



# saia<sup>®</sup> PC

## Speicherprogrammierbare Steuerungen

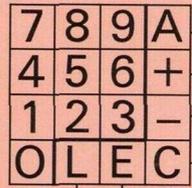
NIVEAU 1

### Betriebsarten und Verwendung des Programmeingabegerätes PCA2.P10

- Betriebswahlschalter**
- RUN
  - BREAK
  - STEP
  - MAN
  - LCM
  - PROG

Ein- und Ausgänge sind aktiv  
 nur PCA2

**Tastenfeld am Programmeingabegerät PCA2.P10**



- Adresse
- Adresse +1
- Adresse -1
- Clear (Löschen)
- Enter (Eingeben)
- LCM (Laden des Kopierspeichers)

#### RUN

**Normaler Programmablauf**  
 Bei nicht angeschlossenem Programmeingabegerät ist die PCA2 unabhängig von der Stellung des Betriebswahlschalters immer auf RUN-Betrieb (im Gegensatz zu PCA1).

#### BREAK

**Unterbruch des Programmablaufes mit anschließendem Einzelschritt**

[+] die zuletzt bearbeitete Programmzeile wird angezeigt

[+] [+] ... Abarbeiten der folgenden Programmzeilen im Einzelschritt

Die ganze Programmzeile sowie der Zustand des Verknüpfungsspeichers ACCU wird angezeigt.

Der Programmablauf im BREAK-Betrieb erfolgt in allen Teilen analog demjenigen in der Betriebsart RUN, d.h. bei

**Setzen eines (Breakpoint)**

[A] 820 [+] Programm läuft selbständig ab (etwa 10mal langsamer als im RUN-Betrieb) bis Schrittadresse 820 und hält dann an

[+] [+] ... Abarbeiten der folgenden Programmzeilen im Einzelschritt

Parallelprogrammen werden alle Programme parallel abgearbeitet.

Bei Rückstellung des Betriebswahlschalters auf RUN wird das Programm da weiterabgearbeitet, wo im Einzelschritt aufgehört wurde.

#### STEP

**Springen auf vorgewählte Schrittadresse**

[+] die zuletzt bearbeitete Programmzeile wird angezeigt

[A] 139 [+] Sprung auf die Schrittadresse 139

[+] [+] ... Abarbeiten der folgenden Programmzeilen im Einzelschritt

Die ganze Programmzeile sowie der Zustand des Verknüpfungsspeichers ACCU wird angezeigt.

Durch Sprung auf die Anfangsschrittadresse eines bestimmten Parallelprogrammes ist es möglich, nur dieses abzuarbeiten.

Bei Rückstellung des Betriebswahlschalters auf RUN wird nur dieses Parallelprogramm da weiterabgearbeitet, wo im Einzelschritt aufgehört wurde.

#### MAN

**Manuelles Abfragen und Setzen/Rücksetzen des logischen Zustandes von Elementen**  
 (Elemente = Eingänge, Ausgänge, Merker, Zähler und Timer)

Abfragen:

[A] 32 Abfrage des Elementes 32. In der letzten Ziffer des Operand wird 1 oder 0 angegeben (High oder Low)

Setzen/Rücksetzen:

[E] gibt die Eingabe für den logischen Zustand frei

[1]/[0] Setzen des Elementes auf High oder Low

#### LCM

**Laden des Kopierspeichers** (Diese Funktion ist nur mit der Baureihe PCA2 möglich.)

**STEP**

[A] XXXX Anfangs-Schrittadresse des umzuladenden Programmes eingeben

**CODE OPERAND**

[E] 00 XXXX End-Schrittadresse eingeben

[L] Starten des Umladevorganges

Umladen, d.h. Kopieren eines Programmes von RAM auf EPROM, EPROM auf EPROM, EPROM auf RAM, RAM auf RAM (Kopierspeicher auf unterem Steckplatz).

Das Umladen des Programmes erfolgt in einer Geschwindigkeit von 1K/100s und kann im Display mitverfolgt werden.

#### PROG

**Programmieren**

**STEP**

[A] XXXX Die Taste [A] setzt das Schrittregister auf 0. Anschliessend Anfangs-Schrittadresse eingeben.

**CODE OPERAND**

[E] XX XXXX Die Taste [E] setzt das Code- und Operandregister auf 0. Eventueller Speicherinhalt wird gelöscht.

