet allgemein:



positive Anwort der CPU, dass Befehl ausgeführt werden konnte (auch CR LF).
NAK (bzw. # CR LF), falls der Befehl nicht ausgeführt werden konnte.

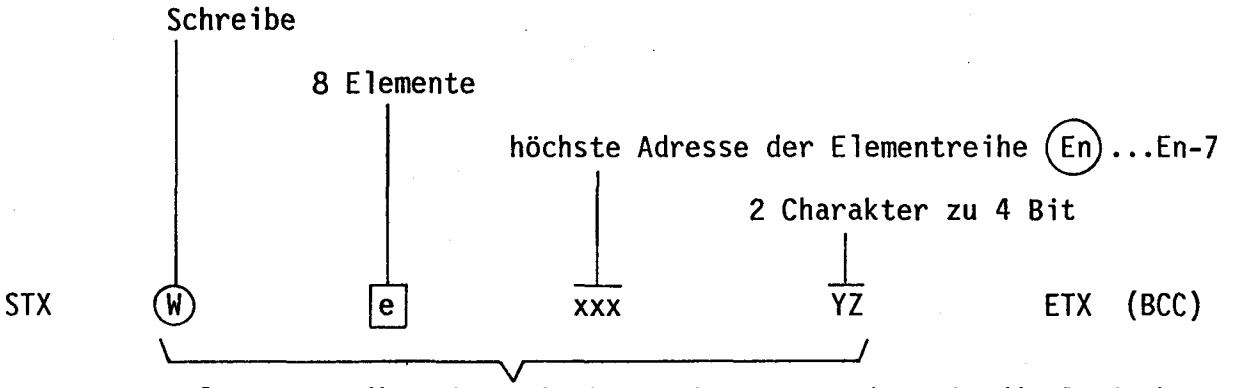
- Setzen einzelner Elemente En (E, A, T, C, M) bezüglich ihres logischen Zustandes (Ø oder 1)



---> setze Element xxx (3 Ziffern) in den logischen Zustand Y (1 Ziffer Ø oder 1)

*) Erfolgt sofort (ohne Abwarten des Zeittaktes).

- Setzen von 8 aufeinanderfolgenden Elementen (A, M) bezüglich ihres logischen Zustandes (Ø oder 1) durch Eingabe von 2 Charaktern zu 4 Bit



---> setze Elementenreihe, deren höchste Adresse xxx ist, in die logischen Zustände, welche durch die Charakter YZ definiert werden.

(Y setzt die Elemente En-4...En-7; Z setzt die Elemente En...En-3)

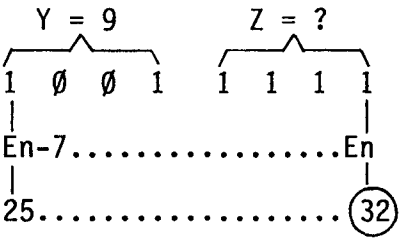
Für Y und Z können folgende Charakter eingegeben werden:

Charakter		Logischer Zustand der Elemente			
	Y	En-7	En-6	En-5	En-4
	Z	En-3	En-2	En-1	En
Ø		Ø	Ø	Ø	Ø
1		Ø	Ø	Ø	1
2		Ø	Ø	1	Ø
3		Ø	Ø	1	1
4		Ø	1	Ø	Ø
5		Ø	1	Ø	1
6		Ø	1	1	Ø
7		Ø	1	1	1
8		1	Ø	Ø	Ø
9		1	Ø	Ø	1
:		1	Ø	1	Ø
;		1	Ø	1	1
<		1	1	Ø	Ø
=		1	1	Ø	1
>		1	1	1	Ø
?		1	1	1	1

Beispiel:

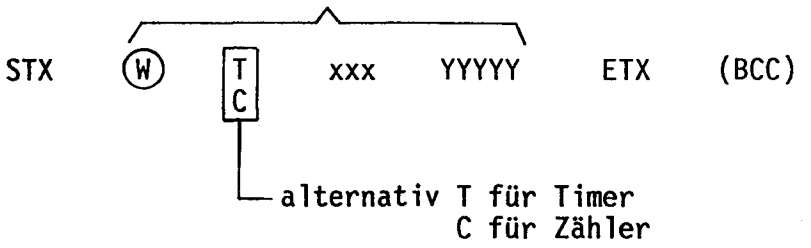
STX W e (32) 9? ETX

bewirkt setzen der Ausgänge 25...(32) wie folgt:



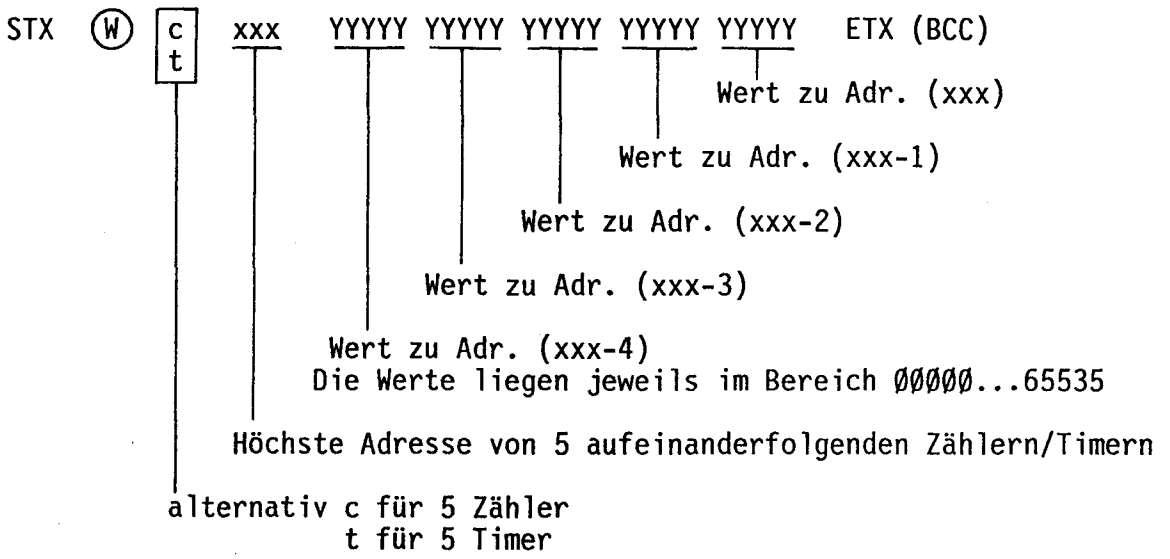
Gegenüber der Bearbeitung von Einzeladressen verkürzt diese Variante die Laufzeit um den Faktor 7.

- Schreiben eines Wertes in das Register eines Timers oder Zählers (Cn)

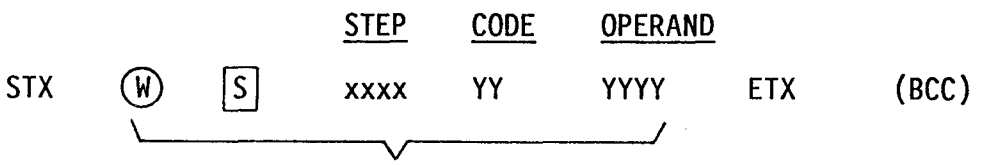


---> Schreibe den Wert YYYYY (5 Ziffern) ins Timer/Zähler-Register mit der Adresse xxx (3 Ziffern)

- Schreiben von 5 Werten in 5 aufeinanderfolgende Zähler oder Timer



- Schreiben einer Programmzeile (STEP) in den RAM-Anwenderspeicher



---> Schreibe in den RAM-Anwenderspeicher auf Schritt (STEP) xxxx (4 Ziffern) den Wert YYYYYY (6 Ziffern, 2 für den CODE Ø...31 und 4 für den Operand Ø...2Ø47)

- Schreiben eines Wertes in das Display Operand-Register

Durch die Bitprozessor-Befehle DOP und DTC können 4-stellige Zahlen in das Operand-Register geschrieben werden, die im OPERAND-Feld des Programeingabegerätes P1Ø angezeigt werden. Der Register-Inhalt wird in der folgenden Sekunde für die Dauer einer Sekunde angezeigt.

STX Ⓦ [O] YYYY ETX (BCC)

---> Schreibe den Wert YYYY (4 Ziffern) ins Operanden-Register

- Schreiben eines 2-stelligen Wertes (1 Registerwort) ins Wortregister Rn * (1 Registerwort)

STX Ⓦ [R] xxx YY ETX (BCC)

---> Schreibe den Wert YY (2 Ziffern) ins Wortregister mit der Adresse xxx (3 Ziffern)

- Schreiben eines 1Ø-stelligen Wertes (1 Registerblock) ins Wortregister * (5 Registerworte)

STX Ⓦ [A] xxx YYYYYYYYYY ETX (BCC)

---> Schreibe den Wert Y...Y (1Ø Ziffern) ins Wortregister, wobei xxx der höchsten Adresse eines Registerblockes entspricht.

- Schreiben eines 3 x 1Ø-stelligen Wertes (3 Registerblöcke) ins Wortregister * (15 Registerworte)

STX Ⓦ [B] xxx Y...(3Øx)...Y ETX (BCC)

---> Schreibe den Wert Y...Y (3Ø Ziffern) ins Wortregister, wobei xxx der höchsten Adresse von 3 Registerblöcken entspricht.

*) nur PCA231/232

- Schreiben eines 2 Byte-Wertes in den Textspeicher

STX Ⓦ [M] xxxx YYYYY ETX (BCC)

5-stelliger Wert ØØØØØ...65535

höhere Adresse des Textspeichers * (ungerade Adr. angeben)

Der Wert YYYYY wird im Textspeicher in binärer Form abgelegt. Auf der höheren Textspeicheradresse (xxxx) befindet sich das "H"-Byte, auf der tieferen Textspeicheradresse (xxxx-1) das "L"-Byte.

---

*) Die PCA2.M32 verfügt bekanntlich über einen 8K-Charakter Text- und über ein 8K-Byte Datenspeicher, welcher ebenfalls als Textspeicher benutzt werden kann. Die an dieser Stelle angegebene Adresse bezieht sich auf den Textspeicher, die Adressierung hat somit ein Bereich von Ø...15999, wobei für 1Ø - 15-Tausender folgende Zeichen stehen müssen:

1Ø : / 11 ; / 12 < / 13 = / 14 > / 15 ?

- Schreiben von 5 x 2 Byte-Werten in den Textspeicher

STX Ⓦ [m] xxxx YYYYY YYYYY YYYYY YYYYY YYYYY ETX (BCC)

Wert zu Adressen (xxxx) und (xxxx-1)

Wert zu Adressen (xxxx-2) und (xxxx-3)

Wert zu Adressen (xxxx-4) und (xxxx-5)

Wert zu Adressen (xxxx-6) und (xxxx-7)

Wert zu Adressen (xxxx-8) und (xxxx-9)

Höchste Adresse von 1Ø aufeinanderfolgenden Charakter-Nummern des Textspeichers (ungerade Adr. angeben)

Die einzelnen Werte YYYYY werden auf dieselbe Weise wie unter Telegramm [M] abgelegt und es gilt ebenfalls die dort unter der Fussnote *) stehende Information.

- Schreiben von 1Ø Digit (1 Registerblock) in den Textspeicher

STX Ⓦ [N] xxxx YYYYYYYYYY ETX (BCC)

- YYYYYYYYYY: Wert zu Adr. (xxxx)
- YYYYYYYYY: Wert zu Adr. (xxxx-1)
- YYYYYYYY: Wert zu Adr. (xxxx-2)
- YYYYYYY: Wert zu Adr. (xxxx-3)
- YYYYYY: Wert zu Adr. (xxxx-4)
- xxxx: höchste Adresse von 5 aufeinanderfolgenden Charakter-Nummern des Textspeichers mit Endziffer 4 oder 9 *) (gemäss Organisation der Registerblöcke)

---

*) Siehe Fussnote *) unter Telegramm M

- Schreiben von 3Ø Digit (3 Registerblöcke) in den Textspeicher

STX Ⓦ [n] xxxx YY.....YY ETX (BCC)

- YY.....YY: 3Ø Digit beginnend mit Wert zu Adresse (xxxx-14), endend mit dem Wert zu Adresse (xxxx)
- xxxx: höchste Adresse von 15 aufeinanderfolgenden Charakter-Nummern des Textspeichers mit Endziffer 4 oder 9 * (gemäss Organisation der Registerblöcke)

*) Siehe Fussnote *) unter Telegramm [M]

Nicht alle hier aufgeführten Telegramme laufen mit jeder CPU der PCA-Reihe. Entnehmen Sie die entsprechenden Informationen aus der Uebersicht im Kapitel K 6.4 "Definition des Modus P".

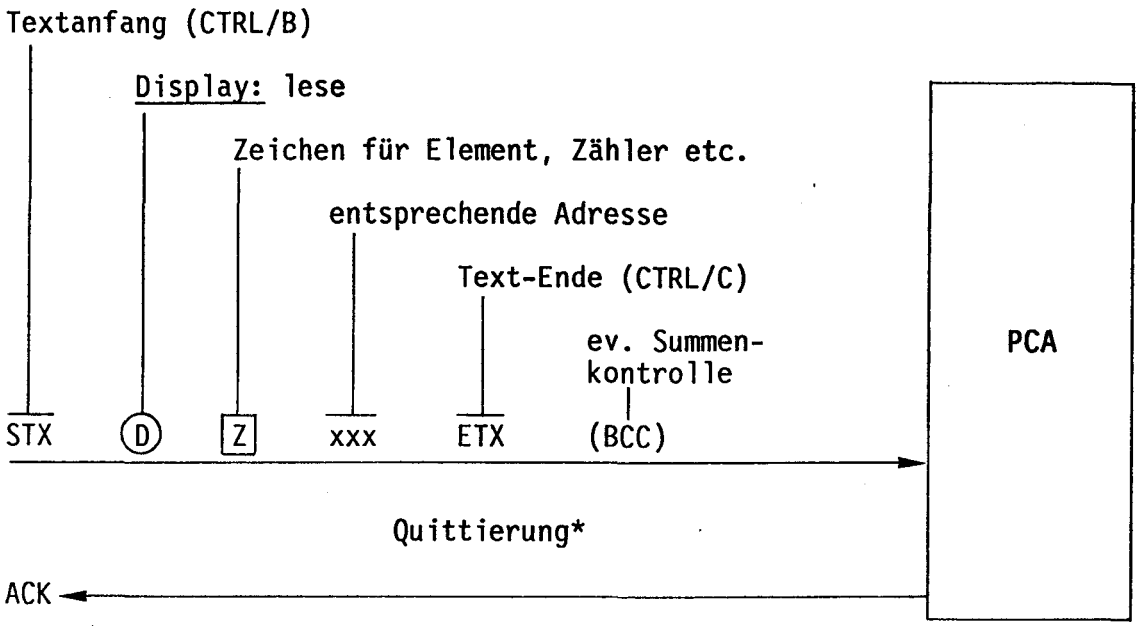
P.S.: Bevor der letzte Charakter eines Telegrammes jeweils übertragen wurde, kann mit ENQ (CTRL/E) ein Telegramm als ungültig bezeichnet werden. Die PCA quittiert dann mit NAK (# CR LF). Anschliessend kann ein neues Telegramm gesendet werden.

