

Ergänzung 5/89 zur Anwendervorschrift Applikationsrechner

1.	Vorbemerkungen	1
2.	Grundsätzliche Hinweise zur Bedienung mit der Tastatur K 7672.03	1
3.	Korrekturen und Ergänzungen neuer Punkt: 7.11. BASIC Programmbeispiel	3 13

1. Vorbemerkungen

Diese Ergänzung bezieht sich auf die Technische Dokumentationen für den AR: C17A, C17B und C17C. Sie wurden vor allem erforderlich wegen der Ablösung der Robotron-Tastatur ANF K 7634.51 durch die Robotron-Tastatur K 7672.03. Weiterhin werden die bis zu diesem Zeitpunkt eingetretenen sonstigen Änderungen eingearbeitet.

Aufgrund der teilweise sehr unterschiedlichen Handhabung beider Tastaturen wird ein Abschnitt zur AR-Bedienung mittels K 7672.03 vorangestellt. Nachfolgend werden seitenweise geordnet die in den Dokumentationen erforderlichen Ergänzungen bzw. Korrekturen aufgelistet.

2. Grundsätzliche Hinweise zur Bedienung mit der Tastatur
K 7672.03

Nach Rechneranlauf bzw. Rechner-Reset und beim Herstellen der Verbindung zwischen Tastaturstecker und Tastaturanschlusskabel durchläuft die Tastatur eine Anlaufroutine mit Selbsttest und Einstellung des erforderlichen Grundzustandes. Die Arbeitsbereitschaft wird akustisch (Pip-Ton) und optisch (READY- und CAPS-Anzeige) gemeldet. Der gleiche Ablauf erfolgt nach Betätigung der Taste "BREAK".

Mit Betätigen der Tasten "CTRL" + "RESET" wird ein internes Tastatur-Reset erzeugt. Danach liefert die Tastatur Kleinbuchstaben. Die Gross-/Kleinbuchstaben-Umschaltung erfolgt mittels der Taste "CAPS LOCK". Der Tastenklick kann mit der Taste "CL" ein- bzw. ausgeschaltet werden.

Das Grundsende des AR wird mit PF9 aufgerufen (bei der alten Tastatur mit PF12). Als Quittungstaste kann "<---|" oder "ENTER" genutzt werden (bei der alten Tastatur nur ENTER). Als Abbruchtaste wird "R" genutzt (entspricht OFF bei der alten Tastatur). Aufgrund der sehr häufigen Verwendung der Tastenbezeichnungen "OFF" und "ENTER" in der Dokumentation werden diese Bezeichnungen n i c h t korrigiert. Mit der Taste "<---|" oder "CE" wird ein Zeichen gelöscht, mit der Taste "DEL" kann ein Bereich gelöscht werden.

Bestimmte Funktionen müssen durch gleichzeitiges Betätigen der Taste "CTRL" und einer Funktionstaste erzeugt werden:

INS MODE	[INS M]	= CTRL + PF1
INS LINE	[INS L]	= CTRL + PF2
DEL LINE	[DEL L]	= CTRL + PF3
ERASE INP	[E INP]	= CTRL + PF4
ERASE EOF	[E EOF]	= CTRL + PF5
DUP		= CTRL + PF6
FM		= CTRL + PF7

Hierfür sind zukünftig entsprechende Vorderflächenbeschriftungen der Tasten vorgesehen. Eingabekombinationen von CTRL + Tasten mit ASCII-Zeichen sollen bei der normalen Anwendung der Tastatur vermieden werden!

Die Tastenbelegung mit der AR-internen Tastencodierung ist dem Korrekturblatt zur Seite 59 zu entnehmen. Hier sind auch Tasten gekennzeichnet, die am AR generell nicht betätigt werden sollen und auch solche Tasten, die nur für spezielle Anwendungen vorgesehen sind. Solche speziellen Anwendungen sind:

- Ein-/Ausschalten des Alternativzeichensatzes mittels "ALT1" (Lämpchen an/aus)

Damit wird bei ausgeschaltetem CAPS LOCK für die ä-, ø-, û-Taste eine Vertauschung der Shift-Wirkung bewirkt:
[<---> ä, \ <---> ø,] <---> û

- Aus-/Einschalten der Umcodierung der direkten Tastencodes in die AR-internen Tastencodes mittels "^S" (Lämpchen an/aus)

Nach einmaliger Betätigung von "^S" (Lämpchen an) werden alle folgenden Eingaben direkt codiert in den Puffer für die Tastatureingabe eingeschrieben.

Bei der normalen Anwendung (Lämpchen aus) erfolgt eine Umcodierung der Steuerzeichen bzw. Steuerfolgen der Tastatur in eine AR-interne Codierung, die den Codes der alten Tastatur entspricht, und unzulässige Steuerzeichen unterdrückt.

- Ein-/Ausschalten der direkten Hexacodierung mittels "ALT"

Damit kann bei gedrückter Taste "ALT" ein beliebiger Code direkt mit 2 Hexazeichen eingegeben werden, der bei Loslassen der Taste "ALT" als Tastencode übertragen wird. Bei Bedarf ist vorher die Taste "^S" zu betätigen (keine Umcodierung/keine Unterdrückung).

- Eingabe von Sondercodes mittels "ESC"

Nach Betätigen der Taste "ESC" kann eine Taste mit normalem ASCII-Zeichen betätigt werden, deren Code dann mit gesetztem BIT 7 in den Tastaturpuffer eingetragen wird (Ohne vorherige Betätigung von "ESC" wird direkt der ASCII-Code in den Tastaturpuffer eingeschrieben).

3. Korrekturen und Ergänzungen

Seite 14, vor Punkt 1.4.

Text-Ergänzung:

- "Bei jeder Umschaltung der Betriebsart wird der Back-up-Kontakt im Fehleranzeigebaustein gesetzt. Eine Freigabe des Back-up-Signals kann durch den Anwender über Kommando Nr. 23 erfolgen"

Seite 37, im Anschluss an die Tabelle

"Hinweis: Die Antworten für die Regime-Nr. 15...27 (6F...7BH) enthalten grundsätzlich ein Statusbyte entsprechend Bild 1.10.1."