Eingeben des Befehls im Zahlencode sowie des Operanden

[C] Löschen einer Fehleingabe

[+] [+] bzw. [-] [-] Programm prüfen

Ein Programm kann nur auf einen RAM-Anwenderspeicher eingegeben oder überschrieben (korrigiert) werden.

# Befehlsübersicht Niveau 1

Schrittadresse  
**STEP**

Befehl im Zahlencode  
**CODE**

Element- oder Sprungadresse  
**OPERAND**

Verknüpfungs-  
speicher  
**ACCU = 1**



0 1 3 9

0 1

0 3 5 6

Anzeige auf  
Programm-  
eingabegerät  
PCA2.P10

Adressbereiche:

EA = Elementadresse  
S = Schrittadresse  
PP = Nummer des  
Parallel-  
programmes  
OP = beliebiger  
Operand  
I = Indexierwert  
(i) = indexierbar

**Wirkung  
bezüglich  
ACCU**

Wird nur ausge-  
führt, wenn ACCU =  
Befehl setzt ACCU =

**Logik-  
befehle**



**Schalt-  
befehle**



**Zeit-  
und  
Zähl-  
befehle**

**Sprung-  
befehle**



**Warte-  
befehle**

**Hilfs-  
befehle**



Zahlen- code	Mnemo- code 2. Zeile	Befehl englisch	Beschreibung	Adressbereiche:	Wird nur ausge- führt, wenn ACCU =	Befehl setzt ACCU =
01 02	STH STL	Start High Start Low	{ Beginn einer Verknüpfung: } High Element abgefragt nach } Low	EA = 0...999 (i)		1/0
03 04	ANH ANL	And High And Low	{ Und-Verknüpfung: } High ACCU mit Element abgefragt nach } Low	EA = 0...999 (i)		1/0
05 06	ORH ORL	Or High Or Low	{ Oder-Verknüpfung: } High ACCU mit Element abgefragt nach } Low	EA = 0...999 (i)		1/0
07	XOR	Exclusive Or	Exklusiv-Oder-Verknüpfung: ACCU mit adressiertem Element vergleichen	EA = 0...999 (i)		1/0
08	NEG	Negate Accu	Umkehrung des ACCU-Inhaltes	0		0/1
09	DYN	Dynamic Control	Flankentriggerung eines Signales bzw. Dynamisierung einer Verknüpfung	EA = 288...999 (i)		1/0
10	OUT	Set Output with Status of Accu	Ausgang oder Merker mit Inhalt des ACCU setzen	EA = 0...255, 288...999 (i)		1/0
11 12	SEO REO	Set Output Reset Output	Ausgang oder Merker speichernd setzen Ausgang oder Merker speichernd rücksetzen	EA = 0...255, 288...999 (i)		1
13	COO	Complement Output	Ausgang oder Merker auf Zustand abfragen und umgekehrt setzen	EA = 0...255, 288...999 (i)		1
14	STR 00 <sup>1)</sup>	Set Timer -	Timer auf vorgewählten Wert setzen und Zeitablauf starten Zeitwert in 1/10s (bzw. 1/100s)	EA = 256...287 (i) 0...2047		1
15	SCR 00 <sup>1)</sup>	Set Counter -	Zähler auf vorgewählten Wert setzen Zählwert <sup>1)</sup> für Werte > 2047 siehe Basis-Handbuch C 7.3.	EA = 256...287 (i) 0...2047		1
16 17 18	- - -	- - -	Externe Werteingabe in BCD Format: Einlesebefehl für 2 x 4 Bit BCD x1 Einlesebefehl für 2 x 4 Bit BCD x10 Einlesebefehl für 2 x 4 Bit BCD x100	EA = 7...255, 295...999		
17 18	INC DEC	Increment Counter Decrement Counter	Erhöht } Erniedrigt } den Zählerstand um 1	EA = 256...287 (i)		1
20	JMP	Unconditional Jump	Unbedingter Sprung auf Schrittadresse	S = 0...2047		1
21 22	JIO JIZ	Jump if Accu is One Jump if Accu is Zero	{ Sprung auf Schrittadresse wenn } ACCU = 1 ACCU = 0	S = 0...2047	1 0	1 1
23 24	JMS RET	Jump to Subroutine Return from Subroutine	Sprung ins Unterprogramm Rücksprung aus Unterprogramm	S = 0...2047 0		1 1
25 26	WIH WIL	Wait if High Wait if Low	{ Wartet solange Element } High Low	EA = 0...999 (i)		1 1
00	NOP	No Operation	Löst keinerlei Funktion aus	0		
19	SEA	Set Accumulator = 1	Setzt ACCU = 1	0		1
16	SEI	Set Index	Setzt Indexregister auf Anfangswert (I0)	I0 = 0...255		1
27 28	INI DEI	Increment Index Decrement Index	Erhöht } Erniedrigt } Indexregister um 1 bis zum Endwert (I)	I > I0 I < I0		1/0 1/0
29	PAS 00	Program Assignment -	Zuweisung des Parallelprogrammes Anfangs-Schrittadresse des Parallelprogrammes	PP = 0...15 S = 0...2047		
30 31	DOP DTC	Display Operand Display Timer/Counter	Anzeige der im Operand stehenden Zahl Anzeige des Timer- oder Zählerstandes	OP = 0...2047 EA = 256...287 (i)	0 1	