K 6.4.4 Lesen von Daten aus der PCA

Diese Art der Abfrage erlaubt Werte aus der PLC herauszulesen wie: logischer Zustand von Elementen En, Registerinhalte von Timern oder Zählern, Schritt-inhalte im Anwenderprogramm oder aus dem Display Operand-Register.

Diese Abfrage erfolgt in zwei Schritten:

(1) Befehl zur Abspeicherung an die CPU M32



Positive Antwort der CPU, dass der Befehl entgegengenommen wurde (ev. auch CR LF). NAK (bzw. # CR LF) bedeutet, dass die Abspeicherung nicht erfolgen konnte.

*) Erfolgt sofort (ohne Abwarten des Zeittaktes)

Die verschiedenen PC-Daten werden wie folgt abgefragt:

- Lesen des logischen Zustandes von einzelnen Elementen En mit Adresse xxx

  En = (E, A, T, C, M)

  STX (D) [E] xxx ETX (BCC) , xxx = Element-Adresse mit 3 Ziffern

- Lesen von 8 aufeinanderfolgenden Elementen (E, A, M) in Form von 2 Charaktern zu je 4 Bit

  STX (D) [e] xxx ETX (BCC) , xxx = höchste Element-Adresse En (3 Ziffern) von 8 aufeinanderfolgenden Elementen

  Die PCA antwortet mit

  STX YZ ETX (BCC bzw. CR LF)

  Beispiel für YZ

Charakter	Y	Z
Bitmuster mit	Ø Ø 1 1	1 1 1 Ø
En = 32	25	 32
ausgegebene Charakter	3	>

- Lesen des Registerinhaltes von Timer oder Zähler (Cn)

  STX (D) [T C] xxx ETX (BCC) , xxx = Timer/Zähler-Adresse mit 3 Ziffern

- Lesen von 5 aufeinanderfolgenden Zählern/Timern

  STX (D) [t c] xxx ETX (BCC) ; xxx = höchste Adresse von 5 aufeinanderfolgenden Timern/Zählern

- Lesen des Inhaltes einer Programmzeile (STEP)

  STX (D) [S] xxxx ETX (BCC) , xxxx = Schritt-Adresse mit 4 Ziffern (bis 8191)

- Lesen des Display Operand-Registers (von DOP und DTC)

  STX (D) [O] ETX (BCC)

- Lesen des Inhaltes des Registerwortes (Rn) von 2 Digit (1 Registerwort)

STX (D) [R] xxx ETX (BCC) , xxx = Registeradresse mit 3 Ziffern

- Lesen des Registerblockes (Rn) von 1Ø Digit (5 Registerworte)

STX (D) [A] xxx ETX (BCC) , xxx = höchste Register-Adresse des Registerblockes

- Lesen von 3 Registerblöcken von 3Ø Digit (15 Registerworte)

STX (D) [B] xxx ETX (BCC) ; xxx = höchste Register-Adresse von 3 Registerblöcken

- Lesen eines 2 Byte-Wertes aus dem Textspeicher

STX (D) [M] xxxx ETX (BCC) ; xxxx = höhere Adresse des Textspeichers (ungerade Adr. angeben)

- Lesen von 5 x 2 Byte-Werten aus dem Textspeicher

STX (D) [m] xxxx ETX (BCC) ; xxxx = höchste Adresse von 1Ø aufeinanderfolgenden Charakternummern des Textspeichers (ungerade Adr. angeben)

- Lesen von 1Ø Digits (1 Registerblock) aus dem Textspeicher

STX (D) [N] xxxx ETX (BCC) ; xxxx = höchste Adresse von 5 aufeinanderfolgenden Charakternummern des Textspeichers mit Endziffer 4 oder 9 (gemäss Organisation der Registerblöcke)

- Lesen von 3Ø Digits (3 Registerblöcke) aus dem Textspeicher

STX (D) [n] xxxx ETX (BCC) ; xxxx = höchste Adresse von 15 aufeinanderfolgenden Charakternummern des Textspeichers mit Endziffer 4 oder 9 (gemäss Organisation der Registerblöcke)

Die in den Lesetelegrammen angeforderten Daten werden im Antwort-Telegramm der CPU im selben Format wie im entsprechenden Schreibtelegramm gesendet.

Nicht alle hier aufgeführten Telegramme laufen mit jeder CPU der PCA-Reihe. Entnehmen Sie die entsprechenden Informationen aus der Uebersicht im Kapitel K 6.4 "Definition des Modus P".

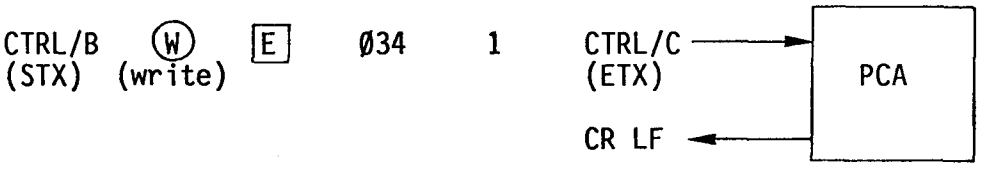
P.S.: Bevor der letzte Charakter eines Telegrammes jeweils übertragen wurde, kann mit ENQ (CTRL/E) ein Telegramm als ungültig bezeichnet werden. Die PCA quittiert dann mit NAK (# CR LF). Anschliessend kann ein neues Telegramm gesendet werden.

Beispiele:

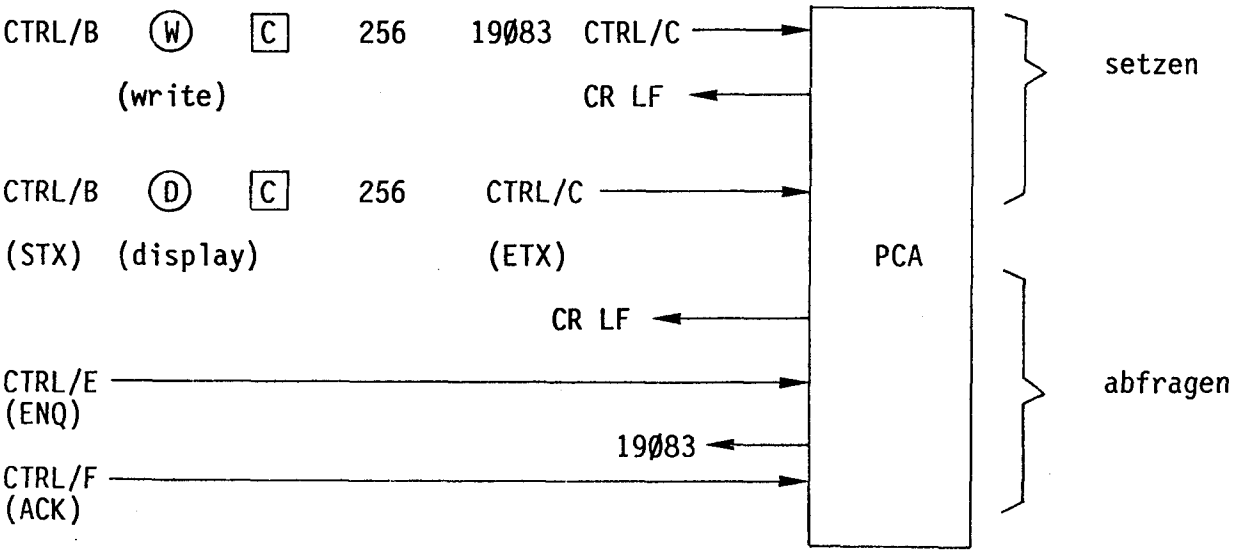
Durch Assignierung der Schnittstelle im P2-Modus (Code ØØ in Zeile 6) können die folgenden beiden Beispiele mit einem einfachen Terminal nachvollzogen werden.

a) Setzen eines Ausganges via P2-Modus

Ausgangsadresse: 34



b) Setzen eines Zählers und Abfrage dieses Zählers via P2-Modus



K 6.5 Die Assignierung für kombinierte Modi

Beim Kombinieren von verschiedenen Modi für senden und empfangen kann in folgenden Fällen mit nur einer Schnittstellen-Assignierung gearbeitet werden:

Menü-Modus (Empfang von Einzelcharaktern und senden von Texten):
Assignierung für Modus C

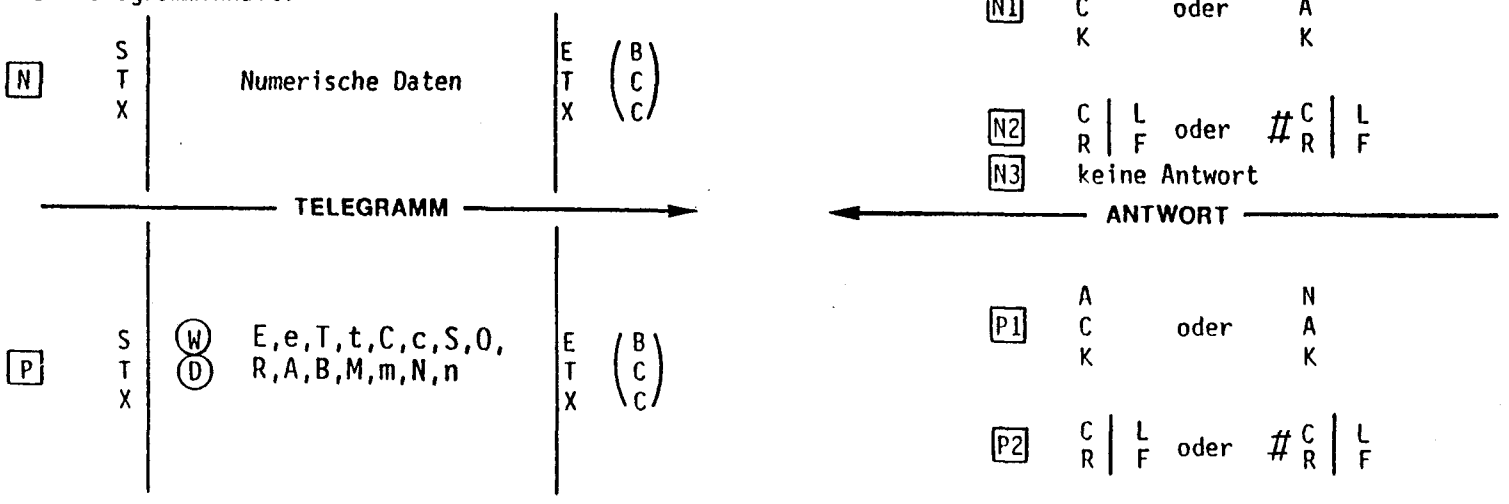
Computer-Modus (Austausch numerischer Daten und PCA-bezogener Daten):
Assignierung für Modus N

Grundsätzlich gilt:

Assignierung	Betrieb möglich im
für Textausgabe	Modus T
für Einzelcharakter	Modi C + T
für numerische Daten	Modi N + T + P
für PCA-Daten	Modi P + T

Kommentar zum CODE:	Zeilen-Nr.	CODE	OP	Kommentar zum OPERAND:
	1:	PAS	1ØØ	Assignierung serielle Datenschnittstelle
	2:	ØØ	xxxx	BAUD, Bits/Charakter, Parity, Stopbit
E — Text-Editor Modus Andere Betriebsarten	3:	(Ø1) ØØ	TXB	Text busy Flag
N — Empfang von 1...31x4 Bit Daten Telegramm C — Empfang von Einzelcharaktern	4:	Ø1...31 ØØ	RBY	Receive Buffer busy Flag
C — Senden von 1...31x4 Bit Daten Telegramm Senden von Einzelcharaktern	5:	Ø1...31 ØØ	TBY	Transmit Buffer busy Flag
N1, P1, N3 Mit BCC für "intelligente" Peripherie N2, P2, N3 Ohne BCC für einfache Peripherie	6:	(Ø1) ØØ	Ø (1) (3)	Ohne Quittierung wird Telegramm wiederholt Telegramm wird nur einmal gesendet Telegramm wird nur einmal gesendet, wobei keine Rückantwort erwartet wird
	7...1Ø:	ØØ	Ø	Leer

N und P unterscheiden sich nur durch den Telegramminhalt:

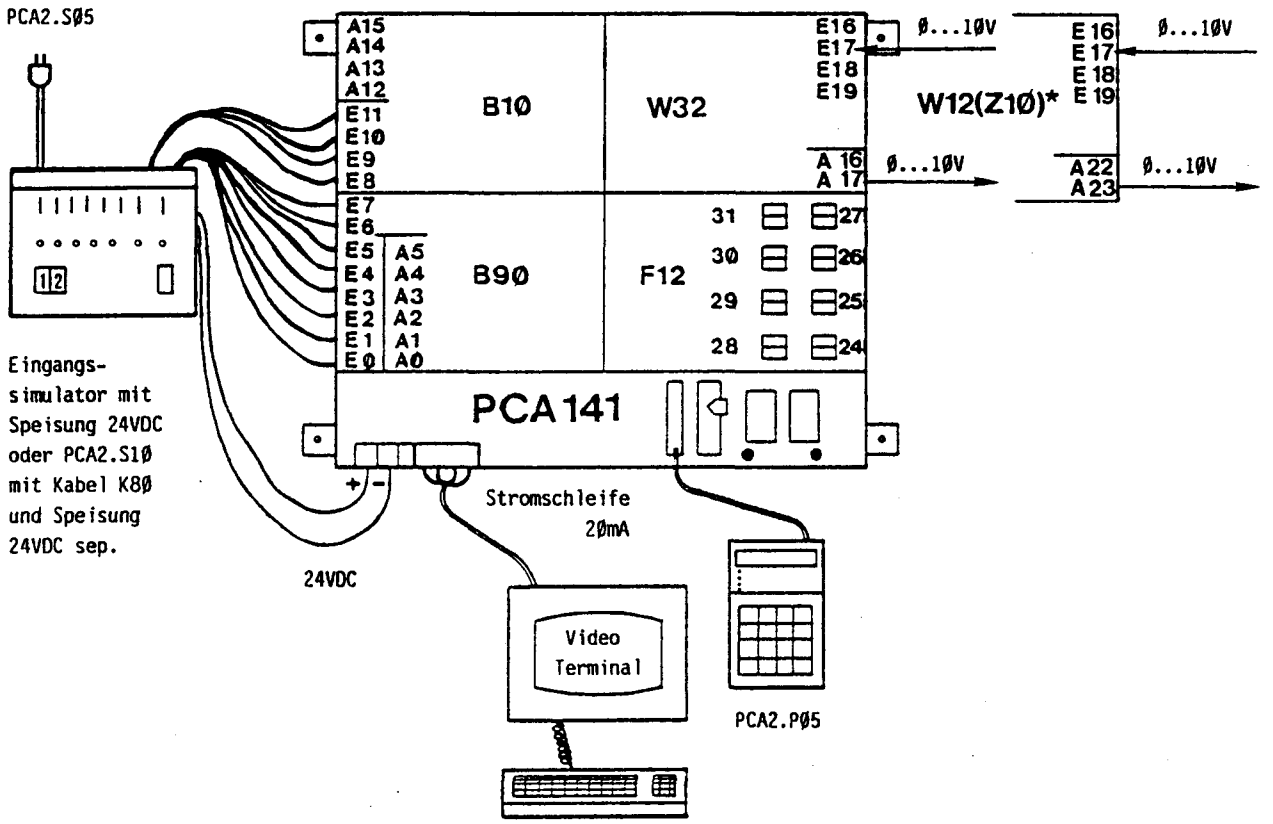


TEIL L PROGRAMMIERBEISPIELE

TEIL L PROGRAMMIERBEISPIELE

In den nachfolgenden Beispielen 1 bis 11 werden aus den vielfältigen Möglichkeiten vor allem typische Lehrbeispiele aus den Gebieten Textausgabe, Kommunikation, PID-Regelung sowie Schiebe- und FIFO-Register behandelt. Weiter finden sich typische Programme zur Behandlung der Datum-Uhr und der Check Sum.