Seite 48, Pkt. 1.13.1, 5. Zeile

"Tastatur ANF K 7634.51" wird ersetzt durch "Tastatur K 7672.03".

Seite 53, im Abschn. ISO*

Adressen-Korrektur:
B413 (alt) ----> B41B (neu)

Seite 54, Seite 56, Seite 59

Werden komplett durch folgende Seiten ersetzt.

Tabelle 1.13.2.

Liste der Steuerzeichen

01	TAB+	Tabulator	Es können max. 7 Tabulatoren in aufsteigender Reihenfolge abgelegt werden. Durch das Betriebssystem wird eine Grundbelegung eingestellt. (Taste: --->)
04	KUO+	Kursor nach oben	(Taste: ↑)
05	KUU+	Kursor nach unten	(Taste: ↓)
06	KUR+	Kursor nach rechts	(Taste: --->)
07	KUL+	Kursor nach links	(Taste: <---)
0A	KNL	Kursor nächsten Zeilenanfang	
0C	DUN	BS dunkler (nur MON2)	
0D	KEA	Kursor am Zeilenanfang	
0E	HEL	BS heller (nur MON2)	
0F	CLR+	Zeichen löschen	(Taste: <--- bzw. CE)
12	KBA	Kursor an Bildanfang (!Bereich wird max gesetzt)	
13	DEL+	Zeile löschen	(Taste: CTRL + PF) = DEL L)
14	BLX	BS löschen, Kursor an Bildanfang (!Bereich wird max gesetzt)	
15	KBR	Kursor an Bereichsanfang	
16	BRO	BS rollen 1 Zeile nach oben, Kursor auf Anfang letzte Zeile	
17	BRU	BS rollen 1 Zeile nach unten, Kursor auf Anfang erste Zeile	
18	KBE	Kursor BS-Ende	
19	KDU	Kursor dunkel schalten	
1B	BLO+	Bereich löschen	(Taste: DEL)

Die mit + gekennzeichneten Codes werden AR-intern auch durch die Tastatur erzeugt. Durch Eingabe mit direkter Hexa-Codierung (ALT, *S) kann tastaturseitig auch jeder andere Code erzeugt werden.

Folgende Tasten der Tastatur können gesondert entschlüsselt werden:

PP 9	----> Anmeldung von VAP 0 (fest)
Monitor	----> Break ----> Arbeit mit Monitor möglich (strukturierbar)
HACO	----> Hardcopy ----> Aufruf Hardcopyprogramm (strukturierbar)

- Die Tastatur wird bei Aufruf der Unterprogramme jeweils einem VAP zugeordnet (eine aktuell mögliche Eingabe). Sondertastensequenzen werden in einem speziellen Puffer abgelegt und sind durch spezielle Kommando-VAP auswertbar (parallel zur eigentlichen Wert- und Texteingabe).

ANRK:	Anreizkennung	/ 1 = Anreiz Funktionstaste \ 0 = keine Eingabe
	Tastencode	
	- Folgetasten -	} Anzahl entsprechend
		} strukturierter Anzahl
	-	} für den Tastencode
		} in der Steuerliste

Die Funktionstasten können durch unterschiedliche VAP ausgewertet werden. In VAP, das als letztes die Eingabe nutzt, ist die Anreizkennung zu löschen.

- Die Unterprogramme mit Unterbrechung sind im Off-line- und On-line-Betrieb nutzbar. Insbesondere für die Eingabe im On-line-Betrieb ist eine Tastatureingabe ohne Warten möglich. Dadurch können z.B. zyklische Anzeigen mit Beeinflussung über Tastatur realisiert werden.

Nutzung:

- Tastaturzuweisung zum VAP über das UP ZEPU im Neustart der Funktion
- zyklische Übernahme des Tastaturpuffers über das UP ABPR im Restartzweig der Funktion

Seite 66, Kommandotabelle

Kommando 42 Spalte Eintritt | Austritt lautet richtig:
"HL: Adr. EASF"

Seite 76, Tabelle 1.13.6 (Ueberschrift)

"Magnetband- und Druckeransteuerung" wird ersetzt durch
"Floppy- und Druckeransteuerung".

Seite 80, vor Abschn. Sprungverbinder

Ergänzung Systemsteuerzellen:
"ANRK 5B97H Anreizkennung Sondertaste"

Seite 84, im Abschn. *HALT, SYST

Der Code D9 ist ersatzlos zu streichen.

Seite 94, nach Abschn. Fehlerzustand

Textergänzung:
"Fehlerrücksetzen"

Das Rücksetzen der Fehler per Bedienhandlung erfolgt in der Fehlerzustandstabelle. Mit dem Cursor wird der jeweilige Fehler angefahren und mit ENTER inaktiv gesetzt (BIT 7 = 1)."

Seite 102, nach Abschn. zur Fehler-Nr. 1C

Ergänzung zur Erläuterung der Systemfehler:
"Beschreibung: Tastaturfehler 1F
Spezifikation: 01 Tastatureigentest

Ursache: Tastatur meldet internen Fehler beim Anlauf
/Reset des Rechners, beim Anstecken der Tas-
tatur, bei Betätigung der BREAK-Taste.