# saia<sup>®</sup> PC

## Speicherprogrammierbare Steuerungen NIVEAU 2

### Eingabe von Texten in den Textspeicher

#### 1 Programmierung der Datenschnittstelle als Editor (mit Programmeingabegerät P10, PROG-Betrieb)

4001	PAS	(29)	100	
4002	00		898	Parameter (entsprechend Peripheriegerät)
4003	01		63	
4004	00		254	} Ein Element das dauernd auf Low bleibt
4005	00		254	
4006	00		0	
4007	00		0	
4008	00		0	
4009	00		0	
4010	00		0	
4011	PAS	(29)	23	} TXn = Text-Nummer
4012	00		TXn	
4013	JMP	(20)	0	} für Textausgabe im STEP-Betrieb
4014	00		4011	

#### 2 Aktivierung des Editors

Im STEP-Betrieb mit Programmeingabegerät P10 die Schrittadresse 4001 anspringen und bis 4011 abarbeiten → Editor ist bereit

#### 4 Eingabe-Modus des Editors (mit Peripheriegerät)

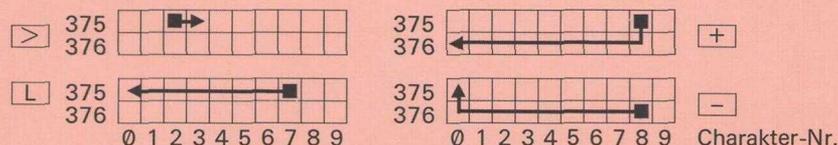
- CTRL T** Texteingabe erfolgt ab Charakter-Nummer, welche im Display-Modus mit dem Cursor vorgewählt wurde. Eingabe des Lauftextes sowie der Steuer- und Funktionszeichen
- CTRL @\*** Abschluss eines Textes
- ^ CR** Wird als <CR> (Carriage Return) in den Textspeicher übernommen
- CR** Wechsel in den Display-Modus

\* vom Peripheriegerät abhängig

#### 3 Display Modus des Editors (Display des Textspeicher-Inhaltes auf Peripheriegerät)

TXn **CR** Textzeile mit Text-Nummer TXn wird angezeigt

Steuern des Cursors



#### 5 Textausgabe zur Kontrolle

Im STEP-Betrieb mit Programmeingabegerät P10 **[+]**-Taste 2mal betätigen, wobei jedesmal der Text zur Kontrolle ausgegeben wird.