Um die Dokumentation nicht unübersichtlich werden zu lassen, sind alle Beispiele auf die Hardware der weitest verbreiteten PCA14 ausgerichtet und zwar in der gleichbleibenden E/A-Konfiguration gemäss nachfolgender Abbildung.



Alle Beispiele lassen sich in ähnlicher Weise mit einer PCA2 durchführen, indem ein digitales E-Modul und ein digitales A-Modul je auf die Basis-Adresse Ø gestellt werden. Als Analog-Modul wird ein PCA2.W14 z.B. auf die Adresse 176 empfohlen.

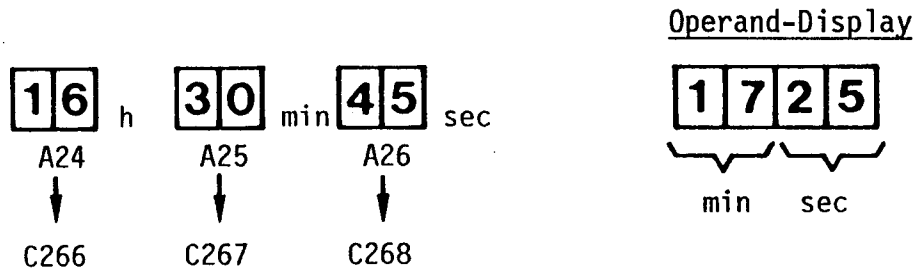
---

*) Das Modul PCA1.W12 Z1Ø erlaubt Eingangsspannungen von Ø...1ØV (anstatt Ø...5V).

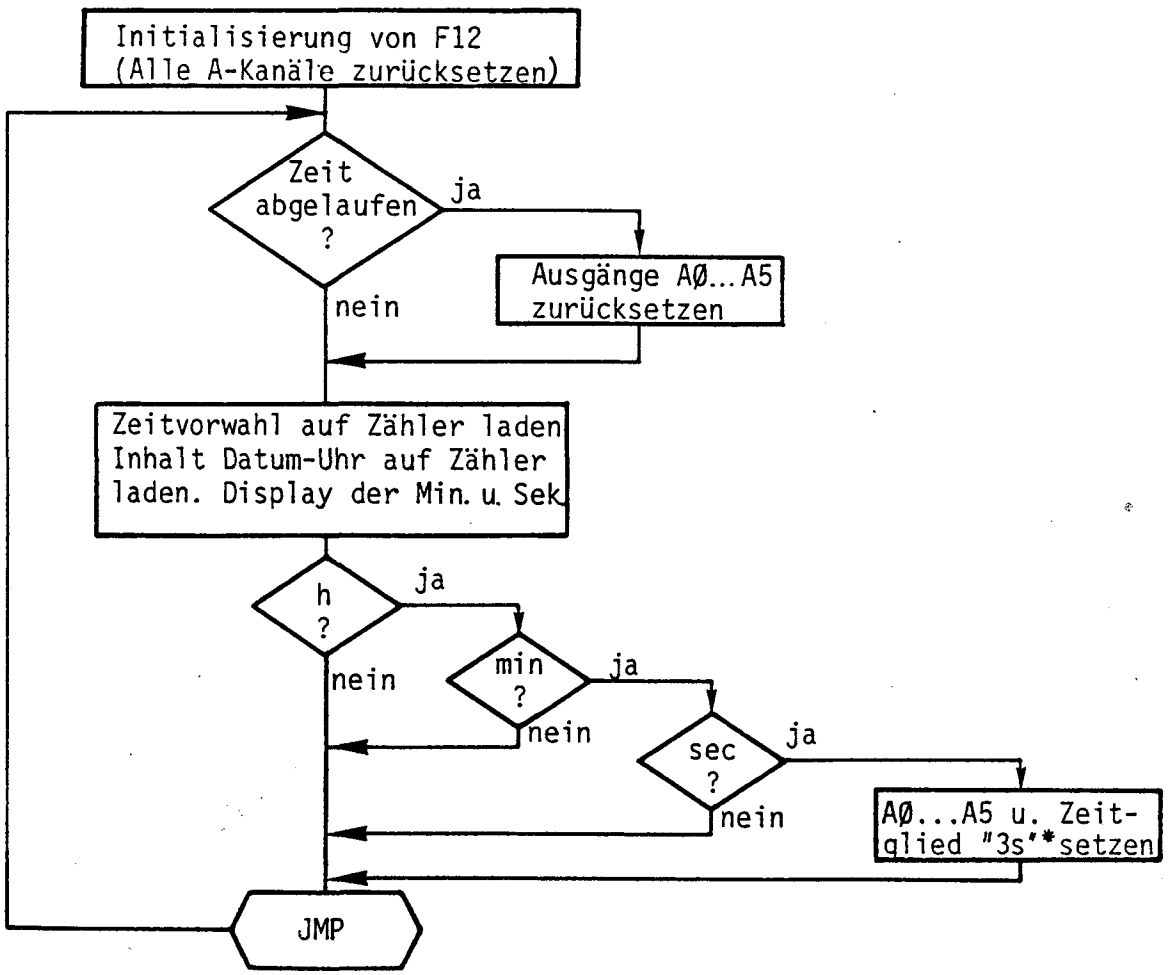
Beispiel 1 Schaltfunktion und Textausgabe zu vorgewählter Uhrzeit
(Verwendung von PAS 5Ø)

Aufgabe 1a

In der Betriebsart MAN soll die Datum-Uhr gestellt werden. Ueber die jeweils 2 Digits der Adressen A24...A26 ist eine Uhr-Zeit vorzuwählen, bei welcher die Ausgänge AØ...A5 für 4s* aufleuchten müssen. Die Minuten und Sekunden der Datum-Uhr sollen im Operand-Display angezeigt werden.



Flussdiagramm



*) Während einer Sekunde wird das Zeitglied immer wieder neu gesetzt. Um die Ausgänge für eine Zeit von 4s zu aktivieren, muss das Zeitglied auf 3s gesetzt werden.

Lösung 1a

```
    360  16  SEI     Ø
┌►  361  12  REO   1Ø24
¦   362  27  INI     7
└─  363  21  JIO   361

    366  Ø1  STH   279
    367  21  JIO   374
    368  16  SEI     Ø
┌►  369  12  REO   1ØØØ
¦   37Ø  27  INI     5
└─  371  21  JIO   369

    374  16  SEI     Ø
┌►  375  11  SEO   1Ø24
¦   376  15  SCR   1266
¦   377  16  16      31
¦   378  12  REO   1Ø24
¦   379  27  INI     2
└─  38Ø  21  JIO   375

    383  29  PAS     5Ø
    384  15  15     27Ø   h
    385  29  PAS     5Ø
    386  16  16     271   min
    387  29  PAS     5Ø
    388  17  17     272   sec

    391  15  SCR   273
    392  31  31    271
    393  15  SCR   273
    394  29  29    1ØØ
    395  15  SCR   273
    396  27  27    272
    397  31  DTC   273

    4ØØ  15  SCR   266
    4Ø1  28  28    27Ø
    4Ø2  Ø2  STL   266
    4Ø3  22  JIZ   366

    4Ø5  15  SCR   267
    4Ø6  28  28    271
    4Ø7  Ø2  STL   267
    4Ø8  22  JIZ   366

    41Ø  15  SCR   268
    411  28  28    272
    412  Ø2  STL   268
    413  22  JIZ   366

    416  16  SEI     Ø
┌►  417  11  SEO   1ØØØ
¦   418  14  STR   279
¦   419  ØØ  ØØ     3Ø
¦   42Ø  27  INI     5
└─  421  21  JIO   417
    422  2Ø  JMP   366
```

Initialisierung von F12

Ausgänge AØ...A5 nach abgelaufener Zeit zurücksetzen

Digits A24...A26 auf Zähler C266...C268 laden

Datum-Uhr auf C27Ø...C272 laden

Minuten und Sekunden der Datum-Uhr auf C273 laden und Display

Vergleich h

Vergleich min.

Vergleich sec.

Ausgänge AØ...A5 und Zeitglied "4s" setzen

Zusatzaufgabe 1b

Bei Uebereinstimmung der Datum-Uhr mit der vorgewählten Zeit soll zusätzlich folgender Text ausgegeben werden:

Datum + Uhrzeit (konventionelle Form)

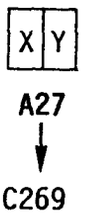
Die SAIA- PLC Reihe PCA14 ist das neueste

SAIA- Produkt innerhalb der SAIA- PLC Familie

SAIA- PLC gibt es in XY Ländern!

SAIA

- SAIA- soll als Textsubroutine geschrieben werden.
- Der Wert XY ist dem BCD-Schalter A27 zu entnehmen (ohne Vornullen).



Lösung 1b

```
431  29   PAS   100 \
432  00    00   902  > 4800 Baud
433  00    00   399 /
434  00    00   254 \
435  00    00   254  > Schnittstellen-
436  00    00     0  > Assignierung
437  00    00     0  >
438  00    00     0  >
439  00    00     0  >
440  00    00     0 /

443  16   SEI     0 \
444  12   REO  1024  > Initialisierung
445  27   INI     7  > von F12
446  21   JIO   444 /

449  01   STH   286 \
450  21   JIO   457  > Ausgänge nach
451  16   SEI     0  > abgelaufener Zeit
452  12   REO  1000  > zurücksetzen
453  27   INI     5  >
454  21   JIO   452 /

457  16   SEI     0 \
458  11   SEO  1024  > Digits
459  15   SCR  1266  > A24...A27 auf
460  16    16    31  > C266...269
461  12   REO  1024  > laden
462  27   INI     3  >
463  21   JIO   458 /

466  29   PAS    50 \
467  15    15   270  > Datum-Uhr auf
468  29   PAS    50  > C270 bis C272
469  16    16   271  > laden
470  29   PAS    50  >
471  17    17   272 /

474  15   SCR   273 \
475  31    31   271  > Min. und Sek.
476  15   SCR   273  > auf C273 laden
477  29    29   100  > und Display
478  15   SCR   273  >
479  27    27   272  >
480  31   DTC   273 /

(1)

483  15   SCR   266 \
484  28    28   270  > Vergleich h
485  02   STL   266  >
486  22   JIZ   449 /

488  15   SCR   267 \
489  28    28   271  > Vergleich min.
490  02   STL   267  >
491  22   JIZ   449 /

493  15   SCR   268 \
494  28    28   272  > Vergleich sec.
495  02   STL   268  >
496  22   JIZ   449 /

499  16   SEI     0 \
500  11   SEO  1000  > Ausgänge
501  14   STR   286  > A0...A5 und
502  00    00    40  > Zeitglied "4s"
503  27   INI     5  > setzen
504  21   JIO   500 /

507  01   STH  287* \
508  21   JIO   449  >
509  29   PAS    23  > Textausgabe
510  00    00     5  >
511  14   STR  287*  >
512  00    00    11  >
513  20   JMP   449 /
```

Text:

```
  5   .$. .0.2.0.$.H^M^J^J
  6   .$. .0.2.0.D.I.E. .$
  7   .L.0.2.3.P.L.C. .R.E
  8   .I.H.E. .P.C.A.1.4.
  9   .I.S.T. .D.A.S. .N.E
 10   .U.E.S.T.E^M^J.$. .0
 11   .2.0.$.L.0.2.3.P.R.O
 12   .D.U.K.T. .I.N.N.E.R
 13   .H.A.L.B. .D.E.R. .$
 14   .L.0.2.3.P.L.C. .F.A
 15   .M.I.L.I.E^M^J.$. .0
 16   .2.0.$.L.0.2.3.P.L.C
 17   .G.I.B.T. .E.S. .I.N
 18   .$.c.2.6.9. .L.A.E.N
 19   .D.E.R.N.!^M^J^J^M^M
 20   ^M.$. .0.5.4^M^J^@^@

 23   .S.A.I.A.-.$.U^@^@^@
```

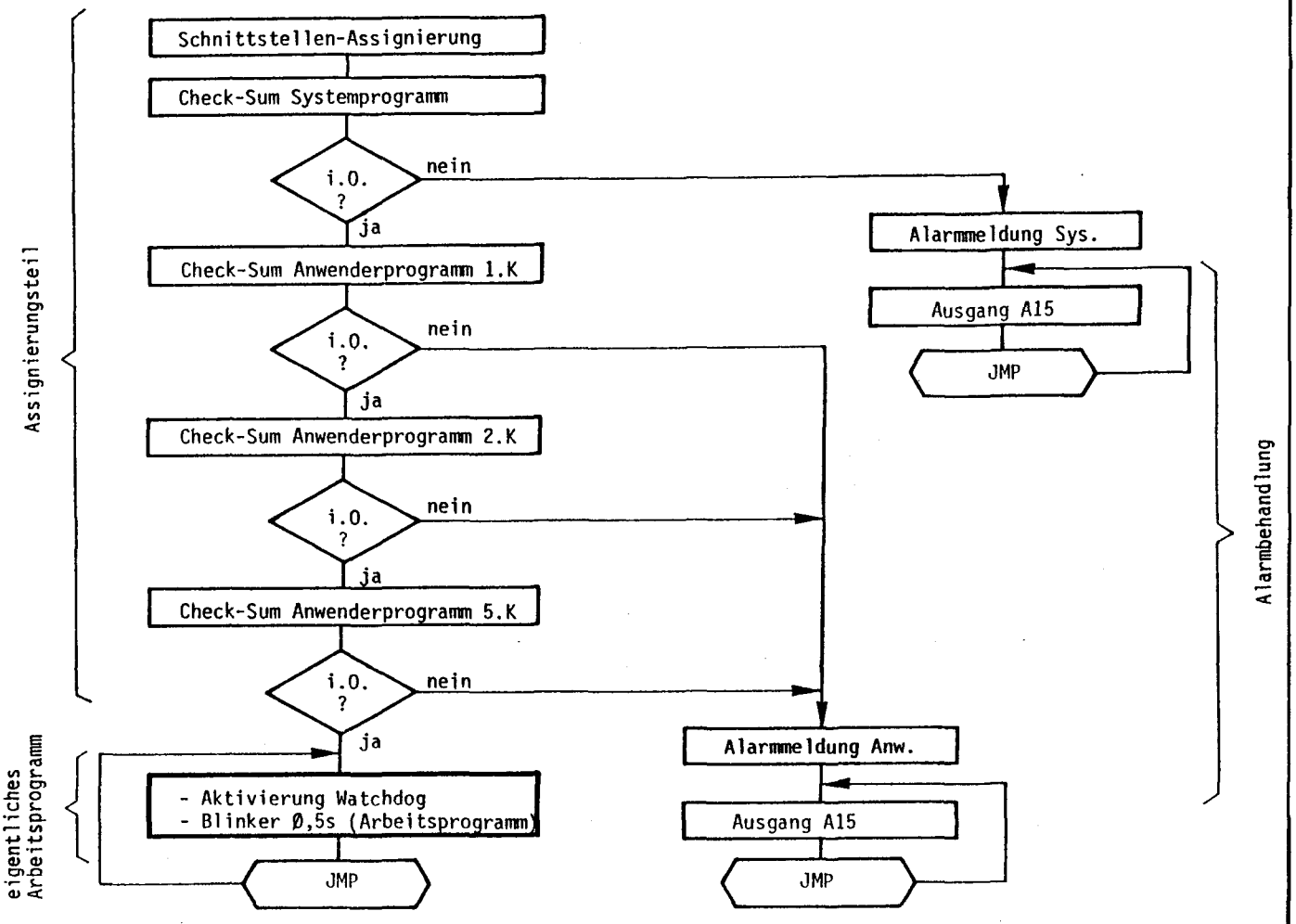
*) Während der Sekunde, bei welcher die Bedingungen für die Textausgabe erfüllt sind, würde der Text mehrmals ausgegeben. Ueber das Zeitglied (1,1s) wird dafür gesorgt, dass der Text nur einmal ausgegeben wird.