Löschen: 20 s nach letzter Fehlererkennung"

Seite 105, nach Fehler-Nr. 1C

Ergänzung Systemfehlermeldungen:
Nr. Fehlerkl. Spezifikation Löschen Fehlerbeschreibung Nr.
"1F 0 Code 20 sek Gerätefehler Tastatur 1F"

Seite 114, Abschn. Zuweisung der Tasten

- Hinter INS ist zu ergänzen: "Tasten: CTRL + PF1"
MODE ---->

- Hinter INS ist zu ergänzen: "Tasten: CTRL + PF2"
LINE ---->

- Hinter DEL ist zu ergänzen: "Tasten: CTRL + PF3"
LINE ---->

Seite 118, Seite 119, Seite 120

Werden komplett ersetzt durch die folgenden Seiten:

3.3.5. Strukturierung des Prozessabbildes (PRAB)

Zur Strukturierung des PRAB wird folgendes Untermenü angeboten:

STRUKTURIERUNG PRAB: X

0 NEUSTART
1 VERWALTUNG
2 STEUERFELDER

Bei der Strukturierung des PRAB erfolgt der Aufbau der Steuerfelder zur Verwaltung (Aktualisierung, Zugriff) des Prozessabbildes und der Anforderungstelegramme zur BSE. Die Länge eines Anforderungstelegramms ist ≤ 256 Byte.

Die strukturierten Bereiche bzw. Listen werden dynamisch verwaltet, das heisst, sie haben variable Längen und werden dichtgespeichert. Bei der Strukturierung werden die Eingaben auf Sinnfälligkeit geprüft. Bei Verletzung der Bereichslängen wird die Eingabe abgelehnt.

3.3.5.1. Neustart

Bei Neustart der Funktion werden die Steuerlisten gelöscht. Die Eingabe ist nur bei Neuzuweisung und nicht bei Korrekturauszuführen. Die Eingabe ist durch weitere Quittungsanforderung ("J") verriegelt.

3.3.5.2. Verwaltung

Die Verwaltung erfolgt in mehreren Stufen:

- Grundzuweisung für Ebene, Adressen, Länge
 - . Ebene (EB) : 1 ... 4
 - . Adresse Prozessabbild (A-PRAB): hexadez.
 - . Länge Prozessabbild (L-PRAB): dezimal
 - . Adresse Telegramme (A-TEL) : hexadez.
 - . Länge Telegramme (L-TEL) : dezimal
- Reservierung der Prozessabbildbereiche mit 1, 2, 3 und 5 Byte (Festlegung der Elementezahl)
 - . PRAB 1 $\hat{=}$ Typ 0 = 1 Byte
 - . PRAB 2 $\hat{=}$ Typ 1 = 2 Byte
 - . PRAB 3 $\hat{=}$ Typ 2 = 3 Byte
 - . PRAB 4 $\hat{=}$ Typ 3 = 5 Byte

Dialog:

GRUNDRESERVIERUNG PRAB

GEN.-DATEN VOM EPROM

B:	A-PRAB:	L-PRAB:	A-TEL:	L-TEL:
X	XXXX	XXXXX	XXXX	XXXXX
PRAB 1: XXX	}			
PRAB 2: XXX	}	Elementezahl		
PRAB 3: XXX	}	für die 4 Bereiche (1/2/3/5 Byte)		
PRAB 4: XXX	}			

- Grundzuweisung von BSE-Nr., PRAB-Typ und Anzahl Blöcke/Anforderungselemente für die 3 verschiedenen Zeittakte (TAKT 1 $\hat{=}$ 2 sek, 2 $\hat{=}$ 6 sek, 3 $\hat{=}$ 60 sek)

Dialog: (Beispiel für Takt 1)

PRAB TAKT 1(2 SEC)

ANZAHL BLOECKE: XX (d.h. Anzahl Telegramme)

----> max. 8 für TAKT 1 und 3

----> max. 12 für TAKT 2

	FE	TYP	ANZAHL		
00	X	X	XX	}	
01	X	X	XX	}	
02	X	X	XX	}	
03	X	X	XX	}	FE = Funktionseinheit 1 ... 4
04	X	X	XX	}	TYP = PRAB-Typ 0 ... 3
05	X	X	XX	}	ANZAHL = Anzahl Anforderungselemente
06	X	X	XX	}	pro Block (Telegramm)
07	X	X	XX	}	----> max. 80

(Nach Quittierung des letzten Zeichens im TAKT 1 bzw. TAKT 2 besteht Dialogbereitschaft für den folgenden TAKT 2 bzw. TAKT 3.)

A C H T U N G !

Die Verwaltung ist bei Korrektur stets komplett abzuarbeiten, da bei jeder Eingabe Steuerlisten korrigiert werden. Dabei sind von vornherein Reserven zu strukturieren, da sonst der Anschluss der Applikationsfunktionen bei jeder Korrektur angepasst werden muss.

3.3.5.3. Steuerfelder

Diese Funktion realisiert den Aufbau der einzelnen Anforderungstelegramme in den 3 verschiedenen Aktualisierungstakten. Ein Anforderungstelegramm besteht aus n Anforderungselementen zu je 3 Byte.

Die Strukturierung der Anforderungs-Elemente beginnt entsprechend Bild 3.4. Wenn ein Anforderungstelegramm gemäss Pkt. 3.3.5.2 reserviert ist, werden nach der Eingabe von Zeit-TAKT (1...3) und Block-NUMMER (z.B. 0...7) die Funktionseinheit, der PRAB-TYP, die reservierte Anzahl der Anforderungselemente und die PRAB-Anfangsnummer angezeigt.

Die einzugebende BELEGUNG kann kleiner oder gleich der reservierten Anzahl Anforderungselemente sein. Nach Quittierung der Belegung erfolgt die Strukturierung der Anforderungselemente in der vorgegebenen Listen-form: Die Anwahl eines Elementes erfolgt mittels "INS MODE Nr.". Das Weiterblättern in der Liste erfolgt mittels "N". Die Listenlänge entspricht der eingegebenen Belegung. Pro Anforderungselement sind einzugeben:

- DUE-Regime-Nr. (Überwachung auf Zulässigkeit)
- IMEN der BSE (0 ... 255)
- Relativadresse im KOM (0 ... 34)

Für verschiedene Felder des Prozessabbildes sind folgende DUE-Regime-Nr'n. zulässig (siehe Punkt 1.11.1.):

PRAB	Regime-Nr.	Zugriff auf	Relativadr. von Bedeutung
1 Byte	15	IW Aggregate KOM	-
	16	IW-Leit KOM	-
2 Byte	17	IW Bin.Geber	-
	18	1 Byte aus KOM	x
	19	bin. Merker	-
3 Byte	21	IW analoger KOM	-
	22	2 Byte aus KOM	x
	23	analoger Merker	-
5 Byte	25	IW Zähler KOM	-
	26	4 Byte aus KOM	x

(IW = Istwert)

Wenn die Relativadresse ohne Bedeutung ist, kann die Belegung 00 bleiben.