### Funktionszeichen im Textspeicher

Alphanumerische Eingabe	Eingabe mit P10	Beschreibung
\$C Cn*	36, 67, Cn	Ausgabe des Inhaltes vom Timer/Zähler-Register Cn
\$c Cn*	36, 99, Cn	5 Dezimal-Ziffern ohne Vornullunterdrückung
\$E En*	36, 69, En	Ausgabe des Zustandes der 8 Elemente En...En-7 (8 Charakter 1 oder 0)
\$e En*	36, 101, En	Ausgabe des ASCII-Charakters, gebildet aus den 8 Elementen En...En-7
\$T	36, 84	<b>Ausgabe des Inhalts der Datum-Uhr:</b> 14 : 04 / 18 : 50 : 42 Woche Tag Stunde Minute Sekunde (14.) (Don.)
\$H	36, 72	84 : 04 : 05 / 18 : 50 : 42 Jahr Monat Tag Stunde Minute Sekunde
\$D	36, 68	84 : 04 : 05 Jahr Monat Tag
\$_ xxx*	36, _, xxx	Repetierte Ausgabe eines Charakters, der unmittelbar nach dem Zeichen \$ folgt, so oft wie in xxx festgelegt: xxx < 127 Repetition entsprechend xxx xxx = 256...319 Repetition entsprechend dem Inhalt (max. 127) des adressierten Timer/Zählers Es dürfen alle ASCII-Zeichen (auch NUL) verwendet werden, ausser den folgenden, welche Funktionen auslösen: \$, A, C, D, E, H, L, R, T, U, a, c, e, r
\$L xxx*	36, 76, xxx	Sprung auf Text-Nr. xxx mit Speicherung der Absprungadresse (Charakter-Nummer)
\$U	36, 85	Rücksprung in die Absprungadresse +1
\$\$	36, 36	\$ (Ausgabe des Zeichens Dollar)

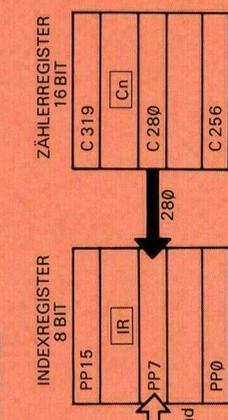
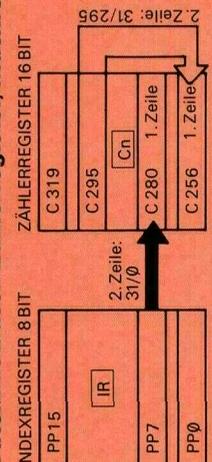
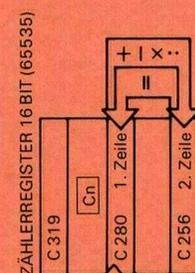
\* Für Cn, En bzw. xxx müssen immer 3 Ziffern eingegeben werden, z. B 027.