Beispiel 2 "Check-Sum" des System- und Anwenderprogrammes sowie Aktivierung des "Watchdog" (Verwendung von PAS 3Ø...38)

Aufgabe 2

Beim ersten Durchlauf dieses Programmes soll eine Summenkontrolle für das Systemprogramm sowie für das Anwenderprogramm (1.K, 2.K und 5.K)* durchgeführt werden. In einem kleinen Arbeitsprogramm soll der Watchdog aktiviert werden und der Ausgang A12 soll im Ø,5s-Takt blinken. Wird im Systemprogramm ein Fehler festgestellt, muss die Meldung "Summenkontrolle Systemprogramm" ausgegeben werden. Wird ein Fehler im Anwenderprogramm festgestellt, muss die Meldung "Summenkontrolle Anwenderprogramm" ausgegeben werden. In beiden Fällen soll Ausgang A15 (Alarm) aktiviert werden.

Flussdiagramm



*) Das 1.K und das 2.K enthalten die Anwenderprogramme, das 5.K die Texte.

Fehler im Systemprogramm können nicht simuliert werden. Hingegen kann im RAM-Speicher des Anwenderprogrammes auf leeren Adressen eine Eingabe gemacht werden, um die Alarmmeldung zu provozieren.

Lösung 2

```
521  29  PAS  100
522  ØØ   ØØ   9Ø2
523  ØØ   ØØ   399
524  ØØ   ØØ   254
525  ØØ   ØØ   254
526  ØØ   ØØ     Ø
527  ØØ   ØØ     Ø
528  ØØ   ØØ     Ø
529  ØØ   ØØ     Ø
53Ø  ØØ   ØØ     Ø

532  29  PAS   3Ø
533  ØØ   ØØ     Ø
534  22  JIZ   554
535  29  PAS   31
536  ØØ   ØØ  1929
537  22  JIZ   559
538  29  PAS   32
539  Ø3   Ø3  1397
54Ø  22  JIZ   559
541  29  PAS   35
542  ØØ   ØØ  1815
543  22  JIZ   559
544  12  REO   15  ;

546  13  COO   255
547  Ø2  STL   285
548  14  STR   285
549  ØØ   ØØ     5
55Ø  13  COO    12
551  2Ø  JMP   546

554  29  PAS    23
555  ØØ   ØØ   1ØØ
556  11  SEO    15
557  2Ø  JMP   556

559  29  PAS    23
56Ø  ØØ   ØØ   1Ø4
561  11  SEO    15
562  2Ø  JMP   561
```

4800 Baud

Schnittstellen-Assignierung

Check-Sum Systemprogramm

Check-Sum 1.K

Check-Sum 2.K

Check-Sum 5.K

Alarm zurücksetzen

Watchdog

Arbeitsprogramm

Textausgabe und Alarmausgang bei Fehler im Systemprogramm

Textausgabe und Alarmausgang bei Fehler im Anwenderprogramm

Erläuterung:
Der Test über die Check-Sum wird normalerweise nur beim Einschalten der PLC durchgeführt. Er gehört daher ebenfalls zum Assignierungsteil des Programmes (Zeilen 521 bis 544).

Das eigentliche Arbeitsprogramm reicht nur von Zeile 546 bis 551. Die restlichen Programmteile dienen der Textausgabe für den Fehlerfall.

Text 100:

```
100  .$.H^M^J.S.U.M.M.E.N
101  .K.O.N.T.R.O.L.L.E.
102  .S.Y.S.T.E.M.P.R.O.G
103  .R.A.M.M..^M^J^J^@^@
```

Text 104:

```
104  .$.H^M^J.S.U.M.M.E.N
105  .K.O.N.T.R.O.L.L.E.
106  .A.N.W.E.N.D.E.R.P.R
107  .O.G.R.A.M.M..^M^J^J
108  ^@.*.U.*.*.U.*.*.U.*
```

*) Das Verbleiben in der Alarmschleife ist gewollt. Verläuft die Check-Sum Prüfung negativ, so ist entweder die CPU oder ein Anwenderspeicher defekt und muss ausgetauscht werden. Durch den abfallenden WD können weitere Sicherheitsmassnahmen ergriffen werden.

Beispiel 3 Text-Sonderzeichen

Aufgabe 3

Alle Text-Sonderzeichen sollen auf einem Bildschirm dargestellt werden.
Es sind dies:

a) Freier Text

b) Ausgabe eines Zählerregisters Cn, ohne Vornullenunterdrückung

c) Ausgabe eines Zählerregisters Cn, mit Vornullenunterdrückung

d) Ausgabe von Elementen (logische Zustände von 8 sich folgenden Elementen)

e) Ausgabe des ASCII-Charakters* (gebildet von 8 sich folgenden Elementen)

f) Inhalt Datum-Uhr, industrielle Form

g) Inhalt Datum-Uhr, konventionelle Form

h) Inhalt Datum-Uhr, Jahr, Monat, Tag

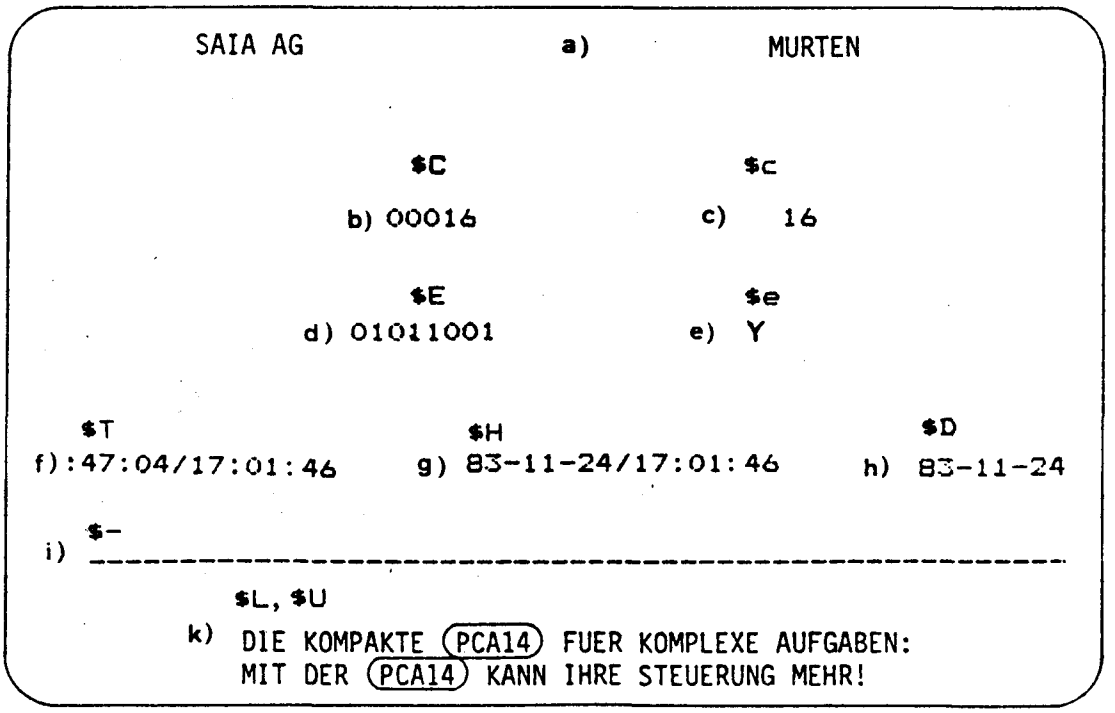
i) Zeichenrepetierung

k) Textsubroutine

l) Ausgabe des Zeichens $

---

*) Ausgewertet werden alle Charakter im Bereich Ø...127. Die sichtbaren Zeichen werden dargestellt, während sich die Steuerzeichen direkt als solche auswirken und in der Darstellung durch SPACE ersetzt werden.



Lösung 3

(Die Parameter sowie die verschiedenen Cursor-Anweisungen beziehen sich auf das Terminal VC 4404).

```
951  29   PAS   100
952  00    00   902 ; 4800 Baud
953  00    00   333 ; TXB
954  00    00   254
955  00    00   254
956  00    00     0
957  00    00     0
958  00    00     0
959  00    00     0
960  00    00     0
962  29   PAS    23
963  00    00     0 ; Text Nr. 0
966  25   WIH   333 ; TXB
967  29   PAS    23
968  00    00    50 ; Text Nr. 50
971  15   SCR   290
972  00    00     0
973  25   WIH   333 ; TXB
974  14   STR   259
975  00    00    10
976  29   PAS    23
977  00    00    30 ; Text Nr. 30
978  25   WIH   259
979  17   INC   290
980  20   JMP   973
```

Texte:

```
→  0  ^X^@^@^@^@^@^@^@^@
→ 30  ^P.#.*.$.C.2.9.0.$.
  31  .0.2.0.$.c.2.9.0^M^P
  32  .(.(.$.E.0.0.7.$. .0
  33  .2.0. ^H.$.e.0.0.7^M
  34  ^P.+. .$.T.$. .0.0.5
  35  .$.H.$. .0.0.5.$.D^J
  36  ^J^J^J^J^J^J^J^J^M^@
→ 50  .$. .0.0.7.S.A.I.A.
  51  .A.G.$. .0.2.0. . .
  52  .M.U.R.T.E.N. . ^M^J
  53  ^J.$. .0.1.1.$.$.C.$
  54  . .0.2.3.$.$.c^M^J^J
  55  ^J^J^J.$. .0.1.1.$.$
  56  .E.$. .0.2.3.$.$.e^M
  57  ^J^J^J.$.$.T.$. .0.1
  58  .8.$.$.H.$. .0.2.0.$
  59  .$.D^M^J^J^J.$.$.-^M
  60  ^J.$.-.0.5.0^M^J^J.$
  61  . .0.0.4.$.$.L.,.$.$
  62  .U^M^J.$. .0.0.4.D.I
  63  .E. .K.O.M.P.A.K.T.E
  64  . .$.L.0.8.0. .F.U.E
  65  .R. .K.O.M.P.L.E.X.E
  66  . .A.U.F.G.A.B.E.N.:
  67  ^M^J.$. .0.0.5.M.I.T
  68  . .D.E.R. .$.L.0.8.0
  69  . .K.A.N.N. .I.H.R.E
  70  . .S.T.E.U.E.R.U.N.G
  71  . .M.E.H.R.!^M^J^J^@
→ 80  ^N.P.C.A.1.4^O.$.U^@
```

Bildschirm löschen

Inhalt von Zählerregister Elementenreihe und Datum-Uhr

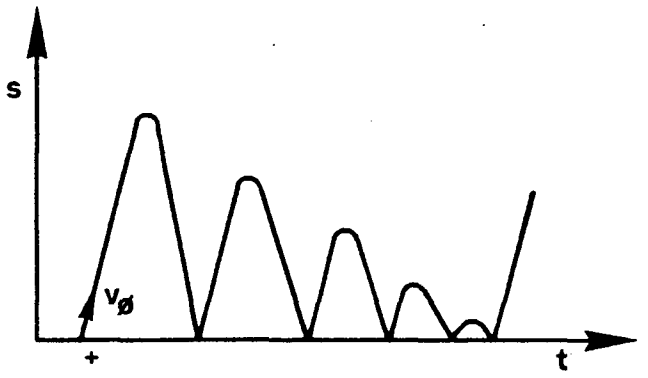
Basis-Text

Textsubroutine

Beispiel 4 Ballspiel

Gegeben:

Ein Ball wird mit der Anfangsgeschwindigkeit $V_ø$ senkrecht nach oben geworfen. Beim Rückfall auf eine horizontale Fläche verliert er jedesmal X % seiner Geschwindigkeit.



$$s = V_ø \times t - \frac{g}{2} \times t^2$$

$$g = 1ø\ m/s^2$$

$$t = t/1ø\ (1/1ø\ s)$$

$$\Longrightarrow s = \frac{v_ø x t}{1ø} - \frac{1ø\ t^2/1øø}{2}\ [m]$$

$$\boxed{s = 1/1ø\ (v_ø x t - t^2/2)}\ [m]$$

Aufgabe 4

man berechne ca. jede 1/1øs die Flughöhe s und stelle diese auf dem Bildschirm dar.

Bei jedem Aufprall am Boden reduziert sich $v_ø$ auf den prozentualen Wert, welcher über A24 (von F12) vorgewählt wird.

8 6 % von $v_ø$

A24

C313

Lösung 4

$v_ø$ wird auf C314 eingelesen.

Die Formel $v_ø$ x $t^2/2$ wird innerhalb der Zählerregister erarbeitet.