Für den gezielten Zugriff auf Werte (ausser Istwert) im KOM sind zur Festlegung der Relativadresse die Angaben in den Bedienungsanleitungen zur BSE (Teil 2 bzw. Teil 6) zu nutzen. Die maximale IMEN der jeweiligen BSE ist vom Anwender zu beachten.

Seite 121, Pkt. 3.3.7

- Der Satz in der 5. Textzeile ("Die Listen ...") ist ersatzlos zu streichen.
- Text-Ergänzung am Ende der Seite:
"Die Steuerlisten befinden sich auf dem Strukturier-EPROM. Der vollständige Aufbau dieses EPROMs ist in der tabellarischen Uebersicht am Ende des Kapitels 4 beschrieben."

Seite 123

- In der 1. Zeile des Bildes 3.4 wird "Anforderungslisten" ersetzt durch "Elemente".
- Zusatz folgender Bemerkung zum Bild 3.4:
"Das Bild gehört zum Punkt 3.3.5."

Seite 127, Pkt. 3.4.2., letzter Absatz

Text-Ergänzung nach:
"Über Werteingabe ist eine Wertänderung möglich."
"Bei binären Merkern ist Maske und Wert einzugeben."

Seite 132, Pkt. 3.4.4, letzte Zeile

Text-Ergänzung:
"Ebenfalls die Quittierungsmöglichkeit der Fehler."

Seite 133, Pkt. 3.4.5.2, Abschn. Bedienung, 5. Zeile

"PFxx" wird ersetzt durch
"CLEAR/Prt Sc".

Seite 144, Pkt. 3.5, 4. Zeile und
Seite 148, Abschn. VAP Abbruch V, 3. Zeile

"ERASE INP" erhält folgende Fussnote:
"ERASE INP = CTRL + PF4"

Seite 135, Adr. HACO Anwahltaste

muss richtig lauten: "Anwahltaste: 516EH"

Seite 156, Tabelle/Funktionskomplex Adressen

"HACO-Gerät" ist ersatzlos zu streichen, vgl. S.158.

Seite 158, Tabelle/Funktionskomplex Adresse

Text-Ergänzung unter Struktur:
"z.B. 80H für 1. AR,
81H für 2. AR

·
· " " ·
·

Seite 159, Ende

Ergänzung folgender Fussnote:
* Ebenenkennzeichen der ISI; Dieser Block wird nur im Anlauf getestet und nicht im zyklischen Speichertest.

Seite 197, Pkt. 7.6.

zusätzliche Funktion:

"INKEY Tastaturabfrage ohne Warten auf Tastendruck, d.h., es wird die momentan gedrückte Taste bzw. das letzte noch nicht abgeforderte Zeichen geliefert.
 Argument 0 %: kein Echo (Vorzugsvariante)
 Argument 1 %: Eingabe mit BS-Echo"

Seite 202, Pkt. 7.7.11.

Textergänzung:

"Wird eine Eingabe mit 'R' (OFF) abgebrochen, kann das über die Systemvariable ERROR% ausgetestet werden."

Seite 207, Pkt. 7.7.34.

muss richtig lauten:

"OPEN {Gerätekennzeichen},{Bytefeldname}"

Seite 207, Pkt. 7.7.36

Satz "Es wird eine Datei mit Kopfblock gelesen."

muss richtig lauten:

"Soll eine Datei mit Kopfblock (,H) absolut gelesen werden (d.h. auf die im Kopfblock eingetragene Adresse), ist für den dritten Parameter eine Variable mit der Adresse 0000H anzugeben. Die Byteanzahl 4. Param. ist dann ohne Bedeutung."

Seite 211, Pkt. 7.9.2.7.

muss richtig lauten:

"... Die Syntax der Zeitdifferenz entspricht der im Punkt 7.9.2.5."

Seite 213, Pkt. 7.9.5.

Text-Ergänzung:

"Wenn bei falscher Strukturierung der Zugriff auf das Abbild fehlerhaft ist, wird bei den Kommandos >PGET, >AGET, >PADDR, >AADDR, >APUT die Systemvariable ERROR% gesetzt."

Seite 215, Pkt. 7.9.6.3.

Im Beispiel muss es richtig heißen:

```
"DIM F%(3)
>SCS F%
```

oder

```
DIM Fu(7)
>SCS Fu"
```

Seite 215, Pkt. 7.9.7.

bei der Aufzählung ergänzen:

"APUT"

Seite 216, Pkt. 7.10.

Es ist der folgende Punkt 7.11. einzufügen:

7.11. BASIC Programmbeispiel

Am Beispiel eines Programmsystems von 3 VAP sollen die Arbeit mit den wichtigsten BASIC-Anweisungen und das Zusammenwirken verschiedener VAP sowie deren Einbindung in das System gezeigt werden. Das System enthält die VAP:

- Messwertanzeige
- Parametereingabe
- Simulation

Diese müssen in dieser Reihenfolge fortlaufend in die VAP-Steuerliste eingetragen werden. Die Programme werden nach Online-Schaltung mit J (Nr. der Messwertanzeige) gestartet.

Hinweis: Werden alle Programme gemeinsam übersetzt, wird von allen die gleiche Systemvariable ERROR% genutzt. Das ist nur in diesem Beispiel möglich, da ERROR% ohne Unterbrechung sofort durch das jeweilige VAP wieder abgefragt wird.

Messwertanzeige

Das Programm realisiert die Anzeige von 10 Messwerten. Es sind die ersten 10 Werte des PRAB Typ 2.

