

C 17C



**Prozeßleitsystem audatec
Technische Dokumentation
Applikationsrechner**

Anwendervorschrift C



Inhaltsverzeichnis

0.	Allgemeines	6
1.	Systembeschreibung	10
1.1.	Komponenten des Betriebssystems	10
1.2.	Anlauforganisation	10
1.3.	Programmabarbeitung	13
1.4.	Stackverwaltung	14
1.5.	Registerrettung	15
1.6.	Verarbeitungsprogramme	15
1.7.	Programmablaufsteuerung	17
1.8.	Ebenenverwaltung	17
1.9.	Zugriff auf periphere Gerate (Drucker, FD)	18
1.9.1.	Kommando zur E/A- Steuerung - EAST	19
1.9.2.	Kommando Warten auf E/A - WEA	25
1.9.3.	Vorrangbelegung	26
1.10.	Datenorganisation/Datentypen	27
1.11.	DUE-Organisation	30
1.11.1.	Prozessabbild	30
1.11.2.	Direkter Zugriff ueber Kommando	32
1.11.2.1.	Grundregime der Kopplung	33
1.11.2.2.	Datenbereitstellungsregime	35
1.11.2.3.	Schreibbedingungen bei KOM-Zugriff	40
1.11.3.	Beispiele zur DUE	41
1.11.4.	Aussagen zur DUE-Leistungsfahigkeit	46
1.12.	RST-Routinen	48
1.13.	Kommandos, Hilfsprogramme, Unterprogramme	48
1.13.1.	Unterprogramme fuer BS/Tastatur	48
1.13.1.1.	BS-Unterprogramme	49
1.13.1.2.	Tastatur UP	55
1.13.2.	Arithmetik- und Konvertierungsprogramme	61
1.13.2.1.	GK-Format	61
1.13.2.2.	Konvertierungsprogramm fuer weitere Formate	61
1.13.2.3.	Konvertierung Integer 2, 4 Byte <=> GK	62
1.13.3.	Uhrzeit/Datum	63
1.13.4.	Kommandos/Systemrufe	63
1.13.5.	Umrechnung zwischen GK und PG	68
1.13.6.	Unterprogramme zur Drucker- und Floppy-Disk-Ansteuerung	69
1.13.6.1.	Ausgabe zum Seriendrucker	69
1.13.6.2.	Floppy-Disk-Programme	70
1.14.	Einbindung von Interrupt-Service-Routinen durch den Anwender	77
1.15.	Zusammenstellung von Systemsteuerzellen/Adressen	79
1.16.	Kenngroessen des Betriebssystems	82
2.	Ueberwachung und Fehlermeldung	83
2.1.	Allgemeines Anzeigekonzept	83
2.2.	Fehleranzeige bei Rechnerhalt	83
2.3.	Beschreibung der Anlaufmerker NUA/PUA/WUED1	86
2.4.	Systemfehlermeldungen	92
2.4.1.	Ueberblick der Systemfehlermeldungen	92
2.4.2.	Systemfehlermeldungen bei laufendem System - Darstellung auf Bildschirm	92
2.4.3.	Erlaeuterungen der Systemfehlermeldungen	95
2.4.4.	Nutzung des internen Fehlersystems durch den Anwender	106

3.	Standardfunktionen des Systems	113
3.1.	Funktionen - Ueberblick	113
3.2.	Grundmenue	113
3.3.	Systemstrukturierung	114
3.3.1.	Funktion 0 bis 2	115
3.3.2.	Systemdaten	115
3.3.3.	Anmeldeliste	116
3.3.4.	Steuerliste VAP	116
3.3.5.	Strukturierung des Prozessabbildes (PRAB)	118
3.3.5.1.	Neustart	118
3.3.5.2.	Verwaltung	118
3.3.5.3.	Steuerfelder	120
3.3.6.	Strukturierung des AR-Abbildes	121
3.3.7.	Strukturierung weiterer Steuerlisten	121
3.4.	Systembedienung	122
3.4.1.	Zugriff auf PRAB und AR-Abbild	122
3.4.1.1.	Anzeige PRAB/AR-Abbild	122
3.4.1.2.	Zugriff auf Prozessabbild und AR-Abbild	124
3.4.2.	Dateneingabe in HSE	124
3.4.3.	Anzeige der Eingabe in Hexa	127
3.4.3.1.	Aufbau der Steuerlisten zur Anzeige	127
3.4.3.2.	Anzeige der Daten	132
3.4.3.3.	Eingabe von Daten	132
3.4.3.4.	Kommando	132
3.4.4.	Fehlerzustand, Fehlerpuffer und Status des AR	132
3.4.5.	Bedienung Seriendrucker	133
3.4.5.1.	Systemfunktion	133
3.4.5.2.	Hardcopy	133
3.4.6.	Bedienung der Floppy-Disk-Einheit	135
3.4.6.1.	Hilfsmenue	H
3.4.6.2.	Bibliothek auflisten	B
3.4.6.3.	Diskette initialisieren	I
3.4.6.4.	Systemdiskette duplizieren	Y
3.4.6.5.	Alle Dateien duplizieren	D
3.4.6.6.	Fehler-/Statusabfrage	X
3.4.6.7.	Laufwerk Freigabe	F
3.4.6.8.	Datei kopieren	K
3.4.6.9.	Datei streichen	C
3.4.6.10.	Datei umbenennen	R
3.4.6.11.	Datei lesen	L
3.4.6.12.	Datei schreiben	S
3.4.6.13.	Datei weiterschreiben	W
3.4.6.14.	Ausgewählte Fehler-/Statusmeldungen	140
3.4.7.	EPROM-Programmierung und Pruefsummenbildung	141
3.4.8.	Uhrzeit/Datum	143
3.5.	Monitor	144
4.	Erstellung und Einbindung von Anwenderprogrammen	150
4.1.	Programmerstellung	150
4.2.	Einordnung in die zeitliche Abarbeitung	150
4.3.	Systemstrukturierung	150
4.4.	Programstest	150
4.5.	Programmeinbindung	150
4.6.	Nutzung des Grundbereiches auf EPROM durch den Anwender	151
4.7.	Zugriff zum PRAB und AR-Abbild	151
4.8.	UP mit Ebenenwechsel (parallel zur Programmebene)	151

4.9.	Einbindung von ISR durch den Anwender	151
4.10.	Aufbau der Pruefsumentabelle auf EPROM 00	152
4.11.	Aenderung des Strukturier-EPROM's	153
4.12.	Zuweisung der Funktionstasten durch den Anwender	153
4.13.	Hinweise fuer die Gestaltung von Anlauf und Verwaltung	153
5.	Gleitkommapprogrammpaket	161
5.1.	Allgemeine Hinweise	161
5.1.1.	Bestandteile des Gleitkommapaketes	161
5.1.2.	Zahlendarstellung	162
5.1.3.	Parameteruebergabe	164
5.1.4.	Fehlermeldungen	164
5.1.5.	Adressen Arithmetik	165
5.1.6.	Hinweise zu Aufwandsangaben	165
5.2.	Programmbeschreibung Arithmetikgrundmodul	166
5.2.1.	Gleitkommaaddition	ADD 166
5.2.2.	Gleitkommasubtraktion	SUB 166
5.2.3.	Gleitkomma multiplikation	MULT 167
5.2.4.	Gleitkomma division	DIV 167
5.2.5.	Quadratfunktion	SQU 168
5.2.6.	Kehrwert	RVS 168
5.2.7.	Betragsfunktion	ABS 169
5.2.8.	Konvertierung 16-Bit-Zweierkomplementdarstellung im Gleitkommaformat	FLOAT 169
5.2.9.	Konvertierung Gleitkomma darstellung in 16-Bit-Zweierkomplementdarstellung	TRUNC/ROUND 170
5.3.	Programmbeschreibung Arithmetikerweiterungsmodul	171
5.3.1.	Polynomrechnung	POLY 171
5.3.2.	Quadratwurzel	SQRT 172
5.3.3.	Exponentialfunktion	EXP 172
5.3.4.	Natuerlicher Logarithmus	LN 173
5.3.5.	Cosinusfunktion	COS 173
5.3.6.	Sinusfunktion	SIN 173
5.3.7.	Tangensfunktion	TAN 174
5.3.8.	Arcustangensfunktion	ARCTAN 174
5.3.9.	Dekadischer Logarithmus	LG 174
5.3.10.	Potenzfunktion	POT 175
5.3.11.	Gleitkommavergleich	COMP 175
5.4.	Programmbeschreibung E/A-Konvertierungsmodul	176
5.4.1.	Eingabekontvertierung	GET 176
5.4.2.	Ausgabekontvertierung	PUT 177
6.	Editor zum BASIC-Compiler	183
6.1.	Einfuehrung	183
6.2.	Textformat	183
6.3.	Bedienung des Editors	183
6.3.1.	Start des Editors	183
6.3.2.	Arbeiten im Menue des Editors	184
6.3.3.	MENU	184
6.3.4.	ADDR	184
6.3.5.	NEW	185
6.3.6.	QUIT	185
6.3.7.	KEY	185
6.3.8.	KEYLIST	185
6.4.	Compileraufruf COMP	186

6.5.	Quelltextbearbeitung EDIT, TOP, BOTTOM	186
6.5.1.	Quelltextdarstellung	187
6.5.2.	Cursorbewegungen	187
6.5.3.	Cursorpositionierung	187
6.5.4.	Einfuegen/Streichen	187
6.5.5.	Copymode	187
6.5.6.	CAPLOOK	188
6.5.7.	Funktionstasten	188
6.5.8.	Verlassen des Edit-Modus	188
6.5.9.	RESET (CTRL)-Modus	188
6.6.	Zugriff auf periphere Gerate	191
6.6.1.	LIST	191
6.6.2.	SAVE	191
6.6.3.	LOAD	191
6.6.4.	FD	191
6.7.	Uebersicht Editor-Fehlermeldungen	192
7.	BASIC-Compiler, Sprachbeschreibung	193
7.1.	Einleitung	193
7.2.	Aufbau einer Programmzeile	194
7.3.	Namen	194
7.4.	Variablen	194
7.5.	Konstanten	195
7.6.	Arithmetische Ausdruecke	196
7.7.	Anweisungen	199
7.8.	Inline-Assembler	208
7.8.1.	Allgemeines	208
7.8.2.	Aktivierung des Assemblers	208
7.8.3.	Sprachumfang	209
7.8.4.	Verfuegbarkeit der Prozessorregister	209
7.9.	Systemkommando-Compiler	210
7.9.1.	Allgemeine Bemerkungen	210
7.9.2.	Steuerung des Echtzeitbetriebssystems	210
7.9.3.	Steuerung der Systemuhr	211
7.9.4.	Steuerung des FS-Status	212
7.9.5.	Zugriff auf Prozess- und Applikationsrechner- abbild	213
7.9.6.	Sonstiges	214
7.9.7.	Fehlerbehandlung bei Systemrufen	215
7.10.	Fehlermeldungen	216
8.	Schematasystem des Applikationsrechners mit Eigenstrukturierung	217
8.1.	Erstellung der statischen Bilder	217
8.1.0.	Zweck des Bildsystems	217
8.1.1.	Speicheraufteilung	217
8.1.2.	Systematik der Speicherung	219
8.1.3.	Dateiverwaltung	219
8.1.4.	Bedienung	219
8.1.5.	Festlegungen zum Zeichengenerator	223
8.1.6.	Sprachbeschreibung fuer statische Bild- informationen	225
8.1.7.	Hinweise zur Strukturierung der statischen Bilder	226

8.2.	Erstellung der dynamischen Bildelemente	227
8.2.1.	Wirkungsweise und Funktion	227
8.2.2.	Speicherfestlegungen und Vereinbarungen	227
8.2.3.	Dialogfuehrung	232
8.2.4.	Bildmodule	235
8.2.4.0.	Allgemeine Festlegungen und Moduluebersicht	235
8.2.4.1.	Einblendung von Bildmodulen	BMOD 240
8.2.4.2.	Einblendung einer Textzeile	BITX 241
8.2.4.3.	Blinken von Bildschirmbereichen	BIBL 243
8.2.4.4.	Einblendung mehrerer Textzeilen	BYTX 244
8.2.4.5.	Auswertung Alarmcode mit Texteinblendung	TXTF 247
8.2.4.6.	Bildeinblendung in Abhaengigkeit vom Alarmcode	PELD 249
8.2.4.7.	Textausgabe von Verarbeitungsprogrammen	HIPU 251
8.2.4.8.	Trenddarstellung von analogen Groessen	KUTR 252
8.2.4.9.	Darstellung von waagerechten und senkrechten Balken analoger Groessen	BALK 253
8.2.4.10.	Zahlendarstellung von Analogsignalen	FLKO 254
8.2.4.11.	Module zur Darstellung des aktuellen Anzeigeanfangs und -bereichs bei Kurzzeitrend und Balkendarstellung	ANBI 256
8.2.4.12.	Zahlendarstellung von Integerwerten	INTE 257
8.2.4.13.	Zahlendarstellung von Prozessgroessen, Gleitkommazahlen und Integerzahlen	ZIFF 258
8.3.	Bedienhandlungen zur Aktivierung der Strukturierung auf dem AR	261
8.4.	Abarbeitung der erstellten Bilder unter dem Echtzeitbetriebsystem des AR	262
8.4.1.	Prinzipielle Einordnung in die ON-LINE-Programmabarbeitung	262
8.4.2.	Notwendige und moegliche Zusatzfunktionen fuer die ON-LINE-Programmabarbeitung (Anwenderunterstuetzungsprogrammssystem)	263
8.4.2.1.	Grundlagen	263
8.4.2.2.	Handhabung des Anwenderunterstuetzungsprogrammssystems	267
8.4.2.3.	Kurzbeschreibung der einzelnen Programmbestandteile	268
8.5.	Anwendungsbeispiel	270
9.	Aenderung oder Erstellung des Zeichengenerators	275
9.1.	Uebersicht	275
9.2.	Startbedingungen	275
9.3.	Bearbeitungsablauf	275
9.4.	Ausgabe des Zeichengenerators	276
9.4.1.	Ausgabe auf Drucker	276
9.4.2.	Ausgabe auf FD/MB	276
9.4.3.	Programmierung der EPROM's	276
9.5.	Zusammenfassung der Bedienkommandos	277

6. Editor zum Basic-Compiler

6.1. Einfuehrung

Der Basic-Compiler verfuegt ueber einen eigenen Editor zur Erstellung und Bearbeitung der BASIC-Quelltexte. Dieser Editor uebernimmt die gesamte Steuerung des Compilers sowie die verschiedenen Funktionen zur Bearbeitung der Quelle und die Bereitstellung wichtiger Informationen fuer den Programmierer.

6.2. Textformat

Der Text stellt im Speicher eine Folge von ASCII-Zeichen dar, wobei das Ende einer jeweiligen Textzeile mit dem Steuerzeichen OAH dargestellt wird. Tabulatoren werden nicht mit verarbeitet.

Quellen, die auf anderen Rechnertypen bzw. auf anderen Textprogrammen erstellt wurden, koennen nur unter der Bedingung, dass sie ebenfalls OAH als Zeilenende enthalten, auf diesem Editor bearbeitet werden.

Das Quellenende wird mit OOH gekennzeichnet.

Auf dem Bildschirm koennen theoretisch beliebig lange Textzeilen bearbeitet werden. Praktisch ist jedoch durch die nachlassende Uebersichtlichkeit und die nachlassende Verarbeitungsgeschwindigkeit eine Grenze fuer den Bearbeiter gesetzt.

6.3. Bedienung des Editors

6.3.1. Start des Editors

Nach dem Einladen ist der Editor mit der Funktion 1 - STRUKTURIERUNG, 5 - STEUERLISTE VAP in die VAP-Steuerliste mit folgenden Adressen einzutragen:

ED	EP	ADR	ST-BYTE	
3	3	1000	0000	; Editor-Neustart (EDIT)
3	3	1010	0000	; Editor-Restart (REEDIT)

Dann kann der Editor ueber J VAP-Nr. gestartet werden.

Beim Neustart ueber EDIT werden wesentliche Systemvariablen des Editors und des Compilers auf sinnvolle Anfangswerte gesetzt.

Dannach meldet sich der Editor mit dem Menue, in dem die verschiedenen Funktionen des Editors aufgelistet werden.

Beim Start ueber REEDIT erfolgt lediglich eine Initialisierung der Systemvariablen, die sich in der Farb- und SW-Version unterscheiden.

Achtung ! Nach dem Einlesen muss der Editor zur ordnungsgemaessenen Funktion unbedingt einmal ueber EDIT gestartet werden.

6.3.2. Arbeiten im Menue des Editors

Die aufgelisteten Kommandos im Menue koennen wahlweise auf verschiedene Arten aufgerufen werden:

- Es wird mit dem Cursor mittels der Steuertasten auf das jeweilige Kommando gefahren und mit <ENTER> die Ausfuhrung verlangt.
- Es wird der vollstaendige Kommandoname oder nur ein Teil von ihm ab Kursorposition auf den Bildschirm geschrieben. Die Eingabe wird mit <ENTER> abgeschlossen.

Achtung ! Stehen zwei Kommandos hintereinander auf dem Bildschirm, die einen sehnlichen Wortanfang haben, so muessen, um das zweite Kommando aufzurufen, mindestens so viele Buchstaben eingegeben werden, dass der Unterschied zwischen den Kommandos sichtbar wird.

Bsp.: >KEY
>KEYLIST
>-

Zum Aufruf von Keylist ist mindestens die Eingabe >KEYL <ENTER> notwendig.

Ausser den Kommandonamen koennen dem Programm gleichzeitig noch bis zu fuef Argumente als maximal vierstellige Hexazahlen uebergeben werden. Diese sind vom Kommandonamen sowie untereinander durch ein Leerzeichen zu trennen. Wie diese Argumente konkret verarbeitet werden, wird unter den einzelnen Kommandos beschrieben. Ueberfluessige Argumente werden ignoriert.

6.3.3. MENU

Das Kommando >MENU arbeitet ohne Argumente. Es bewirkt ein Auflisten der Kommandos des Editors.

6.3.4. ADDR

Nach Aufruf der Funktion wird die Speicheraufteilung fuer die Arbeit mit dem Basis-System angezeigt.

```
SOURCE E3: {Anf.-adr.} : {Akt. Ende}
CODE     E2: {Anf.-adr.} : {Akt. Ende} : {Endadr.}
DATA    : {Anf.-adr.} : {Akt. Ende} : {Endadr.}
IDT     E3: {Anf.-adr.} : {Akt. Ende} : {Endadr.}
```

Der Compiler ueberwacht die Laenge der Bereiche CODE und IDT. Der Bereich DATA wird nicht auf Ueberschreitung der Endadresse ueberprueft. Auf der ersten Bildschirmzeile wird waehrend der Arbeit mit dem Editor staendig das aktuelle Quellenende angezeigt.

Achtung ! Die Identifikatortabelle wird im Speicher von hinten beginnend aufgebaut.

Hinter dem E werden die Ebenen angezeigt, auf denen die Bereiche liegen. Der DATA-Bereich wird erst beim Lauf des MC-Programms belegt und verwendet dann die in die VAP-Steuerliste eingetragenen Werte. Der erzeugte Maschinencode kann ebenfalls auf andere Ebenen geladen werden und ist dort lauffähig. Die angezeigten Adressen koennen ueberschrieben werden. Dazu ist der Cursor mit den Tasten <-- , --> , ↑ , ↓ auf die entsprechende Position zu bewegen. Nach Quittung mit <ENTER> werden die Adressen uebernommen. Die Endadresse fuer die Quelle wird dabei selbstaendig aktualisiert, die anderen Adressen werden erst beim naechsten Compilerlauf aktualisiert. Mit einer beliebigen Taste kann die Funktion wieder verlassen werden.

6.3.5. NEW

>NEW arbeitet ohne Argumente. Nach dem Aufruf erfolgt eine nochmalige Abfrage: 'DELETE ALL (Y/N)?'. Bei Beantwortung dieser Abfrage mit 'Y' wird die gegenwaertig eingestellte Quelle fuer die weitere Bearbeitung geloescht.

6.3.6. QUIT

>QUIT veranlasst den Editor zur Rueckkehr zum Grundmenue des Applikationsrechners. Dieses kann im Editor-Menue auch durch Betaetigen der Taste 'OFF' erreicht werden.

6.3.7. KEY

Mit dem Kommando >KEY ist es moeglich, den Funktionstasten PF1 bis PF8 eine Zeichenkette zuzuordnen. Bei der eigentlichen Textbearbeitung besteht dann die Moeglichkeit, diese Zeichenkette durch den entsprechenden Tastendruck in den Quelltext einzufuegen. Das Argument hinter >KEY wird als Funktionstastenummer interpretiert. Nach dem Aufruf erfolgt der Ausdruck der Funktionstastenummer und der aktuellen Belegung. Der Cursor steht dabei hinter dem letzten Zeichen. Jetzt ist es moeglich, die Belegung der Taste beliebig zu aendern. Nach Quittierung mit <ENTER> wird die Zeichenkette vom Beginn bis zum Zeichen vor dem Cursor in den Puffer uebernommen. Mit 'OFF' kann die Eingabe ohne Veraenderung der Tastenbelegung verlassen werden. Waehrend der gesamten Eingabe wird eine Ueberschreitung des Pufferbereiches verhindert.

6.3.8. KEYLIST

>KEYLIST bewirkt ein Auflisten aller acht Funktionstastenbelegungen.

6.4. Compileraufruf

Der Compiler wird mit dem Kommando >COMP gestartet. Dabei wird ein evtl. angegebenes Argument als neuer Quellenanfang interpretiert. Weitere angegebene Argumente bewirken eine Neueinstellung des Maschinencodanfangs sowie der Maximaladresse des MC.

Nach Beendigung des Uebersetzungsvorganges kehrt der Compiler mit folgenden Varianten zum Editor zurueck:

- a) Der Uebersetzungslauf verlief ohne Fehlererkennung und im Programm kam der Befehl '>PROGRAM' nicht vor.

--> Der Cursor kehrt zurueck, die naechste Kommandoeingabe ist moeglich.

- b) Der Uebersetzungsvorgang verlief ohne Fehlererkennung und im Programm wurde(n) der(die) Befehl(e) '>PROGRAM' gefunden.

--> Es wird getestet, ob die angeforderte(n) Adresse(n) bereits in der VAP-Steuerliste eingetragen ist(sind). Wenn das nicht der Fall ist, erfolgt eine Abfrage, ob eine Eintragung erfolgen soll. Anschliessend ist die naechste Kommandoeingabe moeglich.

- c) Beim Uebersetzungslauf wurde ein Fehler erkannt.

--> Es erfolgt eine Fehlermeldung entsprechend der Fehlercodes des Compilers.

Anschliessend erfolgt eine Abfrage, ob ein Sprung in den Edit-Modus erfolgen soll in der Form:

'SHOW ERROR (Y/N)?'.

Bei Beantwortung dieser Frage mit 'Y' steht der Cursor auf der Stelle im Quelltext, an der der Fehler erkannt wurde.

Um Fehlern beim Programmieren rechtzeitig vorzubeugen, sollte auch darauf geachtet werden, dass die in der ersten Bildschirmzeile angezeigte Editor- und Compilerversion mit den Versionen der jeweiligen Beschreibung uebereinstimmt.

Achtung ! Compilerversionen mit Stand vor September 1988 koennen mit EDBAS VERSION 1.1. nicht gestartet werden.

6.5. Quelltextbearbeitung

Mit den Kommandos >TOP, >BOTOM, >EDIT und >*(Marke) kann der Edit-Modus aufgerufen werden.

Alle vier Kommandos arbeiten ohne die Angabe von Argumenten.

Mit >TOP wird der Cursor auf den Beginn der eingestellten Quelle gesetzt, mit >BOTOM auf das Ende.

Mit >EDIT wird der Cursor auf die Stelle innerhalb der Quelle

gesetzt, auf der er zuletzt gestanden hat.
Mit >*(Marke) wird der Cursor hinter der angegebenen Marke positioniert.

6.5.1. Quelltextdarstellung

Es wird jeweils ein bestimmter Abschnitt der Quelle auf dem Bildschirm eingeblendet. Dabei kann eine Programmzeile auch laenger als eine Bildschirmzeile sein. Das Zeilenende (Code OAH in der Quelle) wird auf dem Bildschirm durch ein '<' in der letzten Bildschirmspalte angezeigt.

6.5.2. Cursorbewegungen

Der Cursor kann mit den Steuertasten in alle vier Richtungen bewegt werden, jedoch nach oben maximal bis zum Quellenanfang und nach unten maximal bis zum Quellenende.

6.5.3. Cursorpositionierung

Mit der Taste '<-->' wird der Bildschirm um ca. die Haelfte nach oben gerollt, mit '|<-->|' nach unten.
Mit <ENTER> gelangt der Cursor an den Anfang der naechsten Zeile. Gleichzeitig wird die aktuelle Zeile mit OAH abgeschlossen.
Mit '|<-->' wird der Cursor auf den Anfang der aktuellen Zeile positioniert.
Die Taste '<-->|' bewirkt einen Cursorsprung auf die naechste Tabulatormarke. Diese Marken sind auf den eingeblendeten "Lineal" auf der dritten Bildschirmzeile durch '|' gekennzeichnet.

6.5.4. Einfuegen / Streichen

Mit der Taste INS MODE kann ein INSERT-Modus eingeschaltet werden. Dieser wird durch eine Ausschrift in der zweiten Zeile gekennzeichnet. Im eingeschalteten Zustand wird jedes Zeichen, das geschrieben wird, auf der Cursorposition in die Zeile eingefuegt.
Durch Betaetigen der Taste DEL wird das Zeichen auf der Cursorposition geloescht und anschliessend die Zeile von rechts verdrichtet.
Zeilen einfuegen kann man mit INS LINE. Ein Zeilenloeschen ist mit DEL LINE moeglich.

6.5.5. Copymode

Mit der Taste DUP wird ein Copy-Modus eingeschaltet. Zunaechst liegt dabei unter dem Normalcursor ein "Copycursor". Dieser kann mit den Cursorstasten auf dem Bildschirm bewegt werden, waehrend der normale Cursor nur noch mit den Tasten '2', '4', '6' und '8' bewegt werden kann.
Mit der Leertaste oder mit der Taste '5' wird ein Zeichen vom Copycursor auf den Normalcursor kopiert. Damit wird es unter anderem moeglich, Linien auch mit nicht repetierenden Tasten

zu ziehen, indem beide Cursor direkt hintereinander gestellt werden.

Der Copy-Modus wird mit <ENTER> wieder verlassen.

6.5.6. CAPSLOCK

Mit 'REC' wird der Editor in einen schreibmaschinenähnlichen Modus versetzt, d.h., Gross- und Kleinbuchstaben werden gegeneinander vertauscht, so dass in der normalen Tastenbelegung die Kleinbuchstaben und in der Shift-Belegung die Grossbuchstaben erreichbar sind. Zahlen und Sonderzeichen bleiben von diesem Modus unangetastet. CAPSLOCK wird durch erneuten Druck auf 'REC' wieder verlassen.

Achtung ! Da von Compiler keine Kleinbuchstaben in Befehlen akzeptiert werden, eignet sich dieser Modus nicht zur Erstellung von BASIC - Programmquellen.

6.5.7. Funktionstasten

Zur Erleichterung der Programmierung ist es möglich, den Funktionstasten PF1 bis PF8 Zeichenketten zuzuweisen (siehe Punkt 3.10.). Während des Schreibvorganges wird bei Auslösung einer dieser Tasten die zugehörige Zeichenkette ab Cursorposition ausgeschrieben. Der evtl. eingeschaltete INSERT-Modus wird dabei berücksichtigt.