Der Text ist dank Verwendung der Sonderzeichen verblüffend kurz.
Nur 16 Charakter!

```
651  29  PAS   100
652  00   00   902  ; 4800 Baud
653  00   00   600  ; TXB
654  00   00   254  ; RBY
655  00   00   254  ; TBY
656  00   00     0
657  00   00     0
658  00   00     0
659  00   00     0
660  00   00     0
661  11  SEO    24          } Einlesen der
662  15  SCR   313          } Amplitude in %
663  16   16    31          } von A24
664  12  REO    24
665  15  SCR   314  <-
666  00   00    11  vø
667  15  SCR   315
668  00   00     0  t = 0
669  15  SCR   317  <-
670  31   31   314  v
671  15  SCR   317  v       } vxt
672  29   29   315  t       }
673  15  SCR   316  <-
674  31   31   315  t
675  15  SCR   316  t       } t²
676  29   29   316  t       }
677  15  SCR   316  t²      } t²/2
678  30   30     2  2       }
679  15  SCR   317  vxt     } vxt - t²/2 = s
680  28   28   316  t²/2    }
681  15  SCR   318  <-
682  31   31   317  s
683  17  INC   315  t + 1
684  21  JIO   692
685  15  SCR   314  v       } vx%
686  29   29   313  %       }
687  15  SCR   314  vx%     } vx%
688  30   30   100  100     } 100
689  01  STH   314          } v = 0 ?
690  21  JIO   667          }
691  20  JMP   661  v = 0 --> Neustart
692  29  PAS    23          } Text Nr.3
693  00   00     3          }
694  25  WIH   600  ; Warten solange
695  00  NOP     0    TXB = H
696  20  JMP   669
```

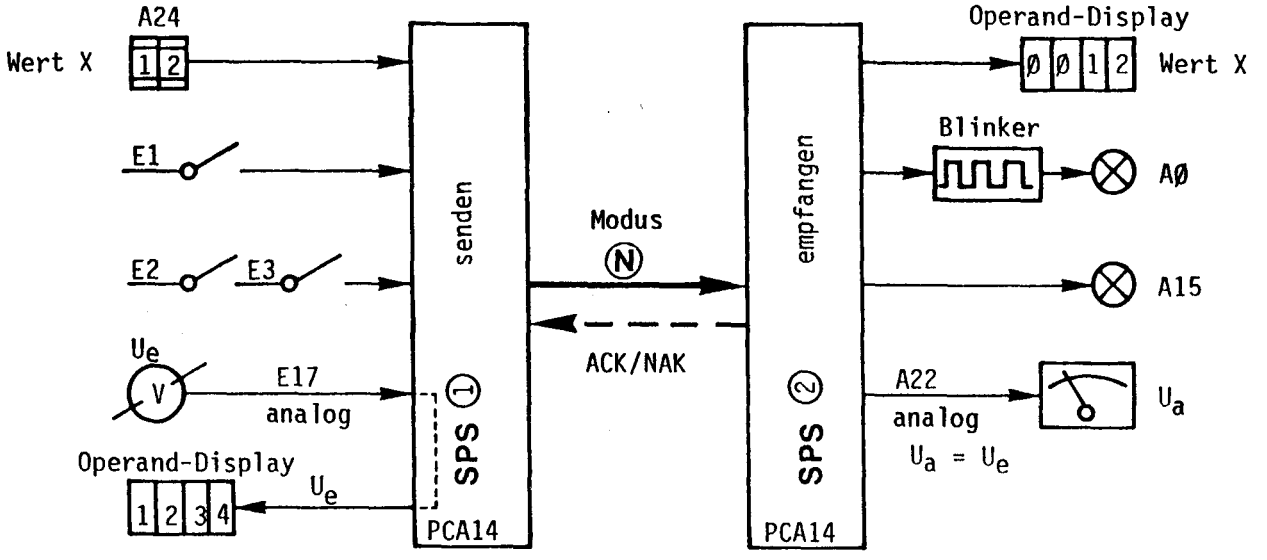
Text: C-Register 318 ausgeben

```
3   .$.c.3.1.8.  .:.$.-.3
4   .1.8.*^M^J^@^@^@^@^@
```

Beispiel 5 Datenaustausch via serielle Datenschnittstelle zwischen zwei PCA14 im N-Modus

Aufgabe 5

Verschiedene Eingaben an der 1. SPS sollen mit Modus N über die serielle Datenschnitstelle übertragen und an der 2. SPS verarbeitet und zur Ausgabe gebracht werden. Es sind dies:



Aufgabe für SPS ① (Sender)

a) PAS 1ØØ definieren beginnend bei Adresse 6Ø1

- TXB = 254
- RBY = 525
- TBY = 425

b) Erstellen des Einleseprogrammes nach obiger Konfiguration. Der über Eingangskanal E17 eingelesene Analogwert U_e soll im Operand-Display als Binärwert angezeigt werden.

c) Telegramm senden.

d) Die Stromschleife soll von der SPS ① gespeist werden (SPS ① aktiv).

Aufgabe für SPS (2) (Empfänger)

a) PAS 1ØØ definieren beginnend bei Adresse 6Ø1

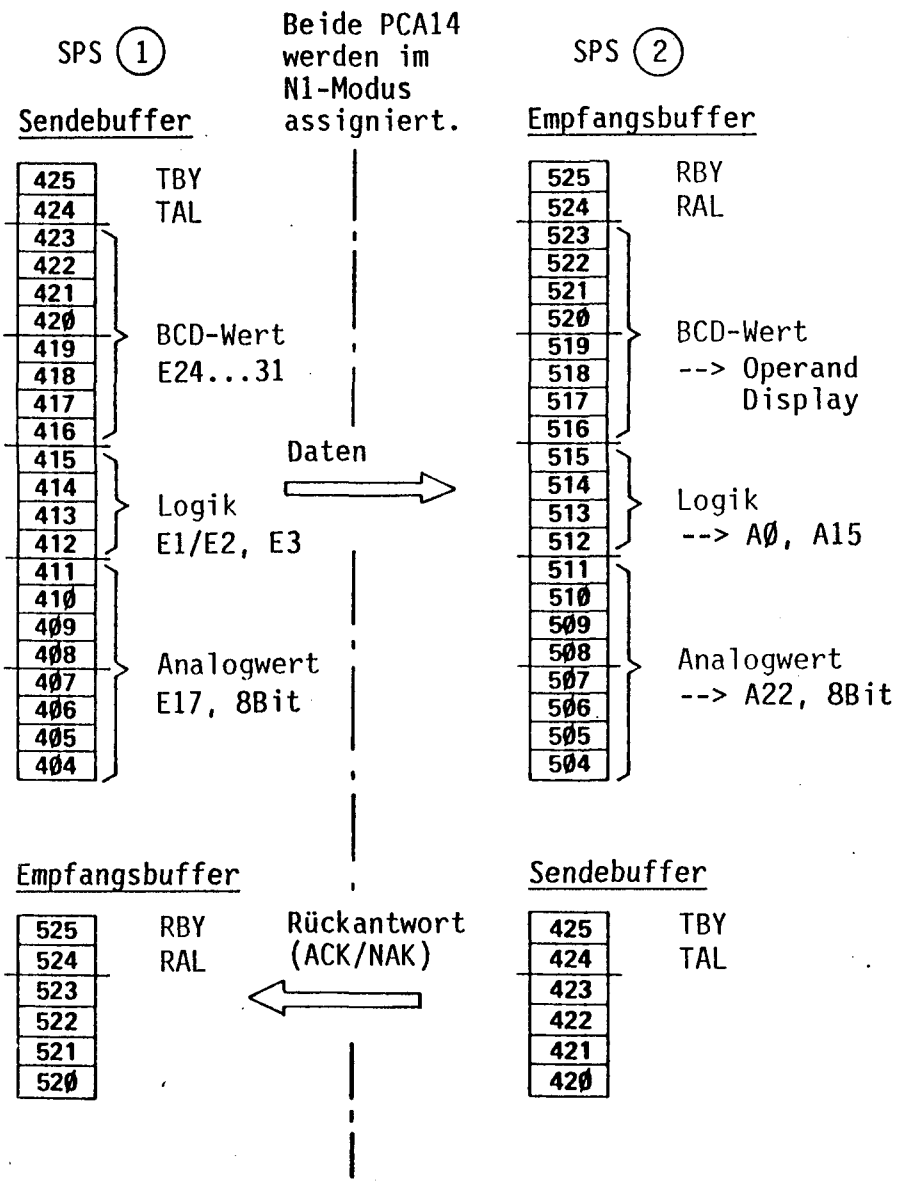
- TXB = 254
- RBY = 525
- TBY = 425

b) Erstellen des Verarbeitungsprogrammes nach obiger Konfiguration. Der Ausgang AØ soll gemäss dem BCD-Wert X blinken.

c) Empfang freigeben.

d) Die SPS (2) soll passiv sein.

Lösung 5



SPS (1)

Sender

```
601  29  PAS   100
602  00   00   902 ;4800 Baud
603  00   00   254 ;TXB
604  01   01   525 ;RBY
605  05   05   425 ;TBY
606  01   01     0 ;BCC
607  00   00     0
608  00   00     0
609  00   00     0
610  00   00     0

613  01  STH   425 } Sender
614  21  JIO   639 } frei?

616  11  SEO    24 }
617  15  SCR   299 } BCD-Wert
618  24   24    31 } von A24 auf
619  12  REO    24 } Sendebuffer
620  15  SCR   299 } übertragen
621  21   21   423 }

623  01  STH     1 }
624  10  OUT   412 } Log. Resultate
625  01  STH     2 } auf Sendebuffer
626  03  ANH     3 } übertragen
627  10  OUT   413 }

629  19  SEA     0 }
630  10  OUT    17 } Analogwert von
631  15  SCR   298 } Kanal E17 ein-
632  24   24    23 } lesen und auf
633  15  SCR   298 } Sendebuffer
634  21   21   411 } übertragen

636  19  SEA     0 } Telegramm
637  11  SEO   425 } senden

639  31  DTC   298 ;Display Analog-
640  20  JMP   613  wert
```

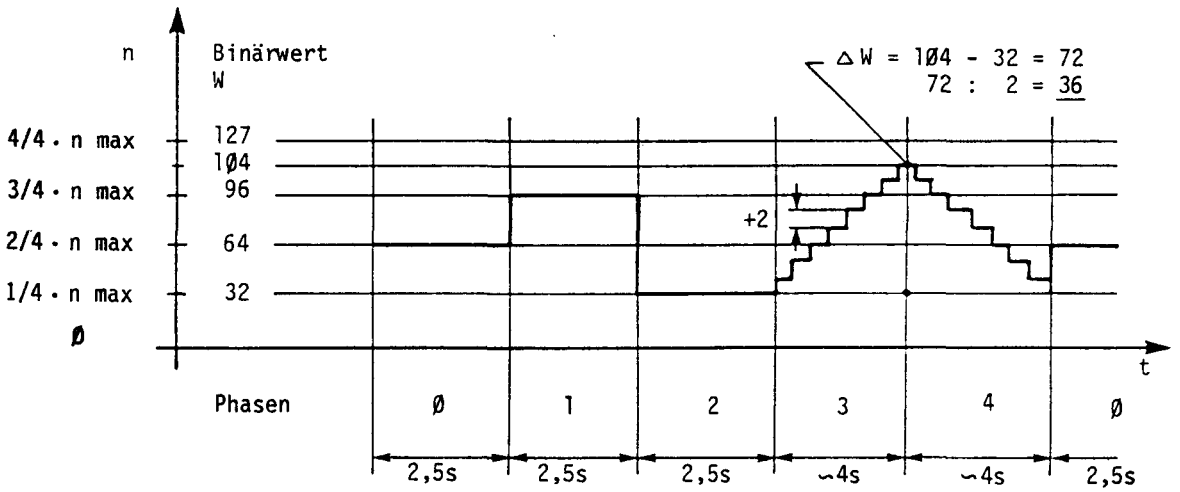
SPS (2)

Empfänger

```
601  29  PAS   100
602  00   00   902 ;4800 Baud
603  00   00   254 ;TXB
604  05   05   525 ;RBY
605  01   01   425 ;TBY
606  01   01     0 ;BCC
607  00   00     0
608  00   00     0
609  00   00     0
610  00   00     0

613  01  STH   525 } Empfänger
614  22  JIZ   633 } frei

616  12  REO   525 ;Empfang beendet

618  15  SCR   257 } BCD-Wert
619  16   16   523 } auf Operand-
620  31  DTC   257 } Display

622  15  SCR   258 }
623  24   24   511 }
624  15  SCR   258 } Analogwert
625  21   21    23 } übertragen
626  11  SEO    23 } auf Kanal A22
627  05  ORH    22 }
628  12  REO    23 }

630  01  STH   513 } Ausgang A15
631  10  OUT    15 }

633  02  STL   512 }
634  11  SEO   400 }
635  01  STH   512 }
636  04  ANL   256 } Blinker auf
637  14  STR   256 } Ausgang A0
638  16   16   523 } (t = BCD-
639  13  COO   400 } Wert)
640  01  STH   400 }
641  10  OUT     0 }
642  20  JMP   613 }
```

Beispiel 6 Schneller PID-Regelkreis mittels PAS 202 auf PCA14 (Motor-Tachogenerator)

Ein Gleichstrommotor mit Tacho-Generator soll entsprechend einer variablen Sollwertkurve W = f (t) geregelt werden. Diese Sollwertkurve hat folgenden Verlauf:

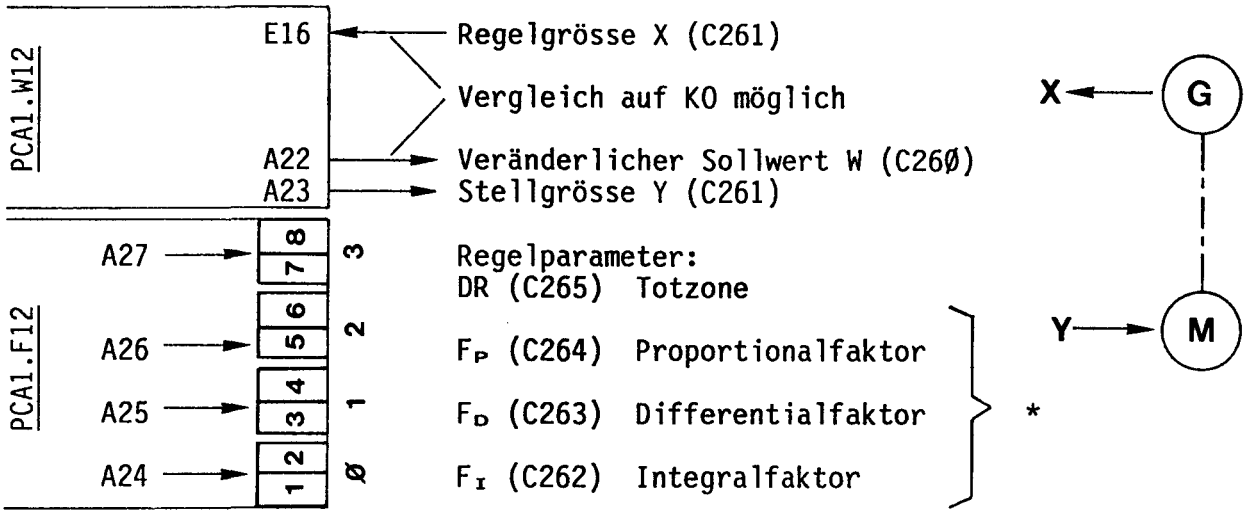


Die fünf zeitlichen Phasen Ø...4 können entsprechend dem folgenden Programm festgelegt werden:

```
100   STR   281   ;  Phase Ø
      ØØ     25
      SCR   26Ø   }  W = 1/2 n max
      ØØ     64   }
  ┌─> JMS   15Ø  --> PID
  │   STH   281
  └── JIO   104

11Ø   STR   281   ;  Phase 1
      ØØ     25
      SCR   26Ø   }  W = 3/4 n max
      ØØ     96   }
  ┌─> JMS   15Ø  --> PID
  │   STH   281
  └── JIO   114

12Ø   STR   281   ;  Phase 2
      ØØ     25
      SCR   26Ø   }  W = 1/4 n max
      ØØ     32   }
  ┌─> JMS   15Ø  --> PID
  │   STH   281
  └── JIO   124
```

```
13Ø   SCR   3ØØ   ;  Phase 3
      ØØ     36
  ┌─> INC   26Ø   }  W + 2
  │   INC   26Ø   }
36x   JMS   15Ø  --> PID
  │   DEC   3ØØ
  │   STH   3ØØ
  └── JIO   132

14Ø   SCR   3ØØ   ;  Phase 4
      ØØ     36
  ┌─> DEC   26Ø   }  W - 2
  │   DEC   26Ø   }
36x   JMS   15Ø  --> PID
  │   DEC   3ØØ
  │   STH   3ØØ
  └── JIO   142

      JMP   1ØØ  -->
```