Im Initialisierungszweig wird das Nachladen des Arithmetikerweiterungsmoduls durch das VAP selbst ermöglicht. Gerümpfehlerrmeldungen (in Textform) werden auf dem Bytefeld ERTX abgelegt und im Fehlerfall über OUTSTR ausgegeben. Es wird die Möglichkeit der strukturierten Programmierung aufgezeigt. Um ein erstmaliges Durchlaufen der WHILE-Schleifen zu ermöglichen, wird I% gesetzt.

Da die Ergänzungsarithmetik ARITHM.ABS eine Datei mit Kopfblock ist, muss als Anfangsadresse eine Variable mit der Adresse 0000H angegeben werden, damit die Datei absolut geladen wird. Die Bytesanzahl ist ohne Bedeutung.

Im Neustartzweig trägt sich das VAP als dialogführendes VAP in das Betriebssystem ein. Damit läuft es nicht weiter, wenn ein anderes VAP als dialogführend eingetragen wird, z. B. das Grundmenü-VAP. Es wird die Messwertanzeige selbst zum zyklischen Restart in Sekundentakt und die Simulation zum Neustart mit zyklischen Restart in Grundtakt (1/3 sec) angemeldet. Da die VAP in aufsteigender Reihenfolge in die Steuerliste eingetragen sind, wird die VAP-Nr. der anderen VAP aus der eigenen bestimmt.

Im Restartzweig erfolgt die Anzeige der Messwerte, wenn das Prozessabbild entsprechend strukturiert ist und wenn deren Status in Ordnung ist.

Des weiteren wird die Taste PFG abgefragt und bei Betätigung die Parametereingabe gestartet (eigene VAP-Nr. + 1). Damit bei laufender Anzeige der Messwerte parallel der Dialog für die Parametereingabe geführt werden kann (d. h., 2 VAP den Bildschirm nutzen), wird der Displaysteuerblock (Cursor-, Fensterinformationen) bei jedem Durchlauf erhalten.

Parametereingabe

Das VAP nutzt parallel zur Messwertanzeige den Bildschirm in einem anderen Fenster. Es hat nur eine Startadresse. Es werden die Messwert-Nr., der Oberwert und der Unterwert angefordert und auf dem AR-AB mit gültigem Status abgelegt. Hier stehen die Werte dem Simulations-VAP zur Verfügung. In diesem Beispiel wäre es auch möglich, die Werte in einem globalen Feld zu übergeben.

Das VAP wird mit OFF, oder auch wenn das AR-AB nicht strukturiert ist, beendet.

Simulation

Zur Simulation wird das Prozessabbild direkt beschrieben. Es muss deshalb auf der gleichen Ebene wie der Datenbereich des VAP liegen (Vgl. Listenbeispiel am Ende des Abschnitts).

Das VAP nutzt selbst den Bildschirm nicht. Es wird die Nutzung von spezifischen Funktionstastenfolgen (Pkt. 1.13.1.2. / 4.1E.) gezeigt. Zur einfachen Auswertbarkeit wird ein Bytefeld auf die Adresse des Eingangspuffers (ANRK) gelegt. Über die Funktionstasten sind drei Arten der Simulation einstellbar:

- PF1 Sprung (Unterwert/Oberwert)
- PF2 Rampe (rauf/runter)
- PF3 Sinus (ein/aus)

Es muss jeweils eine Folgetaste mit der Messwert-Nr. angegeben werden.

Im Neustartsweg wird die Adresse der Funktionstastensteuerliste auf die vorgesehene Betriebssystemadresse gelegt und die Funktionstastensteuerliste in den Grundbereich ungespeichert. Es sind 3 Funktionstasten mit je einer Folgetaste für die Messwert-Nr. vorgesehen.

Die genutzten Felder erhalten definierte Anfangswerte.

Im Restartsweg wird aus der eingegebenen Folgetaste die Messwert-Nr. ermittelt und der Schalter für diesen Wert geschaltet sowie aus der Funktionstaste die Art der Simulation bestimmt. Die Anreiskennung der Funktionstasten wird zurückgesetzt.

Für jeden Messwert wird der zulässige Ober- und Unterwert aus dem Prozessabbild gelesen und bei gültigem Status entsprechend der Art der Simulation der neue Messwert bestimmt. Je nach Schalterstellung wird

- beim Sprung der Ober- oder Unterwert eingestellt,
- bei der Rampe der Wert so lange vergrößert bzw. verkleinert, bis der Ober- bzw. Unterwert erreicht ist und
- die Sinusfunktion ein- oder ausgeschaltet. Das Ergebnis der Sinusfunktion wird auch zwischen Ober- und Unterwert normiert.

Die ermittelten Messwerte werden über eine selbst definierte Funktion in Festkommawerte konvertiert und über Speicherschreiben auf das Prozessabbild geschrieben. Das ist nur für die Simulation erforderlich, da sonst die Werte über die Datenübertragung kommen.

>PROGRAM MESSWERTANZEIGE(MAIST,MANST,MAST)