6.5.8. Verlassen des Edit-Modus

Mit 'OFF' kann die direkte Bearbeitung des Quelltextes abgeschlossen werden. Erst dann ist die Quelle vollständig aktualisiert.

Achtung ! Ein Verlassen des Edit-Modus mit PF12 führt dazu, dass die Quelle nicht vollständig aktualisiert bzw. sogar zerstört wird.

6.5.9. RESET (CTRL) -Modus

Mit der Taste 'RESET' wird der CTRL-Modus (Control) eingeschaltet. Danach erfolgt in der zweiten Bildschirmzeile die Ausschrift 'CTRL:'. Es erfolgt eine erneute Tastaturabfrage. Mit dieser zweiten Eingabe wird das entsprechende Serviceprogramm gesucht und - wenn vorhanden - gestartet.

6.5.9.1. 'RESET' + Taste '--->' (TAB)

Damit ist eine Einstellung der Tabulatormarken moeglich. Der Cursor steht nach Aufruf auf dem "Lineal" in der dritten Bildschirmzeile. Der Cursor kann mit '<---' und '--->' in der Zeile bewegt werden. Mit '!' koennen Tabulatormarken gesetzt werden, mit '-' werden diese geloescht. Ein Loeschen aller Tabulatoren ist mit 'DEL' moeglich. Die Eingabe wird mit <ENTER> abgeschlossen. Danach werden maximal sieben Tabulatoren von links beginnend uebernommen.

6.5.9.2. 'RESET' + 'T' (TOP)

Der Cursor wird wie beim Kommando TOP auf den Beginn der Quelle gestellt.

6.5.9.3. 'RESET' + 'B' (BOT)

Der Cursor wird wie beim Kommando BOTTOM auf das Ende der Quelle gestellt.

6.5.9.4. 'RESET'+ 'P' (PUTLINE)

Die Bildschirmzeile, auf der der Cursor steht, wird in einen Puffer auf der letzten Bildschirmzeile gebracht.

6.5.9.5. 'RESET' + 'G' (GETLINE)

Der Pufferinhalt auf der letzten Bildschirmzeile wird an die Cursorstelle gebracht. Bei eingeschaltetem INSERT-Modus wird zuvor eine Leerzeile erzeugt.

6.5.9.6. 'RESET' + 'F' (FIND)

Diese Funktion ist in Verbindung mit RESET + 'D' bzw. 'U' zum Suchen von Zeichenketten zu verwenden. Der Kursor wird auf den Anfang des Puffers gesetzt. Danach kann die Eingabe einer beliebigen Zeichenkette erfolgen.

6.5.9.7. 'RESET' + 'D' (DOWN)

Die mit 'RESET' + 'F' gepufferte Zeichenkette wird in Richtung Quellenende gesucht. Bei erfolglosen Suchen steht der Kursor am Quellenende.

6.5.9.8. RESET + 'U' (UP)

Die mit 'RESET' + 'F' gepufferte Zeichenkette wird in Richtung Quellenanfang gesucht. Bei erfolglosem Suchen steht der Kursor am Quellenanfang.

6.5.9.9. RESET + 'A'

Es wird eine Anfangsmarkierung fuer den Blocktransport gesetzt. Es ist nur eine Anfangsmarkierung setzbar.

6.5.9.10. RESET + 'E'

Es wird eine Endmarkierung fuer den Blocktransport gesetzt. Es ist nur eine Endmarkierung setzbar.

6.5.9.11. RESET + 'C'

Es wird der mit RESET + 'A' und RESET + 'E' gekennzeichnete Block an die Cursorposition bewegt. Der Block wird eingefuegt. Die Blockmarkierungen werden mit transportiert. Es erfolgt kein Transport, wenn eine Markierung fehlt oder wenn die Endmarkierung kleiner als die Anfangsmarkierung ist.

6.5.10. Uebersicht ueber die Tastenbelegungen im Edit-Modus

Taste	Code	Belegung
<--	07H	Cursor links
-->	06H	Cursor rechts
↑	04H	Cursor hoch
↓	05H	Cursor runter
ENTER	0DH	Cursor Anfang naechste Zeile
<--	0BH	Cursor Anfang der Zeile
-->	04H	Tabulatorsprung
DEL	1BH	Zeichen streichen
INS MODE	02H	Ein- / Ausschalten INSERT-Mode
DEL LINE	13H	Zeile streichen
INS LINE	03H	Einfuegen Zeile
<--'	0AH	Blaettern Seite vorwaerts
<--	0FH	Blaettern Seite rueckwaerts
REC	11H	Ein- / Ausschalten CAPSLOCK
DUP	1CH	Einschalten Copy-Modus
RESET	1FH	Aufruf CRTL-Modus
PF1-PF8	91H-98H	frei belegbare Funktions-tasten

6.6. Zugriff auf periphere Gerate

Alle Geratefehlermeldungen sind mit denen in 1.9.1. genannten identisch.

6.6.1. LIST

Es wird die aktuelle Quelle ausgedruckt. Hinter LIST ist als Argument die Druckernummer 1 oder 2 anzugeben.

6.6.2. SAVE

Diese Funktion dient zum Sichern von Programmen oder Quellen auf Diskette. Als Argument ist die Laufwerksnummer 0 oder 1 anzugeben.

Mit SAVES wird die aktuelle Quelle gesichert.

Wenn das Gerat verfuegbar ist wird der Dateiname angefordert (in SCPA-Format XXXXXXXX.YYY angeben).

Nach Quittung mit <ENTER> wird die Quelle gesichert. Ist schon eine Datei gleichen Namens vorhanden, wird diese zur BAK-Datei umbenannt.

Mit SAVENC wird der aktuelle Maschinencode gesichert. Die Namenseingabe erfolgt wie bei SAVES. Es wird eine Datei mit Kopfblock geschrieben., d.h. der Datei ist ein Block vorangestellt, der Adresse, Ebene und Laenge enthaelt (vgl. 3.4.6.).

6.6.3. LOAD

Die Funktion dient zum Laden von Programmen oder Quellen von der Diskette.

Als Argument ist die Laufwerksnummer 0 oder 1 anzugeben.

Mit LOADS wird auf die aktuell eingestellte Quelladresse (mit ADDR versenderbar) die Quelle geladen. Dazu wird der Dateiname angefordert.

Achtung ! Das Quellenende wird nicht ueberwacht.

Mit LOADNC wird eine Programmdatei geladen, wenn diese einen Kopfblock hat. sie wird auf die Adressen/Ebenen geladen, die im Kopfblock angegeben sind. Der Dateiname wird angefordert.

6.6.4. FD

Fuer alle sonstigen Floppy-Disk-Funktionen (Bibliothek, Streichen, Umbenennen, Initialisieren, usw.) kann die aus der SYSTEMBEDIENUNG bekannte Floppy-Disk Bedienung genutzt werden (vgl. 3.4.6.).

Die Bildschirmanzeige ist ueber FD direkt aufrufbar.

6.7. Uebersicht Editor - Fehlermeldungen

Bei Notwendigkeit von Speicherplatzeinsparungen ist es moeglich, den Editor auch ohne eine Tabelle zur Fehlermitteilung arbeiten zu lassen. In diesem Fall erfolgt bei einem auftretenden Fehler nur eine Ausschrift ERROR und anschliessend der entsprechende Fehlercode als Hexazahl:

Fehlercode:	Beschreibung:
30H	Kommando zu lang
31H	zu viele Stellen im Argument
32H	zu viele Argumente
33H	Kommando existiert nicht
34H	fehlerhaftes Argument
40H	Compiler defekt oder nicht geladen

Geratfehler:

80H	Gerate-Nr. fehlt oder falsch
C1H	Gerat nicht strukturiert
A0H	Systemprogramm der PDE laeuft nicht
20H	Sektor nicht gefunden
	Laufwerk nicht zugewiesen
	Datei existiert schon
	Datei existiert nicht
	Diskette voll
	Dateiverzeichnis voll
08H	Datenuebertragung zur PDE unterbrochen
92H	eigenes Kommando laeuft noch

Eine vollstaendige Liste der Geratfehler ist in 1.9.1. enthalten.

Die Fehlermitteilungen des Compilers sind in der Beschreibung des Compilers gesondert beschrieben (siehe Punkt 7.10.).

7. Basic-Compiler

7.1. Einleitung

Der BASIC-Compiler BC-88/9 ist ein Programm zur Erstellung von Maschinencodeprogrammen in der Sprache des Prozessors U 880 (Z 80). Der hierbei realisierte Sprachumfang der Quellsprache besitzt einige Besonderheiten gegenüber den gebräuchlichen Standardvarianten von BASIC. Es wird eine Teilmenge von Standard-BASIC-Kommandos realisiert, die durch einen integrierten Assembler ergaenzt wird, welcher ein maschinennaehes Programmieren ermoeeglicht. Zusätzlich sind die wichtigsten Systemrufe des Applikationsrechners in den Sprachumfang aufgenommen worden, was die Arbeit mit dem Echtzeitbetriebssystem stark erleichtert.

Der vom Compiler erstellte MC (Maschinencode) ist PRON-fachig, d.h. Programm- und Datenbereich werden getrennt aufgebaut, wobei der Programmereich keine Arbeitsbereiche beinhaltet. Weiterhin ist der erstellte MC wiedereintrittsfachig, was ermoeeglicht, ihn durch Echtzeitbetriebssysteme abzuarbeiten. Fuer die Ausfuehrung von Gleitpunktoperationen programmiert der Compiler CALL-Befehle zu einer im Applikationsrechner vorhandenen 4-Byte-Arithmetik.

Nachteilig ist die fehlende Noeglichkeit, Lademodule zu generieren. Der Compiler arbeitet ausschliesslich speicherorientiert, d.h. sowie Quelle als auch MC muessen im RAM stehen koennen. Hierbei ist zu beachten, dass die Quelle stets auf der Programmebene 3 stehen muss. Der MC wird auf der Programmebene 2 aufgebaut, kann aber auch nach einem Umladen auf anderen Programmebenen abgearbeitet werden. Der vom Compiler generierte Datenbereich ist an keine spezielle Ebene gebunden. Bei Testlaeuften ist jedoch vorzugsweise die Ebene 4 oder 1 zu benutzen, um beim Starten des erzeugten MC den Compiler bzw. den Editor nicht zu zerstoeeren.

Wichtige Einschränkungen im Sprachumfang sind:

- Es gibt keine Zeichenkettenverarbeitung mit Ausnahme von Zeichenkettenkonstanten in PRINT- und INPUT-Anweisungen und Bytefeldern, die als Zeichenketten verwendet werden koennen.

Bemerkung:

Im Folgenden werden zur Darstellung der Syntax der einzelnen Kommandos folgende, nicht zum Zeichenvorrat der Sprache gehoerenden Zeichen verwandt:

```

::=      als "ist definiert als"
{ }     schliessen ein syntaktisches Element ein
|       "oder"

```

7.2. Aufbau einer Programmzeile

Die Programmzeile kann mit einer Zeilennummer im Bereich von 0<=ZN<=65535 begonnen werden. Bei Verwendung eines geeigneten Editors kann diese jedoch auch entfallen. In einer Zeile koennen mehrere Anweisungen geschrieben werden, die dann durch Semikolen (;) zu trennen sind. Den Abschluss bildet (Newline), dargestellt durch den Code OAH. Jeder Anweisung kann eine Marke vorangestellt sein. Ihre Syntax ist

```
{Marke} ::= {Name};
```

7.3. Namen

Ein Name ist eine Folge von 1 bis 5 Buchstaben oder Ziffern, beginnend mit einem Buchstaben. Es sind nur Grossbuchstaben zulassig.

Einige Namen sind fuer den Compiler reserviert und duerfen nicht benutzt werden:

```
SYSH2, SYSD2, SYSCO, SYSNL, SYSTX, SYSIB  
SYSPB
```

Ihre Benutzung fuehrt zur Fehlermeldung durch den Compiler. Weitere Systemvariablen sind ERROR% und ERRCO%. Sie sind vom Typ INTEGER und dienen als Fehlerzellen fuer die Systemkommandos (Abschnitt 9). Sie koennen vom Nutzer abgefragt werden.

7.4. Variablen

7.4.1. Einfache INTEGER-Variablen

Ihre Syntax ist

```
{INTEGER-Variable} ::= {Name}%
```

Sie beinhalten 16-bit-Zahlen mit Vorzeichen. Die Darstellung negativer Werte erfolgt im Zweierkomplement. Der Wertebereich ist:

```
-32768 <= X% <= 32767
```

INTEGER-Variablen werden bei ihrem ersten Auftauchen im Programm automatisch vereinbart.

7.4.2. INTEGER-Felder

Es ist moeglich, mit der DIM-Anweisung (s. Punkt 7.2) ein- oder mehrdimensionale INTEGER-Felder zu vereinbaren. Die Syntax des Feldnamen entspricht der einer einfachen INTEGER-Variablen. Jedes Feldelement belegt 2 Byte im Speicher. Der Index (bzw. die Indizes bei mehrdimensionalen Feldern) wird in runde Klammern gesetzt. Bei der Abarbeitung des uebersetzten HC's erfolgt keine Kontrolle auf Feldgrenzenueberschreitungen.

7.4.3. REAL-Variablen

Eine REAL-Variablen wird bei ihrem ersten Auftreten im Programm vereinbart (Name ohne %). Sie benoetigt im Speicher 4 Byte (vgl. Beschreibung des Arithmetikpakets).

7.4.4. REAL-Felder

Sie werden durch die DIM-Anweisung vereinbart (Name ohne %). Der bzw. die Indizes stehen in runden Klammern. Jedes Feld-element belegt 4 Byte. Bei der Abarbeitung des MCs wird keine Kontrolle der Feldgrenzen ausgefuehrt.

7.4.5. BYTE-Felder

Mit Hilfe der DIM-Anweisung koennen ein- oder mehrdimensionale BYTE-Felder vereinbart werden. Der Name eines BYTE-Feldes besteht aus einem einfachen Variablennamen mit einem angefügten #-Zeichen. In BYTE-Feldern entspricht jedem Feld-element einem Byte mit einem Wertevorrat von 0 bis 255.

7.5. Konstanten

7.5.1. Ganze Dezimalkonstanten

Entspringt eine Konstante dem Wertevorrat -32768 <= X% <= 32767 und beinhaltet sie keinen Dezimalpunkt bzw. keinen Exponenten (E), so handelt es sich um INTEGER-Konstanten. Zur Kennzeichnung kann der Konstante ein % nachgestellt werden (nicht unbedingt erforderlich). Sie belegen 2 Byte im Speicher. Negative Zahlen werden dabei im Zweierkomplement dargestellt.

7.5.2. Hexadezimale Konstanten

Sie werden durch ein nachgestelltes H gekennzeichnet und sind von Typ INTEGER. Wenn die erste Ziffer groesser 9 ist, so ist eine 0 voranzustellen. (Beispiel: 0A000H). Zu beachten ist, dass bei versehentlichem Weglassen der voranzustellenden 0 anstelle der Hexadezimalkonstante eine REAL-Variablen vom Compiler uebersetzt wird und daher kein syntaktischer Fehler auftritt. Der Wertebereich ist 0H <= X% <= 0FFFFH. Hexadezimale Konstanten groesser gleich 8000H entsprechen negativen INTEGER-Werten (Zweierkomplement).

7.5.3. Dualkonstanten

Dualkonstanten wird ein B nachgestellt. Sie sind von Typ INTEGER. Der Wertebereich umfasst 0B <= X% <= 1111111111111111B. Dualkonstanten groesser gleich 1000000000000000B entsprechen negativen INTEGER-Werten.

7.5.4. ASCII-Zeichen

Sie sind vom Typ INTEGER und haben die Syntax

```
{ASCII-Konst.} ::= '{ASCII-Zeichen}' | '{ASCII-Zeichen}{ASCII-Zeichen}'
```

Die Zeichen werden im ASCII-Code dargestellt, wobei bei zwei Zeichen das zweite den hoeherwertigen Teil der Konstante ergibt.

7.5.5. Logische Konstanten

Es sind die logischen Konstanten .TRUE. und .FALSE. zulassig. Sie sind vom Typ INTEGER. Der Wert .TRUE. ist durch gesetztes Bit 0 im 2-Byte-Wort, der Wert .FALSE. durch nicht gesetztes Bit 0 gekennzeichnet. Die Konstanten selbst entsprechen den Konstanten 0 (.FALSE.) und 1 (.TRUE.). Treten logische Werte als Ergebnis von Vergleichsoperationen auf, so sind die Bits 1 bis 15 unbestimmt.

7.5.6. Gleitkommakonstanten

Sie haben die fuer REAL-Konstanten uebliche Syntax mit bis zu 6 gueltigen Stellen. Die Exponent-Schreibweise ist zulassig. Sie muessen entweder einen Dezimalpunkt enthalten, in Exponent-Schreibweise angegeben sein oder ausserhalb des Wertevorrats fuer INTEGER-Konstanten liegen, um als REAL-Konstanten zu gelten. Deshalb wird bei A:=5/4 der Variablen A nur der INTEGER-Wert 1 zugeordnet. Erst nach A:=5./4 ist der Wert von A=1,25.

7.6. Arithmetische Ausdruecke

Der arithmetische Ausdruck hat den fuer hoehere Programmiersprachen typischen Aufbau. Die moeglichen Operationen sind, beginnend mit der niedrigsten Prioritaet:

Prioritaet	Syntax	Operation
bitweise logische Operationen (auch fuer logische Werte!):		
1	.OR.	Bitweise OR-Verknuepfung
2	.AND.	Bitweise AND-Verknuepfung
	.XOR.	Bitweises ausschliessendes OR
Vergleichsoperationen (Ergebnis .TRUE. oder .FALSE.):		
3	<	kleiner
	>	groesser
	=	gleich
	<>	ungleich
	>=	groesser gleich
	<=	kleiner gleich
Vergleichsoperationen fuer INTEGER-Werte:		
	.ULT.	vorzeichenlos kleiner
	.UGT.	vorzeichenlos groesser
	.ULE.	vorzeichenlos kleiner gleich
	.UGE.	vorzeichenlos groesser gleich

Prioritaet	Syntax	Operation
arithmetische Operationen:		
4	+	Addition
	-	Subtraktion
	.ADD.	BCD-Addition
	.SUB.	BCD-Subtraktion
5	*	Multiplikation
	/	Division
6	**	Potenzfunktion

Uebergeordnet sind Standardfunktionen und Klammerausdruecke. Folgende Standardfunktionen sind vorhanden:

ABS	Betrag einer Zahl. Ein INTEGER-Argument wird als Zweierkomplementzahl interpretiert.
SGN	Signum-Funktion. Ein INTEGER-Argument wird als Zweierkomplementzahl interpretiert.
NOT	Bitweise Negation (negiert auch logische Werte)
LOW	Niederwertiger Teil eines INTEGER-Wertes
HIGH	Hoherwertiger Teil eines INTEGER-Wertes
EXC	Austausch von hoehwertigen und niederwertigem Byte eines INTEGER-Wertes
IN	Eingabe von Port, das durch das Argument adressiert wird.
PEEK	Lesen eines Bytes vom Speicher
DEEK	Lesen eines Wortes (2 Byte) vom Speicher
INCHAR	Tastaturabfrage mit Warten auf Tastendruck Argument 0%: Kein Echo Argument 1%: Eingabe mit Echo
INT	Konvertierung REAL=>INTEGER mit Abrunden
ADDR	Diese Funktion liefert die Adresse einer Marke, Variablen oder Zeilennummer (auch Fel(namen zulassig!)

Weiterhin sind die Funktionen SQR, EXP, LN, SIN, COS, TAN, ATN und LOG (Zehnerlogarithmus) vorhanden.

Achtung ! Fuer die Abarbeitung dieser Befehle im erzeugten Maschinenprogramm wird das Zusatzmodul der Applikationsrechner-Arithmetik benoetigt.

Die Funktionen PEEK, LOW, HIGH, INCHAR und IN liefern in hoehwertigen Byte des Ergebnisses Nullen.

Das Argument von Standardfunktionen muss stets in Klammern stehen.

Im Argument der ADDR-Funktion kann das typbestimmende Zeichen in Variablenamen (% bzw. *) weggelassen werden.

Vor der Abarbeitung eines arithmetischen Ausdrucks werden im Bedarfsfall automatische Konvertierungen der Argumente von Operationen und Funktionen von REAL- in INTEGER und umgekehrt vorgenommen. Die Konvertierung erfolgt fuer die einzelnen Operationen entsprechend der nachfolgenden Tabelle:

Operation	Operanden	Konvertierung der Operanden	Resultat
.OR.,.AND.,.XOR.	INTEGER gemischt REAL	nein REAL==>INTEGER REAL==>INTEGER	INTEGER INTEGER INTEGER
<,>=,<>,<=,>=	INTEGER gemischt REAL	nein INTEGER==>REAL nein	INTEGER INTEGER INTEGER (.TRUE./ .FALSE.) INTEGER
+,-,*,/,**	INTEGER gemischt REAL	nein INTEGER==>REAL nein	REAL REAL REAL
.ADD.,.SUB.	INTEGER gemischt REAL	nein REAL==>INTEGER REAL==>INTEGER	INTEGER INTEGER INTEGER
.ULT.,.UGT., .ULE.,.UGE.	INTEGER gemischt REAL	nein REAL==>INTEGER REAL==>INTEGER	INTEGER INTEGER INTEGER

Bei den Standardfunktionen erfolgt die Konvertierung nach folgenden Regeln:

Funktion	Argument	Konvertierung des Arguments	Resultat
ABS,SGN	INTEGER REAL	nein nein	INTEGER REAL
NOT,IN,PEEK,BEEK, LOW,HIGH,EXC	INTEGER REAL	nein REAL==>INTEGER	INTEGER INTEGER
INCHAR	INTEGER REAL	nein REAL==>INTEGER	INTEGER INTEGER
SQR,EXP,LN,SIN, COS,TAN,ATN	INTEGER REAL	INTEGER==>REAL nein	REAL REAL
ADDR	Name	nein	INTEGER

Bemerkung:

INTEGER-Terme von Typ

{INTEGER-Variable}+{INTEGER-Konstante}
 {INTEGER-Variable}-{INTEGER-Konstante}
 {INTEGER-Konstante}*{INTEGER-Variable}
 {INTEGER-Konstante}*{INTEGER-Variable}+{INTEGER-Konstante}

und

{INTEGER-Konstante}*{INTEGER-Variable}-{INTEGER-Konstante}

werden vom Compiler gesondert behandelt und ergeben schnellere und kuerzere Maschinencodes, als die entsprechenden Terme mit vertauschten Operanden.

7.7. Anweisungen

7.7.1. LET {Einfache Variable}|{Feldelement}:= {Arithm. Ausdr.}

Die LET-Anweisung realisiert eine Ergebniszweisung. Das Schluesselwort LET kann weggelassen werden, wenn es kein Schluesselwort gibt, das mit dem Anfang des Namens der Ergebnisvariablen uebereinstimmt.

Stimmt der Typ des Resultats des Ausdrucks nicht mit dem der Ergebnisvariablen ueberein, so wird automatisch konvertiert.

Beachte das zu verwendende Zuweisungszeichen := !

7.7.2. DIM {Feldvereinbarungsliste}

Es koennen statische, ein- oder mehrdimensionale REAL- und INTEGER- und BYTE-Felder vereinbart werden. Die oberen Feldgrenzen sind als positive, vorzeichenlose Dezimal- oder Hexadezimalkonstanten in der DIM-Anweisung anzugeben, die untere Grenze ist 0.

Beispiel: DIM A(10),X%(9),Y1%(19,19),Tn(3,3,3)

Das Feld A ist eindimensional und vom Typ REAL mit 11 Elementen (0 bis 10 !).

X% ist ein eindimensionales INTEGER-Feld mit 10 Elementen.

Y1% ist zweidimensional mit 400 Elementen (20 * 20) und Tn ist ein BYTE-Feld mit 64 Elementen (4 * 4 * 4).

Bei mehrdimensionalen Feldern werden die einzelnen Feldelemente im Speicher hintereinander angeordnet, wobei der am weitesten rechts stehende Index der "niederwertigste" ist. Das Feld Tn ist daher wie folgt aufgebaut:

```
Tn(0,0,0),Tn(0,0,1),Tn(0,0,2),Tn(0,0,3)
Tn(0,1,0),Tn(0,1,1),Tn(0,1,2),Tn(0,1,3)
Tn(0,2,0) usw.
```

Bei der Vereinbarung von mehrdimensionalen Feldern ist es guenstig, Feldgrenzen, die kleiner gleich 30 sind, nach hinten zu nehmen, wenn das der zu realisierende Algorithmus erlaubt. Dadurch kann die Rechengeschwindigkeit des MC erhoeht werden.

DIM A(100,20) ist guenstiger als A(20,100)

Zusaetzlich kann am Anfang der Feldvereinbarungsliste eine Marke stehen:

```
DIM {Name}:{Liste von Feldvereinbarungen}
```

Dabei gibt es zwei Varianten:

- Die Marke ist bereits vor der DIM-Anweisung definiert worden. In diesem Fall werden die Felder beginnend bei der durch die Marke repraesentierten Adresse aufgebaut.

Achtung ! Fehlermöglichkeit wenn die Marke eine Sprungmarke im Programm ist.

- b) Die Marke ist noch nicht definiert. Dabei wird der Marke die Adresse des ersten Feldes in der Liste zugeordnet.
Achtung ! Sprung zu einer solchen Marke fuehrt zum Absturz.

Eine zusaetzliche Erweiterung der DIM-Anweisung besteht darin, dass in der Liste der Felder auch einfache Variable auftauchen koennen (ohne Feldgrenzen).
 Damit ist es moeglich, diese definiert im Speicher zu platzieren.

Beispiele:

```
LABEL M1:1475H
DIM M1:A(4),X,Y,Z$(3,3),C$(10)
```

Das Feld A, die Variablen X und Y und die Felder Z\$ und C\$ werden ab 1475H in der angegebenen Reihenfolge angelegt.

```
DIM M2:B(9)
DIM M2:D,E,F,G,H,I,J,K(2)
```

Die einfachen Variablen D bis J und die Feldelemente K(0) bis K(2) belegen die entsprechenden Speicherplaetze des Feldes B(9).

7.7.3. GOTO {Zeilennummer}|{Marke}

Unbedingter Sprung zur Marke oder zur Zeile mit der angegebenen Nummer. Der Compiler erzeugt einen JMP-Befehl.

7.7.4. GOSUB {Zeilennummer}|{Marke}

Unbedingter Unterprogramm sprung zu einer Marke oder Zeilennummer. Der Compiler erzeugt einen CALL-Befehl.

7.7.5. RETURN

Ruecksprung vom Unterprogramm. Er wird als RET (0C9H) uebersetzt und kann zum Ruecksprung in Programme fuehren, die den vom Compiler generierten MC mit CALL aufgerufen haben.

7.7.6. IF (Arithm. Ausdruck) THEN {Marke}|{Zeilennummer}| DO

Es wird zur Zeile mit der angegebenen Nummer oder zur Marke gesprungen bzw. der nach dem DO folgende Programmabschnitt abgearbeitet, wenn das Resultat des arithmetischen Ausdrucks .TRUE. ist.