Die Regelparameter können mit dem Vorwahlmodul PCA1.F12 eingegeben bzw. laufend angepasst werden. Der Motor-Tachogenerator wird am Analog-Modul PCA1.W12 angeschlossen wie folgt:



Die PID-Subroutine 15Ø lautet demnach wie folgt:

```
--> 15Ø   SEI      Ø
    ----> SEO   1Ø24   FI
          SCR   1262   FD   *
           17*    31   FP
          REO   1Ø24
          INI      2          BCD-Schalter A24...A27 auf
    ----- JIO    151          Zähler C262...265 einlesen!
          SEO     27
          SCR    265   DR
           16     31
          REO     27

   17Ø    WIH    28Ø          Abtastzeit abwarten
          OUT     16          Via E-Kanal 16 Regel-
          SCR    261          grösse X einlesen
           24     23          (auf C261)

   174    PAS    2Ø2
     5     ØØ     31          Datablock 31
     6     ØØ    26Ø          26Ø, 261, 262, 263, 264, 265
     7     ØØ      Ø
     8     ØØ      Ø
     9     ØØ      Ø
   18Ø     ØØ      Ø
     1     ØØ      Ø
     2     ØØ      Ø
     3     ØØ      Ø
     4    JMP    185   **
   185    SCR    261
           21     23
          SEO     23          Stellgrösse Y ausgeben
          ORH     23          auf A-Kanal 23
          REO     23

   19Ø    SCR    26Ø
           21     23          Veränderlicher Sollwert W
          SEO     23          ausgeben auf A-Kanal 22
          ORH     22
          REO     23

   195    STR    28Ø          Abtastzeit
           ØØ      1          TØ = Ø,1 sec
   197    RET      Ø  ------->
```

*) Die BCD-Vorwahlwerte werden mit 1Ø multipliziert in den entsprechenden Zähler eingelesen. Für Vorwahl 5Ø berechnet sich der Praxiswert für F_I, F_D, F_P wie folgt:
5Ø x 1Ø = 5ØØ / 5ØØ : 256 = 1.95

**) Durch JMP wird bewirkt, dass ein PP-Wechsel erfolgen kann.

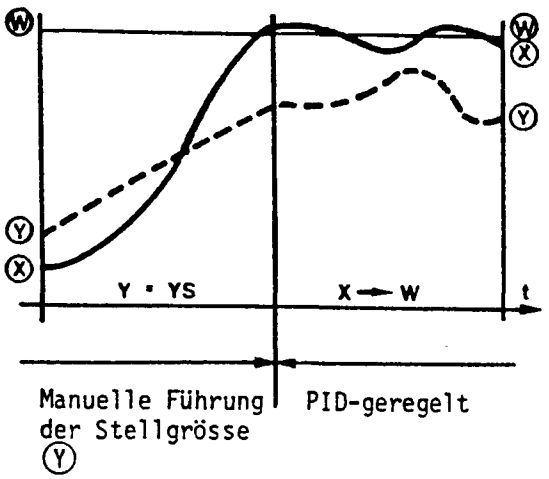
Beispiel 7 Hand/PID-Betrieb mittels PAS 211 auf PCA14 (Analogmodule 12 Bit PCA1.W32)

Aufgabe 7

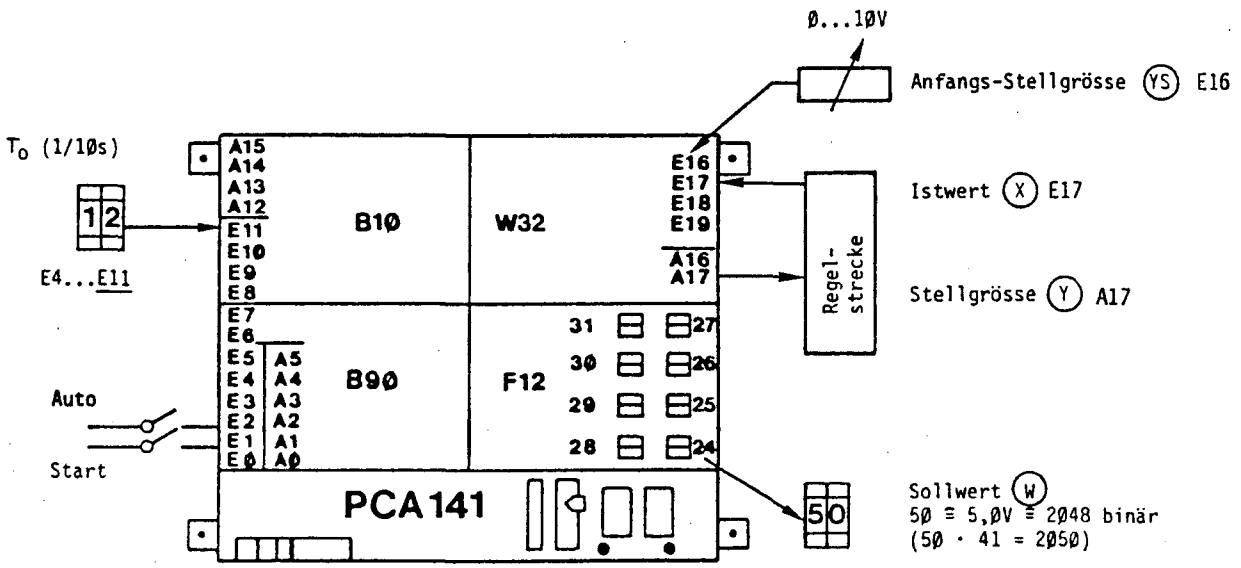
Das folgende Programm soll den Uebergang von der direkten Handsteuerung der Stellgrösse zur automatischen PID-Regelung zeigen.
Als Subroutinen werden die Ein- und Auslese-routinen aus der Dokumentation des Analog-moduls PCA1.W32 verwendet.

Der Sollwert W wird am BCD-Schalter A24 in Ø,1V vorgegeben. Die Abtastzeit $T_ø$ ist am BCD-Schalter E4...11 in Ø,1 sec. einstellbar. Mit EØ wird der Vorgang gestartet.

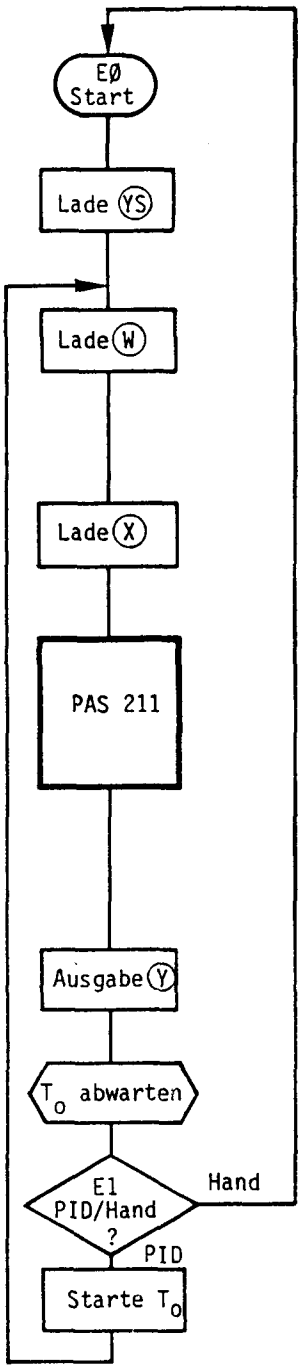
Ist E1 = L, so kann in Handsteuerung (Potentiometer) via Analog-Eingang E16 die Stellgrösse Y solange verändert werden, bis X = W ist.
Mit E1 = H wird auf PID-Regelung umgeschaltet, womit Störgrössen automatisch ausgeregelt werden
X ---> W.



Hardware-Anordnung



Lösung 7




```
50   WIL    0      Start

51   JMS    550**  } Stellgrösse (YS) über Analogkanal
52   SCR    282    } E16 einlesen und auf C282
53    31    300    } übertragen

54   SEO     24    }
55   SCR    276    } BCD-Schalter A24
56    16     31    } Sollwert (W)
57   REO     24    } in C276 laden
58   SCR    276    } mal 41, d.h. auf
59    29     41    } Binärwert umrechnen

60   JMS    500*   } Istwert (X) über Analogkanal E17
61   SCR    277    } einlesen und auf C277
62    31    300    } übertragen

63   PAS    211
      12     31     12 Bit, Datablock 31
      00    276     (W)
      00    277     (X) / (Y)
        FI
        FD
        FP
        DR
      00    282     (YS)
      00      0

73   SCR    301    } Stellgrösse (Y)
74    31    277    } auf Analogkanal
75   JMS    700*   } A17 ausgeben

76   WIH    256    Abtastzeit Tø abwarten

77   STH      1    } PID/
78   JIZ     50    } Handabfrage

79   STR    256    } Abtastzeit Tø von BCD-
80    16     11    } Schalter laden und starten

81   JMP     54
```

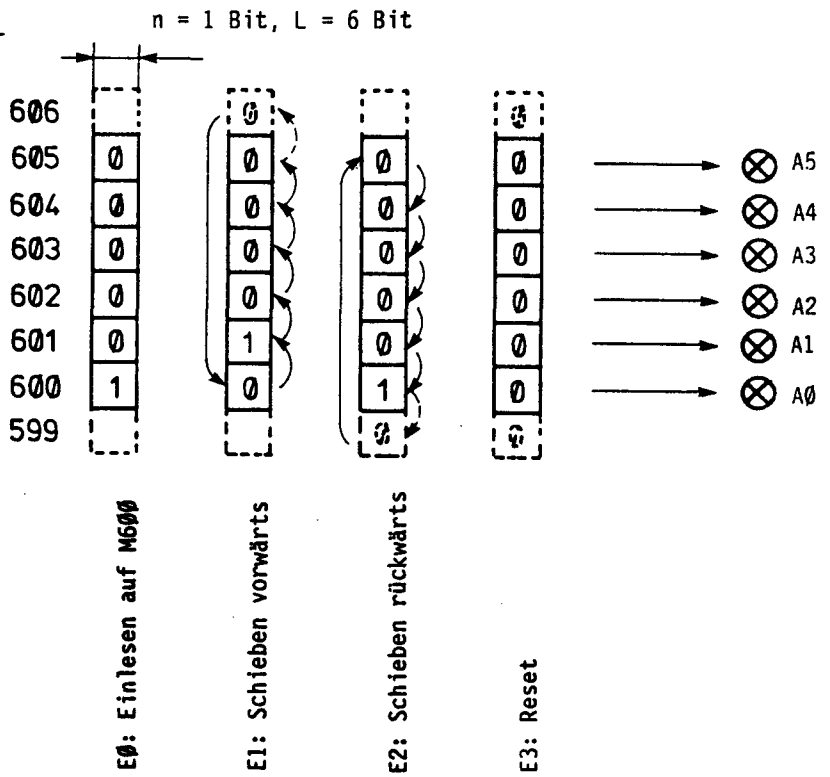
Wie in der Aufgabenstellung beschrieben, kann mit diesem Programm durch Veränderung des Potentiometers (YS) (E16) der Istwert von Hand in die Nähe des Sollwertes (W) gelegt werden. Anschliessend wird mit E1 stossfrei auf PID-Regelung umgeschaltet.

---

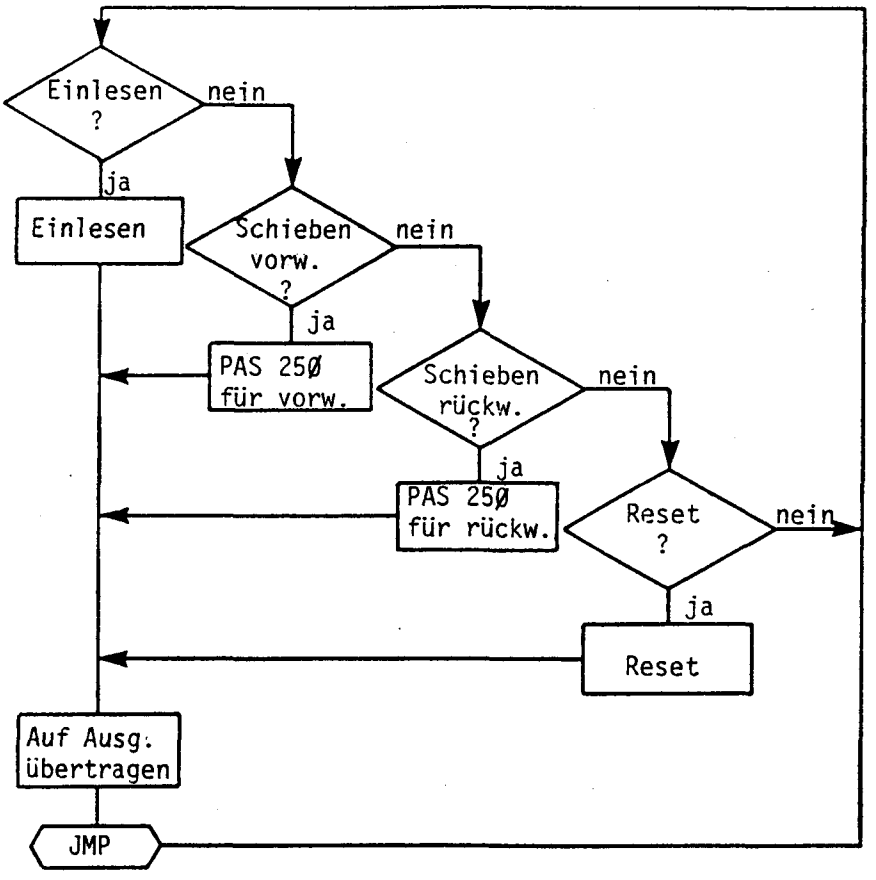
*) Unterprogramme siehe Kapitel "PCA1.W3.." im Handbuch Hardware PCA1.
**) Das Unterprogramm 550 entspricht dem Unterprogramm 500 indem dort die zweite Zeile durch OUT 16 ersetzt werden muss.

