

C 11B



**Technische Dokumentation**

**Baueinheiten der Prozeßleitebene**

**BP 31/1, BP 31, WRE/1, WRE/2, KE, DSS**

**Teil 9 B: Schnittstellenbeschreibung  
für Anwenderprogramme  
Wartenrechnereinheit WRE/1  
und WRE/2**



## Anlage 1 : Kennblätter der Unterprogramme

Kennblatt- Nr.	UP- Name	Kurscharakteristik	UP- Nr.	Seite
1.		Aufruforganisation von Verarbeitungs- programmen		
1. 1.	APAMNS	Anmeldung Neustart sofort	2	86
1. 2.	APAMNE	Anmeldung Neustart nach Bootzeit	3	87
1. 3.	APAMNR	Anmeldung Neustart nach Relativzeit	4	88
1. 4.	APAMRS	Anmeldung Restart zyklisch sofort	5	89
1. 5.	APABPG	Abmeldung Programm	6	90
1. 6.	APLEWI	Lesen des Wiederanlaufbits	7	91
1. 7.	APRUVF	Rücksprung aus dem VAP zum SPS	8	92
1. 8.	APLESP	Lesen des Steuerbytes für Parametersatz	9	93
1. 9.	APAMRE	Anmeldung Restart zyklisch	1	94
1.10.	APWART	Warten eine Zeitscheibe	0	95
2.		BS-Funktionen		
2. 1.	BSSTT	Schlüsselschaltertest	12	96
2. 2.	BSDAT	Ausgabe des Datums	13	97
2. 3.	BSUHR	Ausgabe der Uhrzeit	14	98
2. 4.	BSEULE	Lesen Uhrzeit	15	99
2. 5.	BSDALE	Datum lesen	16	100
2. 6.	BSEBE1	Ebeneninstellung	17	101
2. 7.	BSEBRE	Ebene retten	19	103
2. 8.	BSEBRU	Ebene rückstellen	20	104
2. 9.	BSREZ	Bestimmung Relativzeit aus Echtzeit	21	105
2.10.	BSWIAN	Mitteilung VAP-Nr. für Wiederan- lauf der WRE	18	106
3.		Zugriff auf prozessorientierte Daten		
3. 1.	PDPALE	Prozessabbild Lesen	24	107
3. 2.	PDPAAD	Prozessabbild Adressenberechnung	25	108
3. 3.	PDPIAD	Berechnung problemorientierter Information aus Adresse	26	109
3. 4.	PDWALE	Lesen Wartenrechnerabbild	27	111
3. 5.	PDWASR	Schreiben Wartenrechnerabbild	28	112
3. 6.	PDWAAD	Berechnung Wartenrechnerabbild- Adresse	29	113
3. 7.	PDBINP	Berechnung BSE, IMEN aus n-POM	30	114
3. 8.	PDEZDI	Zugriff auf Dimensionstyp	31	115
4.		Arithmetik- und Konvertierungs-UP'e		
4. 1.	GKADD	Addition zweier GK-Zahlen	37	116
4. 2.	GKSUB	Subtraktion zweier GK-Zahlen	38	117
4. 3.	GKMUL	Multiplikation zweier GK-Zahlen	39	118
4. 4.	GKDIV	Division zweier GK-Zahlen	40	119
4. 5.	GKSQU	Quadratbildung einer GK-Zahl	41	120
4. 6.	GERVS	Kehrwert einer GK-Zahl	42	121
4. 7.	GKABS	Betragsbildung für eine GK-Zahl	43	122
4. 8.	GKPLOAT	Konvertierung 16 Bit - Zweier- komplement in Gleitkomma- darstellung	44	123