Der Programmabschnitt nach DO erstreckt sich bis DOEND. Diese Konstruktion ist ein DO-Block. Er kann mit ELSE in zwei Abschnitte unterteilt werden:

Der Abschnitt zwischen DO und ELSE wird abgearbeitet, wenn der Ausdruck .TRUE. ergibt, der Abschnitt von ELSE bis DOEND wenn .FALSE. heraus kam.

DO-Blocke koennen beliebig verschachtelt werden.

Beispiele:

```
IF A>B.AND.X%(1%)<=-1 THEN DO;PRINT "SCHOCK";DOEND
```

Es wird das Wort SCHOCK ausgegeben, wenn die Bedingung erfuellt ist.

```
W%:=A>B;IF W% THEN M1
```

Es wird zur Marke M1 gesprungen, wenn A>B ist. W% spielt hier die Rolle einer logischen Variablen.

7.7.7 FOR (Einf. Variable):={arithm. Ausdr.} TO {arithm. Ausdr.}

Die FOR-Schleife arbeitet nur mit positiver Schrittweite. Ohne Angabe wird 1 angenommen. Es kann STEP {arithm. Ausdr.} zur Festlegung der Schrittweite zugefuegt werden.

Die Schleife wird mit NEXT ohne Angabe der Laufvariablen abgeschlossen.

FOR-Schleifen koennen beliebig verschachtelt werden. Als Laufvariable koennen einfache INTEGER- und REAL-Variable verwendet werden.

Die Ausdruecke fuer die Abbruchbedingung (hinter TO) und die Schrittweite (hinter STEP) werden bei jedem Schleifen-durchlauf neu berechnet. Soll dies vermieden werden, so muessen Hilfsvariablen eingefuehrt werden:

```
FOR X:=0 TO EXP(Y)
```

bewirkt, dass die Abbruchbedingung jedes mal neu berechnet wird. Soll dies nicht verdergruendig erreicht werden, so ist

```
Y1:=EXP(Y);FOR X:=0 TO Y1
```

besser. Die Abarbeitung erfolgt so wesentlich schneller.

Schleifen, bei denen der Anfangswert der Laufvariablen groesser als der Endwert ist, werden einmal durchlaufen. Bei Angabe der Schrittweite laeuft die Schleife mit der Laufvariablen, die erstmals groesser oder gleich den Endwert ist das letzte mal.

7.7.8 LABEL (Name):(Ausdruck)

Der angegebene Name wird als Marke mit der dem Wert des Ausdruckles entsprechenden Adresse vereinbart.

Wobei gilt:

```
{Ausdruck}::={Konstante}||{vorher definierte Marke}+|-  
                  {Konstante}||{vorher definierte Marke}+|-...
```

Durch Komma getrennt, koennen mehrere Vereinbarungen vorgenommen werden. Die Anweisung arbeitet aehnlich der EQU-Anweisung bei Assemblern.

7.7.9. ON {arithm. Ausdr.} GOTO {Markenliste}

Es wird zu dem durch den Ausdruck adressierten Element der Markenliste gesprungen. Die erste Marke wird bei einem Ergebnis des Ausdrucks von 1 angesprungen. Ist das Ergebnis kleiner als 1 oder grösser als die Anzahl der Marken in der Liste, so wird zur nachfolgenden Anweisung uebergegangen.

7.7.10. PRINT {PRINT-Liste}

Die PRINT-Anweisung fuehrt zu einer Ausgabe zum Bildschirm oder/und zum Drucker. PRINT-Listenelement kann sein:

/	an Zeilenende: kein Newline
//	Ausfuehrung von Newline
	Bildschirmloeschen bzw.
	Seitenschaltung
*	Tabulator
String-Konstante in "	Textausgabe
arithm. Ausdr.	Ausgabe als REAL-Wert (bei P- und E-Format) oder Hexadezimalzahl (bei H-Format)

Listentrennzeichen ist nur Komma. Ein Komma am Ende der Liste bewirkt die Unterdrueckung des Newlines, das normalerweise erzeugt wird. PRINT ohne Parameter bewirkt kein Newline. Hinter arithm. Ausdruecken kann eine Formatangabe erfolgen. Syntax:

```
{Format} ::= :Fn.m - Festkommadarstellung
              n      Anzahl der Stellen insgesamt
              m      Anzahl der Stellen nach dem Punkt ( m<8 )
{Format} ::= :En      - Exponent-Darstellung
              n      Stellen nach dem Komma ( n<8 )
{Format} ::= :H      - Hexadezimale Ausgabe
              als vierstellige Hexadezimalzahl
```

Bei Formatfehlern erscheinen Doppelkreuze.

7.7.11. INPUT {INPUT-Liste}

Es wird eine REAL-Eingabe von der Tastatur realisiert. INPUT-Listenelemente sind

{Feldelement}	
{Einfache Variable}	
String-Konstante	bewirkt Textausgabe zum Display (bzw. zum Drucker)
/, // und *	wie bei PRINT

Werden INTEGER-Variablen angegeben, so wird der eingegebene Wert nach INTEGER konvertiert. Vor dem Abschluss der Eingabe mit ENTER kann der Wert mit der Taste "Cursor links" korrigiert werden. Die eingegebenen Werte werden auch bei angewählten Drucker (s. SWITCH) nur auf dem Display angezeigt.

7.7.12. OUTCHAR {arithm. Ausdr.}

Der errechnete Wert wird als ASCII-Zeichen interpretiert und zum Display und/oder zum Drucker ausgegeben.

7.7.13. INSTR {Bytefeld-Name},{Ausdruck}

Es wird die Eingabe einer Zeichenkette erwartet, die zunächst auf dem Bildwiederholungspeicher abgelegt (Betriebs-system-Ruf 0B445H) und nach dem Quittieren in das Bytefeld gelesen wird. Die Anzahl der Zeichen gibt der arithmetische Ausdruck an.

7.7.14. OUTSTR {Bytefeld-Name},{Ausdruck}

Die im Ausdruck angegebene Anzahl von Zeichen wird dem Bytefeld entnommen und zum angewählten Ausgabekanal (SWITCH) ausgegeben.

7.7.15. OUT {Ausdr.1},{Ausdr.2}

Ausgabe des niederwertigen Bytes des zweiten Ausdrucks zum Port, adressiert durch den ersten Ausdruck.

7.7.16. POKE {Ausdr.1},{Ausdr.2}

Das niederwertige Byte des Ergebnisses des zweiten Ausdrucks wird auf den vom ersten Ausdruck adressierten Speicherplatz geschrieben.

7.7.17. DOKE {Ausdr.1},{Ausdr.2}

Das Ergebnis des zweiten Ausdrucks wird auf die vom ersten Ausdruck adressierten zwei Speicherplaeetze in der Reihenfolge "niederwertiges Byte", "hoehwertiges Byte" geschrieben.

7.7.18. END

Logisches Ende des Programms. Bis hier hin erfolgt die Uebersetzung. END darf nicht weggelassen werden.

7.7.19. MOVE {Ausdr.},{Ausdr.},{Ausdr.}

Es wird ein "intelligenter" Umladebefehl programmiert. Der erste Ausdruck liefert die Ziel- der zweite die Quelladresse und der dritte Ausdruck die Anzahl der umzuladenden Bytes. Ist die Anzahl 0 so erfolgt kein Umladen.

7.7.20 SL {Ausdr.}({Liste einfacher INTEGER-Variablen})

Der Befehl bewirkt eine bitweise Linksverschiebung der sich aus den in der Liste angegebenen INTEGER-Variablen ergebenden Bitkette. Von rechts werden Nullen eingeschoben. Die Anzahl der Verschiebungen gibt der Ausdruck an. Ist dessen Ergebnis Null, so wird 256 mal geschoben.

7.7.21. SR {Ausdr.}({Liste einfacher INTEGER-Variablen})

Diese Anweisung arbeitet analog zu SL, nur dass nach rechts geschoben wird.

7.7.22. CLS

Bildschirmloeschen bzw. Seitenvorschub.

7.7.23. REM

Kommentareinfuegung. Das Ende wird durch Semikolon oder Newline bestimmt.

Zu Testzwecken ist es oft guenstig, eine ganze Textzeile voruebergehend zu einem Kommentar zu machen (auszublenden). Da REM nur bis zu einem evtl. vorhandenen Semikolon wirkt, existiert ausserdem das Ausrufungszeichen (!) als Definition fuer einen Kommentar bis zum naechsten Newline.

7.7.24. DATA {Feldname}({Dimension}),{Liste der Feldelemente}

Der DATA-Befehl generiert wie der DIM-Befehl ein Feld mit dem angegebenen Namen, Typ und den gewünschten Dimensionen. Dieses Feld befindet sich aber im MC. Ein "Hineinlaufen" des Programmes wird durch automatisches Ueberspringen verhindert. Die Elemente der Liste werden nacheinander in das Feld eingetragen.

Der Compiler fordert genau soviele Elemente, wie bei der Dimensionierung angegeben wurden.

Bsp: DATA F1(2,1),1,'A',3,"CDE"

Listenelemente koennen sein:

Bei Typ BYTE oder INTEGER:

ASCII-Zeichen	-	'A'
Zeichenketten	-	"ABCDE"
Einbytekonstanten	-	1
Zweibytekonstanten	-	1000
	-	1000H
	-	001101101110011B

Bei Bytefeldern wird nur das Low-Byte von Zweibytekonstanten benutzt.

Bei Typ REAL:

INTEGER- oder REAL-Konstanten (keine Hexa-Konstanten)

Wird eine Zeile mit Komma abgeschlossen, so kann das Konstantenfeld auf der naechsten Zeile fortgesetzt werden.

7.7.25. LOCATE {Ausdruck},{Ausdruck}

Der Cursor wird in die angegebene Spalte (erster Ausdruck) und die angegebene Zeile (zweiter Ausdruck) gesetzt.

7.7.26. COLOR

Das Kommando COLOR kann in drei Varianten genutzt werden:

- a) COLOR {Farbbyte}
Stellen von Vordergrund- und Hintergrundfarbe sowie Blink- und Breitschriftkennzeichen fuer den ganzen Bildschirm
- b) COLOR {Spalte},{Zeile},{Farbbyte}
Wie a), aber nur fuer das durch {Spalte} und {Zeile} adressierte Bildelement
- c) COLOR {Spalte},{Zeile},{Breite},{Hoehe},{Farbbyte}
Wie a), aber nur in einen Rechteck mit angegebener Hoehe und Breite, das sich von dem durch {Spalte} und {Zeile} angegebenen Element nach unten rechts erstreckt.

Alle Argumente koennen arithmetische Ausdruecke sein.

7.7.27. WINDOW

Das Kommando WINDOW grenzt einen Bildschirmbereich ein, in dem sich der Cursor bewegen kann. Es existiert in zwei Varianten:

WINDOW
legt als Bereich den gesamten Bildschirm fest.
WINDOW {Spalte},{Zeile},{Breite},{Hoehe}
legt einen Bereich in Analogie zu COLOR fest.

Alle Argumente koennen arithmetische Ausdruecke sein.

7.7.28. CLW

Loeschen des aktuellen WINDOWS.

7.7.27. DEFFN {Funktions-Name}({einfache Variable})

Mit dieser Anweisung beginnt eine Funktionsdefinition. Es wird die Funktion mit den angegebenen Namen definiert. Die Variable in Klammern kann eine einfache INTEGER- oder REAL-Variable sein. Sie dient der Parameteruebergabe, ist jedoch eine globale Variable, d.h. eine gleichnamige Variable im uebrigen Programm wird beim Aufrufen der Funktion veraendert! Der Typ der Funktion, d.h. der Typ des Ergebnisses entspricht stets dem des Parameters.

Hinter der DEFFW-Anweisung beginnt der Programmabschnitt zur Berechnung des Funktionswerts. Die Funktionsdefinition wird automatisch umsprungen.

7.7.30. FENND {Ausdruck}

Ende der Funktionsvereinbarung. Durch diese Anweisung wird der Funktion der Wert des Ergebnisses des Ausdrucks zugewiesen. Der Typ des Ergebnisses entspricht dem des Parameters der Funktion, gegebenenfalls wird konvertiert.

7.7.31. WHILE {Ausdruck}

WHILE ist eine Schleifen-Anweisung, die ein Programmabschnitt bis zum dazugehörigen WEND solange wiederholt abarbeitet, wie der Ausdruck hinter WHILE den logischen Wert .TRUE. ergibt. Bei .FALSE. wird hinter die WEND-Anweisung gesprungen. WHILE-Schleifen koennen geschachtelt werden.

7.7.32. WEND

Ende des WHILE-Blocks.

7.7.33. SWITCH {Liste von Geratekennzeichen}

SWITCH schaltet den Ausgabekanal, der von den Kommandos PRINT, OUTCHAR, CLS und gegebenenfalls INPUT auf Display oder die vorhandenen Druckerkanale. Eine Mehrfachzuweisung ist durch Angabe mehrerer Geratekennzeichen moeglich. Diese sind:

CON	- fuer die Konsole (Display)
LST	- fuer den Drucker 1 (SGN 3)
LST1	- ebenfalls fuer den Drucker 1
LST2	- fuer den Drucker 2 (SGN 4)

Listentrennzeichen ist Komma.

In einem Programm, in dem die Kommandos benutzt werden, die einen Ausgabekanal benoetigen, ist vor der ersten Ausgabe (am besten am Anfang des Programms) unbedingt ein SWITCH vorzusehen, da sonst die Zuweisung unbestimmt ist.

Die Verwendung der List- (LST-) kanale erfordert weiterhin die Kommandos OPEN und CLOSE.

7.7.34. OPEN {Gerätekennzeichen},{Integer-Variable}

Dieses Kommando reserviert das angegebene Gerät. Danach kann eine Ein-, bzw. Ausgabe vorgenommen werden.

Die fuer OPEN moeglichen Gerätekennzeichen sind:

LST	- fuer den Drucker 1 (SGN 3)
LST1	- ebenfalls fuer den Drucker 1
LST2	- fuer den Drucker 2 (SGN 4)
PDO	- fuer das Floppy-Laufwerk 0 (SGN 1)
PD1	- fuer das Floppy-Laufwerk 1 (SGN 2)

Fuer die Ausgabe ueber Display (CON) ist kein OPEN erforderlich.

Die Integer-Variable gibt die Adresse an, auf der Fehlermitteilungen durch die Geræte (PD Klartext bis 64 Byte, Drucker Code 2-Byte) abgelegt werden.

7.7.35. CLOSE {Gerätekennzeichen}

Das Kommando gibt das durch OPEN reservierte Gerät wieder frei.

7.7.36.

LOAD {Gerätekennz.},{Bytefeld-Name},{Var.-name},{Ausdr.}

Die im Ausdruck angegebene Anzahl von Bytes wird von der Diskette in den Speicher beginnend mit der Adresse der angegebenen Variablen gelesen. Das Lesen beginnt am Anfang der Datei. Der Dateiname muss in Bytefeld (2. Parameter) bereitgestellt werden. Das Gerätekennzeichen gibt das Laufwerk an:

PDO	Laufwerk 0
PD1	Laufwerk 1

Es wird eine Datei ohne Kopfblock gelesen.

7.7.37. CONTLOAD {Parameterliste wie bei LOAD}

Bei CONTLOAD wird das Lesen an der Stelle der Datei fortgesetzt, bis zu der ein anderes LOAD- oder CONTLOAD-Kommando gelesen hat.

7.7.38. SAVE {Parameterliste wie LOAD}

Die im Ausdruck angegebene Anzahl von Bytes wird aus dem Speicher, beginnend mit der Adresse der angegebenen Variablen, auf die Diskette geschrieben. Der Dateiname muss im Bytefeld (2. Parameter) bereitgestellt werden. Das Schreiben beginnt am Anfang der Datei. Das Gerätekennzeichen gibt das Laufwerk an:

PDO	Laufwerk 0
PD1	Laufwerk 1

Es wird eine Datei ohne Kopfblock geschrieben.

7.7.39. CONTSAVE (Parameterliste wie LOAD)

Eine durch ein anderes SAVE- oder CONTSAVE-Kommando begonnene Datei wird fortgesetzt.

Bemerkungen:

Bei den Kommandos LOAD, SAVE, CONTLOAD und CONTSAVE koennen auch Dateien mit Kopfblock behandelt werden. Dazu wird hinter dem letzten Parameter der Parameterliste ein

,H

angefuegt (Head).

Soll mit BAK-Datei geschrieben werden, ist hinter der Parameterliste ein

,B

anzufuegen. Es sind auch beide Angaben moeglich:

,H,B

Beim Lesen und Schreiben von Datenfeldern ist zu beachten:

- Die Anzahl der zu lesenden (schreibenden) Bytes ist von Typ des Feldes bzw. der Variablen abhaengig. Fuer Bytefelder ist pro Feldelement 1 Byte, fuer Integerfelder oder -variablen 2 Byte und fuer Realfelder bzw. -variablen 4 Byte vorzusehen.
- Felder bzw. Variablen, die durch eine DIM-Anweisung aufeinanderfolgend angelegt wurden, koennen mit einem Disketten-Kommando behandelt werden. Dazu ist als Variablenname der des ersten Feldes bzw. der ersten Variablen anzugeben und die Byteanzahl entsprechend zu berechnen.
- Bei allen Feldern ist zu beachten, dass die untere Grenze des Index 0 ist.

Beispiel:

DIM X(100) kann mit SAVE PDS,NAME=X,404 vollstaendig geschrieben werden.

7.8. Inline-Assembler

7.8.1. Allgemeines

Der Compiler verfuegt ueber einen integrierten Assembler, der es ermoeglicht, Programmabschnitte in Maschinencode einzufuegen. Hierbei wird die MAFS-Mnemonik des V 830 (VEB Robotron, s. IDAS, EBAS) verstanden.

7.8.2. Aktivierung des Assemblers

- a) Wird eine Anweisung mit / begonnen, so wird sie als Assemblerbefehl uebersetzt.

- b) Der Compiler kann durch den Befehl /ASM in einen speziellen Assemblermodus versetzt werden. Alle nachfolgenden Befehle werden als Assemblerbefehle interpretiert. Dieser Modus wird mit /BASIC wieder verlassen. Im Assemblermodus fuehrt der Gebrauch von/ver Anweisungen zu Fehlermeldungen.

7.8.3. Spachumfang

Der Assembler uebersetzt alle Befehle. Relativspruenge koennen nur im Assemblermodus benutzt werden. Der Marke ist dabei ein \neq nachzustellen. Als Einbytekonstanten koennen keine symbolischen Marken verwendet werden. Konstanten sind alle im BASIC definierten INTEGER-Konstanten. Alle im BASIC-Teil vereinbarten Namen sind fuer den Assembler symbolische Marken, wobei das typbestimmende Zeichen weggelassen werden darf. Als Markenausdruck ist {Konstante} oder {Marke} oder {Marke}+{Konstante} oder {Marke}-{Konstante} zulassig.

Beispiel:

```
/LD HL,X1%   Adresse der INTEGER-Variablen X1% ==> HL
/LD DE,X3    Adresse der REAL-Variablen X3    ==> DE
/LD IX,(X1%) Wert der INTEGER-Variablen X1%   ==> IX
/LD (P%+4),HL wenn P% ein INTEGER-Feld ist, so wird
              HL ==> Wert von P%(2)
/LD A,(X1%)  LOW(X1%) ==> A
```

Achtung! Bei Spruengen werden auch solche ohne Fehlermeldung uebersetzt, die als Sprungziel Variablenamen angeben.

Als Pseudobefehle koennen verwendet werden:

/DB {Liste mit Komma als Trennzeichen}

==> als Listenelemente sind zugelassen:

- Zeichenketten - "ABCD"
- Einbytekonstanten - 'A','"',10,30H

/DW {Liste mit Komma als Trennzeichen}

==> als Listenelemente sind zugelassen:

- Zeichenketten - "ABCD"
- Einbytekonstanten - 'A','"',10,30H
- Zweibytekonstanten - 1000H
- Marken
- Ausdruecke - {Marke}+|-{Konstante}

Bei Zeichen oder Zeichenketten erfolgt ein lockeres Eintragen (ein ASCII-Code auf zwei Byte).

7.8.4. Verfuegbarkeit der Prozessorregister

In Maschinencodeabschnitten koennen alle Register, ohne deren Inhalte retten zu muessen, verwendet werden. BASIC-Anweisungen veraendern die Inhalte der Vordergrundregister sowie IX und IY.

7.9. Systemkommando-Compiler

7.9.1. Allgemeine Bemerkungen

Das Betriebssystem des AUDATEC-Applikationsrechners kennt eine Reihe von Systemrufen bzw. Kommandos, die dem Nutzer zugänglich sind. Sie werden durch einen Befehl RST 20H mit nachfolgender Rufnummer (DB n) realisiert. Der Compiler BC-88.9 erlaubt die effektive Nutzung dieser Systemrufe durch Erweiterung des Sprachumfangs um entsprechende Schluesselwörter. Der Compiler fuer die Systemrufe wird mit dem Zeichen > aktiviert, es ist somit dem Schluesselwort des jeweiligen Kommandos voranzustellen.

7.9.2. Steuerung des Echtzeitbetriebssystems

7.9.2.1. >PROGRAM {Programmname}({Marke 1},{Marke 2},{Marke 3})

Diese Anweisung generiert die einen fuer das Betriebssystem verstaendlichen Kopfblock mit den angegebenen, aus maximal 30 Zeichen bestehenden Programmnamen. Der Name taucht nach dem Eintragen des Programms in die VAP-Liste in Menue auf. Die Marken bezeichnen die Eintrittspunkte in das Programm bei

Initialisierungsstart:	Marke 1
Neustart:	Marke 2
Restart:	Marke 3

Es koennen in einem BASIC-Programm mehrere PROGRAM-Anweisungen enthalten sein, wodurch mehrere VAP generiert werden.

7.9.2.2. >CONTINUE

Fortsetzen des laufenden VAP's mit Meldung an das Betriebssystem (Kommando 0).

7.9.2.3. >STOP

Abbruch des laufenden VAP's (Kommando 1).

7.9.2.4. >START {Ausdr.}

Anmeldung des VAP's mit der im Ausdruck angegebenen Nummer zum Neustart (Kommando 2).

7.9.2.5. >START {Ausdr. 1} AFTER {Zeitdiff.}

Anmeldung des VAP's mit der im ersten Ausdruck angegebenen Nummer zum Neustart mit Zeitverzoeigerung (Kommando 3). Die Verzoeigerung {Zeit} besitzt folgende Syntax:

```
{Zeitdiff.} ::= {Ausdr. fuer Verzoeigerung in Grundtakten}
                SEC {Ausdr. fuer Verzoeigerung in Sekunden}
                MIN {Ausdr. fuer Verzoeigerung in Minuten}
```

7.9.2.6. >START {Ausdr.} AT {Zeitpunkt}

Anmeldung des VAP's mit der im ersten Ausdruck angegebenen Nummer zum Neustart mit Uhrzeit (Kommando 4). Die Uhrzeit hat folgende Syntax:

{Zeitpunkt} ::= {Ausdr.},{Ausdr.}

Der erste Ausdruck gibt die Stunde, der zweite die Minute des Startzeitpunkts an.

7.9.2.7. >START {Ausdr.} CYCLE {Zeitdiff.}

Anmeldung des angegebenen VAP's zum zyklischen Restart (Kommando 5). Die Syntax der Zeitdifferenz entspricht der im Punkt 7.9.2.4.

Wird anstelle des ersten Ausdrucks ein * geschrieben, so wird das laufende VAP zum zyklischen Restart angemeldet.

7.9.2.8. >RESTART {Ausdr.}

Anmeldung des angegebenen VAP's zum Restart ohne Zyklus (Kommando 6). Ein Stern anstelle der VAP-Nummer kennzeichnet das laufende VAP.

7.9.2.9. >CSBT {Ausdr.} CYCLE {Zeitdiff.}

Setzen des Zykluszaehlers (Kommando 7). Der Ausdruck gibt die VAP-Nummer an.

7.9.2.10. >TERMINATE {Ausdr.}

Abmelden des angegebenen VAP's. Wird anstelle des Ausdrucks ein * geschrieben, so wird das laufende VAP abgemeldet.

7.9.2.11. >SUSPEND FOR {Zeitdiff.}

Unterbrechung des VAP's fuer die in der {Zeitdiff.} angegebene Anzahl von Sekunden oder Minuten (Kommando 9). Eine Angabe in Grundtaktten wird nicht unterstuezt.

7.9.2.12. >SUSPEND

Unterbrechung des VAP's fuer einen Grundtakt (Kommando 10).

7.9.3. Steuerung der Systemuhr

7.9.3.1. >DATESTR {Bytefeld-Name}

Das Datum wird als Zeichenkette auf das angegebene Bytefeld abgelegt (Kommando 15). Das Feld muss mindestens 8 Byte lang sein.

7.9.3.2. >TIMESTR {Bytefeld-Name}

Die Uhrzeit wird als Zeichenkette auf das angegebene Bytefeld abgelegt (Kommando 16). Das Feld muss mindestens 8 Byte lang sein.

7.9.3.3. >GETDATE {Variable 1},{Variable 2},{Variable 3}

Das Datum wird an die drei angegebenen Variablen uebergeben (Kommando 18). Es sind sowie einfache Variablen als auch Feldelemente zulaessig. Die Reihenfolge der Zuordnung ist: Jahr, Monat, Tag.

7.9.3.4. >GETTIME {Variable 1},{Variable 2},{Variable 3}

Die Uhrzeit wird an die drei angegebenen Variablen uebergeben (Kommando 17). Es sind sowie einfache Variablen als auch Feldelemente zulaessig. Die Reihenfolge der Zuordnung ist: Stunde, Minute, Sekunde.

7.9.3.5. >PUTDATE {Ausdr.1},{Ausdr.2},{Ausdr.3}

Das Datum wird entsprechend dem Ergebnis der drei Ausdruecke gestellt (Kommando 20). Die Reihenfolge der Zuordnung ist: Jahr, Monat, Tag.

7.9.3.6. >PUTTIME {Ausdr.1},{Ausdr.2},{Ausdr.3}

Die Systemuhr wird entsprechend dem Ergebnis der drei Ausdruecke gestellt (Kommando 19). Die Reihenfolge der Zuordnung ist: Stunde, Minute, Sekunde.

7.9.4. Steuerung des FE-Statuses

7.9.4.1. >GETSTAT {Variable}

Der FE-Status wird der Variablen als logischer Wert uebergeben. .TRUE. bedeutet On-Line .FALSE. Off-Line (Kommando 21).

7.9.4.2. >ONLINE

FE-Status wird On-Line (Kommando 22).

7.9.4.3. >OFFLINE

FE-Status wird Off-Line (Kommando 22).

7.9.5. Zugriff zu Prozess- und Applikationsrechnerabbild

7.9.5.1. >PGET

Es werden die Daten vom Prozessabbild mit der angegebenen Nummer und dem gegebenen Typ (0, 1, 2 oder 3) gelesen (Kommando 25). Das Ergebnis wird den entsprechenden Variablen zugeordnet. Es sind einfache Variablen und Feldelemente zulässig. Der ersten Variable wird stets das erste Byte des Abbilds zugeordnet, welches die Status-Information enthält. Es existieren folgende Varianten:

```
>PGET ({Nr.},T0),{Stat.}
>PGET ({Nr.},T1),{Stat.},{Byte}
>PGET ({Nr.},T2),{Stat.},{Wort}
>PGET ({Nr.},T2:UI),{Stat.},{X}          => Zaehlwert => Real
>PGET ({Nr.},T2:FP),{Stat.},{X}         => FK-Zahl  => Real
>PGET ({Nr.},T3:),{Stat.},{Wort},{Wort} => 2 x 16 Bit
>PGET ({Nr.},T3:UI),{Stat.},{X}         => Zaehlwert => Real
>PGET ({Nr.},T3:R),{Stat.},{X}         => Real-Groesse
```

7.9.5.2. >AGET

Es werden die Daten vom Applikationsrechnerabbild mit der angegebenen Nummer und dem angegebenen Typ (0, 1 oder 2) gelesen (Kommando 26).