!-----

```

DIM ERRT(40)
DIM DPLSB(7)
DATA ARITH(11),"ARITHE.ABS "
LABEL NUL:0
DIM NUL:NULL ;! VARIABLE MIT ADRESSE 0000H

! BEIM INITIALISIERUNGSSTART KANN DER ARITHMETIKERWEITERUNGS-
! MODUL NACHGELADEN WERDEN

MAIST: SWITCH CON
FOR I%:=0 TO 40
  ERRT(I%)=' ' ;! FEHLERTEXTFELD LOESCHEN
NEXT
PRINT /,"IST DER ARITHMETIKERWEITERUNGSMODUL GELADEN ? : J"
IF INCHAR(0) <> 'J' THEN DO
  I%='J' ;! ANFANGSWERT F. 1. DURCHLAUF
  WHILE I%='J'
    WHILE I%='J'
      PRINT "SYSTEMDISK IN LW 0 EINLEGEN: J"
      WHILE INCHAR(0)<>'J' ;! WARTEN AUF RESTARTIGUNG
      WEND
      PRINT "BITTE WARTEN"
      OPEN FDO,ERRT ;! GERÄTEZUWEISUNG
      GOSUB BRAUS
      WEND
      IF I% = 'OK' THEN DO
        LOAD FDO,ARITH=NULL,0,H
        ! DAMIT WIRD AUF ADR AUS KOPFBLOCK GELESEN
        GOSUB BRAUS
      DOEND
      CLOSE FDO ;! GERÄTEFREIGABE
      WEND
      IF I%<>'OK' THEN DO
        PRINT "PP3 (SIN) NICHT VERWENDEN: J"
        I%:= INCHAR(0)
      DOEND
    DOEND
  >CONTINUE ;! ENDE INITIALISIERUNGSSTART
  ! NEUSTART FOLGT

! DIESES VAP VERFÜHRT ÜBER DEN BS UND ORGANISIERT DEN START
! DER ANDEREN VAP

MANST: SWITCH CON
CLS
VAPNR%=-PEEK(5402H) ;! EIGENE VAP NR. LESEN
POKE 55KPH,VAPNR% ;! ALS DIALOG VAP EINTRAGEN
COLOR 2 ;! GRÜN
PRINT "MESSWERTE:",/,/
FOR I%:=0 TO 9 ;! STATISCHER BILDANTEIL
  PRINT I%:P1.0
NEXT
>START * CYCLE SEC 1 ;! ANZEIGE ALLE SEC
>CSST VAPNR%+2 CYCLE 1
>START VAPNR%+2 ;! SIMULATION ALLE 1/3 SEC
>CONTINUE ;! ENDE NEUSTARTZWEIG
! RESTARTZWEIG FOLGT

```

```

MARST: >SCS DPLSB=
! DISPLAYSTUERBLOCK SETZEN, DAMIT DIE PARAMETEREINGABE
! (VAPNR+1) AUCH DAS DISPLAY NUTZEN KANN
WINDOW 5,3,20,11 ;! FENSTER F. MESSWERTANZEIGE
FOR IX:= 0 TO 9
>PGET (IX,T2:FP),STAT%,MW ;! PRAB TYP 2 LESEN
IF ERROR% THEN DO
PRINT "NST " ;! FEHLER PRAB NICHT STRUKTUR.
COLOR 5,IX+3,3,1,10H ;! GRUEN UNTERLEGT
ELSE
IF (STAT%.AND.0COH) = 0 THEN DO
PRINT 100*MW:P5.2 ;! WENN STATUS I.O. ANZEIGE IN %
COLOR 5,IX+3,6,1,2 ;! GRUEN
ELSE
PRINT "GST " ;! SONST GESTOERT UND
COLOR 5,IX+3,6,1,1 ;! ROT
DOEND
DOEND
NEXT
>LCS DPLSB=
WHILE INKEY(0) = 98H ;! PPS
>START VAPNR%+1 ;! PARAMETEREINGABE
WEND
>STOP

```

! UNTERPROGRAMM ZUR FEHLERAUSGABE

```

BRAUS: IF ERROR% THEN DO
OUTSTR ERRT=,40 ;! AUSGABE DES FEHLERTEXTES
PRINT "WIEDERHOLEN ? : J"
IX:=-INCHAR(0)
ELSE
IX:='OK'
DOEND
RETURN

```

```

>PROGRAM PARAMETEREINGABE(PANST,PANST,PANST)
!-----

```

! DIE KOMMUNIKATION ZWISCHEN DEN VAP ERFOLGTE UEBER DAS AR-AB

```

PANST: SWITCH CON
WINDOW 0,15,64,10 ;! FENSTER F. PARAMETEREINGABE
CLW
PAIN1: INPUT "NR: " ,NR%
IF ERROR% THEN PAEND ;! WENN OFF
IF NR% < 0 .OR. NR% > 10 THEN PAIN1
PAIN2: INPUT " OBERWERT: " ,OW%
IF ERROR% THEN PAEND ;! WENN OFF
IF (OW% < 0 .OR. OW% > 100) THEN PAIN2
PAIN3: INPUT " UNTERWERT: " ,UW%
IF ERROR% THEN PAEND ;! WENN OFF
IF (UW% < 0 .OR. UW% >= OW%) THEN PAIN3
>APUT (NR% * 2,T1,0,OW%);!PARAMETERUEBERGABE IN AR-AB
IF ERROR% THEN PAEND
>APUT (NR%*2+1,T1,0,UW%)
IF NOT(ERROR%) THEN PAIN1
PAEN1: PRINT " AR-AB NST"
PAEND: >STOP