Kennblatt- Nr.	UP- Name	Kurzcharakteristik	UP- Nr.	Seite
4. 9.	GKTRUNC	Konvertierung Gleitkommaformat in	45	124
	GKROUND	16 Bit - Zweierkomplement (ganz- zählig) rundend/abschneidend	46	124
4.10.	KOUMAG	Zahlenkonvertierung 1	49	125
4.11.	KOUMGA	Zahlenkonvertierung 2	50	127
4.12.	KOPUTI	Ausgabekonvertierung einer binä- ren Gleitkommazahl	53	128
4.13.	KOAINI	Ausgabekonvertierung Integer in eine SIF 1000-Zeichenkette	54	130
4.14.	KONOBA	Ausgabekonvertierung von Analog- werten in eine SIF 1000-Zeichenkette	55	131
4.15.	KOAKBI	Ausgabekonvertierung eines maskier- ten Binärwertes in eine SIF 1000- Zeichenkette	56	132
4.16.	KOAKZP	Ausgabekonvertierung eines Zähl- wertes des Prozessabbildes in eine SIF 1000-Zeichenkette	57	133
4.17.	KOPUTWA	Ausgabekonvertierung einer binä- ren GK-Zahl des WR-Abbildes in eine SIF 1000-Zeichenkette	58	134
4.18.	KOGET	Eingabekonvertierung einer dezimalen Gleitkommazahl	59	136
4.19.	KOSINT	Eingabekonvertierung einer SIF 1000- Zeichenkette in eine Integer-Zahl	60	137
4.20.	KOZAGK	Konvertierung eines Zählwertes in eine binäre GK-Zahl	51	138
4.21.	INMU11	Integermultiplikation	62	139
4.22.	INDI21	Integerdivision	63	140
5.		Kommunikations-UP'e		
5. 1.	KMAUTA	Textausschrift aus Textspeicher auf Monitor bzw. anderen Speicherbereich	65	141
5. 2.	KMLOEM	Löschung des Bildbereiches von 2. - 29. Zeile des Farbmonitors	66	145
5. 3.	KMLOBB	Löschung von Bereichen auf dem Farbmonitor	67	146
5. 4.	KMTAPL	Übergabe des Tastaturpuffers aus dem System-RAM (Tastaturpuffer Lesen)	68	147
5. 5.	KMTAPS	Tastaturpuffer schreiben	69	149
5. 6.	KMINFO	Ausgabe Informations- und Melde- puffer	70	150
5. 7.	KMCODE	Codewortvergleich	71	153
5. 8.	KMINFM	Ausgabe Informations- und Melde- puffer mit Löschung des Übergabe- bereiches	22	154
5. 9.	DUROG	Organisation des Datenaustausches zwischen WRE und einem Fremdrechner	10	155
6.		Routinen zur DÜ und E/A		
6. 1.	DUSA	Anmeldung und Steuerung der Daten- Übertragung von Prozessdaten (KOM-Daten)	74	157
6. 2.	DULAPI	Laden Informationspuffer PIPU durch PDVP	75	163

Kennblatt- Nr.	UP- Name	Kurzcharakteristik	UP- Nr.	Seite
6. 3.	DULOPA	Bereitstellen Anforderungspuffer PSR-Dialog (Datenquerverkehr)	76	165
6. 4.	DUABPD	Abmelden des dialogführenden Rahmenprogrammes (Datenquerverkehr)	77	167
6. 5.	EAST	Steuerung der Ein-/Ausgabe (E/A-Anforderungen an die E/A- Geräte)	78	168
6. 6.	EAGFG	Freigabe Ein-/Ausgabegeräte	79	175
6. 7.	DUSZYK	Zeigerlesen zu Mitteilungen aus der Systemkommunikation	73	176
6. 8.	DUSYKO	Lesen des Änderungspuffers zu Mitteilungen aus der System- kommunikation an BSE'n	80	177

AFAMNS

UP-Nr. 2

1.1.

UP Anmeldung Neustart sofort

1.1. Programmname : AFAMNS1.2. Funktion des Programmes:

Mit dem JP AFAMNS kann durch ein Verarbeitungsprogramm ein sofortiger Neustart eines Verarbeitungsprogrammes angemeldet werden.

1.3. Schnittstellenbeschreibung1.3.1. Eingangsgrößen : B := VAP-Nr.

1.3.2. Ausgangsgrößen : Im Statusbyte des anzumeldenden VAP wird das Neustartbit gesetzt (80H). Der Zykluszähler im Anmelderegister wird Null.

1.3.3. Unveränderte Register : IX, IY, B1.4. Fehler : CY=1 wenn VAP-Nr.>591.5. Rechenzeit : 0,405 ms2. Beispiel: Anmeldung VAP 5 zum Neustart

```
LD    B,5
RST  10H
DEFB  2 ; AFAMNS
```

AFAMNEUP-Nr. 31.2.