Es existieren folgende Varianten:

```
>AGET ({Nr.},T0),{Stat.},{Byte}
>AGET ({Nr.},T1),{Stat.},{Wort}
>AGET ({Nr.},T1:UI),{Stat.},{X}          => Zaehlwert => Real
>AGET ({Nr.},T1:FP),{Stat.},{X}         => FK-Zahl  => Real
>AGET ({Nr.},T2:),{Stat.},{Wort},{Wort} => 2 x 16 Bit
>AGET ({Nr.},T2:UI),{Stat.},{X}         => Zaehlwert => Real
>AGET ({Nr.},T2:R),{Stat.},{X}         => Real-Groesse
```

7.9.5.3. >PADDR({Nr.},Tn),{Adr.}

Es wird die Adresse des Prozessabbildes mit der angegebenen Nummer und dem gegebenen Typ (0, 1, 2 oder 3) ermittelt (Kommando 27) und der angegebenen Variablen (auch Feldelement moeglich) uebergeben.

7.9.5.4. >AADDR({Nr.},Tn),{Adr.}

Es wird die Adresse des Prozessabbildes mit der angegebenen Nummer und dem gegebenen Typ (0, 1 oder 2) ermittelt (Kommando 28) und der angegebenen Variablen (auch Feldelement moeglich) uebergeben.

7.9.5.5. >PREAD

Es werden die Daten vom Prozessabbild mit den Varianten von >PGET gelesen. Es muss aber statt {Nr.} ein Adressausdruck angegeben werden (Kommando 29). Das Ergebnis wird, analog zu >PGET den Variablen der Liste zugeordnet.

7.9.5.6. >AREAD

Es werden die Daten vom Prozessabbild mit den Varianten von >AGET gelesen. Es muss aber statt {Nr.} ein Adressausdruck angegeben werden (Kommando 30). Das Ergebnis wird, analog zu >AGET den Variablen der Liste zugeordnet.

7.9.5.7. >APUT

Schreiben von Daten ins AR-Abbild mit der angegebenen Nummer und den angegebenen Typ (Kommando 31). Die nachfolgenden 2 bis 3 Ausdrücke stellen die zu schreibende Information dar.

Es existieren folgende Varianten:

```
>APUT ({Nr.},T0,{Stat.},{Byte})
>APUT ({Nr.},T1,{Stat.},{Word})
>APUT ({Nr.},T1:UI,{Stat.},{I})
>APUT ({Nr.},T1:FP,{Stat.},{X})
>APUT ({Nr.},T2:{Stat.},{Word},{Word})
>APUT ({Nr.},T2:UI,{Stat.},{I})
>APUT ({Nr.},T2:R,{Stat.},{X})
```

>APUT ist als Umkehrung von >AGET zu betrachten.

7.9.5.8. >AWRITE

>AWRITE existiert in den Varianten wie >APUT. Es muss jedoch statt {Nr.} eine Adresse angegeben werden. Diese kann auch durch einen arithmetischen Ausdruck dargestellt werden.

7.9.6. Sonstiges

7.9.6.1. >BACKUP ON

Back Up einschalten (Kommando 23).

7.9.6.2. >BACKUP OFF

Back Up abschalten (Kommando 23).

7.9.6.3. >SCS {Integer- oder Bytefeldname}

>SCS (Save Control of Screen) uebertraegt den Inhalt des Displaysteuerblocks (s. Systembeschreibung) in das angegebene Feld. Dieses muss mindestens 8 Byte lang sein:

```

    DIM P%(3)
    >SCS
oder
    DIM Pa(7)
    >SCS

```

7.9.6.4. >LCS {Integer- oder Bytefeldname}

>LCS (Load Control of Screen) laedt einen durch >SCS geretteten Displaysteuerblock zurueck.

7.9.7. Fehlerbehandlung bei Systemrufen

Die Kommandos START, START APTEN, START AT, START CYCLE, RESTART, CSET, TERMINATE, PGET, AGET, PADDR, AADDR, OPEN, PRINT, LOAD, CONTLOAD, SAVE, CONTSAVE liefern eine Fehlermeldung, die dem Nutzer ueber die Systemvariable ERROR% zugaeenglich ist. Sie muss nicht vereinbart werden. Ein Fehler wird durch den logischen Wert .TRUE. angezeigt, dadurch kann ein einfacher Test realisiert werden:

```
IF ERROR% THEN .....
```

Die Bedingung ist erfuehlt, wenn ein Fehler vorliegt.

7.10. Fehlermeldungen

Der Compiler erkennt stets nur einen Fehler je Uebersetzungslauf. Er erzeugt einen Fehlercode entsprechend der folgenden Tabelle:

Fehlercode	Fehler
01	Fehlerhafter Name / END fehlt
02	Falscher Variablentyp
03	Klammerfehler
04	Falsches Argument in ADDR-Funktion
05	Fehler bei Markenvereinbarung
06	Mehrfach definiertes Symbol
07	Syntaxfehler
08	Falsches Trennzeichen
09	Falsche Feldgrenze
0A	DOEND oder ELSE ohne DO
0B	FOR-DO-WHILE- Verschachtelung
0C	Nicht definierte Marke
0D	Fehler in Parameterliste
0E	NEXT ohne FOR
0F	NEXT fehlt
10	DOEND fehlt
11	WEND ohne WHILE
12	Fehler in Format
13	FNEND ohne DEFFN
20	Mnemonik-Fehler (Assembler)
21	Fehler im Assemblerbefehl
22	Fehler im Assemblerbefehl
23	Fehler im Assemblerbefehl (Relativsprung)
FE	Nicht definierte Marke
FF	Systemfehler, Speicherbereichsueberschreitung, sonstige Fehler

8. Schematasystem des Applikationsrechners mit Eigenstrukturierung

8.1. Darstellung der statischen Bilder

8.1.0. Zweck des gesamten Schematasystems

Zur Darstellung technologischer Schemata mit aktuellen Prozessinformationen müssen Bildschirmhalte strukturiert und im Speicher abgelegt werden. Zur direkten Uebernahme eines statischen Bildes in den Speicher werden entsprechend dem Bildschirmwiederholungspeicher 4 KByte benötigt. Deshalb wird ueber das Programmsystem der reale Bildschirminhalt in einem speziellen Code verdichtet und abgelegt, so dass sich eine Reduzierung des Speicherbedarfs auf ca. 30 % (statistisches Mittel) ergibt. Durch spezielle Sonderfunktionen, z. B die Mehrfachnutzung von Bildteilen als "Bildmodul", laesst sich ueber Strukturiermassnahmen der Speicherbedarf weiter senken.

Dynamische Informationen, d. h. binäre oder analoge Prozessinformationen lassen sich prinzipiell mittels beliebiger Programme selbst realisieren. Das Bildschirmssystem enthaelt fuer die wesentlichen Darstellungsarten Programmbausteine, die nur aufgerufen und parametrisiert werden muessen, um die gewuenschte Information darzustellen.

Darueber hinaus erlaubt das Bildsystem relativ komfortabel den Entwurf, die Bildeingabe mit dynamischen Informationen, einen Test ueber die Gesamtaufteilung und Manipulation zur Speicherminimierung. Der fuer die Bildinformationen reservierte Speicher kann waehrend der Bearbeitung in 1K-Stufen dem Bedarf angepasst werden.

Das umfangreiche Programmpaket fuer die Bildstrukturierung (ca. 16 KByte) kann im Bedarfsfall geladen werden, so dass der Speicher bei On-line-Betrieb fuer andere Programme verwendet werden kann. Im ON-Line-Betrieb arbeitet ein angepasster Bildinterpret.

8.1.1. Speicheraufteilung

Fuer die Bilder stehen max. 3 Speicherebenen mit je 16 KByte zur Verfuegung ab Adr. 1000H. Jede der Speicherebenen wird in Aufbau der Videothek gesondert behandelt. Bildspeicherebenen sind A=2, B=3, C=4.

Auf jeder Ebene sind Adressplaetze fuer 64 Bildmodule und fuer 25 Bilder fest vereinbart. Dabei gelten fuer

Ebene A:	Bild 1 ... 25;	Modul 1 ... 64
Ebene B:	Bild 26 ... 50;	Modul 65 ... 128
Ebene C:	Bild 51 ... 75;	Modul 129 ... 192

Die Ebenenumschaltung erfolgt anhand der Bild- und Bildmodulnummern bei der Manipulation mit der Videothek.

Der Dateiaufbau in der Ebene hat die Form nach Bild 8.1.4

Ist ein Bild oder Bildmodul nicht vorhanden, so ist die Verweissadr. 0000H. Bei Bildanwahl wird dann ein Standardbild "NICHT VORHANDEN" angezeigt.

Der Code der Bildmodule und Bilder ist verschieblich, d. h., es sind in ihm keine absoluten Adressen enthalten.

Die Codes stehen entsprechend ihrer Nummer geordnet; beim Löschen eines Bildes wird nicht gespeichert, bei Uebertragen in die Videothek wird entsprechend der Ordnungszahl das Bild bzw. Modul eingefügt und die Startadressen werden aktualisiert.

Die Datenebene 1 wird zur Bildstrukturierung als Arbeitsspeicher benötigt und ist deshalb vorher auf MB-Kassette zu sichern. Zum statischen Bildaufbau werden auf dieser Ebene benötigt:

Adr: 1000H 5 KByte Code statisch
 3000H 4 KByte Code dynamisch
 4000H 4 KByte Bildspeichermaskierung

Die Systemprogramme zur Bildstrukturierung sind auf der Programmebene 2 ab 7000H abgelegt.

Adr: 1000	[Anzahl Ebenen (2 Byte)]
1002	[Endadr. Reservierung]
1004	[Anfadr. freier Bereich]

1006	[Adr. 1. Bildmodul]
	["]
	["]
1084	[Adr.64. Bildmodul]

1086	[Adr. 1. Bild stat.]
1088	[Adr. 1. Bild dyn.]
	["]
	["]
	["]
10B6	[Adr.25. Bild stat.]
10B8	[Adr.25. Bild dyn.]

10BA	[Code]
	[Bildmodule]

	[Code Bilder]
	[statisch/dynamisch]

Anf.Adr. freier Ber.	[freier Bereich]

Bild 8.1.1.: Dateiaufbau in einer Datenebene

Hinweis: Das n-te Bildmodul bzw. Bild ist das n-te Bildmodul bzw. Bild auf dieser Datenebene.
 absolute Modul-Nr. = $n + 64 * (\text{Ebenen-Nr.} - 2)$
 absolute Bild-Nr. = $n + 25 * (\text{Ebenen-Nr.} - 2)$

8.1.2. Systematik der Speicherung

Die interne Codierung geht in der Bildbeschreibung vom linken Rand der Zeile 2 aus und setzt sich ueber relativ aufsteigende Adressen fort bis zum rechten Rand der Zeile 32. Relativ aufsteigend heisst, die Startadresse einer n-ten Bildfunktion ergibt sich aus der Summe der Laengen der n-1 vorhergehenden Funktion zuzueglich der Anfangsadresse der Zeile 2 des Bildspeichers. Voraussetzung der Decodierung ins Bild ist ein auf gruen/schwarz geloeschter Bildschirm. Der Bildaufbau kann auch vor Ende der Zeile 32 durch den Code "statisches Bildende" abgeschlossen werden.

Abschn. 8.1.6. zeigt die verwendeten Elemente der Codierung. Die Sonderfunktionen duerfen nur am Kopf des Bildcodes stehen vor der ersten Normalfunktion. Da so die Sonderfunktionen als erste aufgebaut werden, koennen sie bei Bedarf mit beliebigen Zeichen ueberschrieben werden.

Die Codierung geht davon aus, dass Breitschrift und Blinken in statischen Bild nicht verwendet werden.

Bildmodule werden wie Bilder erstellt. Ihr Bezugspunkt ist immer links oben. Bei der Einbindung in Bilder und andere Module wird die aktuelle Position dazuzueaddiert.

8.1.3. Dateiverwaltung

Die Bilder und Bildmodule sind entsprechend dem Schema nach Bild.MI im Videothekspeicher abgelegt. Zur Bearbeitung im Bildsystem wird jeweils ein Bild oder Bildmodul in den Arbeitsspeicher umgeladen und ueber das Bilderstellungsprogramm bearbeitet. Zu jedem Zeitpunkt kann das Bild an beliebige freie Plaetze unter einer Bild-/modulnummer wieder in der Videothek abgelegt werden.

8.1.4. Bedienung

Zur Bedienung der Menue- und Normalfunktionen wird ausschliesslich die oberste Zeile des Bildschirms verwendet, die im on-line-Betrieb fuer allgemeine Systemmeldungen reserviert ist.

Nach dem Laden meldet sich das System nach Start ueber das Systemmenue mit dem Menue zur Bild-Strukturierung

```
0 BILD 1 DYN 2 ZUOR 3 VID 4 CODE 5 ANZ N NEUS
```

```
0 BILD  Bildstrukturierung
```

```
-----
```

Mit Aufruf der Strukturierung "Bild" wird das im Arbeitsspeicher vorhandene Bild/Bildmodul angezeigt. In der ersten Zeile werden Farbe, Cursorposition und die gewaehlte Zeichenzuordnung ausgegeben.

Zur Eingabe und Aenderung des Bildes werden die alphanumerischen Tasten und Sonderzeichen verwendet. Einige Tasten haben dabei besondere Funktionen der Eingabesteuerung:

Cursortasten: Die vier Cursortasten setzen den Cursor jeweils ein Zeichen in die angegebene Richtung weiter. Bei "rechts" erscheint der Cursor nach Zeilenende auf dem Anfang der naechsten Zeile, bei Zeile 32 am Anfang der Zeile 2. Bei "links" ist die Ausfuehrung synonym, bei "hoeher" springt der Cursor von Zeile 2 auf Zeile 32, bei "tiefer" umgekehrt. Die Stellung des Cursors wird oben rechts (Zeile 1) als Zeile/Spalte angezeigt.

Kleinbuchstaben: Mit der Eingabe von Kleinbuchstaben wird auf dem Bildschirm das zugeordnete Zeichen eingesetzt.

Cursorzuordnung: Mit Betaetigung der Taste wird dem Cursor [ⓐ] das zuletzt eingegebene Zeichen zugeordnet. Mit Betaetigung der Cursortaste wird das Zeichen auf dem Bildschirm eingetragen. (Verwendung fuer Wiederholungen senkrecht und waagerecht, z. B. Linien ziehen)

Cursor freisetzen: Diese Taste entbindet den Cursor von der [ⓐ] vorher genannten Eingabefunktion.

Farbschaltung: Mit dieser Funktion koennen Farbaenderungen [RESET] und spezielle Manipulationen im Bildschirm ausgefuehrt werden.

- a) Zur Farbschaltung sind nach [RESET] zwei die Farbe bezeichnende Buchstaben einzugeben, und zwar

S = schwarz	G = gruen	P = purpur
W = weiss	B = blau	Y = gelb
R = rot	C = cyan	(yellow)

- b) Wird nach [RESET] ein "F" (Faerben) eingegeben, so wird durch Eingabe weiterer beliebiger Zeichen nur die gewahlte Farbe eingetragen. Die Funktion wird durch Neueingabe der Farbe nach [RESET] verlassen (Anzeige "F" vor der Farbe). Ist vorher dem Cursor ein Zeichen zuegordnet worden, so koennen die Cursortasten zum Faerben verwendet werden.

- c) Nach [RESET] kann durch Eingabe spezieller Zeichen der Bildinhalt manipuliert werden:
- "I" INS - Einfuegen eines Leerzeichens bis Zeilenende.
 - "D" DEL - Tilgen eines Zeichens und Heranruecken des Zeilenrests
 - "L" INS LINE - Einfuegen einer Zeile und Verschieben der weiteren Zeilen nach unten.
 - "X" DEL LINE - Tilgen einer Zeile

Die Funktion wird durch Eingabe eines nicht relevanten Zeichens verlassen (z.B. Leertaste) oder durch die Anweisung einer neuen Farbe oder "F".

Hinweis: Die Vereinbarung von Sonderfunktionen
(Modul, Rechteck) wird dabei nicht beruecksichtigt, d.
h., durch Verschieben sind die Sonderfunktionen und der
Bildinhalt nicht mehr deckungsgleich, und die Codierung
erfasst alle nicht deckungsgleichen Teile zusaetzlich.
Durch Aenderung der Sonderfunktionen oder Loeschen oder
Neueintragen kann diese ungewollte Erweiterung korri-
giert werden.

Codierung: Das angezeigte Bild wird automatisch codiert.
[ENTER] Nach Ende der Funktion wird die aktuelle Laenge
auf der Zeile 1 hexadezimal angezeigt.

Fuer die Sonderfunktionen beim Bildaufbau werden Kommandos verwendet. Das Menue wird ueber die Kommandotaste "<-" aufgerufen.

Kommandos:

MOD(NR)/R.ECK(M-RAND/HOCH/BREIT/FA:FELD/RAND)/LOESCH

M: Mit diesem Kommando wird ein gespeichertes Bildmodul gerufen und entsprechend der Cursorstellung ins Bild eingetragen. Der Aufruf erfolgt ueber die Nummer. Wird vor oder nach der Auswahl nichts eingetragen, geht die Steuerung zum Bild zurueck. Ueber "L" kann ein Bildmodul wieder geloescht werden (vorher Cursor auf Modulposition).

Fehler: Eingabe rot: Modul nicht vorhanden.

R: Kommandos zur Vereinbarung eines Rechteckes oder einer senkrechten Zeichenwiederholung ("Linie senkrecht") bei der aktuellen Cursorposition. Mit Eintragung "L" wird ein auf der Cursorposition eingetragenes Rechteck geloescht.

- Linie senkrecht: Nach Fuehren des Cursors auf das Zeichen, das senkrecht vervielfaeltigt werden soll, ist das Kommando zu rufen und die Zahl einzutragen.
- Rechteck mit Rand innen: Eingabe von Hoehe, Breite, Farbe Feld, Farbe Rand. Der Cursor bildet die linke obere Ecke des Rechteckes.
- Rechteck ohne Rand: Die Randfarbe ist mit "SS" zu vereinbaren.
- Rechteck mit Mittelrand: Es ist "RM" am Anfang anzugeben.

Hinweis:

Mit Ausfuehrung der Kommandos wird der Bildcoder ge-
startet.
Die Codierung ist nach Abschluss der Funktion voll-
staendig, jedoch nicht laengenoptimal, wenn identische
Ueberlappungen zum Code vor der Funktionsvereinbarung
auftreten.

L : Loeschen: die erste im Code vereinbarte Sonderfunktion fuer diese Cursorposition wird geloescht.
Fehler:"L" wird rot, wenn auf dieser Cursorposition keine Sonderfunktion vereinbart ist.

leer: Bei leerem Eingabefeld erfolgt die Rueckkehr ohne Kommandoausfuehrung (gilt auch fuer nicht identifizierbare Eintragungen).

2 ZUOR Zuordnungsliste

=====

Im Bild wird der gesamte Vorrat der generierten Zeichen der Bildschirmsteuerung angezeigt.

Durch Eintragen des Hexa-Code der gewaehlten Zeichen entsprechend der Buchstabenreihenfolge koennen den Kleinbuchstaben der Tastatur beliebige Zeichen zugeordnet werden.

3 VID Videotheks-Bedienung

=====

Die Videothek umfasst den Speicher fuer Bildmodule (Modul) und Bilder (Picture).

Die Funktion wird ueber Kennbuchstaben und Bild-Nummer eingegeben.

"XY ZZZ"	X	Kennbuchstabe fuer die Funktion I/O/D
	Y	P = Picture; M = Modul
	Z	Bild-/Modul-Nummer

I: INS Einspeichern in die Videothek

O: OUT Ausgabe aus der Videothek

D: DEL Loeschen aus der Videothek

Meldung: "NO IN": gewuenschte Bild-/Modulnummer ist bereits belegt.

"ENDVI": Die Einordnung des Bildes oder Moduls in die Videothek wuerde zu Speicherueberlauf fuehren.

Die Videothekslaenge auf den max. 3 Ebenen wird nach Ausfuehrung der Anweisung angezeigt.

4 ANZ Anzeige der Videothek

=====

Anzeige der Bild- und Bildmodulplaetze mit den Startadressen in den Ebenen. "Umblaettern" erfolgt ueber "ENTER" und "Cursor hoch".

5 CODE Bildcode-Anzeige

=====

Der Code des jeweils aktuell bearbeiteten Bildes wird mnemotechnisch entsprechend Abschn. 8.1.6. angezeigt. Ueber die Cursortasten "hoeher" und "tiefer" kann das Bild gerollt werden.

Mit der Anzeige von Zeile und Spalte kann die genaue Position der Sonderfunktionen am Kopf des Bildcode entnommen werden.

N NEUS Neustart

=====

Mit der Funktion "Neustart" wird zu Beginn der Bilderstellung oder im Verlauf der Bearbeitung der Speicherplatz der Videothek festgelegt und gleichzeitig damit der zu sichernde Bereich formiert.

Hinweis: Nach Neuaufruf des Bildsystems ist unbedingt zu-
erst "Neustart" anzuwaehlen, auch wenn die Speicherreser-
vierung nicht geaendert wird.

Zum Neustart ist die Anzahl der gewuenschten n KByte Speicherreserve als Parameter einzutragen (1..48). Der Zusatz "L" loescht dabei den gesamten Bereich der Videothek und formiert ihn neu. Ohne diesen Zusatz wird nur die Endadresse neu vorgegeben und die Differenz formiert, so dass die bisherige Videothek erhalten bleibt.

Beispiele:

1) Neustart 37 : 42

Der Videotheksbereich wird von bisher 37 KByte auf 42 KByte erweitert. Die damit ab 37 KByte evtl. vorhandenen Anwenderprogramme werden geloescht.

Die Zahl 37 verdeutlicht den bisher reservierten Bereich.

2) Neustart 37 : L42

Ein Bereich von 42 KByte wird geloescht und neu formiert. Evtl. vorhandene Bilder sind danach geloescht.

3) Neustart 37 : 24

Der Videotheksbereich wird von bisher 37 KByte auf 24 KByte gekuerzt. Ist die aktuelle Laenge der Videothek groesser als 24 KByte, so wird die Anweisung nicht ausgefuehrt. Es erfolgt eine Fehlermeldung. Zur Ausfuehrung muessen vorher ebenenbezogen Bilder und Module bis zur gewuenschten Laenge ueber die Videotheksbedienung gestrichen werden.

8.1.5. Festlegungen zum Zeichengenerator

Fuer Grossbuchstaben und gebrauchliche Sonderzeichen wird der ISO-Code vorausgesetzt (z. B. "A" = 41H).

Fuer die Sonderfunktion "Rechteck" sind die Codezeichen

05 06 07 8F 90 91 92 C1 C5 C9 (hexadezimal)

fest vereinbart nach dem Muster in Bild 8.1.2. Andere Festlegungen haben Programmaenderungen zur Folge. Ohne Vereinbarung entsprechender Symbole ist die Sonderfunktion "Rechteck" nicht nutzbar.



Bild 6.1.2.: Zeichen fuer die Sonderfunktion Rechteck

3.1.6. SPRACHBESCHREIBUNG fuer statische Bildinformation

Mnemonic	Bezeichnung		Bit- codierung	Hexa- code	Summand Bildadr.
COL	Color	Farbschaltung x:-Farbinformation Vorder-/Hintergrund	xxxxxxII	--	0
JMP	Jump	Sprung auf neue Bildadresse x:=L-1 L:=Laenge (max.64)	xxxxxxOI	--	L
REP	Repetition	Wiederholung eines Zeichens x:=A-1 A:=Anzahl (max.64) Y:=Zeichen	xxxxxxIO YYYYYYYY	--	A
QUE	Queue	Zeichenreihe x:=Z-1 Z:=Laenge (max.64)	xxxxxIOO	--	Z
NOOP	No Operation	keine Bild- funktion	00000000	--	0
JMPZ	Jump Zeile	Zeilensprung Y:=Zeilenzahl	000II000 YYYYYYYY	16 YY	y*64
ENDS	Ende Static	Ende statischer Bildinhalt	0000I000	08	0

S O N D E R F U N K T I O N E N
(immer am Bildcode-Anfang)

RECK	Rechteck	Rechteck / Linie senkrecht	00I0I000 nnnnnnnn nnnnnnnn nnnnnnnn pppppppp qqqqqqqq rrrrrrrr yyyyyyyy	28 1M 1M 1M PP QQ RR YY	0
		N:=Relativadresse fuer Rechteck N+128:=Mittelrand P:=Breite Q:=Farbe Innenflaeche R:=Farbe Rand, bei "SS" ohne Rand	zum Bildanfang	fuer Linie senkrecht N:=Hoeche P:=<2 Q:=Zeichenelement R:=Farbe	
BIM	Bild- modul	Bildmodulaufruf N:=Modulnummer M:=Relativadresse zum Bildanfang	00I00000 nnnnnnnn nnnnnnnn nnnnnnnn	20 1M 1M 1M	0

8.1.7. Hinweise zur Strukturierung des statischen Bildes

Bei der Bildstrukturierung sollte von moeglichst exakten Zeichnungen ausgegangen werden, wenn der Speicherbedarf minimal gestaltet werden muss.

Aus den Vorlagen lassen sich am besten Wiederholungen in einem oder in mehreren Bildern feststellen und dann als Bildmodul eintragen. Unterscheiden sich zwei Bilder nur an wenigen Stellen, so kann auch das ganze Bild als Modul vereinbart werden, wobei der groesste Spareffekt erreicht wird. Wegen der Moeglichkeit der mehrfachen Schachtelung lassen sich auch in einem Modul andere Module bei Wiederholung definieren.

Die Definition von Bildmodulen, die nur einmal in einem Bild verwendet werden, vermindert den Speicherbedarf nicht. Bildmodule mit weniger als vier Zeichen bringen selten Spareffekte.

Module und Bilder sind nacheinander einzugeben und in der Videothek zu speichern. Die Sonderfunktion "Rechteck" kann bedarfsweise vereinbart werden. Rechtecke koennen auch nachtraeglich im Bild definiert werden. Dabei ist darauf zu achten, dass vorher dieselbe Art Randlinie verwendet worden ist, sonst wird das Rechteck nachtraeglich durch die anderen Zeichen ueberschrieben und der Speicherbedarf erhoeht sich.

Auch Bildmodule koennen nachtraeglich definiert und im anderen Bild installiert werden.

Beim sonst fertigen Bildmodul oder Bild kann durch Aussehen und Vereinbaren senkrechter Linien eine weitere Speicherminderung erreicht werden, besonders bei zusaeztlichen Farbaenderungen in der Bildzeile. Senkrechte Linien koennen mit beliebigen senkrechten Zeichenreihen vereinbart werden.