### ASCII CODE-Tabelle

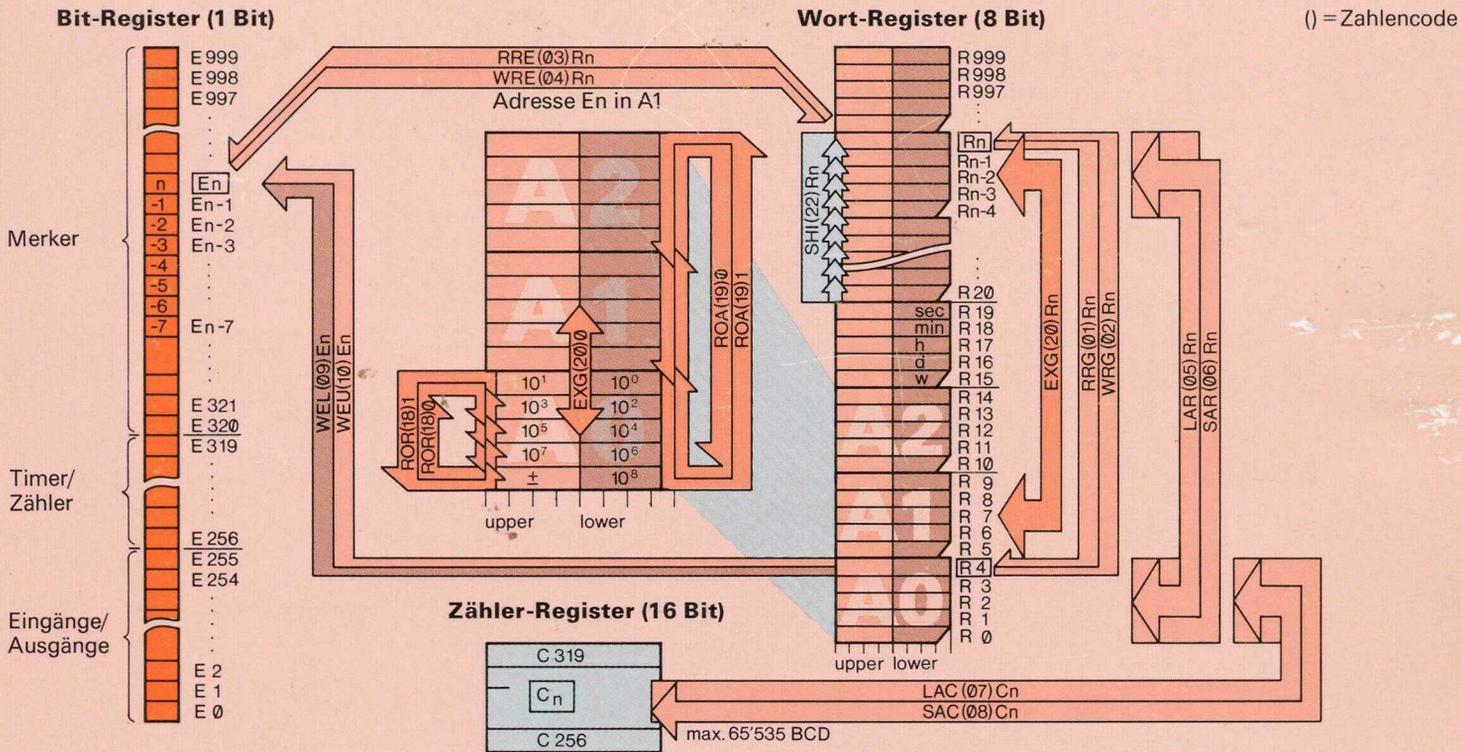
B7 BITS	B6 B5	0 0	0 1	1 0	1 1	1 0	1 1	1 0	1 1
B4B3B2B1		CONTROL		SYMBOLS		UPPERCASE		LOWERCASE	
0000	0	NUL	DLE	SP	0	a	P	'	p
0001	1	SOH	DC1	!	1	A	Q	a	q
0010	0	STX	DC2	"	2	B	R	b	r
0011	1	ETX	DC3	#	3	C	S	c	s
0100	0	EOT	DC4	\$	4	D	T	d	t
0101	1	ENQ	NAK	%	5	E	U	e	u
0110	0	ACK	SYN	&	6	F	V	f	v
0111	1	BEL	ETB	'	7	G	W	g	w
1000	0	BS	CAN	(	8	H	X	h	x
1001	1	HT	EM	)	9	I	Y	i	y
1010	0	LF	SUB	*	10	J	Z	j	z
1011	1	VT	ESC	+	11	K	[	k	{
1100	0	FF	FS	,	12	L	]	l	}
1101	1	CR	GS	-	13	M	^	m	~
1110	0	SO	RS	.	14	N	_	n	DEL
1111	1	SI	US	/	15	O	~	o	DEL

\* 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.  
So wird <FF> (Form Feed, No 12) durch gleichzeitiges Betätigen der Tasten **CTRL** und **L** (No 76) erzeugt.