Beispiel 8 1-Bit-Rotationsregister vorwärts/rückwärts mit Reset (Verwendung von PAS 25Ø)

Funktionsweise:



Flussdiagramm:



Lösung 8

```
1001  01   STH     0
1002  09   DYN   500
1003  22   JIZ  1007
1004  10   OUT   600
1005  20   JMP  1055 --->

1007  01   STH     1
1008  09   DYN   501
1009  22   JIZ  1026

1011  29   PAS   250
1012  00    00   600 ; SADD
1013  00    00   605 ; DADD
1014  00    00     1 ; 1 Bit
1015  00    00     0
1016  00    00     0
1017  00    00     0
1018  00    00     0
1019  00    00     0
1020  00    00     0

1022  20   JMP  1055 --->

1026  01   STH     2
1027  09   DYN   502
1028  22   JIZ  1044

1031  29   PAS   250
1032  00    00   605 ; SADD
1033  00    00   600 ; DADD
1034  00    00     1 ; 1 Bit
1035  00    00     0
1036  00    00     0
1037  00    00     0
1038  00    00     0
1039  00    00     0
1040  00    00     0

1042  20   JMP  1055 --->

1044  01   STH     3
1045  09   DYN   503
1046  22   JIZ  1055 --->
1047  16   SEI     0
1048  12   REO  1599
1049  27   INI     7
1050  21   JIO  1048

1055  16   SEI     0
1056  01   STH  1600
1057  10   OUT  1000
1058  27   INI     5
1059  21   JIO  1056
1060  20   JMP  1001
```

Einlesen

Vorwärts schieben

Rückwärts schieben

Reset

Merker auf Ausgänge A0...A5

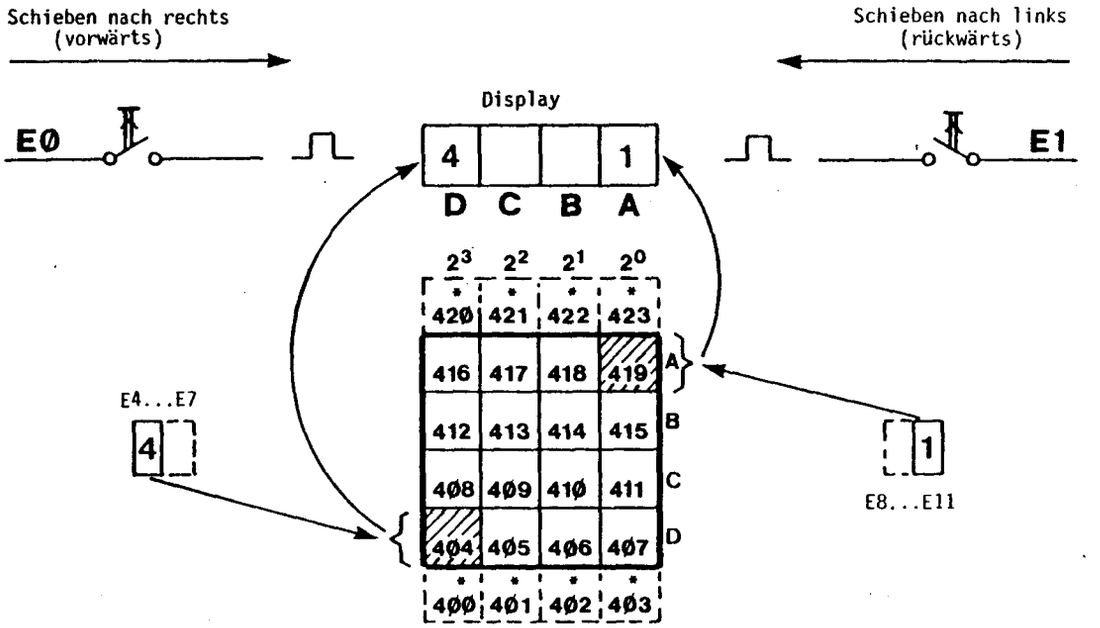
Beim vorliegenden Beispiel handelt es sich um ein einfaches Prinzipmodell. Darin ist der Aufwand zum Ergebnis relativ hoch. Die Vorteile des PAS 250 kommen bei der Ausführungszeit (für dieses Beispiel ca. 250µs) und bei grösseren Registerstrukturen zum Tragen.

Beispiel 9 BCD-Schieberegister vorwärts/rückwärts
(Verwendung von PAS 25Ø)

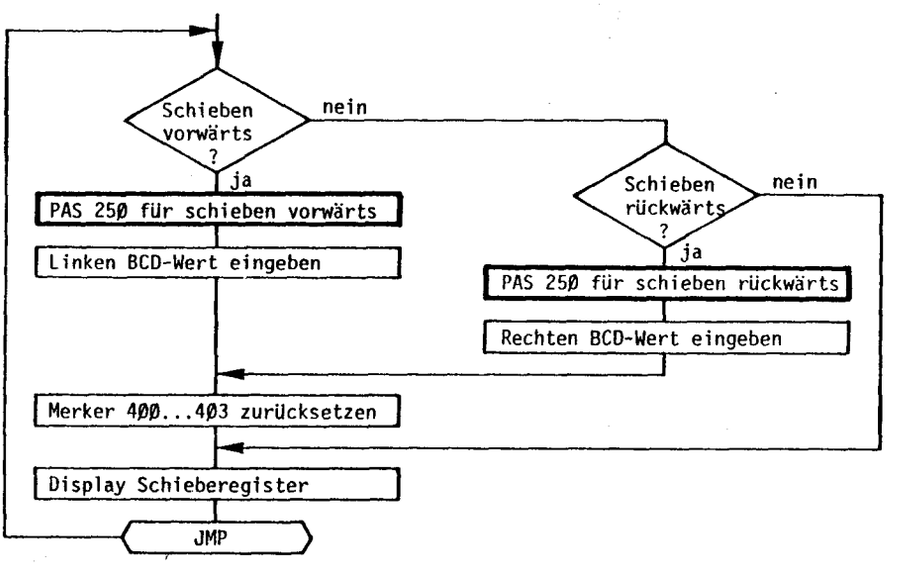
Beschreibung

Das Schieberegister ist 4 Bit lang und 4 Bit breit. Zur Status-Eingabe werden die BCD-Schalter auf den Eingängen E4...E11 verwendet, die linke Ziffer (E4...E7) für schieben vorwärts, die rechte Ziffer (E8...E11) für schieben rückwärts.

Das Register besteht aus den Merkern 4Ø4...419. Der Schiebevorgang wird ausgelöst durch den Eingang EØ (schieben vorwärts) resp. durch E1 (schieben rückwärts). Die Ziffern werden im Operand-Display angezeigt.



Flussdiagramm



*) Die Merker 4ØØ...4Ø3 und 42Ø...423 müssen reserviert werden, da sie als Umlaufbuffer verwendet werden.

Lösung 9

```
1101  01  STH     0
1102  09  DYN   511
1103  22  JIZ  1131

1111  29  PAS   250
1112  00   00   404   ; SADD
1113  00   00   419   ; DADD
1114  00   00     4   ; N = 4 Bit
1115  00   00     0
1116  00   00     0
1117  00   00     0
1118  00   00     0
1119  00   00     0
1120  00   00     0

1123  16  SEI     0   Wert des
1124  01  STH  1004   linken
1125  10  OUT  1404   BCD-
1126  27  INI     3   Schalters
1127  21  JIO  1124   ein-
1128  20  JMP  1161   lesen

1131  01  STH     1
1132  09  DYN   512
1133  22  JIZ  1173

1141  29  PAS   250
1142  00   00   419   ; SADD
1143  00   00   404   ; DADD
1144  00   00     4   ; N = 4 Bit
1145  00   00     0
1146  00   00     0
1147  00   00     0
1148  00   00     0
1149  00   00     0
1150  00   00     0

1153  16  SEI     0   Wert des
1154  01  STH  1008   rechten
1155  10  OUT  1416   BCD-
1156  27  INI     3   Schalters
1157  21  JIO  1154   einlesen

1161  16  SEI     0
1162  12  REO  1400   Merker 400...403
1163  27  INI     3   rücksetzen *
1164  21  JIO  1162

1171  15  SCR   300
1172  19   19   419   Schieberegister auf Zähler
1173  31  DTC   300   und Display übertragen
1174  20  JMP  1101
```

schieben vorwärts

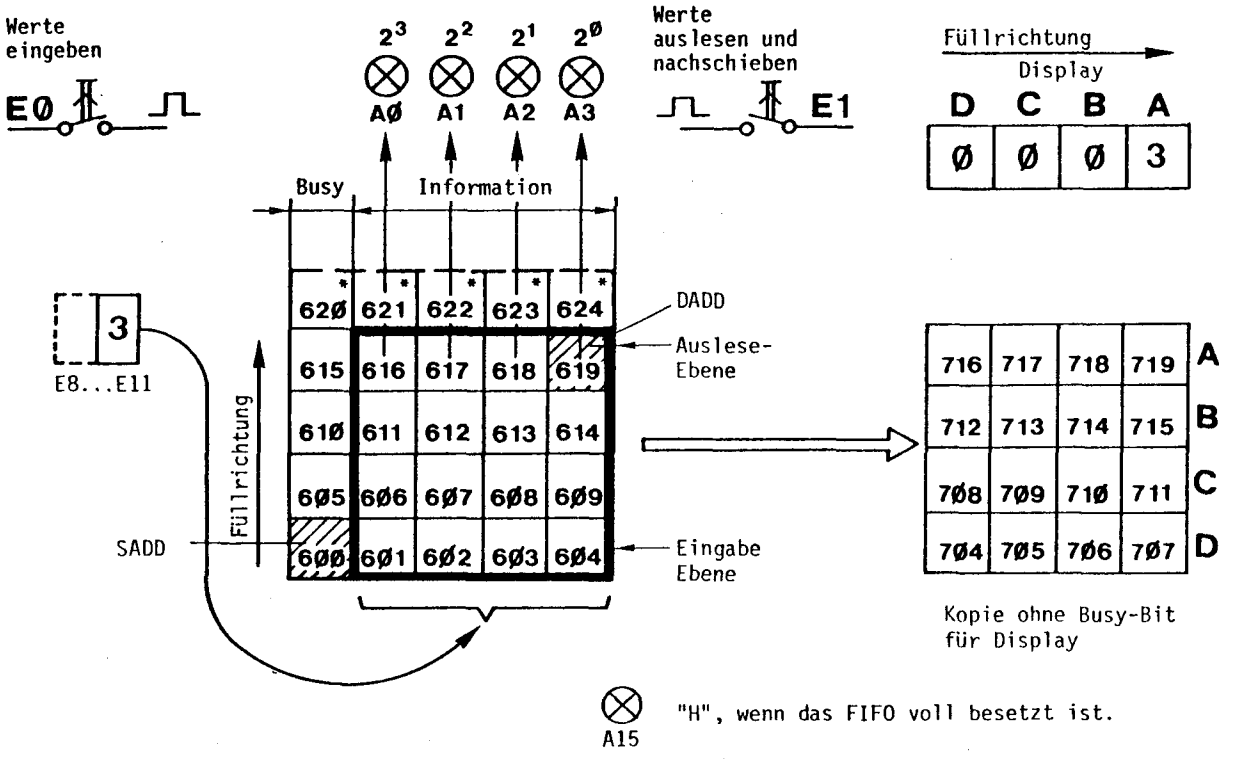
schieben rückwärts

*) Die Merker 400...403 müssen auf Null gesetzt werden, da mit dem Code 19 zusammenhängend 20 Bit BCD in den Zähler eingelesen werden.

Beispiel 1Ø FIFO-Register (Verwendung von PAS 251)

Beschreibung

Das FIFO-Register ist 4 Bit lang und 4 Bit breit. Zur Eingabe wird der BCD-Schalter auf den Eingängen E8...E11 verwendet. Das Register besteht aus den Merkern 6ØØ...619. Ueber den Eingang EØ wird ein BCD-Wert ins FIFO geschrieben, mit E1 wird ein Wert auf die Ausgänge AØ...A3 ausgelesen. Der Ausgang A15 wird aktiviert, wenn das Register voll besetzt ist. Die Ziffern werden im Operand-Display angezeigt.



Um den Inhalt des FIFOs im Operand-Display anzeigen zu können, muss es in eine Form gebracht werden, die nur aus dem Informationsteil besteht. Dazu wird das Register, ohne das "Busy"-Bit, auf einen anderen Merkerbereich kopiert. Nun kann es mit Code 19 (5 x 4 Bit BCD) auf einen Zähler geladen und angezeigt werden.

---

*) Die Merker 62Ø...624 müssen reserviert werden, da sie beim PAS 25Ø als Umlaufbuffer verwendet werden.