Sind bei Bildmodulen, Rechtecken oder senkrechten Linien Teile durch andere Zeichen belegt, so spielt das fuer ihre Installation im Bild keine Rolle. Die Sonderfunktionen werden immer als erste im Bild aufgebaut, so dass die nicht passenden Zeichen an den betreffenden Stellen darueber geschrieben werden und somit erhalten bleiben. Eine Ausnahme bilden evtl. Leerzeichen (DCH). In diesem Fall ist ein anderes Zeichen (z. B. Vollzeichen mit Vordergrund schwarz) einzutragen.

Beim Loeschen von Sonderfunktionen sollte zur Ermittlung der genauen Cursorposition der Bildcode angezeigt werden.

Bei Sonderfunktionen sollte auf ihre nachtraegliche Sichtbarkeit geachtet werden. Es ist z. B. moeglich, beliebig viele senkrechte Linien mit Leerzeichen zu vereinbaren, die ausser einer Erweiterung des Bildcode keinen Effekt zeigen.

Vor der Minimierung sollten fertige Bilder in der Videothek gespeichert werden, um bei Misserfolg oder Fehlbedienung den Urzustand wieder abrufen zu koennen. Nach der Bearbeitung kann in der Videothek das alte Bild geloescht und das minierte an seiner Stelle eingetragen werden.

8.2. Erstellung der dynamischen Bildelemente

8.2.1. Wirkungsweise und Funktion

Das Programm zur Dynamik ermöglicht die Strukturierung bzw. Darstellung von dynamischen Informationen in Fließbildern. Zur Darstellung dieser Informationen wurden elementare Bildmodule geschaffen, die den Anschluss einer dynamischen Informationsquelle gestatten und diese Information als grafische Anzeige oder Anzeigeneuerung auf den Fließbild ermöglichen. Diese elementaren Bildmodule sind als Unterprogramme aufzufassen, die von Fall zu Fall ueber einen entsprechend eingerichteten und zugehoerigen Modulaufrufblock (MAB) mit den entsprechenden Parametern versorgt werden. Der MAB enthaelt in einer festgelegten Reihenfolge Plaetze fuer alle Parameter, die fuer die ordnungsgemaesse Funktion des Bildmoduls und Darstellung auf dem Bildschirm notwendig sind. Fuer jedes Fließbild kann eine Kette von MAB's eingerichtet werden, die in ihrer Auswertung mit den zugehoerigen Bildmodulen fuer die Anzeige aller dynamischen Informationen sorgen. Die ON-Line Abarbeitung ruft die zugehoerige Dynamik-Modulkette auf, ermittelt die Anfangadressen der MAB und startet die zugehoerigen Bildmodulprogramme. Diese Arbeitsweise erfolgt seriell und zyklisch. Eine wesentliche Qualitaet dieses Programmpaketes bildet die Aenderbarkeit (Strukturierung) dieser Dynamik-Modulketten auch auf dem Applikationsrechner. Diese Funktion ist wesentlicher Bestandteil neben der Aenderbarkeit des statischen Bildaufbaus.

8.2.2. Speicherfestlegungen und Vereinbarungen

Fuer das komplexe Zusammenwirken der Softwarebestandteile zur statischen und dynamischen Fließbilddarstellung, aber auch im Zusammenwirken mit Applikationsprogrammen des Nutzers zur Uebergabe bestimmter dynamischer Informationen ist im Grundbereich ein Systemblock eingerichtet.

Dieser Systemblock mit der Anfangsadresse 6400H hat folgenden Aufbau:

6400H	[]	Bildanwahldaten werden durch das System benutzt und angelegt
		- Nummer des angewählten Bildes
		- Ebenen-Nummer des angew. Bildes
6405H	[]	\ Anfangsadresse / Dynamik-Block Dynamik-MAB-Anwahldaten werden durch das System benutzt u. angel.
6409H		Steuerfeld fuer Konvertierungen wird durch das System benutzt und angelegt
6414H	[]	Puffer fuer Datenerfassung wird durch das System benutzt und angelegt
641AH	[]	\ 3 Anfangsadressen f. Hinweispufer, \ ANBI-Byte-Tabelle und \ Steuerfeldtabelle Kurzzeitrend werden durch \ Applikationsprogrammierer festgelegt.
641CH		
641EH		

Der Systemblock setzt sich aus Speicherplatzreservierungen zusammen, die durch das Softwaresystem selbstaendig eingerichtet und verwaltet werden und aus Vereinbarungen, die bei der Applikationsprogrammierung festgelegt werden. Durch den Applikationsprogrammierer ist nur der Teil der 3 Anfangsadressen (Blockadressbereich 641AH - 641FH) und die jeweils angewählte Bildnummer (Blockadresse 6404H) vorzugeben. Der Adressbereich von 6400H - 6419H ist freizuhalten.

Systemteilblock der 3 Anfangsadressen:

[]	ANIPU	Anfangsadresse fuer Hinweispufer (z.B.: 6500H)
[]	AANBI	Anfangsadresse der ANBI-Byte-Tabelle (z.B.: 6430H)
[]	ASTKU	Anfangsadresse der Steuerfeldtabelle (z.B.: 6460H) fuer Kurzzeitrend

Die erwachten Bereiche koennen nur im Grundbereich angeordnet werden.

Zentrale Schnittstelle zum Anlegen von Bild-Modulaufruf-bloessen (MAB) sowie zur Abarbeitung dieser MAB fuer eine Anzeige im Fliehbild bildet die MAB-Tabelle. Die Einrichtung dieser Tabelle ist bereits beim statischen Teil beschrieben worden. Unmittelbar den Informationen fuer den statischen Bildaufbau folgen die Informationen fuer den dynamischen Bildaufbau.

Dieser Bereich hat damit folgenden Aufbau:

---	KOPF	Kopfinformationen systembedingt --> 0000H
---	LAENG(1)	Laenge MAB 1
---	MAB(1)	Bereich fuer Bildmodul-MAB MO(x)
:		
---	LAENG(2)	Laenge MAB 2
---	MAB(2)	Bereich fuer Bildmodul-MAB MO(x)
:		
:		
---	LAENG(n)	Laenge MAB n
---	MAB(n)	Bereich fuer Bildmodul-MAB MO(x)
:		
---	ENDE	Endekennzeichen MAB-Teil --> 00H

$$LAENG = \langle \text{Laenge des MAB von MO(x)} \rangle + 1$$

Die MAB-Tabelle entsteht beim Strukturieren der Dynamikmodule. Zusammen mit den statischen Bildinformationen wird die sich ergebende MAB-Tabelle bildbezogen abgespeichert und steht so bei der ON-Line-Abarbeitung wieder zur Verfuegung. In der Reihenfolge, wie die einzelnen MAB in der Tabelle angeordnet sind erfolgt auch die ON-Line-Abarbeitung. Befinden sich keine Dynamikmodule im Bild, so besteht das Bild nur aus statischen Elementen und in der MAB-Tabelle fuer die Dynamikmodule erscheint nur die Kopfinformation KOPF und das Endekennzeichen ENDE, also 3 Byte mit 00H.

Fuer die Moeglichkeit der Darstellung eines sogenannten Kurstrends in Fliehbildern (Modul 8) ist die Festlegung des Kurstrendpuffers (enthaelt die im Trend ermittelten Daten) und des Kurstrendsteuerfeldes (enthaelt die Informationen zum Fuehren des Kurstrends und zum Aufruf) erforderlich.

Die Anfangsadresse der Kurstrend-Steuerfeldtabelle ist in Systemblock unter ASTKU zu vereinbaren.

Die Steuerfeldtabelle hat folgenden Aufbau:

[]	ANZA	Anzahl der Kurztrends
[]	STP(1)	Steuerfeld fuer Trendpuffer 1
[]	:	
[]	STP(2)	Steuerfeld fuer Trendpuffer 2
[]	:	
[]	:	
[]	STP(m)	Steuerfeld fuer Trendpuffer m
[]	:	

Speicherplatzabschaetzung: STP-Tabelle = $m * 6 + 1$ [Byte]

Im Steuerfeld sind fuer den jeweiligen Kurztrendpuffer Informationen zur Anfangsadresse sowie zum aktuellen Puellstand enthalten.

Das Steuerfeld hat folgenden Aufbau:

[]	EBENE	Ebenen-Nr. des Trendpuffers
[]	ADR	Anfangsadresse des Trendpuffers
[]	ANZA	Anzahl der Werte des Trendpuffers
[]	LWERT	Nummer des letzten Wertes
[]	AWERT	Nummer des aktuellen Wertes

Der eigentliche Kurztrendpuffer, der die den Trend repraesentierenden Werte fuehrt, ist ein Speicherbereich, der auf einer Datenebene liegen kann. Im Trendpuffer werden die Werte im rechnerinternen Datenformat als 2-Byte-Festkomma-werte abgelegt. Der Trendpuffer ist fuer die Anzahl von Werten ausgelegt, die in ANZA im Steuerfeld vereinbart sind. Unter EBENE und ADR des Steuerfeldes sind die Datenebene und die Anfangsadresse des Trendpuffers abgelegt. Damit gestaltet sich der Aufbau des Trendpuffers wie folgt:

[]	Platz fuer Wert 1	Trendpuffer fuer x-Werte als
[]	Platz fuer Wert 2	2-Byte-Festkomma-Werte
[]	Platz fuer Wert 3	
[]	:	
[]	Platz fuer Wert x	

Speicherplatzabschaetzung: Trendpuffer = $x * 2$ [Byte]

Die Anzahl der Kurstrendpuffer ist in der Steuerfeldtabelle unter ANZAT abgelegt. Die Puffer fuer die einzelnen Signale koennen unterschiedlich gross sein (entsprechend ANZA). Fuer eine effektive Kurstrendfuehrung und -auswertung befindet sich in LWERT die Nummer des Wertes, bis zu der man bei der letzten Auswertung des Bildmoduls zur Anzeige gekommen ist. Ist zwischen den beiden Bildmodulauswertungen eine Aktualisierung erfolgt, so ist aus AWERT die Nummer des Wertes ersichtlich, bis zu der die Aktualisierung erfolgte. Nach erfolgter Auswertung durch das Bildmodul und einer entsprechenden Anzeige wird die Nummer in AWERT auf LWERT uebernommen. Eine Auswertung des Kurstrends erfolgt durch das Bildmodul MOB nur dann, wenn zwischen LWERT und AWERT eine Differenz besteht. Der Trendpuffer arbeitet als Umlaufspeicher; bei Erreichen der letzten Wertenummer beginnt die Behandlung wieder von Wertenummer 1. Das Fuehren des Kurstrends hat durch geeignete Applikationsprogramme zu erfolgen, die dann die Zelle AWERT entsprechend behandeln und die Zuordnung der zu erfassenden Daten organisieren.

Fuer die Einblendung von Informationen aus Applikationsprogrammen kann die Moeglichkeit des Hinweispuffers genutzt werden (Modul 7). Der Hinweispuffer ist ein Speicherbereich auf dem Text-Informationen abgesichert werden koennen, die durch das Modul MO7 auf Bildschirm ausgegeben werden koennen. Durch Steuerzeichen <'> kann das Fortschreiben des Textes aus dem Hinweispuffer auf der naechsten Bildschirmzeile angewiesen werden und durch Steuerzeichen <"> wird das Textende markiert. Damit ist die Ausgabe beliebig langer Texte auf mehreren Zeilen moeglich. Die Anfangsadresse des Hinweispuffers ist durch den Programmierer im Systemblock unter AHIPU zu vereinbaren.

Grundsatzlicher Aufbau des Hinweispuffers:

	Hinweispuffer	1.Platz fuer SIF-1000 Zeichen
		2.Platz fuer SIF-1000 Zeichen
		:
		:
	Hinweispuferende	Steuerzeichen <">

Die Module mit balkenmaessigen Darstellungen (MO8 und MO9) arbeiten mit dem sogenannten ANBI-Byte. Die entsprechenden ANBI-Bytes, die Einfluss auf den Anzeigeanfang und Anzeigebereich nehmen, sind in der ANBI-Byte-Tabelle angeordnet. Die Anfangsadresse dieser Tabelle ist im Systemblock unter AANBI zu vereinbaren. Durch entsprechend festgelegte ANBI-Byte-Nummern im jeweiligen Modul erfolgt die Zuordnung des ANBI-Bytes aus der Tabelle zur Anzeigegrafik. Ein zusaetzliches Dialogprogramm ermoeglicht durch Aendern der ANBI-Bytes in der ANBI-Byte-Tabelle auch waehrend der ON-Line-Arbeitung, den Anzeigebereich/Anzeigeanfang durch Systemkommunikation zu veraendern. Das Bildmodul MO11 ermoeglicht die entsprechende Beschriftung in der Anzeigegrafik.

Grundsatzlicher Aufbau der ANBI-Byte-Tabelle:

	ANBI-Byte Nummer 1
	ANBI-Byte Nummer 2
:	
	ANBI-Byte Nummer n

Der Zugriff zum ersten ANBI-Byte erfolgt ueber die im Bildmodul strukturierte ANBINR = 1.

Das ANBI-Byte beeinflusst den Anzeigebereich wie folgt:

ANBI entspricht den KONS-bezogenen ANBI-Byte fuer Analogwerte im PGR

ANBI Bit 7,6 --> ANBE	00	100%	\	} Anzeigebereich
	01	50%	/	
	10	25%	/	
	11	12,5%	/	

ANBI Bit 5 bis 0 --> ANZA entspricht nach Multiplikation mit 2 dem hoeherwertigen Teil eines 2-Byte-Festkomma-Wertes, der den Anzeigeanfang als %-Wert repraesentiert, z.B.:

XX00.0000 % .0%
 XX10.0000 % 50%

Die Grundbelegung der ANBI-Byte-Tabelle (z.B. mit 00) muss durch das Applikationsprogramm vorgegeben werden.

8.2.3. Dialogfuehrung

Die Dialogfuehrung zum Anlegen und Aendern von dynamischen Bildmodulketten ist als Teil des Programmpaketes zur Dialogfuehrung fuer die Aenderung des statischen Bildaufbaus zu betrachten. Grundsatzliche Dialoggepflogenheiten sind auch hier uebernommen worden, so dass auf deren Ausfuehrung hier verzichtet wird. Es wird daher nur auf die notwendigen Bedienhandlungen eingegangen.

Ein entsprechendes Menue inerhalb des Dialogpaketes bietet die Moeglichkeit der Aenderung von dynamischen Fliebsbildinformationen an:

Menue-Ebene 1

	0	BILD	1	DYN	2	ZUOR	3	VID	4	CODE	5	ANZ	N	NEUS
--	---	------	---	-----	---	------	---	-----	---	------	---	-----	---	------

Durch Betaetigen von <1> erfolgt die Anwahl und es wird Menue-Ebene 2 erreicht. Die bereits vorhandene Bildmodulkette wird aufgerufen und ein entsprechendes Menue zur Behandlung der MAB angeboten:

Menue-Ebene 2

E	ERWIT	L	LOESCH	P	PARAM	S	SIMUL		
DYNAMIK - MODULKETTE									
001	XXXX	021	XXXX	041	XXXX	061	XXXX	081	XXXX
002		022							
003									
:									
020	XXXX	040	XXXX	060	XXXX	080	XXXX	100	XXXX
SPEICHERBEDARF DYNAMIK: RESERVE:									

Ist eine Dynamik-Modulkette vorhanden, so wird deren Folge auf dem Bildschirm dargestellt. Fuer jedes Modul erscheinen dabei eine dreistellige Platznummer (von 1 bis 100) sowie das Kennwort fuer das entsprechende Modul. Es erscheinen nur so viele Elemente wie auch Module in der Kette vorhanden sind. Existiert noch keine Kette, so bleibt der Bildschirm leer. Im unteren Teil des Bildschirms erscheint eine Aussage zur Speicherbilanz mit bereits vergebenen Plaetzen, sowie noch verfuegbaren Reserven (Differenz zu 3 KByte). In der Menue-Zeile erscheint ein Auswahlangebot. Grundlage fuer dieses Angebot bildet die entsprechende Kettenstruktur wie sie auf dem Bildschirm ersichtlich ist. Es koennen folgende Handlungen vorgenommen werden:

E ERWIT - Erweitern der Modulkette
L LOESCH - Loeschen der Modulkette
P PARAM - Parametrieren der Modulkette
S SIMUL - Simulation der Dynamikmodule

Wird die Taste <E> betaetigt, kann die bereits vorhandene Modulkette erweitert werden. Eine Erweiterung ist nur durch Anfuegen weiterer Dynamikmodule an das Ende der Kette moeglich. Ist keine Dynamik-Modulkette vorhanden, so kann eine Kette ab Platz 001 vollkommen neu eingegeben werden. Nach Betaetigen der Taste erscheint zur Kontrolle der entsprechende Teil des Menues farblich hinterlegt. In der Zeile unterhalb des Menues baut sich ein Dialogfeld auf. Es zeigt auf den ersten freien Platz in der Modulkette. In der Folge wird nun die Eingabe des Kennwortes fuer das gewuenschte Bildmodul erwartet. Nach erfolgter Eingabe wird dieses Modul uebernommen und erscheint in der Darstellung der Dynamik-Modulkette auf dem entsprechenden Platz. Der Dialog kehrt auf das Menue-Niveau zurueck und mit Betaetigen der Taste <E> kann nun ein weiteres Bildmodul an die Kette plaziert werden.

Beispiel: <E> <ENTER> Auswahl der Funktion ERWEITERN
 anschliessend wird auf der Dialogzeile eine
 Eingabe erwartet:
 <I> <T> <X> <ENTER> Eingabe Bildmodul BITX

Wird die Taste <L> betätigt, so koennen beliebige Plaetze
 in der Dynamik-Modulkette geloescht werden. Hierbei ist
 unmittelbar nach L die Platznummer des zu loeschenden
 Dynamikmoduls einzugeben. Nach Eingabe der 3-stelligen
 Platznummer wird das Dynamik-Modul an dieser Stelle in der
 Kette geloescht und alle folgenden Dynamik-Module ruecken
 einen Platz auf, so das wieder eine geschlossene Keytte
 entsteht. Dieses wird sofort in der Darstellung der Dynamik-
 Modulkette ersichtlich. Mittels Eingabe von LA kann die
 gesamte Dynamikkette auf einem mal geloescht werden. Als
 gewisse Sicherheit wird das Gesamtloeschen nur nach zwei-
 maliger Eingabe von LA ausgefuehrt. Nach erfolgter Eingabe
 kehrt der Dialog wieder auf das Menue-Niveau zurueck.

Beispiel: <L> <1> <0> <ENTER> Loeschen Dynamikmodul an
 Platz 10
 <L> <A> <ENTER> Kette wird noch nicht ge-
 geloescht, erst wenn un-
 mittelbar darauf wiederum
 <L> <A> <ENTER> folgt, wird das Loeschen
 ausgefuehrt

Wird die Taste <P> betätigt, so koennen die Dynamik-Module
 der angezeigten Modulkette parametrieret werden. Beim Param-
 etrieren werden alle relevanten Informationen (Parameter) die
 fuer die ordnungsgemaeasse und gewuenschte Anzeigefunktion in
 Verbindung mit dem Bildmodul noetig sind, eingegeben. Grund-
 lage hierzu bildet die vorher durch -E- eingegebene oder
 erweiterte und/oder durch -L- korrigierte Modulkette.

Nach Betätigung der Taste wird zur Kontrolle der entspre-
 chende Teil des Menues farblich hinterlegt und in der Zeile
 unterhalb des Menues erscheint ein Dialogfeld, welches die
 Eingabe der Platznummer des zu parametrierenden Dynamik-
 Moduls erwartet. Nach Eingabe der Platznummer wird der Bild-
 schirmbereich mit der Darstellung der Dynamik-Modulkette
 geloescht und es wird die Parametrierungsliste des entspre-
 chenden Bildmoduls dargestellt. Zur Kontrolle erscheint
 Platznummer und Kennwort des Bildmoduls am Anfang der Para-
 metrierliste. Wird das Bildmodul zum ersten mal param-
 etriert, so befindet sich in der Parameterliste fuer jede
 Position eine Anfangsbelegung. Diese kann nun ueber Dialog
 veraendert werden.

Dazu wird mittels eines Cursors der entsprechende Parameter
 angefahren und es kann nun ein neuer Wert eingegeben werden.
 Sind alle gewuenschten Eintragungen erfolgt, so wird die
 Eingabe abgeschlossen. Dies geschieht durch Betätigen der
 <Enter>-Taste bei einer neuen Cursorposition ohne vorherige
 Eingabe von Zeichen fuer eine Parameterraenderung. Der Dialog
 kehrt dann zum Menue-Niveau zurueck. Die Bedeutung und die
 Darstellung der Parametrierlisten fuer die einzelnen Bild-
 module sind in Abschnitt 3.2.4.4...ausgefuehrt.

Beispiel: <P> <2> <0> <ENTER> Parametrieren des Dynamikmoduls am Platz 20, der Cursor wird auf den ersten Parameter plaziert

Wird Taste <S> betätigt, so ist eine Kontrollanzeige der bisher plazierten Dynamikmodule zusammen mit den statischen Bildelementen moeglich. Hierzu wird das jeweilige statische Bildschirmbild aufgebaut und die Plaetze auf denen die dynamischen Informationen im ON-Line-Betrieb gelangen sind farbig hinterlegt. Mittels <ENTER> kann zum Ausgangs-Menue zurueckgekehrt werden und bei Bedarf entsprechende Korrekturen mittels <P> durchgefuehrt werden.

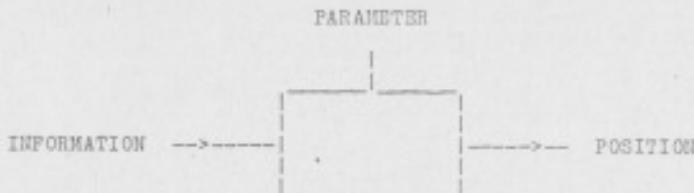
Beispiel: <S> <ENTER> Anzeige des Simulationbildes
 <ENTER> Rueckkehr ins Ausgangsmenue und loeschen der Simulation

Es ist die Besonderheit zu beachten, dass das Modul -MO11- 2 Plaetze mit der gleichen Positionsnummer einnimmt. Die Menue-Ebene 2 kann mittels <PFEIL NACH LINKS ANSCHLAG> verlassen werden. Damit gelangt man wieder in den Menue-Teil fuer den statischen Bildaufbau, mit dem dann die Rueckspeicherung der erzeugten gesamten Fließschildinformation vorzunehmen ist.

8.2.4. Bildmodule

8.2.4.0. Allgemeine Festlegungen und Moduluebersicht

Die Bildmodule sind allgemein gehaltene Programme, die quasi als Unterprogramme die elementaren Anzeigefunktionen im Fließbild ermoeglichen. Die Versorgung dieser Programme mit den konkreten Anzeigeeinformationen erfolgt wie bereits erwahnt ueber Modulaufrufbloecke (MAB), die ja als Dynamik-Modulkette in der MAB-Tabelle abgelegt sind. Das Bildmodul muss im allgemeinen einen Anschluss fuer die Informationsquelle als Eingangsgroesse und einen Anschluss fuer die Plazierung auf dem Bildschirm als Ausgangsgroesse enthalten. Weitere Parameter koennen Einfluss auf die Gestalt auf dem Bildschirm oder zur internen Umrechnung besitzen.



Als Informationsquellen fuer dynamische Grosseen koennen benutzt werden:

- Prozessabbild
- Applikationsrechnerabbild
- Kurztrendpuffer
- Hinweispuuffer

Der Zugriff zum Prozessabbild und zum Applikationsrechnerabbild erfolgt gemass den Festlegungen und auch im Umfang wie sie fuer die Applikationsrechnerprogrammierung definiert sind. Beim Zugriff auf den Kurztrend, auf den Hinweispuuffer und auf die ANBI-Bytes werden die Speicherfestlegungen und der damit verbundene Speicheraufbau vorausgesetzt wie er in dieser Unterlage erlaeutert ist. Aus Sicht der Bildmodule ergeben sich folgende sinnvolle Datentypverschluesselungen:

Typ	Bildmodule (MO)												
	01	02	03	04	05	06	07	08	09	10	11	12	13
0	x	x	x
1	x	x	x	x
2	x	x	.	.	x
3	x	x
4	x	x	x	x	x
5	x	x	.	.	x
6	x	x

Uebersicht der Datentypverschluesselungen:

Code	Bedeutung	Byte	AR-Typ
0	Prozessabbild Statusbyte	1 Byte	Typ 0
1	Prozessabbild Byte	2 Byte	Typ 1
2	Prozessabbild Analog	3 Byte	Typ 2
3	Prozessabbild Zaehlwert	5 Byte	Typ 3
4	AR-Abbild Integer	2 Byte	Typ 0
5	AR-Abbild Festkomma	3 Byte	Typ 1
6	AR-Abbild Gleitkomma	5 Byte	Typ 2

Der Code des Datentyps ist bei der Strukturierung beim Parameter TYP einzutragen.

Der bei den Modulen 01, 02, 03 moegliche bitweise Zugriff wird bei der Strukturierung mittels Bitmaske und Auswahlbyte bzgl. Statusbyte oder Wertbyte ermoeglicht.

Weitere Zugriffsarten der Bildmodule sind:

	01	02	03	04	05	06	07	08	09	10	11	12	13
A	x	x
H	x
K	x
AB	(x)	(x)	.	x	.	.

A = Alarmcode

H = Hinweispuffer

K = Kurztrend

AB = Anbi-Byte-Tabelle

Die Angaben zur Positionierung der gewünschten Information auf dem Bildschirm erfolgt bei der Strukturierung als Zeilennummer und Spaltennummer. Diese beiden Angaben werden immer fuer die linke obere Ecke des Feldes gemacht, welches durch die Anzeigefunktion des Bildmoduls eingenommen wird.