Mnemo-code 2. Zeile	Beschreibung	Operand	Zahlen-code	Mnemo-code 2. Zeile	Beschreibung	Operand	Zahlen-code
STR 00...31 <sup>1)</sup>	<b>ZEIT- UND ZÄHLBEFEHLE</b> Adressierten Timer auf vorgewählten Zeitwert setzen und Zeitablauf starten. Zeitwert in 1/10 s (bzw. 1/100 s)	256...287 (i)	29	PAS	<b>PAS-BEFEHLE</b> Exklusiv-Betrieb eines Parallelprogrammes (mit blockierter Zeitbasis, Software Datum-Uhr und serieller Datenschnittstelle) 2. Zeile ist immer 0	16	29
SCR 00...31 <sup>1)</sup>	Adressierten Zähler auf vorgewählten Zählwert setzen Zählwert	256...319 (i)	29	PAS 00	Ende des Exklusiv-Betriebes 2. Zeile ist immer 0	0	29
00 : 15	<sup>1)</sup> <b>Werte 0 ... 32 767</b> 00 (≙ 0) 01 (≙ 2048) 02 (≙ 4096) 03 (≙ 6144) 04 (≙ 8192) 05 (≙ 10240) 06 (≙ 12288) 07 (≙ 14336) 08 (≙ 16384) 09 (≙ 18432) 10 (≙ 20480) 11 (≙ 22528) 12 (≙ 24576) 13 (≙ 26624) 14 (≙ 28672) 15 (≙ 30720)	+ Wert im Operand 0...2047	29	PAS 00	Begrenzt die Anzahl der aktiven, assignierten Parallelprogramme 2. Zeile bestimmt die aktiven Parallelprogramme	17 0	29
16 17 18 19 24 25 26	<sup>1)</sup> <b>Einlesebefehle</b> Einlesebefehl für 2mal 4 Bit BCD x1 Einlesebefehl für 2mal 4 Bit BCD x10 Einlesebefehl für 2mal 4 Bit BCD x100 Einlesebefehl für 5mal 4 Bit BCD x1 Einlesebefehl für 8 Bit binär Einlesebefehl für 12 Bit binär Einlesebefehl für 16 Bit binär	7...999 (i)	29	PAS 00	Befehl zur Textausgabe 2. Zeile bestimmt die Anfangs-Text-Nummer Ausgabe erfolgt bis zum Zeichen <NUL> im Text	23 0...818	29
27 28 29 30	<sup>1)</sup> <b>Arithmetische Funktionen</b> Addition Subtraktion Multiplikation Division	höchste Adresse der Elementenreihe (LSB = Least Significant Bit)	29	PAS 00 PAS xx	Überprüfen des Systemprogrammes durch <Check-Sum> Überprüfen des Anwenderprogrammes durch <Check-Sum> (31...38 ≙ 1..K...8..K) 2. Zeile enthält die Prüfsumme	30 0 31...38 xxxx	29
31	<sup>1)</sup> <b>Datenaustausch Zählerregister/Indexregister</b> INDEXREGISTER 8 BIT INDEXREGISTER 16 BIT ZÄHLERREGISTER 8 BIT ZÄHLERREGISTER 16 BIT Zähler wird mit dem Wert des Indexregisters geladen Zähler der 1. Zeile wird mit dem Wert des adressierten Timer/Zählers geladen Zähler der 2. Zeile wird mit dem Wert des adressierten Zählers geladen	1...255 256...319 (i)	29	PAS xy	Datenaustausch zwischen Datum-Uhr und Zählerregister xy bestimmt die Art der Funktion: x: 0 ≙ Schreiben in das Register der Datum-Uhr 1 ≙ Lesen aus dem Register der Datum-Uhr y: 0 ≙ Woche des Jahres (Wert: 1...53) 1 ≙ Tag der Woche (Wert: 1...7) 2 ≙ Jahr (Wert: 0...99) 3 ≙ Monat (Wert: 1...12) 4 ≙ Tag des Monats (Wert: 1...31) 5 ≙ Stunden (Wert: 0...23) 6 ≙ Minuten (Wert: 0...59) 7 ≙ Sekunden (Wert: 0...59) Cn bestimmt den Timer/Zähler, aus welchem der in das Register der Datum-Uhr zu schreibende Wert entnommen wird bzw. in welchen der aus dem Register der Datum-Uhr gelesene Wert abzulegen ist.	50 Cn	29
MP/JIO IZ/JMS 00 <sup>2)</sup>	<b>SPRUNGBEFEHLE</b> 2) Schrittdressen ≤ 2047 können ein- oder zweizeilig eingegeben werden Schrittdressen > 2047 müssen zweizeilig eingegeben werden	0 0 0...8191 <sup>2)</sup>	29	PAS	10-zeiliger Befehl zum Festlegen der Übertragungsparameter der seriellen Datenschnittstelle und des Modus T, E, C, N oder P	100	29
SEI	<b>HILFSBEFEHLE</b> Indexregister wird mit dem Wert aus dem Operand (Konstante) geladen Indexregister wird mit dem Wert aus adressiertem Timer/Zähler geladen Der Endwert für das Indexregister steht im Operand (Konstante) Der Endwert für das Indexregister steht im adressierten Timer/Zähler	0...255 256...319	29	PAS 00	10-zeiliger Befehl für das Interrupt-Management (schnelle Reaktion durch Exklusiv-Betrieb gemäss Interrupt-Service-Routine) Ende des Interrupt-Managements 2. Zeile ist immer 0	190 19	29
INI DEI	Der Endwert für das Indexregister steht im Operand (Konstante) Der Endwert für das Indexregister steht im adressierten Timer/Zähler	0...255 256...319	29	PAS PAS	10-zeiliger Befehl für die Parameter von maximal 32 PID-Regelkreisen	200 201 202	29
		0...255 256...319	29	PAS PAS	10-zeiliger Befehl für die Definition und Aktivierung eines Rotations- oder Schieberegisters 10-zeiliger Befehl für die Definition und Aktivierung eines Stapelregisters (FIFO)	250 251	29