```



```

>PROGRAM SIMULATION(SINST,SINST,SIRST)
!-----
! ZUR SIMULATION WIRD DAS PROZESSABBILD DIREKT BESCHRIEBEN
! ES MUSS DESHALB AUF DER GLEICHEN EBENE WIE DER DATEN-BEREICH
! DES BASICPROGRAMMS LIEGEN

LABEL ASTLI:5B95H          ;! ADR. STEUERLISTE FUNKTIONST.
LABEL STLI: 6F70H         ;! STEUERLISTE IN GRUNDBEREICH
DATA DSTLI=(6),91H,1,92H,1,93H,1,0FFH
! DATEN F. STEUERLISTE: PF1,PF2,PF3 MIT JE 1 POGETASTE
LABEL AANRK:5B97H         ;! ADR. ZUR AUSWERT.FUNKTIONST.
DIM AANRK:ANRK=(2)
! 3 BYTE: ANREIZKENNUNG, TASTENCODE, FOLGETASTE
DIM SIMA=(9)             ;! ART DER SIMULATION
DIM SALT=(9)             ;! SCHALTER F. SIMULATION
DIM SWE(9)               ;! WERT ZUR SIMULATION

DEFPH INTFK(XI)          ;! KONVERT.INTEGER -> FESTKOMMA
IF XI >= 100 THEN DO
  XI:=32767.
ELSE
  XI:=XI*327.68          ;! 0 ...100 -> .0000 ... .9999
DOEND
FTHEND XI

SINST:  DOKE ADDR(ASTLI),ADDR(STLI)
! ADR. FUNKTIONSTASTENSTEUERLISTE AN SYSTEM UEBERGEBEN
MOVE ADDR(STLI),ADDR(DSTLI),7
!FUNKTIONSTASTENSTEUERLISTE IN GRUNDBEREICH VERSCHIEBEN
FOR J%:=0 TO 9
  SIMA(J%):=1           ;! ART DER SIMULATION: SPRUNG
  SALT(J%):=0           ;! SCHALTER: UNTEN/AUS
NEXT
>CONTINUE

SIRST:  IF ANRK=(0) THEN DO ;! WENN FUNKTIONSTASTE WAR
  SWE%:= ANRK=(2)-30H ;! AUS POGETASTE: MESSWERTNR.
  IF (SWE% >= 0 .AND. SWE% < 10) THEN DO
    SALT=(SWE%):=SALT=(SWE%).XOR. 1
    SIMA=(SWE%):=ANRK=(1)-90H ;! AUS PF ART DER SIMULATION
  DOEND
  ANRK=(0):=0          ;! ANREIZKENNUNG RUECKSETZEN
DOEND
>PADDR(0,T2),PRAB% ;! ADR. PRAB BEGIN BESTIMMEN
FOR J%:=0 TO 9
  >AGET(J%*2,T1),STO%,XO% ;! OBERWERT
  IF NOT(ERROR%) THEN DO
    >AGET(J%*2+1,T1),STO%,XU% ;! UNTERWERT
  DOEND
  IF NOT(ERROR%).OR.(((STO%.OR.STU%).AND.OCOH)=0)THEN DO
    ON SIMA=(J%) GOTO SSPR,SRAMP,SSIN
    ! VERTEILUNG NACH ART DER SIMULATION

SSPR:  IF SALT=(J%)THEN DO ;! **SPRUNG**
  SWH:=-XO%
ELSE
  SWH:=XU%
DOEND
SWE(J%):=SWH
GOTO SEND

```

```

SRAMP:  IF SWE(J%) > X0% THEN DO          ;! **RAMPE**
        SWE(J%) := X0%
        DOEND
        IF SWE(J%) < X0% THEN DO
        SWE(J%) := XU%
        DOEND
        IF SALT=(J%) THEN DO
        IF SWE(J%) < X0% THEN DO
            SWE(J%) := -SWE(J%) + (X0% - XU%) / 50. ;! AUFWAERTS
        DOEND
        ELSE
            IF SWE(J%) > XU% THEN DO
            SWE(J%) := -SWE(J%) - (X0% - XU%) / 50. ;! ABWAERTS
            DOEND
        DOEND
        SWH := -SWE(J%)
        GOTO SEND

SSIN:   IF SWE(J%) >= 3141.59 THEN DO     ;! **SINUS**
        SWE(J%) := 0
        DOEND
        IF SALT=(J%) THEN DO
        SWH := -(SIN(SWE(J%)) * (X0% - XU%) + X0% + XU%) / 2.
        SWE(J%) := -SWE(J%) + .05
        ELSE
        SWH := (X0% + XU%) / 2.
        DOEND

SEND:   DOKE PRAB% + J% * 3 + 1, INTFK(SWH) ;! WERT SETZEN UND
        DOEND
        POKE PRAB% + J% * 3, STO%.OR.STU%.OR.ERROR% * 60H ;! STATUS
        NEXT
        >STOP

END

```

Beispiel für zu strukturierende Listen

- GRUNDMENUE (entsteht beim Uebersetzen)

```

0 GRUNDMENUE
1 STRUKTURIERUNG
2 SYSTEMBEDienung/ANZEIGE
3 HACO
4 EDIT
5 EDITR
6 NESSWERTANZEIGE
7 PARAMETEREINGABE
8 SIMULATION

```

- SYSTEMSTRUKTURIERUNG

. LISTE PRIO 1 (muss strukturiert werden)

```

YAP-NUMMER
0 0
1 1
2 2
3 3
4 6
5 7
6 8

```

. SYSTEMDATEN (entsteht beim Uebersetzen)

MAX-VAP 8

. VAP- STEUERLISTE (entsteht beim Uebersetzen)

	ED	EP	ADR.	ST-BYTE
0	1	1	A570	0000
1	1	1	7021	0000
2	1	1	8149	0000
3	1	1	A750	0000
4	3	3	1000	0000
5	3	3	1010	0000
6	1	2	7000	0000
7	1	2	77??	0000
8	1	2	77??	0000

? - Adresse ergibt sich beim Uebersetzen

- GRUNDRESERVIERUNG PRAB (muss strukturiert sein)

GEN.-DATEN VOM EPROM:

EB:	A-PRAB:	L-PRAB:	A-TXL	L-TXL:
1	2000	1000	3000	1000
PRAB 0:	50			
PRAB 1:	50			
PRAB 2:	120			
PRAB 3:	20			

PRAB TAKT 1(2 SEC.)
ANZAHL BLOECKE 1

	FE	TYP	ANZAHL
00	1	2	10

- GRUNDRESERVIERUNG AR-AB (muss strukturiert sein)

GEN.-DATEN VOM EPROM

EB:	A-MAB:	L-MAB:
1	4000	1000
MAB1:	100	
MAB2:	100	
MAB3:	100	

Seite 219. Pkt. 8.1.4. Bedienung

Textergänzung unter der Menüleiste:

"In dieses Menü kann immer mit der Taste "|<-" zurückgekehrt werden"

Seite 221. Über "Kommandos"

"Kommandotaste "<->" " wird ersetzt durch:

"Kommandotaste "DEL LINE / CTRL + PF3" "

Seite 235. Pkt. 8.2.3. letzter Absatz

"< PF8 NACH LINKS ANSCHLAG >"

wird ersetzt durch "|<-"

Seite 263. Pkt. 8.4.2.1. Grundlagen

Text-Ergänzung:

"Das Anwenderunterstützungsprogramm "APIC", das auf der Systemdiskette bzw. Kassette mitgeliefert wird, bietet dem Anwender einen ON-LINE-lauffähigen Applikationsrechner mit folgenden Funktionen:

- Bildanzeige der selbst erstellten Schemata mit Bildschirm/Tastaturdialog (Meldung mit Bild 1 der Videothek)
- Verwaltung von 30 Bytes der ANBI-Byte-Tabelle zur Veränderung des Anzeigeanfangs und des Anzeigebereichs bei Balken- und Kurzsichtrenddarstellungen
- Aufbau von 20 Trendpuffern mit je 15 Werten strukturierbarer Trendzeit (1 - 255 Sek.) und Zuordnung zur Abbildnummer
- Start des Anwenderprogramms Nr. 9 mit Tastenbedienung PF1 02
Hiermit kann das Anwenderprogramm Nr. 9 mit dem gleichen Bedienschema wie die Bildanwahl gestartet werden.
- Unspeichern der Datentelegramme zur Prozessabbildaktualisierung von 8800H auf 3000H Datenebene 1 (Länge 1 K Byte) nach Rechneranlauf. Mit Hilfe der Strukturierung (siehe Pkt. 3.3.5.) muss die gewünschte Zugriffsbelegung im RAM ab 3000H Datenebene 1 eingetragen werden. Zur Sicherung der Eintragungen wird dann von 3000H - 33FFF ein EPROM programmiert, der auf 8800H Programmebene 2 gesteckt wird (Diskettensicherung ebenfalls möglich). Für das Prozessabbild ist eine Grundreservierung mit folgenden Längen eingetragen:

PRAB 0:	20
PRAB 1:	20
PRAB 2:	100
PRAB 3:	20

Die aktuellen Telegrammlängen sind mit 0 strukturiert, damit erfolgt keine Datenübertragung. Das Applikationsrechnerabbild ist mit je 100 Werten strukturiert.

- Umschaltung des Applikationsrechners auf ON-LINE nur nach Rechneranlauf
- Zyklische Anzeige der Uhrzeit der Fehlermeldungen (AR-Statusbyte) in der 1. Bildschirmzeile
- Beispielvideothek für Datenebene 2 mit einem Inhaltsverzeichnis (Bild 1) und mit 6 Kurzzeittrends (Bild 2) zur zeitlichen Darstellung von analogen Größen aus dem Prozessabbild."

Seite 266, Streichung der VAP-Steuerliste

Durch die Erweiterung haben sich die Startadressen geändert. Das Programmsystem steht unter dem Namen APIC.ABS zusammen mit der RAM-Version des Generierdatensatzes (Datei APIC000.ABS) auf der Diskette bzw. Kassette bereit.

Seite 267, Streichung der Prüfsummen

Durch die Erweiterungen sind die Prüfsummen ungültig.

Seite 267, Strukturierung Trendsueordnung und Trendszeit

Die Eingabe PF1 01 bewirkt den Aufruf der Anzeige der Trendsseiten in folgender Tabellenform:

TR-Nr.	ZYKL.	TYP	NR.
1	60	2	1
2	60	2	2
3	60	2	3
4	60	2	4
5	60	2	5
6	60	2	6
7	60	2	7
8	60	2	8
9	60	2	9
10	60	2	10
11	60	2	11
12	60	2	12
13	60	2	13
14	60	2	14
15	60	2	15
16	30	2	16
17	30	2	17
18	30	2	18
19	30	2	19
20	30	2	20

KORREKTUR.
10 2 2 20

Der Zyklus ist die Zeit zwischen 2 Trendwerten in Sekunden. Mit Hilfe der Korrektur kann jeder Trend verändert werden (Trendzeit: 1-255 Sekunden, Typ: 2 oder 5, Nr.: Prozessabbild oder Applikationsrechnerabbild). Die Korrektur wird mit dem Cursor geführt und jeweils mit der ENTER-Taste quittiert. In Anwenderhilfssystem ist die oben gezeigte Grundbelegung im Bereich Programmebene 2 8000H bis 83FFH hinterlegt. Der Bereich wird ungespeichert und bei Korrekturen (Trendzeit, Zuordnung) ständig aktualisiert (bei Programmablage auf RAM). Wenn der Applikationsrechner mit der korrigierten Belegung anlaufen soll, muss der Bereich der Programmebene 2 8000H bis 83FFH auf Diskette/Kassette oder EPROM gesichert und vor dem Anlauf geladen bzw. gesteckt werden.

Seite 267, Start Anwenderprogramm Nr. 9

Die Eingabe PF1 02 bewirkt den Neustart des Anwenderprogramms Nr. 9. Damit wird die Möglichkeit angeboten, ein Anwenderprogramm für spezielle Zwecke (z. B. Simulationseingaben, Protokollausgaben usw.) zu starten. Der zyklische Restart wird nicht angemeldet. Er muss wie bei anderen Programmen über die Anmeldeleiste oder durch das Programm (Kommando) selbst organisiert werden.

Seite 269, Erweiterung Kommando-VAP

Folgende Funktionen wurden zusätzlich aufgenommen:

- Strukturierung der Trendzeit und der Zuordnung Trend/Prozessabbild-Nr. mit Funktionstaste PF1 01
- Neustart Anwenderprogramm Nr. 9 mit Funktionstaste PF1 02

Seite 269, VAP-Datenbereitstellung

Da hier eine Programmänderung durchgeführt wurde, ist der alte Text ungültig. Das Programm realisiert durch die Erweiterung die Aktualisierung von max. 20 Trendpuffern mit zugeordneten 2 Byte-Festkommawerten des Prozess- oder Applikationsrechnerabbildes im strukturierten Zeittakt (1-255 Sekunden). Die Trendzeit und die Zuordnung sind strukturierbar.

Seite 275, 7. Zeile von unten

"blinkt invers" wird ersetzt durch:
"blinkt rot"