Ueber den Parameter FARBE wird in den einzelnen Dynamik-Modulen Vordergrund- und Hintergrundfarbe fuer die dynamischen Ausgaben festgelegt. Die Farbinformation wird wie folgt verschlüsselt:

D	B	H	H	H	V	V	V
2. Hexazahl				1. Hexazahl			

V - Vordergrundfarbe

H - Hintergrundfarbe

B - Blinken

D - doppelte Zeichen-
breite

} wird nicht
benutzt,
} ist 0

0 - schwarz

1 - rot

2 - gruen

3 - gelb

4 - blau

5 - purpur

6 - cyan

7 - weiss

Funktionen und Parameter der einzelnen Bildmodule sind nachfolgender Uebersicht zu entnehmen:

Uebersicht_Dynamik_Bildmodule

Be- zeichn.	Funktion	Parameter	Nutzungs- beispiel
BMOD (1)	Einblendung eines Bildmoduls in Abhaengigkeit eines Signals (0/1)	. BS-Position . Bildmodulnummer 1/2 . Signalszugriff	offenes/geschlossenes Ventil, Wegblenden geschlossener Rohrleitungen
BITX (2)	Einblenden von Zeichenketten in Abhaengigkeit eines Signals (0/1)	. BS-Position . Farbinformation 1/2 . Zeichenkette 1/2 . Signalszugriff	Anzeige fester Texte in Abhaeng.v.Signal z.B. "VENTIL GESCHLOSSEN"
BIBL (3)	in Abhaengigkeit eines Signals erfolgt das Blinken eines BS-Bereiches	. BS-Position . BS-Bereich . Signalszugriff	Zusatzmarkierung von Anzeigen (Texte, Werte)
BYTX (4)	Einblenden von Textzeilen in Abhaengigkeit eines Signals (8 Bit)	. BS-Position . Steuerbyte f.Anzeige Textlaenge (max.7 Zeichen) . TEXT 0 bis Text 7 . Signalszugr.	Anzeige des Grenzwertbytes oder des Betriebsartenbytes einer KOMS
TXTP (5)	Einblenden eines Textes in Abhaengigkeit des Alarmcodes	. BS-Position . BS-Bereich . Steuerbyte zur Anzeige . Farbinformationen . Text 0 bis 3 f.Alarmcode 0 bis 3	z.B. "WARNUNG" "ALARM"
FELD (6)	Einblenden eines Bildmoduls in Abhaengigkeit des Alarmcodes	. BS-Position . Bildmodulnr. 1 bis 4 . Signalszugr.	Darstellung v.Behaeltern Rohrleitungen unterschiedlicher Farbe
HIPU (7)	Anzeige eines von Anwender gefuellten dynamischen Hinweispuffers	. BS-Position . BS-Bereich . Farbe	spezielle Bedienerhinweise in Abg.v. Prozesszustand

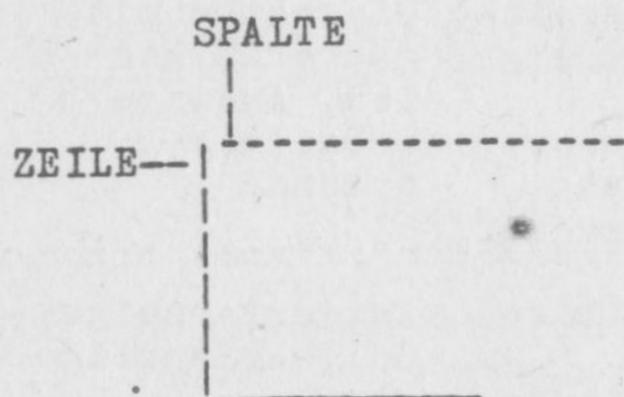
Bezeichnung	Funktion	Parameter	Nutzungsbeispiele
KUTR (8)	Kurztrend	. BS-Position . Wertanzahl . Hoehe . Farbe . Anzeigebereichssteuerung . Trend-Nr.	Trendverlauf v. mehreren Messwerten in einer Darstellung
BALK (9)	Balkendarstellung in Abhaengigkeit eines analogen Signals	. BS-Position . Lage . Balkenbreite . Balkenlaenge . Farbe . Anzeigebereichsteuerung . Signalausgriff	Istwertbalken, Stellwertbalken Diagramme mit mehreren Werten, Anzeige Behaelterstaende
FLKO (10)	numerische Darstellung von Analogsignalen	. BS-Position . Farbe . Dimensionierungskonstant. . Signal (Zugriff)	Istwertanzeige, Sollwertanzeige
ANBI (11)	numerische Darstellung . aktueller Anzeigeanfang und . aktueller Anzeigebereich	. BS-Pos. f. Anfang und Bereich . Farbe . Art Bereichsanzeige . Anzahl Skalentelle . Dim-Konstanten . Zugriff Anbi-Byte	Bereichsbeschriftung bei Kurztrend- oder Balkendarstellung
INTE (12)	numerische Darstellung von 4 Byte Zahlwerten	. BS-Position . Farbe . Signalausgriff	Zahlwertanzeige
ZIFF (13)	numerische Darstellung von . Analogwerten . Gleitkommawerten . Integer (1/2/4 Byte)	. BS-Position . Datenort . Alarmcodeauswertung . Farbe . Dim-Konstant. . Signalausgriff	Anzeige von numerischen Werten aus der BSE und dem AR

8.2.4.1. Einblendung von Bildmodulen - BMOD

Kurzwort: BMOD / M01
 Bedeutung: Bildmodul
 Zugriff: Bit
 MAB-Laenge: 9 Byte

Leistung:

In Abhaengigkeit des Signals wird ein Bildmodul eingeblendet. Ist das Signal 0 so erscheint das Bild BILD1, ist das Signal 1 erscheint das Bild BILD2. Unter BILD1 und BILD2 sind Nummern von Bildern aus der Videothek abgelegt.

Anzeigefeld:

Die Groesse des Anzeigefeldes richtet sich nach dem Bildteil wie er unter der entsprechenden Nummer in der Videothek aufgebaut und abgelegt wurde.

MAB-Aufbau:

Byte 0	MTYP	Modultyp	03H
Byte 1	ZEILE	Zeilenposition	
Byte 2	SPALTE	Spaltenposition	
Byte 3	BILD1	Bildmodulnummer	1
Byte 4	BILD2	Bildmodulnummer	2
Byte 5	MASKE	\	Zugriff
Byte 6	TYP	>	auf
Byte 7/8	NR	/	Informationsquelle

Dialogdarstellung:

mmm BMOD

01	ZEILE	xx	< INT 1...32 >
02	SPALTE	xx	< INT 1...64 >
03	BILD1	xxx	< INT 0...192 >
04	BILD2	xxx	< INT 0...192 >
05	MASKE	xxH	< HEX 0...FFH >
06	S	x	< INT 0, 1 >
07	TYP	xx	< INT 0...6 >
08	NR	xxx	< INT 0...999 >

Bedeutung:

ZEILE	Zeilennummer
SPALTE	Spaltennummer
BILD1	Bildnummer aus Videothek fuer Signal = 0
BILD2	Bildnummer aus Videothek fuer Signal = 1
MASKE	Maske zur Bitauswahl (entsprechendes Bit ist gesetzt)
S	Kennzeichen fuer Bitanwahl aus Status
TYP	Datentypverschlüsselung
NR	Interne Wertnummer

S = 0 Bit wird aus dem Wertbyte ausgewertet
 S = 1 Bit wird aus dem Statusbyte ausgewertet

8.2.4.2. Einblendung einer Textzeile - BITX

Kurzwort: BITX / M02
 Bedeutung: Bit-Text
 Zugriff: Bit
 MAB-Laenge: 26 Byte

Leistung:

In Abhaengigkeit eines Signals wird eine Textzeile eingeblendet. Die Textzeile kann maximal eine Laenge von 8 Zeichen haben. Wird Text benoetigt, der kuerzer als 8 Zeichen ist, so wird ueber LAENGE eine entsprechende Laengenangabe angeben. Nicht benoetigte Zeichen sind mit Leerzeichen auszufuellen.

Ist das Signal 0 wird die Farbinformation aus FARBE1 und ist das Signal 1 aus FARBE2 zur Darstellung genommen. Loeschen von Feldern bei Signal gleich 0 oder 1 erfolgt ueber die zugehoerige Farbinformation (Vordergrundfarbe gleich Hintergrundfarbe).

Anzeigefeld:

```

          SPALTE
          |
          |
ZEILE---|-----|
          |-----|
          |-----|
  
```

Das Anzeigefeld ist einzeilig. Die Laenge richtet sich nach der Angabe in LAENGE, kann aber nicht laenger als 8 Zeichen sein.

MAB-Aufbau:

Byte 0	MTYP	Modultyp 06H
Byte 1	ZEILE	Zeilenposition
Byte 2	SPALTE	Spaltenposition
Byte 3	LAENGE	Textlaenge
Byte 4	FARBE1	Farbe fuer Text 1
Byte 5	FARBE2	Farbe fuer Text 2
Byte 6	TEXT1	Bereich fuer Zeichenkette Text 1
:		
Byte 14	TEXT2	Bereich fuer Zeichenkette Text 2
:		
Byte 22	MASKE \	Zugriff
Byte 23	TYP >	auf
Byte 24/25	NR /	Informationsquelle

Dialogdarstellung:

mmm BITX

01	ZEILE	xx	< INT 1...32 >
02	SPALTE	xx	< INT 1...64 >
03	LAENGE	xx	< INT 1...8 >
04	FARBE1	xxH	< HEX 0...4FH >
05	FARBE2	xxH	< HEX 0...4FH >
06	TEXT1	xxxxxxxx	< TXT 8 SIF-Zeichen >
07	TEXT2	xxxxxxxx	< TXT 8 SIF-Zeichen >
08	MASKE	xxH	< HEX 0...FFH >
09	S	x	< INT 0, 1 >
10	TYP	xx	< INT 0...6 >
11	NR	xxx	< INT 0...999 >

Bedeutung:

ZEILE	Zeilennummer
SPALTE	Spaltennummer
LAENGE	Laenge des Textes
FARBE1	Farbe fuer Signal = 0
FARBE2	Farbe fuer Signal = 1
TEXT1	Text fuer Signal = 0
TEXT2	Text fuer Signal = 1
MASKE	Maske zur Bitauswahl (entsprechendes Bit ist gesetzt)
S	Kennzeichen fuer Bitanwahl aus Status
TYP	Datentypverschlueselung
NR	Interne Wertnummer

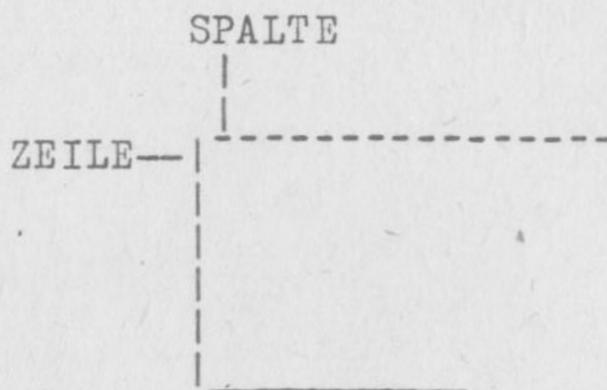
S = 0	Bit wird aus dem Wertbyte ausgewertet
S = 1	Bit wird aus dem Statusbyte ausgewertet

8.2.4.3. Blinken von Bildschirmbereichen - BIBL

Kurzwort: BIBL / M03
 Bedeutung: Bit-Blink
 Zugriff: Bit
 MAB-Laenge: 9 Byte

Leistung:

In Abhaengigkeit eines Signals erfolgt Blinken eines Bildschirmbereiches. Der Bereich hat die Laenge LAENGE und die Hoehe HOEHE. Mittels Parameter kann angegeben werden ob der Bereich bei Signal 0 oder Signal 1 blinkt.

Anzeigefeld:

Die Groesse des Anzeigefeldes wird von den Parametern LAENGE und HOEHE bestimmt.

MAB-Aufbau:

Byte 0	MTYP	Modultyp	09H
Byte 1	ZEILE	Zeilenposition	
Byte 2	SPALTE	Spaltenposition	
Byte 3/7	N	Negationsbit	
Byte 3/6-0	LAENGE	Feldlaenge	
Byte 4	HOEHE	Feldhoehe	
Byte 5	MASKE	Zugriff	
Byte 6	TYP	auf	
Byte 7/8	NR	Informationsquelle	

Dialogdarstellung:

mmm BIBL

01	ZEILE	xx	< INT 1...32 >
02	SPALTE	xx	< INT 1...64 >
03	N	x	< INT 0, 1 >
04	LAENGE	xx	< INT 1...64 >
05	HOEHE	xx	< INT 1...32 >
06	MASKE	xxH	< HEX 0...FFH >
07	S	x	< INT 0, 1 >
08	TYP	xx	< INT 0...6 >
09	NR	xxx	< INT 0...999 >

Bedeutung:

ZEILE	Zeilennummer
SPALTE	Spaltennummer
N	Neagationskennzeichen
LAENGE	Laenge des Blinkfeldes
HOEHE	Hoehe des Blinkfeldes
MASKE	Maske zur Bitauswahl (entsprechendes Bit ist gesetzt)
S	Kennzeichen fuer Bitanwahl aus Status
TYP	Datentypverschluesselung
NR	Interne Wertnummer
N = 0	bei Signal = 0 kein Blinken, bei Signal = 1 Blinken
N = 1	bei Signal = 0 Blinken, bei Signal = 1 kein Blinken
S = 0	Bit wird aus dem Wertbyte ausgewertet
S = 1	Bit wird aus dem Statusbyte ausgewertet

8.2.4.4. Einblendung mehrerer Textzeilen

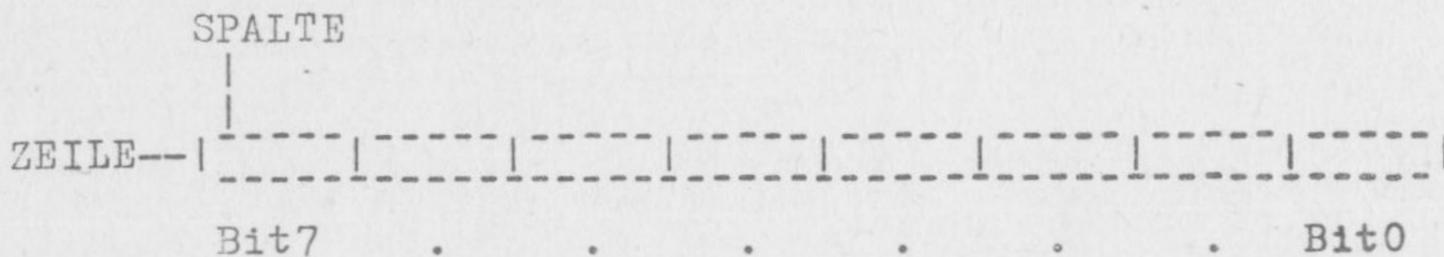
Kurzwort: BYTX / M04
 Bedeutung: Byte-Text
 Zugriff: Byte
 MAB-Laenge: 64 Byte

Leistung:

In Abhaengigkeit der Belegung des Signals werden Textzeilen eingeblendet. Jedes Bit des Signals ist einem Text zugeordnet. Alle Texte haben die Laenge LAENGE. Bit0 bis Bit7 sind den Texten TEXT0 bis TEXT7 zugeordnet. Es ist eine Parallele Anzeige moeglich, wobei dann fuer alle Bits, welche gesetzt sind die entsprechende Textinformation auf dem zugeordneten Platz erscheint. Es besteht die Auswahl zwischen der parallelen Anzeige in einer Zeile oder in einer Spalte (alle Texte nebeneinander oder untereinander). Die andere Anzeigemoeeglichekeit sieht fuer alle 8 Texte nur einen Platz auf dem Bildschirm vor. Zur Anzeige kommt dann, wenn mehrere Bits im Signal gesetzt sind immer der Text vom Bit mit der hoeheren Prioritaet (Bit7 hoechste Prioritaet).

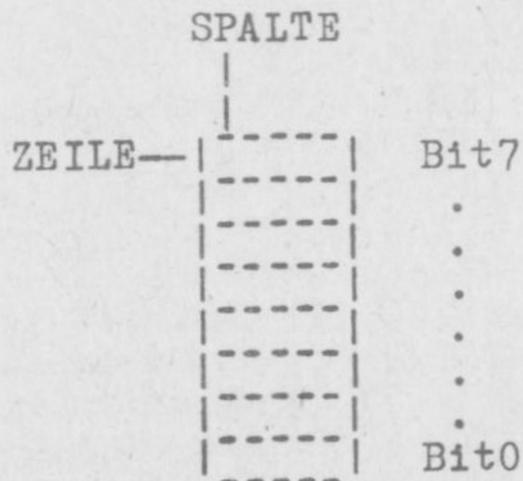
Anzeigefeld:

parallele Anzeige nebeneinander



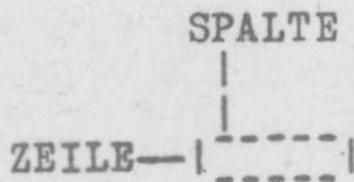
Die Anzeige erfolgt einzeilig. Alle Textinformationen erhalten in der Folge von Bit0 bis Bit7 nebeneinander ihren zugeordneten Platz. Zwischen den Plaetzen wird selbstaendig ein Leerzeichen eingefuegt. Die Laenge dieses Feldes haengt von LAENGE ab, kann jedoch maximal mit hoechstens 7 Zeichen je Text 63 nicht ueberschreiten.

parallele Anzeige untereinander



Die Anzeige erfolgt in 8 Zeilen untereinander. Die Breite des Feldes richtet sich nach LAENGE. Maximal kann das Feld nicht breiter als 7 Zeichen werden.

Platzanzeige



Fuer alle Texte erfolgt die Anzeige auf einem Platz in einzeiliger Form. Die Laenge richtet sich nach dem Parameter LAENGE kann aber jedoch nicht laenger als 7 Zeichen werden.

MAB-Aufbau:

Byte 0	MTYP	Modultyp	OCH
Byte 1	ZEILE	Zeilenposition	
Byte 2	SPALTE	Spaltenposition	
Byte 3/7	B	Betriebsartenbit	
Byte 3/6	P	Prioritaetsbit	
Byte 3/5-0	LAENGE	Textlaenge	
Byte 4	FARBE	Farbe fuer Text	
Byte 5	TEXT0	Zeichenkette fuer Bit 0	
:			
Byte 12	TEXT1	Zeichenkette fuer Bit 1	
:			
Byte 19	TEXT2	Zeichenkette fuer Bit 2	
:			
Byte 26	TEXT3	Zeichenkette fuer Bit 3	
:			
Byte 33	TEXT4	Zeichenkette fuer Bit 4	
:			
Byte 40	TEXT5	Zeichenkette fuer Bit 5	
:			
Byte 47	TEXT6	Zeichenkette fuer Bit 6	
:			
Byte 54	TEXT7	Zeichenkette fuer Bit 7	
:			
Byte 61	TYP	\	Zugriff auf
Byte 62/63	NR	/	Informationsquelle

Dialogdarstellung:

mmn BYTX

01	ZBILB	xx	< INT 1...32 >
02	SPALTE	xx	< INT 1...64 >
03	B	x	< INT 0, 1 >
04	P	x	< INT 0, 1 >
05	LAENGE	xx	< INT 1...7 >
06	FARBE	xxH	< HEX 0...4FH >
07	TEXT0	xxxxxxx	< TXT 7 SIP-Zeichen >
08	TEXT1	xxxxxxx	< TXT 7 SIP-Zeichen >
09	TEXT2	xxxxxxx	< TXT 7 SIP-Zeichen >
10	TEXT3	xxxxxxx	< TXT 7 SIP-Zeichen >
11	TEXT4	xxxxxxx	< TXT 7 SIP-Zeichen >
12	TEXT5	xxxxxxx	< TXT 7 SIP-Zeichen >
13	TEXT6	xxxxxxx	< TXT 7 SIP-Zeichen >
14	TEXT7	xxxxxxx	< TXT 7 SIP-Zeichen >
15	TYP	xx	< INT 0...6 >
16	NR	xxx	< INT 0...999 >

Bedeutung:

ZBILB	Zeilennummer
SPALTE	Spaltennummer
B	Betriebsartenkennzeichen
P	Prioritaetskennzeichen
LAENGE	Laenge eines Textelements
FARBE	Textfarbe
TEXT0	Text fuer Signal-Bit 0
TEXT1	Text fuer Signal-Bit 1
TEXT2	Text fuer Signal-Bit 2
TEXT3	Text fuer Signal-Bit 3
TEXT4	Text fuer Signal-Bit 4
TEXT5	Text fuer Signal-Bit 5
TEXT6	Text fuer Signal-Bit 6
TEXT7	Text fuer Signal-Bit 7
TYP	Datentypverschlueselung
NR	Interne Wertnummer

P = 0 Texte untereinander/ nebeneinander je nach Betriebsart

P = 1 Texte auf Textplatz entsprechend Prioritaet

B = 0 Texte nebeneinander wenn P = 0

B = 1 Texte untereinander wenn P = 0

8.2.4.5. Auswertung Alarmcode mit Texteinblendung - TXTF

Kurzwort: TXTF / H05

Bedeutung: Textfeld

Zugriff: Alarmcode

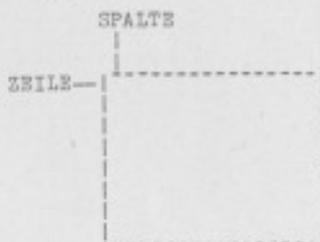
NAB-Laenge: 70 Byte

Leistung:

In Abhaengigkeit des Alarmcodes erfolgt Einblendung eines Textes auf einem Feld mit der Laenge LAENGE der Hoehe HOEHE. Der Text kann Steuerzeichen enthalten, womit der darzustellende Text in mehreren Zeilen stehen kann. Als Steuerzeichen sind ebenfalls Zeichen des SIP-1000-Zeichenvorrats festgelegt. Hierzu sind fuer New-Line das Zeichen <'> (27H) oder fuer Textende das Zeichen <"> (22H) festgelegt. Bei der Parametereingabe fuer diesen Text erscheinen somit die Steuerzeichen ebenfalls mit auf dem Bildschirm. Bei der GN-Line-Darstellung werden diese Zeichen entsprechend ausgewertet und nicht dargestellt.

Die Alarmcodes koennen mittels B0 bis B3 fuer die Anzeigenauswertung ausgewaehlt werden. Wird ein Alarmcode erfasst, der nicht fuer die Anzeige vorgesehen ist, so wird mittels LPARBE das Anzeigefeld geloescht.

Anzeigefeld:



Die Groesse des Anzeigefeldes richtet sich nach dem Text und den Steuerzeichen die dort enthalten sind. Das Feld kann jedoch die Groesse von 16 x 16 Rasterfeldern nicht ueberschreiten.

MAB-Aufbau:

Byte 0	MTYP	Modultyp	OPH
Byte 1	ZEILE	Zeilenposition	
Byte 2	SPALTE	Spaltenposition	
Byte 3/7-4	HOEHE	Feldhoehe	
Byte 3/3-0	LAENGE	Feldlaenge	
Byte 4/4	P	Prioritaetsbit	
Byte 4/3	B3	Betriebsartenbit fuer Alarmcode 3	
Byte 4/2	B2	Betriebsartenbit fuer Alarmcode 2	
Byte 4/1	B1	Betriebsartenbit fuer Alarmcode 1	
Byte 4/0	B0	Betriebsartenbit fuer Alarmcode 0	
Byte 5	PARBE	Farbe fuer Text	
Byte 6	LPARBE	Loeschfarbe	
Byte 7	TEXT0	Zeichenkette fuer Text	ALARMCODE 0
:			
Byte 22	TEXT1	Zeichenkette fuer Text	ALARMCODE 1
:			
Byte 37	TEXT2	Zeichenkette fuer Text	ALARMCODE 2
:			
Byte 52	TEXT3	Zeichenkette fuer Text	ALARMCODE 3
:			
Byte 67	TYP	\	Zugriff auf
Byte 68/69	NR	/	Informationsquelle

Dialogdarstellung:

mmn TYP

01	ZEILE	xx	< INT 1...32 >
02	SPALTE	xx	< INT 1...64 >
03	LAENGE	xx	< INT 1...16 >
04	HOEHE	xx	< INT 1...16 >
05	P	x	< INT 0, 1 >
06	B0	x	< INT 0, 1 >
07	B1	i	< INT 0, 1 >
08	B2	x	< INT 0, 1 >
09	B3	x	< INT 0, 1 >
10	PARBE	xxH	< HEX 0...4FH >
11	LPARBE	xxH	< HEX 0...4FH >
12	TEXT0	XXXXXXXXXXXXXXXXXX	< TXT 15 SIP-Zeichen >
13	TEXT1	XXXXXXXXXXXXXXXXXX	< TXT 15 SIP-Zeichen >
14	TEXT2	XXXXXXXXXXXXXXXXXX	< TXT 15 SIP-Zeichen >
15	TEXT3	XXXXXXXXXXXXXXXXXX	< TXT 15 SIP-Zeichen >
16	TYP	xx	< INT 0...6 >
17	NR	xxx	< INT 0...999 >

Bedeutung:

ZEILE	Zeilennummer
SPALTE	Spaltennummer
LAENGE	Feldlaenge
HOEHE	Feldhoehe
P	Prioritaetskennzeichen
B0	Betriebsart fuer Alarmcode 0
B1	Betriebsart fuer Alarmcode 1
B2	Betriebsart fuer Alarmcode 2
B3	Betriebsart fuer Alarmcode 3
FARBE	Textfarbe
LFARBE	Loeschfarbe wenn B(n)=0
TEXT0	Text fuer Alarmcode 0
TEXT1	Text fuer Alarmcode 1
TEXT2	Text fuer Alarmcode 2
TEXT3	Text fuer Alarmcode 3
TYP	Datentypverschlueselung
NR	Interne Wertnummer

- P = 0 Darstellung des Textes mit Farbe FARBE
 P = 1 Darstellung des Feldhintergrunds in der Prioritaetsfarbe und des Textes in der Farbe FARBE

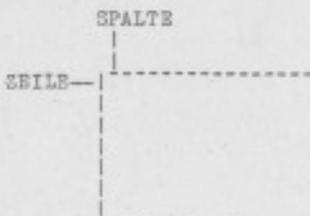
- B(n) = 0 Fuer den entspr. Alarmcode erscheint bei Alarm kein Text. Das Feld wird mit LPARBE geloescht.
 B(n) = 1 Fuer den entspr. Alarmcode erscheint bei Alarm der Text.

3.2.4.6. Bildeinblendung in Abhaengigkeit vom Alarmcode-FELD

Kurzwort: FELD / M06
 Bedeutung: Bildfeld
 Zugriff: Alarmcode
 MAB-Laenge: 10 Byte

Leistung:

In Abhaengigkeit des Alarmcodes erfolgt Einblendung eines Bildes der Videothek. Bei Alarmcode 0 erscheint Bild BILDO usw. bis Alarmcode 3

Anzeigefeld:

Die Groesse des Anzeigefeldes richtet sich nach dem Bildteil, wie er unter der entsprechenden Nummer in der Videothek aufgebaut und abgelegt wurde.

MAB-Aufbau:

Byte 0	HTYP	Modultyp 12H
Byte 1	ZEILE	Zeilenposition
Byte 2	SPALTE	Spaltenposition
Byte 3	BILDO	Bildmodulnummer fuer Alarmcode 0
Byte 4	BILD1	Bildmodulnummer fuer Alarmcode 1
Byte 5	BILD2	Bildmodulnummer fuer Alarmcode 2
Byte 6	BILD3	Bildmodulnummer fuer Alarmcode 3
Byte 7	TYP	Zugriff auf
Byte 8/9	NR	Informationsquelle

Dialogdarstellung:

mmn FELD

01	ZEILE	xx	< INT 1...32 >
02	SPALTE	xx	< INT 1...64 >
03	BILDO	xxx	< INT 0...192 >
04	BILD1	xxx	< INT 0...192 >
05	BILD2	xxx	< INT 0...192 >
06	BILD3	xxx	< INT 0...192 >
07	TYP	xx	< INT 0...6 >
08	NR	xxx	< INT 0...999 >

Bedeutung:

ZEILE	Zeilennummer
SPALTE	Spaltennummer
BILDO	Bildnummer aus Videothek fuer Alarmcode 0
BILD1	Bildnummer aus Videothek fuer Alarmcode 1
BILD2	Bildnummer aus Videothek fuer Alarmcode 2
BILD3	Bildnummer aus Videothek fuer Alarmcode 3
TYP	Datentypverschlüsselung
NR	Interne Wertnummer

MAB-Aufbau:

Byte 0	HTYP	Modultyp 12H
Byte 1	ZEILE	Zeilenposition
Byte 2	SPALTE	Spaltenposition
Byte 3	BILDO	Bildmodulnummer fuer Alarmcode 0
Byte 4	BILD1	Bildmodulnummer fuer Alarmcode 1
Byte 5	BILD2	Bildmodulnummer fuer Alarmcode 2
Byte 6	BILD3	Bildmodulnummer fuer Alarmcode 3
Byte 7	TYP	Zugriff auf
Byte 8/9	NR	Informationsquelle

Dialogdarstellung:

mmn FELD

01	ZEILE	xx	< INT 1...32 >
02	SPALTE	xx	< INT 1...64 >
03	BILDO	xxx	< INT 0...192 >
04	BILD1	xxx	< INT 0...192 >
05	BILD2	xxx	< INT 0...192 >
06	BILD3	xxx	< INT 0...192 >
07	TYP	xx	< INT 0...6 >
08	NR	xxx	< INT 0...999 >

Bedeutung:

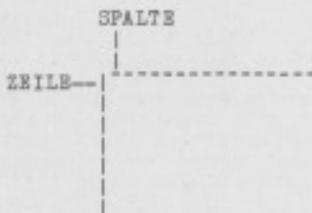
ZEILE	Zeilennummer
SPALTE	Spaltennummer
BILDO	Bildnummer aus Videothek fuer Alarmcode 0
BILD1	Bildnummer aus Videothek fuer Alarmcode 1
BILD2	Bildnummer aus Videothek fuer Alarmcode 2
BILD3	Bildnummer aus Videothek fuer Alarmcode 3
TYP	Datentypverschlüsselung
NR	Interne Wertnummer

8.2.4.7. Textausgabe von Verarbeitungsprogrammen - HIPU

Kurzwort: HIPU / M07
 Bedeutung: Hinweispuffer
 Zugriff: Hinweispuffer
 MAB-Laenge: 6 Byte

Leistung:

Verarbeitungsprogrammspezifische Textinformationen koennen an zentraler Stelle im sogenannten Hinweispuffer abgelegt werden. Diese Textinformation wird auf einem Feld mit der Laenge LAENGE und der Hoehe HOEHE ausgegeben. Die Textinformation kann Steuerzeichen beinhalten, so dass sich der auszugebende Text auch ueber mehrere Zeilen erstrecken kann. Fuer die Steuerzeichen gelten die gleichen Festlegungen wie bei M05 (<'> fuer Zeilenende und <"> fuer Textende).