Befehlssatz im Wortprozessor



Datentransfer-Befehle

Zahlen-code	Mnemo-code	Operand	Befehl englisch	Beschreibung	Daten-format
01	RRG	Rn	Read Register	Lese Wort Rn und speichere es in R4	8 Bit i
02	WRG	Rn	Write into Register	Schreibe in Rn das Wort von R4	8 Bit i
03	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
04	WRE	Rn	Write Register with Elements	Schreibe in Rn den Inhalt der 8 Elemente En...En-7, adressiert durch A1	8 Bit i
05	LAR	Rn	Load A0 with Registers	Lade A0 mit Registerblock Rn	5 x 8 Bit i
06	SAR	Rn	Store A0 into Registers	Speichere A0 in den Registerblock Rn	5 x 8 Bit i
07	LAC	Cn	Load A0 with Counter	Lade A0 mit Zähler Cn	BCD i
08	SAC	Cn	Store A0 into Counter	Speichere A0 in den Zähler Cn	BCD i
09	WEL	En	Write Elements with { Lower } Digit	Schreibe das Digit { R4 (10 <sup>0</sup> ) } in die Elemente	4 Bit i
10	WEU	En			

Behandlung der Arithmetik-Register

18	ROR	0/1	Rotate Register	Rotiere R4...R3...R2...R1...R0 bzw. R0...R1...R2...R3...R4	
19	ROA	0/1	Rotate A	Rotiere A0...A1...A2 bzw. A2...A1...A0	
20	EXG	0	Exchange A1 with A0	Tausche A1 mit A0	5 x 8 Bit
		Rn	Exchange A1 with Rn...Rn-4	Tausche A1 mit Rn...Rn-4	5 x 8 Bit i

i = indexierbar

Funktionszeichen im Textspeicher (ergänzend zu Niveau 2)

Alphanumerische Eingabe	Eingabe mit P10	Beschreibung
\$A Rn*	36, 65, ...	Inhalt des Registerblockes Rn, 10 Dezimal-Ziffern ohne Vornullunterdrückung**
\$a Rn*	36, 97, ...	Inhalt des Registerblockes Rn, 9 Dezimal-Ziffern (mit Vorzeichen, falls negativ) mit Vornullunterdrückung (Spaces)**
\$R Rn*	36, 82, ...	Inhalt des Registerwortes Rn, 2 Dezimal-Ziffern**
\$r Rn* y	36, 114, ..., 0/1	Inhalt des LOWER bzw. UPPER-Teiles des Registerwortes, 1 Dezimalziffer, y = 0 -> LOWER, y = 1 -> UPPER**

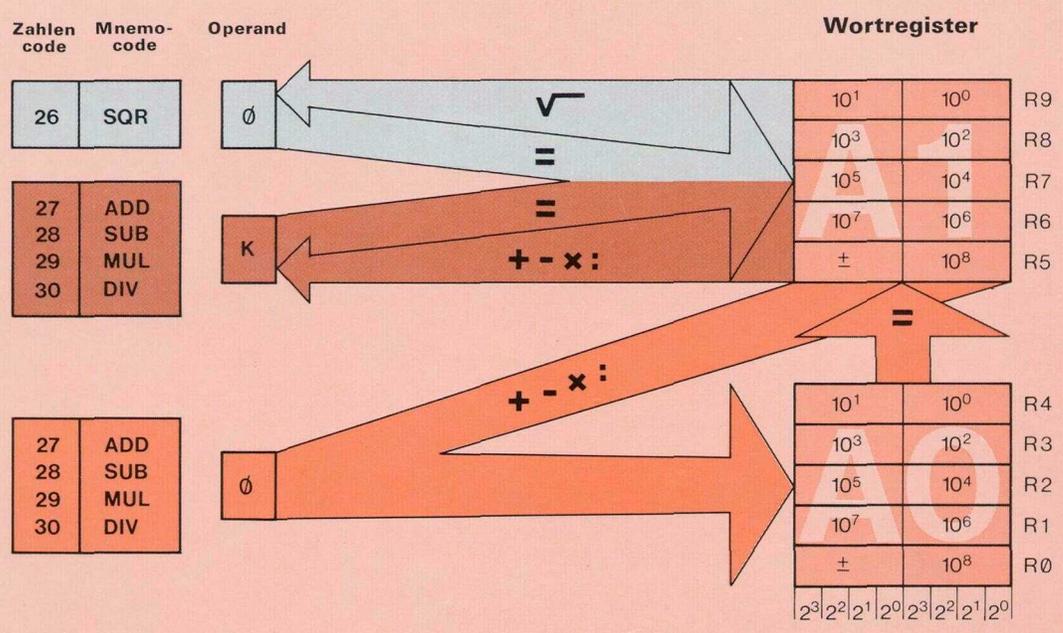
\* Für Rn müssen immer 3 Ziffern eingegeben werden, z. B. 027