Lösung 1Ø

```
1201  Ø1  STH    Ø
1202  Ø9  DYN   52Ø

1203  22  JIZ  1227
1204  11  SEO   6ØØ  ; Busy setzen
1205  16  SEI     Ø
1206  Ø1  STH  1ØØ8
1207  1Ø  OUT  16Ø1      BCD-Wert
1208  27  INI     3      eingeben
1209  21  JIO  12Ø6

1211  29  PAS   251
1212  ØØ  ØØ    6ØØ  ; SADD
1213  ØØ  ØØ    619  ; DADD
1214  ØØ  ØØ      5  ; 5 Bit (4 + Busy)
1215  ØØ  ØØ      Ø
1216  ØØ  ØØ      Ø
1217  ØØ  ØØ      Ø
1218  ØØ  ØØ      Ø
1219  ØØ  ØØ      Ø
122Ø  ØØ  ØØ      Ø

1222  Ø1  STH    6ØØ  ; Busy ?
1223  1Ø  OUT     15  ; FIFO voll
1224  2Ø  JMP   126Ø  --->

1227  Ø1  STH    1
1228  Ø9  DYN   521
1229  22  JIZ  1292

1231  16  SEI     Ø
1232  Ø1  STH  1616      BCD-Wert
1233  1Ø  OUT  1ØØØ      auslesen
1234  27  INI     3
1235  21  JIO  1232

1241  29  PAS   25Ø
1242  ØØ  ØØ    6ØØ  ; SADD
1243  ØØ  ØØ    619  ; DADD
1244  ØØ  ØØ      5  ; N = 5 Bit
1245  ØØ  ØØ      Ø
1246  ØØ  ØØ      Ø
1247  ØØ  ØØ      Ø
1248  ØØ  ØØ      Ø
1249  ØØ  ØØ      Ø
125Ø  ØØ  ØØ      Ø

1252  16  SEI     Ø
1253  12  REO  16ØØ
1254  27  INI     4
1255  21  JIO  1253
1256  12  REO    15  ; FIFO frei
```

Wert in die Eingabe-Ebene einlesen

Auslesen und auf Ausgänge AØ...A3 übertragen

Nachschieben & Null eingeben

(1)

Kopieren des FIFOs auf den Merkerbereich 7Ø4...719 für DTC-Display

Mit Hilfe des PAS 25Ø (alternativ) Bit um Bit

```
1260  16  SEI     Ø
1261  Ø1  STH  16ØØ
1262  1Ø  OUT  17ØØ
1263  27  INI    19
1264  21  JIO  1261
```

FIFO kopieren

```
1270  16  SEI     Ø
1271  29  PAS   25Ø
1272  ØØ  ØØ    719
1273  ØØ  ØØ   17Ø1
1274  ØØ  ØØ      4
1275  ØØ  ØØ      Ø
1276  ØØ  ØØ      Ø
1277  ØØ  ØØ      Ø
1278  ØØ  ØØ      Ø
1279  ØØ  ØØ      Ø
1280  ØØ  ØØ      Ø
1281  27  INI     3
1282  21  JIO  1271
```

Eliminieren des Busy-Bit 1)

```
1260  16  SEI     3
1261  Ø1  STH  1616
1262  1Ø  OUT  1716   } A
1263  28  DEI     Ø
1264  21  JIO  1261
```

```
1266  Ø1  STH  1611
1267  1Ø  OUT  1712   } B
1268  27  INI     3
1269  21  JIO  1266
```

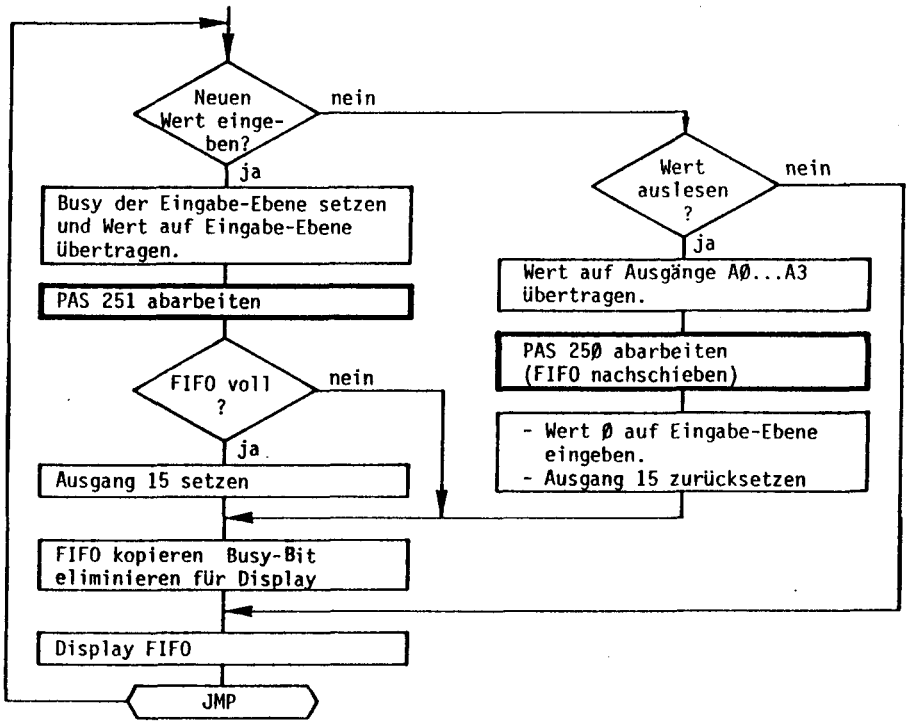
```
1271  Ø1  STH  16Ø6
1272  1Ø  OUT  17Ø8   } C
1273  28  DEI     Ø
1274  21  JIO  1271
```

```
1276  Ø1  STH  16Ø1
1277  1Ø  OUT  17Ø4   } D
1278  27  INI     3
1279  21  JIO  1276
```

```
1285  12  REO  17ØØ   } 2)
1286  28  DEI     Ø
1287  21  JIO  1285
```

```
1290  15  SCR   3Ø1
1291  19   19   719   } 3) FIFO auf Zähler und Display
1292  31  DTC   3Ø1
1293  2Ø  JMP  12Ø1
```

Flussdiagramm



---

1) Die 4 Merker 697...7ØØ müssen als Merker-Buffer reserviert werden, um die einwandfreie Abarbeitung des Befehls PAS 25Ø zu gewährleisten.

2) Die Merker 7ØØ...7Ø3 müssen auf Null gesetzt werden, da mit dem Code 19 zusammenhängend 5 x 4 Bit BCD in den Zähler eingelesen werden.

3) Das ganze Programm dieser Seite ist nur erforderlich, wenn die Werte im FIFO via DTC angezeigt werden sollen.

Beispiel 11 Störmeldungen
(Speicherung im FIFO-Register mittels PAS 251)

Beschreibung

Verschiedene Störungen sollen im Klartext ausgegeben werden. Es ist darauf zu achten, dass die Störungen in der Reihenfolge ausgedruckt werden, wie sie auftreten. Ferner dürfen, währenddem eine Störmeldung ausgedruckt wird, weiter eintreffende Störmeldungen nicht verloren gehen.

Aufgabe 11

Ueber die Eingänge Ø...3 sollen 4 verschiedene Störungen simuliert und entsprechend der Reihenfolge ihres Auftretens, mit Datum und Uhrzeit versehen, ausgedruckt werden, und zwar wie folgt:

(Der Einfachheit halber wird die Uhrzeit angegeben, bei welcher die entsprechende Störung ausgedruckt wird und nicht, wann sie aufgetreten ist.)

EØ : Datum, Uhrzeit
Alarm Ø
Kommentar

E1 : Datum, Uhrzeit
Alarm 1
Kommentar

E2 : Datum, Uhrzeit
Alarm 2
Kommentar

E3 : Datum, Uhrzeit
Alarm 3
Kommentar

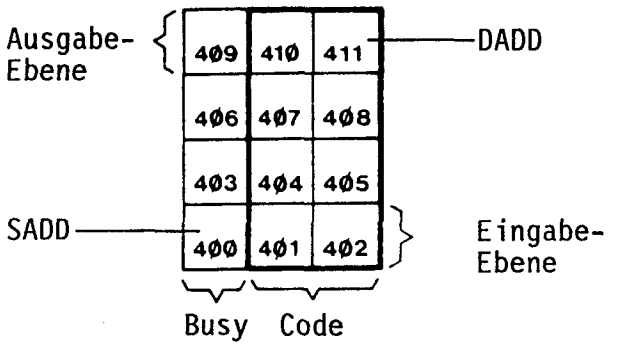
Lösung 11

Die Aufgabe wird mit Hilfe eines FIFO-Registers gelöst. Jeder Störmeldung wird ein bestimmter Erkennungscode zugeteilt. Tritt nun eine Störung auf, wird der entsprechende Code auf die Eingabe-Ebene des FIFOs eingelesen und gleichzeitig das Busy-Bit (SADD) auf "H" gesetzt. Anschliessend wird der PAS 251 abgearbeitet und somit die Information (Code + Busy) bis zum letzten freien Platz des FIFOs geschoben (wenn FIFO leer: bis zur Ausgabe-Ebene).
Ist das Busy-Bit der Ausgabe-Ebene auf "H", wird anhand des Codes untersucht, um welche Störung es sich handelt. (Code ØØ bedeutet Alarm Ø, Code 1Ø ---> Alarm 1, Code Ø1 ---> Alarm 2, Code 11 ---> Alarm 3).
Je nach Code wird der Zähler C319 auf den Wert Ø, 1Ø, 2Ø oder 3Ø gesetzt.
Anschliessend wird das Indexregister mit dem Wert des Zählers C319 geladen.
Durch die indexierte Textausgabe

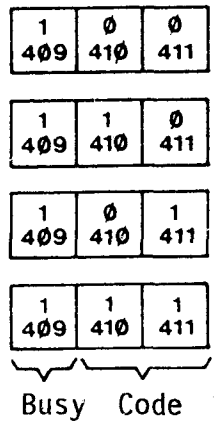
```
PAS   23
 ØØ 111Ø
```

wird zur Textnummer 11Ø der Wert des Indexregisters addiert (Ø, 1Ø, 2Ø oder 3Ø) und somit Text Nr. 11Ø, 12Ø, 13Ø oder 14Ø ausgegeben, was den Alarmmeldungen Ø, 1, 2 oder 3 entspricht.
Nach erfolgter Textausgabe wird mittels PAS 25Ø das FIFO-Register geschoben.

Das FIFO sieht demnach wie folgt aus:



Interpretation der Ausgabe-Ebene:



Alarm Ø ---> C319 mit Wert Ø laden

Alarm 1 ---> C319 mit Wert 1Ø laden

Alarm 2 ---> C319 mit Wert 2Ø laden

Alarm 3 ---> C319 mit Wert 3Ø laden

Programm

```
1301  29   PAS   100
1302  00    00   902 ; 4800 Baud
1303  00    00   425 , TXB
1304  00    00   254
1305  00    00   254
1306  00    00     0
1307  00    00     0
1308  00    00     0
1309  00    00     0
1310  00    00     0

1321  01   STH     0 ; Alarm 0
1322  09   DYN   420
1323  22   JIZ  1327
1324  11   SEO   400 ; SADD
1325  12   REO   401 } Code für
1326  12   REO   402 } Alarm 0
1327  01   STH     1 ; Alarm 1
1328  09   DYN   421
1329  22   JIZ  1333
1330  11   SEO   400 ; SADD
1331  11   SEO   401 } Code für
1332  12   REO   402 } Alarm 1
1333  01   STH     2 ; Alarm 2
1334  09   DYN   422
1335  22   JIZ  1339
1336  11   SEO   400 ; SADD
1337  12   REO   401 } Code für
1338  11   SEO   402 } Alarm 2
1339  01   STH     3 , Alarm 3
1340  09   DYN   423
1341  22   JIZ  1345
1342  11   SEO   400 ; SADD
1343  11   SEO   401 } Code für
1344  11   SEO   402 } Alarm 3
1345  01   STH   400 ; SADD = "H"?
1346  22   JIZ  1361

1351  29   PAS   251 ; FIFO
1352  00    00   400 ; SADD
1353  00    00   411 ; DADD
1354  00    00     3 ; Breite = 3 Bit
1355  00    00     0
1356  00    00     0
1357  00    00     0
1358  00    00     0
1359  00    00     0
1360  00    00     0

1361  02   STL   409 ; Lese-Flag 0
1362  05   ORH   425   oder TXB
1363  21   JIO  1321
1364  15   SCR   319
1365  00    00     0
1366  01   STH   410 ; DADD-1
1367  15   SCR   319
1368  27    27    10
1369  01   STH   411 ; DADD
1370  15   SCR   319
1371  27    27    20
1372  16   SEI   319
1373  29   PAS    23
1374  00    00  1110 ; Text 110+C319
1375  12   REO   409 ; Busy Ausgabe-
                       Ebene

1381  29   PAS   250 ; Nachschieben
1382  00    00   400 ; SADD
1383  00    00   411 ; DADD
1384  00    00     3 ; Breite = 3 Bit
1385  00    00     0
1386  00    00     0
1387  00    00     0
1388  00    00     0
1389  00    00     0
1390  00    00     0
1391  20   JMP  1321
```

```
110   .$.H^M^J.A.L.A.R.M.
111   .0^M^J.$.L.1.4.5^M^@

120   .$.H^M^J.A.L.A.R.M.
121   .1^M^J.$.L.1.4.5^M^@

130   .$.H^M^J.A.L.A.R.M.
131   .2^M^J.$.L.1.4.5^M^@

140   .$.H^M^J.A.L.A.R.M.
141   .3^M^J.$.L.1.4.5^M^@

145   .K.O.M.M.E.N.T.A.R.:
146   .$.X.0.7.0^M.$.  .0.1
147   .0.$.Y.0.7.0^M.$.  .0
148   .1.0.$.Z.0.7.0.$.U^@
```

Alphabetische Befehlsübersicht 2 + 3

Notizen:

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

Weitere Vertretungen

Belgique Landis & Gyr Belge SA, Dépt. Industrie
Avenue des Anciens Combattants 190, B-1140 Bruxelles
☎ 02 729 02 11, Tx 65 929, Fax 02 242 88 31

Danmark E. Friis-Mikkelsen A/S
Krogshojvej 51, DK-2880 Bagsvaerd
☎ 045 42 98 63 33, Tx 37 350, Fax 045 42 98 81 40

Deutschland SAIA GmbH
Daimlerstrasse 1 K, D-6072 Dreieich
☎ 06103 8906–0, Fax 06103 89 06 66

España Landis & Gyr BC SA
Batalla del Salado 25, Apartado 575, 28045 Madrid
☎ 91 467 19 00, Tx 22 976, Fax 91 239 44 79

France SAIA Sàrl.
10, Blvd. Louise Michel, F-92230 Gennevilliers
☎ 1 4086 03 45, Tx 613 189, Fax 1 4791 40 13

Great Britain Burgess-SAIA 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 48600600, Fax 02 48600692

Nederland Landis & Gyr BV, Div. Electrowater
Kampenringweg 45, Postbus 444, NL-2800 AK-Gouda
☎ 01820 65 685, Tx 20 657, Fax 01820 32 437

Norge Malthe Winje & Co A/S
Cort Adelersgt. 14, Postboks 2440, Solli, N-0202 Oslo 2
☎ 02 55 86 40, Tx 19 629, Fax 02 55 22 11

Österreich Landis & Gyr Gesellschaft m.b.H
Breitenfurterstrasse 148, Postfach 9, A-1230 Wien
☎ 0222 80 108-0, Tx 132 706, Fax 0222 00 100 313

Portugal Infocontrol Electronica e Automatismo LDA.
Av. da Igreja No 68-1° Esq., P-1700 Lisboa
☎ 01 7751 61-65, Tx 63 454, Fax 01 7756 87

Suomi Finnland Landis & Gyr Suomi OY
SF-02430 Masala
☎ 8 029731, Tx 121 039, Fax 8 02975531

Sverige Beving Elektronik AB
St. Eriksgatan 113a, Box 21 104, S-10031 Stockholm
☎ 08 15 17 80, Tx 10 040, Fax 08 33 68 63

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 (Australia) Pty Ltd
411 Ferntree Gully Road, P.O. Box 202, Mount Waverley, Vic. 3149
☎ 3 544-2322, Tx 32 244, Fax 3 543 7496

Argentina Electromedidor S.A.I.y C.
Defensa 320, RA-1065 Buenos Aires
☎ 1 337125, Tx 23 377, Fax 1 3319582

Gedruckt in der Schweiz 26/722 D2 1.91 K03