Anzeigefeld:

Die Groesse des Anzeigefeldes richtet sich nach dem auszugebenden Text aus dem Hinweispuffer. Dabei sind die Steuerzeichen zu beruecksichtigen fuer eine mehrzeilige Ausgabe. Das Anzeigefeld kann als Maximum die gesamte Bildschirmgroesse einnehmen.

MAB-Aufbau:

Byte 0	MTYP	Modultyp 15H
Byte 1	ZEILE	Zeilenposition
Byte 2	SPALTE	Spaltenposition
Byte 3	LAENGE	Feldlaenge
Byte 4	HOEHE	Feldhoehe
Byte 5	FARBE	Farbe

Dialogdarstellung:

mm HIPU

01	ZEILE	xx	< INT 1...32 >
02	SPALTE	xx	< INT 1...64 >
03	LAENGE	xx	< INT 1...64 >
04	HOEHE	xx	< INT 1...32 >
05	FARBE	xxH	< HEX 0...4PH >

Bedeutung:

ZEILE	Zeilennummer
SPALTE	Spaltennummer
LAENGE	Feldlaenge
HOEHE	Feldhoehe

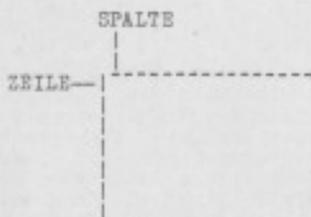
8.2.4.8. Trenddarstellung von analogen Groessen - KUTR

Kurswort: KUTR / NO8
 Bedeutung: Kurstrend
 Zugriff: Trend
 MAB-Laenge: 8 Byte

Leistung:

Dieses Modul gestattet die Ausgabe einer kompakten Trenddarstellung auf dem Bildschirm. Es erfolgt eine Anzeige mit der Werteanzahl ANZAHL (entsprechende Zahl von senkrechten Balken). Die Balkenhoehe bei Endwert ist HOEHE. Die Darstellung erfolgt in der Farbe FARBE.

Das Koordinatensystem ist bei Bedarf durch statische Bildelemente zu realisieren.

Anzeigefeld:

Die Groesse des Anzeigefeldes richtet sich nach dem Umfang des auszugehenden Kurstrends und nimmt genau die Flaeche von ANZAHL x HOEHE ein

MAB-Aufbau:

Byte 0	MTYP	Modultyp 16H
Byte 1	ZEILE	Zeilenposition
Byte 2	SPALTE	Spaltenposition
Byte 3	ANZAHL	Werteanzahl
Byte 4	HOEHE	Endwert
Byte 5	FARBE	Farbe
Byte 6	ANBINR	ANBI-Byte-Nummer
Byte 7	NR	Zugriff auf Informationsquelle (Trendpuffer)

Dialogdarstellung:

mm KUTR

01	ZEILE	xx	< INT 1...32 >
02	SPALTE	xx	< INT 1...64 >
03	ANZAHL	xx	< INT 1...64 >
04	HOEHE	xx	< INT 1...32 >
05	FARBE	xxxH	< HEX 0...4FH >
06	ANBINR	xxx	< INT 1...255 >
07	NR	xxx	< INT 1...255 >

Bedeutung:

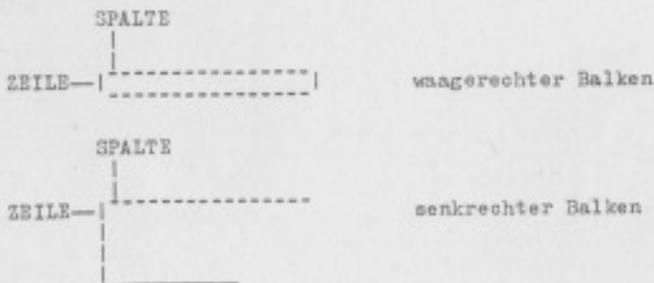
ZEILE	Zeilennummer
SPALTE	Spaltennummer
ANZAHL	Anzahl der Trendbalken (Feldlaenge)
HOEHE	Balkenhoehhe (Feldhoehe)
FAERBE	Farbe
ANBINR	Nummer des ANBI-Bytes aus der ANBI-Byte-Tabelle
NR	Trendpuffernummer

8.2.4.9. Darstellung von waagerechten und senkrechten Balken analoger Groessen - BALK

Kurzwort: BALK / M09
 Bedeutung: Balken
 Zugriff: Festkomma
 MAB-Laenge: 10 Byte

Leistung:

In Abhaengigkeit eines analogen Signals erfolgt die Darstellung als Balken. Der Balken kann waagerecht oder senkrecht dargestellt werden. In der senkrechten Darstellung kann der Balken eine beliebige Breite aufweisen.

Anzeigefeld:

Die Groesse des Anzeigefeldes richtet sich nach Art und Groesse des darzustellenden Balkens. Beim waagerechten Balken erfolgt eine einzeilige Darstellung, wobei die Laenge des Feldes durch den Parameter LAENGE bestimmt wird. Bei der senkrechten Darstellung kann der Balken unterschiedliche Breiten haben. Die Feldgroesse wird in der Breite durch BREITE bestimmt und in der Hoehe durch LAENGE.

HAB-Aufbau:

Byte 0	MTYP	Modultyp 1BH
Byte 1	ZEILE	Zeilenposition
Byte 2	SPALTE	Spaltenposition
Byte 3/7	S	Senkrecht-Bit
Byte 3/6-0	BREITE	Balkenbreite
Byte 4	LAENGE	Balkenlaenge
Byte 5	FARBE	Farbe
Byte 6	ANBINR	ANBI-Byte-Nummer
Byte 7	TYP \	Zugriff auf
Byte 8/9	NR /	Informationsquelle

Dialogdarstellung:

==== BALK

01	ZEILE	xx	< INT 1...32 >
02	SPALTE	xx	< INT 1...64 >
03	S	x	< INT 0, 1 >
04	BREITE	xx	< INT 1...64 >
05	LAENGE	xx	< INT 1...32 >
06	FARBE	xxH	< HEX 0...4FH >
07	ANBINR	xxx	< INT 1...255 >
08	TYP	xx	< INT 0...6 >
09	NR	xxx	< INT 0...999 >

Bedeutung:

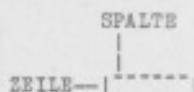
ZEILE	Zeilennummer
SPALTE	Spaltennummer
S	Senkrechtkennzeichen
BREITE	Balkenbreite bei senkrechter Darstellung
LAENGE	Balkenlaenge
FARBE	Farbe
ANBINR	Nummer des ANBI-Bytes in der ANBI-Byte-Tabelle
TYP	Datentypverschlueselung
NR	Interne Wertnummer
S = 0	Waagerechter Balken
S = 1	Senkrechter Balken

9.2.4.10. Zahlendarstellung von Analogsignalen - PLKO

Kurswort:	PLKO / H010
Bedeutung:	Flieas komma / Prozessgroesse
Zugriff:	Festkomma
HAB-Laenge:	12 Byte

Leistung:

Dieses Modul realisiert numerische Darstellungen von Analogsignalen. Die Darstellung erfolgt auf einer Zeile mit einer Breite von 6 Zeichen (1 Vorzeichen, 5 Wertzeichen, wobei der Dezimalpunkt eines der 5 Wertzeichen ist).

Anzeigefeld:

Die Anzeige erfolgt einseitig in einer Laenge von 6 Rasterplaetzen.

MAR-Aufbau:

Byte 0	NTYP	Modultyp 188
Byte 1	ZBILZ	Zeilenposition
Byte 2	SPALTE	Spaltenposition
Byte 3	FARBE	Farbe
Byte 4	DK1	Dimensionierungskonstante K1
:		
Byte 6	DK2	Dimensionierungskonstante K2
:		
Byte 8	E	Exponent
Byte 9	TYP	Zugriff auf
Byte 10/11	NR	Informationsquelle

Dialogdarstellung:

msm PLKO

01	ZBILZ	xx	< INT 1...32 >
02	SPALTE	xx	< INT 1...64 >
03	FARBE	xxH	< HEX 0...4PH >
04	K1	-.xxxx	< PLK -1...+1 >
05	K2	-.xxxx	< PLK -1...+1 >
06	E	x	< INT 1...5 >
07	TYP	xx	< INT 0...6 >
08	NR	xxx	< INT 0...999 >

Bedeutung:

ZBILZ	Zeilennummer
SPALTE	Spaltennummer
FARBE	Farbe
K1	Dimensionierungskonstante 1
K2	Dimensionierungskonstante 2
E	Zehnerexponent
TYP	Datentypverschlüsselung
NR	Interne Wertnummer

8.2.4.11. Modul zur Darstellung des aktuellen Anzeigeanfangs und -Bereichs bei Kurzzeittrend und Balkendarstellung

Kurzwort: ANBI/NO11
 Bedeutung: Anzeigeanfang/-Bereich
 Zugriff: ANBI-Byte
 MAB-Laenge: 14 Byte

Leistung:

Das Modul realisiert die numerische Darstellung des Anfangswertes und des Bereiches der Kurstrend- und Balkendarstellung. Die Darstellung erfolgt auf 2 unabhangig voneinander wahlbaren Bildschirmpositionen fuer den Anzeigeanfang (6 Zeichen/Prozessgroesse) und den Anzeigebereich (6 Zeichen/Prozessgroesse bzw. 3 Zeichen/Prozentwert). Bei der Bereichsanzeige kann man zwischen 3 Typen wahlen:

- Prozentwert
- Absolutwert
- Absolutwert pro Skalenteil

Anzeigefeld:

- Anzeigeanfang bzw. Bereich (Absolutwert)

```

      Spalte
      |
Zeile ---|-----| (6 Rasterplaetze einseitig)
  
```

- Bereich (Prozentwert)

```

      Spalte
      |
Zeile ---|---| (3 Rasterplaetze einseitig)
  
```

MAB-Aufbau:

Byte 0	MTYP	Modultyp 21H	
Byte 1	Z ANP	Zeilenposition	\ Anfang
Byte 2	SP ANP	Spaltenposition	\
Byte 3	Z BER	Zeilenposition	\ Bereich
Byte 4	SP BER	Spaltenposition	/
Byte 5	PARBE	Farbe	
Byte 6	ANZTYP	Anzeigetyp	
Byte 7	SKALE	Anzahl Skalenteile	
Byte 8	DK1	Dimensionierungskonstante 1	
Byte 10	DK2	Dimensionierungskonstante 2	
Byte 12	E	Exponent	
Byte 13	ANBINR	ANBI-Byte-Nummer	

Dialogdarstellung:

mm ANBI

01	Z	ANP	xx	<INT 1...32>
02	SP	ANP	xx	<INT 1...64>
03	Z	BER	xx	<INT 1...32>
04	SP	BER	xx	<INT 1...64>
05	FARBE		xxH	<HEX 0...4FH>
06	ANZTYP		x	<INT 0...2>
07	SKALE		xx	<INT 0...32>
08	DK1		-.xxxx	<PLK -1...+1>
09	DK2		-.xxxx	<PLK -1...+1>
10	E		x	<INT 0...5>
11	ANBINR		xxx	<INT 1...255>

Bedeutung:

Z ANP	Zeilennummer Anzeigeanfang
SP ANP	Spaltennummer Anzeigeanfang
Z BER	Zeilennummer Anzeigebereich
SP BER	Spaltennummer Anzeigebereich
FARBE	Farbe
ANZTYP	Anzeigetyp 0: Prozentwert 1: Absolutwert 2: Absolutwert pro Skalenteil
SKALE	Anzahl Skalenteile fuer KUTR bzw. BALK
DK1	Dimensionierungskonstante 1 (MB-Anfang)
DK2	Dimensionierungskonstante 2 (Messbereich)
E	Zehnerexponent
ANBINR	Nr. des ANBI-Bytes in der ANBI-Byte-Tabelle

8.2.4.12. Zahlendarstellung von Integerwerten - INTE

Kurzwort: INTE / MD12
 Bedeutung: Zahlwert/Integer
 Zugriff: Integer
 MAB-Laenge: 7 Byte

Leistung:

Dieses Modul realisiert numerische Darstellungen von Zahlwerten (4-Byte-Integerwerte). Die Darstellung erfolgt in einer Zeile mit einer Breite von 9 Zeichen.

Anzeigefeld:

```

      SPALTE
      |
ZEILE--|-----|
  
```

Die Anzeige erfolgt einzeilig in einer Laenge von 9 Rasterplaetzen.

MAB-Aufbau:

Byte 0	MTYP	Modultyp 24H
Byte 1	ZBIL	Zeilenposition
Byte 2	SPALTE	Spaltenposition
Byte 3	PARBE	Farbe
Byte 4	TYP	Zugriff auf
Byte 5/6	NR	Informationsquelle

Dialogdarstellung:

mm INTE

01	ZBIL	xx	< INT 1...32 >
02	SPALTE	xx	< INT 1...64 >
03	PARBE	xxH	< HEX 0...4FH >
04	TYP	xx	< INT 0...6 >
05	NR	xxx	< INT 0...999 >

Bedeutung:

ZBIL	Zeilennummer
SPALTE	Spaltennummer
PARBE	Farbe
TYP	Datentypverschlüsselung
NR	Interne Wertnummer

8.2.4.13. Zahlendarstellung von Prozessgrößen, Gleitkommazahlen und Integerzahlen - ZIFF

Kurzwort: ZIFF/M013
 Bedeutung: Ziffern
 Zugriff: Prozessgröße, Gleitkomma, Integer
 MAB-Länge: 14 Byte

Leistung:

Dieses Modul realisiert numerische Darstellungen von Signalen, die rechnerintern als Prozessgröße (Analogsignal und zugehörige Dimensionierungskonstanten), als Gleitkommazahl (AR-Gleitkommatformat) oder als Integerzahl (1-, 2- und 4-Byte) vorliegen können. Die Darstellung erfolgt auf einer Zeile. Die Breite des Platzes ist von der jeweiligen Datenart abhängig. Die Darstellung erfolgt in der Farbe PARBE. Bei Bedarf kann der Hintergrund in der Farbe der Alarm-Prioritätsfarben dargestellt werden.

Anzeigefeld:

```

      SPALTE
      |
ZEILE--|-----|

```

Die Anzeige erfolgt einseitig in einer Laenge, die vom gewählten Datenformat abhaengig ist.

- Prozessadresse 6 Rasterplaetze
- Gleitkomma abhaengig von Gleitkommaformat
- Integer 1-Byte 4 Rasterplaetze
- Integer 2-Byte 6 Rasterplaetze
- Integer 4-Byte 9 Rasterplaetze

MAB-Aufbau:

Byte 0	MTYP	Modultyp 1EH	
Byte 1	ZEILE	Zeilenposition	
Byte 2	SPALTE	Spaltenposition	
Byte 3	DART	Datenart	
Byte 4/3	B3	Betriebsartenbit fuer Alarmcode	3
Byte 4/2	B2	Betriebsartenbit fuer Alarmcode	2
Byte 4/1	B1	Betriebsartenbit fuer Alarmcode	1
Byte 4/0	B0	Betriebsartenbit fuer Alarmcode	0
Byte 5	FARBE	Farbe	
Byte 6	DK1	Dimensionierungskonstante K1	
:			
Byte 8	DK2	Dimensionierungskonstante K2	
:			
Byte 10	E	Exponent bei PG / Datenart bei GK	
Byte 11	TYP	Zugriff auf	
Byte 12/13	NR	Informationsquelle	

Dialogdarstellung:

01	ZEILE	xx	< INT 1...32 >
02	SPALTE	xx	< INT 1...64 >
03	DART	x	< INT 1...5 >
04	B0	x	< INT 0, 1 >
05	B1	x	< INT 0, 1 >
06	B2	x	< INT 0, 1 >
07	B3	x	< INT 0, 1 >
08	FARBE	xxH	< HEX 0...4FH >
09	K1	-.xxxx	< FLK -1...+1 >
10	K2	-.xxxx	< FLK -1...+1 >
11	E	x	< HEX 0...FFH >
12	TYP	xx	< INT 0...6 >
13	NR	xxx	< INT 0...999 >

Bedeutung:

ZEILE	Zeilennummer
SPALTE	Spaltennummer
DART	Datenart
B0	Betriebsart fuer Alarmcode 0
B1	Betriebsart fuer Alarmcode 1
B2	Betriebsart fuer Alarmcode 2
B3	Betriebsart fuer Alarmcode 3
FARBE	Farbe
K1	Dimensionierungskonstante 1
K2	Dimensionierungskonstante 2
E	Zehnerexponent / GK-Datenart
TYP	Datentypverschluesselung
NR	Interne Wertnummer

DART = 1	Prozessgrosse
DART = 2	Gleitkomma
DART = 3	Integer 1-Byte
DART = 4	Integer 2-Byte
DART = 5	Integer 4-Byte

bei DART = 2 Gleitkomma besteht die Moeglichkeit ueber den Parameter E verschiedene Gleitkommadarstellungen zu wahlen, wobei der Parameter in 2 Halbbyte geteilt ist (-EH- Bit4 bis Bit7 und -EL- Bit0 bis Bit3)

EH = 0	Exponentendarstellung, der Rasterplatzbedarf ergibt sich aus Rasterplaetze = EH + 6
EH	Ansonsten gibt EH die Anzahl der Ziffern in der Dezimaldarstellung an, der Rasterplatzbedarf ergibt sich aus Rasterplaetze = EH + 2
EL	Gibt die Anzahl der Ziffern nach dem Komma an, gleichermaßen fuer die Exponent- als auch fuer die Dezimaldarstellung
B(n) = 0	Fuer den entspr. Alarmcode wird der Wert in der Farbe FARBE dargestellt
B(n) = 1	Fuer den entspr. Alarmcode erscheint als Hintergrund die Prioritaetsfarbe. Der Wert selber wird in der Farbe FARBE dargestellt

B.3. Bedienhandlungen zur Aktivierung der Strukturierung auf dem AR

- Um das Bildstrukturierprogramm "PIC25" zusätzlich zu laden, ist die max. VAP-Nr. zu kontrollieren und ggf. zu erhöhen.

Bedienfolge: J 1 ----> 3 ----> VAP-Nr. (z.B.9) eintragen!

- Eintragen der Startadresse fuer "PIC25" auf eine freie Position in der VAP-Steuerliste, naemlich

. Datenebene ED 2
 . Programmebene EP wahlweise, z.B. 3
 . Startadresse ADR 70E0

Bedienfolge: J 1 ----> 5 ----> INS MODE VAP-Nr. (z.B.9)
 ----> ED, EP, ADR eintragen!

- Einlesen von "PIC25" (Dateityp P) auf die entsprechende Programmebene (z.B. 33)

Bedienfolge: entsprechend der AR-Beschreibung zur Bedienung KMSE bzw. Diskettenlaufwerk

Anschliessend erscheint im Grundmenue hinter der gewaehlten VAP-Nr. der Programmname "BILD-STRUKTURIERUNG".

- Start des Bildstrukturierprogramms "PIC25" im OFF-Line-Betrieb mit der Bedienfolge: J VAP-Nr.!

Es erscheint anschliessend das Menue zur Bilderstrukturierung in der obersten BS-Zeile. Die folgenden Bedienhandlungen sind entspr. Abschn. 8.1. (Stat. Bilder) bzw. 8.2. (dynamische Bildelemente) auszufuehren. Die im Rahmen der Statik strukturierten Module sind innerhalb der Menuefunktion 3 (VID) mit dem Kommando "IM Modul-Nr." in der Videothek zu sichern und die strukturierten Bilder mit dem Kommando "IP Bild-Nr." Falls bereits aeltere Versionen in der Videothek stehen, sind diese vorher mit "DM Modul-Nr." bzw. "DP Bild-Nr." zu loeschen. Die strukturierten Dynamik-Ketten sind ebenfalls mit "IP Bild-Nr." in der Videothek zu sichern, nachdem eine ggf. aeltere Version mit "DP Bild-Nr." geloescht wurde.

Die strukturierten Bilder werden auf der 2. Datenebene (und ggf. den folgenden Datenebenen) abgelegt, d. h. auf Adresse 1000H... Von dort aus koennen die generierten Bilder zur Sicherung auf Band bzw. Diskette ausgelagert werden.

Das Bildstrukturierprogramm "PIC25" ist EPROM-faehig und kann auch als EPROM-Version erstellt und gesteckt werden (genutzter Adressbereich 7000 - 9FFFH), so dass ein Laden von Band/Diskette entfaellt.

8.4. Abarbeitung der erstellten Bilder unter dem Echtzeit- betriebsystem des AR

8.4.1. Prinzipielle Einordnung in die ON-LINE-Programmabar- beitung

- Das eigentliche ON-LINE-Verarbeitungsprogramm fuer Anlagenbilder (Bildinterpreter) ist Bestandteil des Anwenderunterstuetzungsprogrammsystems "APIC". Es ist EPROM-faehig und nutzt den Adressbereich 7000 - 7D63H in der Programmebene 2. Will man es ausserhalb des Gesamtprogrammsystems fuer die ON-LINE-Abarbeitung nutzen, sind folgende Vereinbarungen in der AR-Systemstrukturierung vorzunehmen:

- . In die LISTE PRIO 1 ist ausser den VAP-Nr'n 0-3 noch die VAP-Nr. fuer Anlagenbilder (z.B.7) aufzunehmen. (Die VAP-Nr'n 4, 5, 6, 8 des Gesamtprogrammsystems sind ggf. zu loeschen.)

Bedienfolge: J 1 ----> 0 ----> INS LINE lfd. Nr. (z.B.4)
----> INS MODE lfd. Nr. (z.B.4) ----> z.B. 7
eintragen!

- . In den SYSTEMDATEN ist NR-OFF = 0 zu vereinbaren.

Bedienfolge: J 1 ----> 3 ----> ----> 0 eintragen!

- . In der ANMELDELISTE ist die VAP-Nr. fuer die Anlagenbilder (z.B.7) mit Neustart und zyklischem Restart zu vereinbaren.

Bedienfolge: J 1 ----> 4 ----> INS LINE lfd. Nr. (z.B.0)
----> INS MODE lfd. Nr. (z.B.0) ---->
Eintragen: VAP = z. B. 7,
STB1 = 80H, STB2 = 20H, TK = 2, ZAEHL = 2,
TK = 2, ZAEHL = 2!

- . In der VAP-STEUERLISTE ist fuer die VAP-Nr. fuer Anlagenbilder (z.B.7) die Startadresse einzutragen, naemlich

Datenebene ED	2
Programmebene EP	2
Startadresse ADR	7000

Bedienfolge: J 1 ---->5 ----> INS MODE VAP-NR. (z.B.7)
----> ED, EP, ADR eintragen!

- Bei ON-LINE-Abarbeitung ausserhalb des Anwenderunterstuetzungsprogramm-systems sind entsprechend den Hinweisen in Abschn. 8.2.3.
 - . die gewuenschte (strukturierte) Bild-Nr. (01, 02 ...) auf die Systemblockadresse 6401H einzutragen,
 - . die Anfangsadressen des Hinweis-puffers, der ANBI-Byte-Tabelle, der Steuerfeldtabelle zum Kurzzeittrend auf die Systemblockadressen 641AH ... einzutragen (z.B. die Adressen 6500H, 64E0H, 6460H),
 - . die ANBI-Byte-Tabelle bei Verwendung der Dynamik-Module -M08-, -M09-, -M011- mit einer Grundbelegung zu fuellen (z.B.00H),
 - . bei Verwendung des Moduls -M08- die Steuerfeldtabelle zum Kurzzeittrend zu fuellen. (Der Trendpuffer kann zu Testzwecken mit geeigneten Testwerten beschrieben werden. Fuer eine reale Kurzzeitrenddarstellung ist ein gesondertes Datenbereitstellungsprogramm erforderlich!)
- Die gemass vorangegangenen Abschnitten strukturierten Anlagenbilder muessen fuer die ON-LINE-Abarbeitung auf der 2. Datenebene ... stehen.
- Nach Schaffung der genannten Voraussetzungen kann die ON-LINE-Umschaltung des AR erfolgen und damit verbunden der automatische Programmstart "Anlagenbilder" mit zyklischer Aktualisierung der dynamischen Bildinformationen (entsprechend Anmelde-liste alle 2 sek).

Bedienfolge: G ENTER!

8.4.2. Notwendige und moegliche Zusatzfunktionen fuer die ON-LINE-Programmabarbeitung (Anwenderunterstuetzungsprogramm-system)

8.4.2.1. Grundlagen

Die in Abschn. 8.4.1. dargestellte Verfahrensweise ist zwar handhabbar, jedoch ist sie von der Vorbereitung her relativ aufwendig, und es fehlt ein spezifischer Bedienungskomfort. Kurzzeittrenddarstellungen sind generell ohne Zusatzprogramm nicht moeglich.

Das Anwenderunterstützungsprogrammssystem "APIC" umfasst neben der eigentlichen Bildverarbeitung einige Zusatzprogramme, die einen effektiveren Umgang mit der ON-LINE-Abarbeitung der Bilder erlauben. Das Programmsystem ist EPROM-fähig und belegt die Programmebene 2 (7000 - APFFH).