\*\* Erreichen binär formatierte 4-Bit-Wörter Werte zwischen 10 und 15, so werden folgende Charaktere ausgegeben: 10 -> 11 -> 12 -> 13 -> 14 -> 15 ->

Zahlen-code	Mnemo-code	Operand	Befehl englisch	Beschreibung	Daten-format
-------------	------------	---------	-----------------	--------------	--------------

### Arithmetik-Befehle

25	CMP	En	Compare A0 with A1	Vergleiche A0 mit A1 $A1 > A0 \Rightarrow En = 1$ $A1 = A0 \Rightarrow En-1 = 1$ $A1 < A0 \Rightarrow En-2 = 1$	BCD	i
26	SQR	0	Square Root of A1	$\sqrt{A1} \Rightarrow A1$ , nur ganzzahlig	BCD	
27	ADD	0 K	Add A0 to A1 Add K to A1	$A1 + A0 \Rightarrow A1$ , <carry> $A1 + K \Rightarrow A1$ , <carry> (K = 1 ... 2047)	BCD BCD	
28	SUB	0 K	Subtract A0 from A1 Subtract K from A1	$A1 - A0 \Rightarrow A1$ , <carry> $A1 - K \Rightarrow A1$ , <carry> (K = 1 ... 2047)	BCD BCD	
29	MUL	0 K	Multiply A1 by A0 Multiply A1 by K	$A1 \cdot A0 \Rightarrow A1$ , <carry> $A1 \cdot K \Rightarrow A1$ , <carry> (K = 1 ... 2047)	BCD BCD	
30	DIV	0 K	Divide A1 by A0 Divide A1 by K	$A1 : A0 \Rightarrow A1$ , Rest in A0, <carry> $A1 : K \Rightarrow A1$ , Rest in A0, <carry> (K = 1 ... 2047)	BCD BCD	



### Behandlung der Arithmetik-Register

14	CLA	0/1/2	Clear A	Lösche Register A0 oder A1 oder A2		
15	LAI	K	Load A immediately	wenn $K \leq 99 \rightarrow$ Lade R4 mit Data wenn $K = 100 \dots 2047 \rightarrow$ Lade A0 mit Data		
16	DBN	0	Decimal to binary	Wandle A0 dezimal in A0 binär		
17	BND	0	Binary to decimal	Wandle A0 binär in A0 dezimal		

### Zähl- und Sprung-Befehle

11	INR	Rn	Increment Register	Incr. } BCD-Wert Rn um 1 { $> 99$ und setze <carry>		i
12	DER	Rn	Decrement Register	Decr. } falls Resultat { $< 0$		i
13	SNC	0 En	Skip if no <carry> Skip if $En = 0$	Überspringe nächsten { <carry> = 0 Befehl, wenn { $En = 0$		i
24	SEW	0 En	Skip to EWP if no <carry> Skip to EWP if $En = 0$	Überspringe bis EWP { <carry> = 0 oder NOP 1248, wenn { $En = 0$		i

### Diverse Befehle

00	NOP	1248	-	Wechsel von Bit- nach Wortprozessor setzt ACCU = 1		
00	NOP	0	No operation	Keine Operation		
21	CLK	En	Clock source	Zuweisung einer Zeitimpulsquelle		
22	SHI	Rn	Shift registers	Schiebe Wörter ab R20 bis Rn um 1 Adresse nach oben		i
23	TXT	Txn	Text	Start der Textausgabe		i
31	EWP	0	End Word Processor	Ende der Arbeit im Wortprozessor		

i = indexierbar