Entsprechend AR-Grundmenue sind nach Laden des Programmsystems (Dateityp P, Ebene 22) folgende Programme verfügbare:

GRUNDMENUE

- | | | | |
|----|-------------------------|---|----------------|
| 0 | GRUNDMENUE | | |
| 1 | STRUKTURIERUNG | | |
| 2 | SYSTEMBEDIENUNG/ANZEIGE | | |
| 3 | HACO | | |
| 4 | KOMMANDO-VAP | \ | |
| 5 | ANLAUF-VAP | | |
| 6 | DATENBEREITSTELLUNG | > | Programmsystem |
| 7 | ANLAGENBILDER | | "APIC" |
| 8 | ANBI-DIALOG | / | |
| 9 | | | |
| 10 | | | |
| 11 | | | |
| 12 | | | |
| 13 | | | |
| 14 | | | |
| 15 | | | |
| 16 | | | |
| 17 | | | |
| 18 | | | |
| 19 | | | |

#M: J 3

Weiterhin ist der Datensatz "B000H" verfügbare, welcher auf die Unspeicheradresse fuer den BO-EPROM, 5000-53PPH, geladen wird (Dateityp P, Ebene 11). Dieser enthaelt die fuer die Abarbeitung des Programmsystems "APIC" erforderliche spezifische AR-Systemstrukturierung.

Im einzelnen sind folgende Festlegungen getroffen:

LISTE PRIO
VAP-NUMMER

0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	

#8:

SYSTEMDATEN

MAX-VAP	8				
NR-OFF	5				
GER 1:	00000110				
GER 2:	00011000				
UNTERSTATIONSADRESSEN	US1	US2	US3	US4	
	FF	FF	FF	FF	

ANMELDELISTE OFF---->ON

	VAP	STB 1	STB 2	TK	ZAEHL	TK	ZAEHL
0	4	10000000	00100000	2	1	2	1
1	5	10000000	00100000	2	1	2	1
2	6	00000000	10000000	2	1	2	2
3	7	00000000	00000000	0	0	2	2
4	3	00000000	10000000	2	1	0	0
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							

#E:

VAP- STEUERLISTE

	ED	EP	ADR.	ST-BYTE
0	1	1	8320	0000
1	1	1	7021	0000
2	1	1	8149	0000
3	1	1	8750	0000
4	1	2	9180	0000
5	1	2	9120	0000
6	1	2	80A5	0000
7	2	2	7000	0000
8	1	2	7E00	0000
9	5	5	0000	0000
10	5	5	0000	0000
11	5	5	0000	0000
12	5	5	0000	0000
13	5	5	0000	0000
14	5	5	0000	0000
15	5	5	0000	0000
16	5	5	0000	0000
17	5	5	0000	0000
18	5	5	0000	0000
19	5	5	0000	0000

#E:

Die Prüfsummen der Speicherbereiche des gesamten Programmsystems sind:

2.7000	CCD6	
2.7400	56AB	
2.7800	8665	
2.7C00	6CD3	
2.8000	2306	
2.8400	25EA	
2.8800	0C2E	
2.8C00	D949	"APIC"
2.9000	6744	
2.9400	B12B	
2.9800	7101	
2.9C00	67EE	
2.A000	86FE	
2.A400	F216	
2.A800	BF1E	
2.AC00	60C8	

G.5000	B907	"B0000"

8.4.2.2. Handhabung des Anwenderunterstützungsprogrammsystems

Das Programmsystem organisiert nach Rechnerzuschaltung einen sofortigen Übergang des AR in den ON-LINE-Betrieb. Vorausgesetzt, dass auf Datenebene 2 ff die strukturierten Bilder geladen sind, erscheint bei ON-LINE-Umschaltung automatisch das bei der Strukturierung als Bild 1 in der Videothek abgelegte Bild. Die Auswahl weiterer strukturierter Bilder erfolgt mittels Taste PF1 und Zifferneingabe, und zwar wie folgt:

(PF1 - 00 ----> Bild 1)
 PF1 - 03 ----> Bild 2
 PF1 - 04 ----> Bild 3

Die Eingabe PF1 - 01 bzw. PF1 - 02 ist fuer spezielle Sonderfunktionen reserviert. Das Bild 1 ist auch ueber die Taste PF2 aufrufbar. Falls mehrere Bilder strukturiert sind, wird empfohlen, als Bild 1 ein Menue abzulegen, aus dem ersichtlich wird, mit welcher Zifferneingabe welches Bild aufgerufen werden kann. Die Taste PF2 sollte dann zum Aufruf dieses Menues genutzt werden.

Bei Verwendung der Kurzzeittrenddarstellung (Dynamikmodul 8) bzw. Balkendarstellung (Dynamikmodul 9) besteht die Moeglichkeit, den Anzeigebereich (ANBE) und Anzeigenanfang (ANZA) im ON-LINE-Betrieb zu aendern. Diese Funktion wird mittels PF1-98 aufgerufen. In der 1. BS-Zeile erscheint TRNR: ____.

In der Cursorposition ist die Nr. der Trend- bzw. Balkendarstellung einzu/geben. Anschliessend erscheint ANBE mit

0	≅ 100 %
1	≅ 50 %
2	≅ 25 %
3	≅ 12,5 %

und ANZA als %-Anzeige. Nunmehr existiert Eingabebereitschaft fuer ANBE und nach Betaetigung von ENTER fuer ANZA. Wiederum nach Betaetigung von ENTER wird der Anfangswert uebernommen, und eine neue Nr. kann angewaehlt werden. Nach OFF-Betaetigung wird ohne Eingabeuebernahme zur TRNR-Eingabe zurueckgesprungen. Nach nochmaliger Betaetigung von OFF erlischt generell die Eingabebereitschaft.

Es wird empfohlen, im Rahmen der statischen Bild-Strukturierung jedem Trend/Balken eine ueber alle Bilder fortlaufende Nr. zuzuordnen und anzuzeigen. In der gleichen Reihenfolge (Nr. 1...) muessen die zu den einzelnen Trends/Balken zugehoerigen ANBI-Bytes in der ANBI-Byte-Tabelle abgelegt sein. Nur somit ist eine eindeutige Zuordnung beim ANBI-Dialog moeglich.

Einige Einschraenkungen und Randbedingungen bei der Anwendung des Programmsystems sind nachfolgender Kurzbeschreibung der Einzelprogramme zu entnehmen.

8.4.2.3. Kurzbeschreibung der einzelnen Programmbestandteile

Anlauf-VAP

Das Programm ist als OFF-LINE-VAP in die Abarbeitung des AR eingetragen. Es erfuehlt folgende Aufgaben:

- Setzen von Anfangbedingungen fuer das VAP "Anlagenbilder" nach Rechneranlauf, z. B.
 - . Eintragen der Anfangsadresse des Hinweispuffers (6500H), der ANBI-Byte-Tabelle (64E0H), des Kurzzeittrendsteuerfeldes (6460H) in den Systemblock
 - . Loeschen der ANBI-Byte-Tabelle (Laenge 30 Byte), d.h. Eintragen der Grundbelegung (00H)
- Umspeichern der Datentelegramme zur Prozessabbildaktualisierung von ACOOH auf 3000H (Laenge 1 kByte) nach Rechneranlauf
- Umschaltung des AR auf ON-LINE nur nach Rechneranlauf
- Zyklische Anzeige der Uhrzeit und der Fehlermeldungen (AR-Statusbyte) im Restart-Teil (ON-LINE)

Kommando-VAP

Das Programm realisiert den Bedienerdialog ueber die Auswertung der Funktionstasten PF1 und PF2 sowie der numerischen Tasten. Es organisiert das Anmelden des VAP's "Anlagenbilder" zum zyklischen Restart (2 sek) und dessen Neustart mit Darstellung des Bildes Nr. 1 nach ON-LINE-Umschaltung bzw. einer anderen angewaehlten Bild-Nr. Weiterhin loest es die Anmeldung des "ANBI-Dialog"-VAP's aus.

VAP Anlagenbilder

Dieses VAP ist das eigentliche ON-LINE-Verarbeitungs-Programm zur Darstellung der strukturierten Bilder. Im Neustartteil werden die statischen Bilder mittels Bildinterpreter zur Anzeige gebracht und die Zeiger auf die Position des 1. MAB der Dynamikmodulkette gestellt. Im Restartteil werden in der Reihenfolge der Strukturierung die dynamischen Bildmodule aufgerufen und abgearbeitet.

Der Sprungverteiler fuer die Elementarbildmodule befindet sich auf Adresse

```
7334H: JMP M01
7337H: JMP M02
      :
7358H: JMP M013.
```

Beim Aufruf der Elementarbildmodule wird im Register IX die Adresse des 1. strukturierbaren Parameters im MAB uebergeben. Die Rueckkehr in das Rahmenprogramm erfolgt mit JMP 72EFH.

VAP Datenbereitstellung

Das Programm realisiert die Aktualisierung von max.20 Trendpuffern mit den zugeordneten 2 Byte-Festkommawerten des Prozessabbildes im Minutentakt.

Im Neustartteil erfolgt das Umspeichern der Verbindungsliste zwischen Trendpuffer-Nr. und PRAB-Nr. sowie der Aufbau der Steuerlisten fuer den Kurzzeittrend. Die Verbindungsliste wird von Adresse 90DAH ... auf Datenebene 1/Adresse 1C10H... umgespeichert und hat folgenden Aufbau:

```
DB 14H          ; Laenge 20 Trends
DA 0101H       ; HWT=Nr.Trend, NWT=NR.PRAB
DA 0202H
      .
      .
      .
DA 1414H
```

\
|
|
| > 41 Byte
|
|
/

Das heisst, dass der Wert aus PRAB-Nr. 1 bei der Aktualisierung in den Trendpuffer Nr. 1 uebernommen wird ... und der aus PRAB-Nr. 20 in den Trendpuffer Nr. 20. Die Kurzzeit-trend-Steuerliste wird entsprechend Beschreibung in Abschn. 8.2. ab Adresse 6460H aufgebaut. Der Trendpuffer wird ab

Adresse 1039H bis 1090H (1. Datenebene) eingerichtet. Damit sind 20 Trends zu max. 30 Byte (15 Pk-Werte) moeglich. Im Restartteil erfolgt im Minutentakt die Umspeicherung des aktuellen PRAB-Wertes in den Trendpuffer.

ANBI-Dialog-VAP

Dieses VAP realisiert innerhalb des ON-LINE-Betriebes den Dialog mit dem ANBI-Byte und ermoeoglicht damit die in vorangegangenen Abschn. dargestellte Beeinflussung von ANZA oder/und ANBE bei der Kurzzeittrend- und Balkendarstellung.

8.5. Anwendungsbeispiel

Mit dem Bildstrukturierprogramm "PIC25" wurden z.B. 14 technologische Schemata mit eingeblendeten Messwerten und dazugehoerigen Kurzzeittrends fuer einen Anwendungsfall im 100 MW-Kraftwerksblock generiert und mit dem Programmsystem "APIC" im ON-LINE-Betrieb verarbeitet. Die BS-Darstellung nach Anwahl der PIC-Menuefunktion 4 (ANZ) beinhaltet die vollstaendige Speicheruebersicht aller vorhandenen 14 Bilder und der dabei verwendeten Statikmodule. Dies zeigt folgende Hardcopy-Abschrift:

ANZEIGE VIDEOTHEK: EBENE A
BILDADR.STAT./DYN.

1	12FAH	1584H	1	10BAH	17	12BOH	33	----	49	----
2	1587H	165AH	2	110AH	18	12D4H	34	----	50	----
3	1858H	1A70H	3	1125H	19	12DDH	35	----	51	----
4	1BECH	1D25H	4	1160H	20	12E4H	36	----	52	----
5	1DEBH	1FE0H	5	116BH	21	12EBH	37	----	53	----
6	2075H	2221H	6	117BH	22	12EPH	38	----	54	----
7	22F6H	23ECH	7	11BBH	23	12F0H	39	----	55	----
8	2536H	25ACH	8	11C9H	24	12F5H	40	----	56	----
9	268DH	27D1H	9	11D6H	25	----	41	----	57	----
10	2942H	29E1H	10	11E5H	26	----	42	----	58	----
11	2AC2H	2B6CH	11	11F2H	27	----	43	----	59	----
12	2C4DH	2D00H	12	11FCH	28	----	44	----	60	----
13	2D5CH	2FE6H	13	1207H	29	----	45	----	61	----
14	2FE9H	2FEAH	14	123CH	30	----	46	----	62	----
15	----	----	15	127FH	31	----	47	----	63	----
16	----	----	16	129EH	32	----	48	----	64	----
17	----	----								
18	----	----								
19	----	----								
20	----	----								
21	----	----								
22	----	----								
23	----	----								
24	----	----								
25	----	----								

Im Bild Nr. 2 wird fuer das Frischdampf-System z. B. der rechte und linke Ueberhitzer-Strang mit den jeweiligen Einspritzstellen schematisch dargestellt und beschriftet. Fuer beide Straenge werden je 8 Messwerte eingeblendet und zyklisch alle 2 sek aktualisiert. Weiterhin werden in diesem Bild 3 Kurzzeittrends ueber eine Zeitachse von 15 min dargestellt (s. Bild 8.5.).

TRNR : 2 ANBE : 1 ANZA : 50.0

Frischdampfsystem

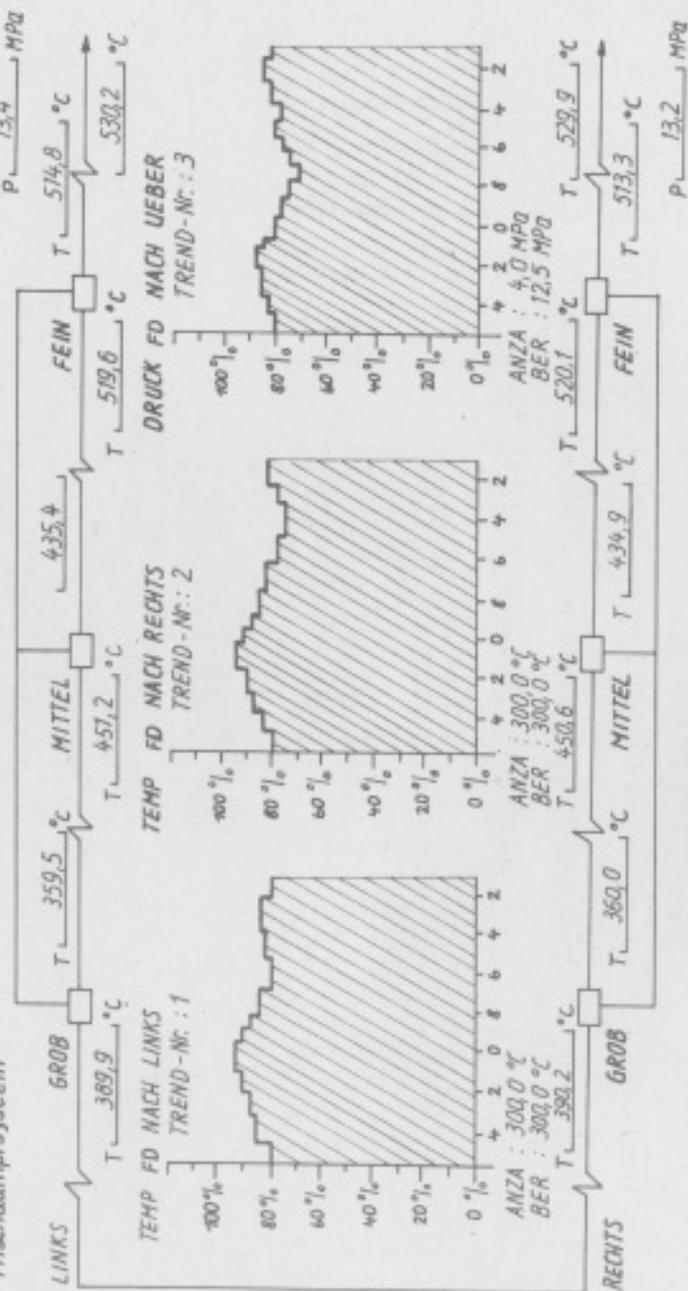


Bild 8.5 : Frischdampfsystem

Sie beinhalten

- die technologische Bezeichnung
- die Achsendarstellung
- die Achsenbeschriftung mit
 - . Zeitbereich (statisch)
 - . Anzeigebereich (dynamisch)
 - . Anzeigeanfang absolut (dynamisch).

Eine Uebersicht ueber den statischen Aufbau des Bildes Nr. 2 gibt die BS-Anzeige in der Menufunktion 5 (CODE):

BILDCODE ANZEIGE

0000

CO-AD	ZE/SP	PA	CODE	SPEZIFIZIERUNG
1C00H	6	1	GS	MOD 12
1C04H	28	1	GS	MOD 13
1C08H	7	1	GS	MOD 14
1C0CH	27	1	GS	MOD 15
1C10H	9	1	GS	MOD 15
1C14H	29	1	GS	MOD 14
1C18H	30	1	GS	MOD 12
1C1CH	12	2	GS	MOD 3
1C20H	12	23	GS	MOD 3
1C24H	12	44	GS	MOD 3
1C28H	8	1	GS	MOD 13
1C2CH	9	1	GS	MOD 15
1C30H	2	1	GS	JHP 1
1C34H	2	2	GS	COL YS
1C38H	2	2	YS	REP 63
1C3CH	3	1	YS	JMP 1
1C38H	3	2	YS	COL WS
1C3EH	3	2	WS	QUE 17 FRISCHDAMPFSYSTEM
1C40H	3	19	WS	JMP 64
1C44H	4	19	WS	JHP 61
1C48H	5	16	WS	JHP BS
1C4CH	5	16	BS	REP 16 A
1C4EH	5	35	BS	REP 17 A
1C50H	5	52	BS	JMP 4
1C54H	5	56	BS	COL 80
1C58H	5	56	BS	REP 1 P
1C5CH	5	57	BS	COL 80
1C5EH	5	57	GS	JMP 64

BILDCODE ANZEIGE

0000

CO-AD	ZE/SP	PA	CODE	SPEZIFIZIERUNG
1CD2H	31	57	BS	ENDS

Die dynamischen Funktionen werden in Bild 2 durch eine Kette von 22 Bildmodulen realisiert. Dies zeigt die Hardcopy-Abschrift nach Auswahl der Menufunktion 1 (DYN):

E ERWEIT L LOESCH P PARAM S SIMUL

DYNAMIK MODULKETTE

1	KUTR	21	ANBI
2	KUTR	22	FLKO
3	KUTR		
4	FLKO		
5	FLKO		
6	FLKO		
7	FLKO		
8	FLKO		
9	FLKO		
10	FLKO		
11	FLKO		
12	FLKO		
13	FLKO		
14	FLKO		
15	FLKO		
16	FLKO		
17	FLKO		
18	FLKO		
19	ANBI		
20	ANBI		

SPICHERBEDARF	DYNAMIK	:0119H
	RESERVE	:0A02H

Fuer die 16 Messstellen ist je ein Modul FLKO (MO10) zustaendig, fuer die 3 Kurzzeittrends je ein Modul KUTR (MO8) und fuer den veraenderbaren Anzeigeanfang und Anzeigebereich der Trends je ein Modul ANBI (MO11). Die Module Nr.1 (KUTR), 4 (FLKO) und 19 (ANBI) in der gezeigten Modulkette sind beispielweise wie folgt parametrisiert:

E ERWBIT L LOBACH P PARAM S SINUL

0000

DYNAMIK MODULKETTE

1	KUTR		
	1	ZEILE	14
	2	SPALTE	3
	3	ANZAHL	15
	4	HOEHE	10
	5	FARBE	02H
	6	ANBINR	1
	7	NR	1

E ERWEIT L LOESCH P PARAM S SIMUL

DYNAMIK MODULKETTE

4 FLKO

1	ZEILE	7
2	SPALTE	19
3	FARBE	10H
4	K1	.0000
5	K2	.6000
6	E	3
7	TYP	2
8	NR	66

E ERWEIT L LOESCH P PARAM S SIMUL

DYNAMIK MODULKETTE

19 ANBI

1	Z ANY	25
2	SP ANP	16
3	Z BER	13
4	SP BER	10
5	FARBE	07H
6	ANZTYP	1
7	SEALE	10
8	K1	.0000
9	K2	.5000
10	E	3
11	ANBINR	1

9. Aenderung oder Erstellung des Zeichengenerators

9.1. Uebersicht

Mit dem Programm Zeichengenerator ist es moeglich, einen Zeichengenerator fuer die Farbvariante des Applikationsrechners mit der Bildschirmansteuerung ABS K 7029.05 zu erarbeiten. Es kann ein neuer Zeichengenerator erstellt, aber auch ein schon vorhandener Zeichengenerator bearbeitet werden. Der erstellte Zeichengenerator kann auf Drucker oder externe Speicher ausgegeben werden. Es ist auch moeglich, den Zeichengenerator auf EPROM zu programmieren.

9.2. Startbedingungen

Das Programm ist nachzuladen und bei der Strukturierung des Applikationsrechners als Verarbeitungsprogramm (VAP) mit Adresse 3006H einzutragen. Das Programm belegt in der zugewiesenen Ebene den Speicherbereich von 1000H bis 3740H. Soll ein bereits vorhandener Zeichengenerator bearbeitet werden, so ist dieser in den Arbeitsspeicher des Programms (ab Adresse 1000H) einzulesen.

9.3. Bearbeitungsablauf

Beim Start des Programms erscheint in den ersten acht Zeilen des Bildschirms der gesamte Zeichenvorrat des zur Zeit in der Bildschirmansteuerung vorhandenen Zeichengenerators. In der unteren Haelfte des Bildschirms erscheint ein Grossfeld, in dem ein Zeichen vergraessert dargestellt wird (je Bildpunkt ein Zeichen). Neben dem Grossfeld wird die Hexa-Codierung des jeweiligen Zeichens angezeigt. Das im Grossfeld dargestellte Zeichen ist im Zeichenvorrat durch Blinken markiert. Die im Grossfeld angezeigten Zeichen entsprechen den im Arbeitsspeicher enthaltenen Zeichengenerator. Eine Uebereinstimmung des im Grossfeld dargestellten Zeichens mit dem im Zeichenvorrat markierten Zeichen ist nur dann gegeben, wenn das Zeichen des Zeichengenerators im Arbeitsspeicher und das Zeichen des Zeichengenerators in der Bildschirmansteuerung mit derselben Hexa-Codierung identisch sind. Durch Eingabe von "L" kann der gesamte ZG geloescht werden. Das Programm befindet sich im Auswahlmodus. Mit Hilfe der Cursortasten ist es moeglich, jedes Zeichen aus dem Zeichenvorrat anzusteuern und es somit im Grossfeld erscheinen zu lassen. Durch Eingabe von "A" kann in den Aenderungsmodus gegangen werden. Dabei wird die Cursorsteuerung vom Zeichenvorrat auf das Grossfeld umgeschaltet, so dass mit den Cursortasten jeder Bildpunkt des Zeichens im Grossfeld angesteuert werden kann. Der jeweilige Bildpunkt blinkt invers. Mit der Betaetigung der Taste "I" wird der angesteuerte Bildpunkt gesetzt, mit Betaetigung der Taste "O" wird der Bildpunkt geloescht. Ist die Bearbeitung eines Zeichens abgeschlossen, muss mit <ENTER> quittiert werden. Dabei werden alle Eintragungen bzw. Aenderungen in den Arbeitsspeicher uebernommen. Die Cursorsteuerung ist wieder auf den Zeichenvorrat umgestellt,

so dass weitere Zeichen angesteuert werden koennen.

Mit "Z" kann das angezeigte Zeichen geloescht werden.

9.4. Ausgabe des Zeichengenerators

9.4.1. Ausgabe auf Drucker

Der im Arbeitsspeicher enthaltene Zeichengenerator kann graphisch auf Drucker ausgegeben werden. Dazu ist im Auswahlmodus ein "D" einzugeben. Der Bildschirm wird geloescht und es erscheinen 16 Grossfelder auf dem Bildschirm. Nach Eingabe der Druckernummer kann der Umfang der auszugebenen Zeichen festgelegt werden. Es werden jeweils acht Zeichen nebeneinander ausgegeben. Die Zeichen werden blockweise (16 Zeichen) gedruckt. Zur Ausgabesteuerung sind einzugeben:

- "1" Ausgabe der Zeichen von 00H bis 7FH
- "2" Ausgabe der Zeichen von 80H bis FFH
- "3" Ausgabe der Zeichen von 00H bis FFH
- "4" Ausgabe der Zeichen von 00H bis FFH mit Abfrage vor jedem Block (Eingabe "P" - Block drucken, Eingabe "N" - naechster Block).

Sind alle gewuenschten Zeichen ausgegeben, so koennen nach Eingabe der Druckernummer weitere Ausgaben durchgefuehrt werden.

Soll in den Ausgabemodus zurueckgesprungen werden, so ist nach einer abgeschlossenen Ausgabe zweimal mit <ENTER> zu quittieren. Zur Kontrolle der Druckerausgabe werden auf dem Bildschirm Druckerfehler und Druckpuffer angezeigt.

9.4.2. Ausgabe auf Magnetband/FD

Der Zeichengenerator kann unter Benutzung des KMBG- oder FD-Programms des Applikationsrechners auf eine Digitalkassette oder Diskette abgespeichert werden. Dazu muss das ZG-Programm verlassen werden und das MB/FD-Programm des Applikationsrechners aufgerufen werden. Als Ebene ist die Ebene anzugeben, auf der das ZG-Programm abgespeichert wurde. Die Anfangsadresse des abzuspeichernden Bereiches ist 1000H und die Endadresse ist 1BFFH.

9.4.3. Programmierung auf EPROM

Auf dem Applikationsrechner ist es moeglich, den Zeichengenerator direkt auf EPROM auszugeben. Das fuer die EPROM-Programmierung zustaeendige Systemprogramm ist dafuer nach dem Verlassen des ZG-Programms anzuwaehlen. Als Ebene ist die Ebene anzugeben, auf der das ZG-Programm abgelegt ist. Die EPROM's sind wie folgt zu programmieren:

- EPROM 1 Speicherbereich 1000H bis 13FFH,
- EPROM 2 Speicherbereich 1400H bis 17FFH,
- EPROM 3 Speicherbereich 1800H bis 1BFFH.

9.5. Zusammenfassung der Bedienkommandos

- Auswahlmodus: - Cursortasten (Steuerung des Cursors im Zeichenvorrat zur Zeichenauswahl)
 - "A" (Uebergang in den Aenderungsmodus)
 - "D" (Uebergang in den Druckmodus)
 - "L" (Loeschen des Zeichengenerators im Arbeitsspeicher)
 - "Z" (Loeschen des angezeigten Zeichens)
- Aendermodus : - Cursortasten (Anwahl eines Bildpunktes)
 - "0" (Loeschen eines Bildpunktes)
 - "1" (Setzen eines Bildpunktes)
 - <ENTER> (Uebergang in den Auswahlmodus)
- Druckmodus : - "1" (Drucken der Zeichen von 00H bis 7FH)
 - "2" (Drucken der Zeichen von 80H bis FFH)
 - "3" (Drucken der Zeichen von 00H bis FFH)
 - "4" (Drucken der Zeichen von 00H bis FFH blockweise)
 - "P" (einen Block drucken bei blockweisen Druck)
 - "N" (naechster Block bei blockweisen Druck)
 - <ENTER> (Uebergang in den Auswahlmodus)

9.6. Adressen

1000H bis 1BFFH	Zeichengenerator
3006H	VAP-Startadresse
3000H bis 3740H	Programmteil

VEB Geräte- und Regler-Werke „Wilhelm Pieck“ Teltow

Betrieb des VEB Kombinat Automatisierungsanlagenbau

DDR · 1530 Teltow, Oderstraße 74-76 · Telefon 440 · Telex 015441



Nachdruck bzw. Vervielfältigung ist nur mit Genehmigung des VEB GRW Teltow zulässig. Änderungen im Sinne des technischen Fortschritts vorbehalten.

AUSGABE: Januar 1989