UP Anmeldung Neustart nach Echtzeit

1.1. Programmname : AFAMNE1.2. Funktion des Programmes :

Mit Hilfe des UP AFAMNE kann aus einem VAP mittels Echtzeit-angabe ein VAP mit Std.-Min.-Vorgabe gestartet werden. Dieser Neustart ist mit einer Genauigkeit von einer Minute festlegbar.

1.3. Schnittstellenbeschreibung

1.3.1. Eingangsgrößen: B := VAP-Nr.  
 DE := Startzeit (D:=Stunden,  
 E:=Minuten je als  
 Integerangabe)

1.3.2. Ausgangsgrößen:

Im Statusbyte des anzumeldenden VAP wird das Neustartbit gesetzt (80H). In den Zyklusähler des Anmelderegisters des VAP wird die Relativzeit vom Anmeldezeitpunkt bis zur Startzeit in Minuten eingeschrieben.

1.3.3. Unveränderte Register: IY, B

1.4. Fehler : CY = 1 bei VAP-Nr. >59  
 Z = 1 bei Fehler in der Zeitangabe  
 (Std.  $\geq$  24 , Min.  $\geq$  60)

1.5. Rechenzeit : 0,674 ms2. Beispiel:

Anmeldung VAP 33 zum Neustart um 13,30

```
LD   B,33
LD   D,13 ; Std.
LD   E,30 ; Min.
RST  10H
DEFB 3 ; AFAMNE
```

-----APAMNR-----UP-Nr.: 4-----1:3-----

UP Anmeldung Neustart nach Relativzeit

1.1. Programmname : APAMNR

1.2. Funktion des Programmes :

Das UP APAMNR kann von einem VAP aus dem Neustart eines VAP nach Relativzeitangabe initiieren. Die Relativzeit kann in Minuten oder Sekunden vorgegeben werden.

1.3. Schnittstellenbeschreibung :

1.3.1. Eingangsgrößen: B := VAP-Nr.  
 DE := Relativzeit (1...32767)  
 DE: Bit 15 = 0 Relativzeit in Sekunden  
 Bit 15 = 1 Relativzeit in Minuten

1.3.2. Ausgangsgrößen:

Im Statusbyte des anzumeldenden VAP wird das Neustartbit gesetzt (80H). In den Zykluszmähler des Anmelderegisters wird die Relativzeit eingetragen mit der Kennung für Minuten- bzw. Sekundenangabe.

1.3.3. Unveränderte Register: IY, B

1.4. Fehler: CY= 1 wenn VAP-Nr. >59  
 Z = 1 bei Relativzeit > 32767

1.5. Rechenzeit: 0,674 ms

2. Beispiel:

Anmeldung VAP 33 zum Neustart nach genau 12 h

```
LD   DE, 02D0H
LD   B, 33
RST  10H
DEFB 4           ; APAMNR
```

----- APAMRS                      UP-Nr.: 5                      1.4. -----

UP Anmeldung Restart zyklisch sofort

1.1. Programmname : APAMRS

1.2. Funktion des Programmes :

Mittels UP APAMRS kann der zyklische Restart eines Verarbeitungsprogrammes gemäß einer Zykluszeit, die in Minuten oder Sekunden angegeben werden kann, ausgelöst werden. Der Restart und die neue Zykluszeit werden sofort wirksam. (Vgl. hierzu im Gegensatz APAMRE.)

Anmerkung: Ist das Verarbeitungsprogramm dem Zeitszyklus von  $1/3 \text{ s}$  zugeordnet, wird ein zyklischer Restart mit einer Zykluszeit von  $1/3 \text{ s}$  realisiert. Die in DE eingezeichnete Zykluszeit hat keine Bedeutung.

1.3. Schnittstellenbeschreibung :

1.3.1. Eingangsgrößen : B        := VAP-Nr.  
                                   DE        := Zykluszeit (1...32767)  
                                   DE: Bit 15 = 0    Zykluszeit in Sekunden  
   Bit 15 = 1    Zykluszeit in Minuten

1.3.2. Ausgangsgrößen :

Im Statusbyte des anzumeldenden VAP wird 40 H eingetragen. Ausserdem wird die Zykluszeit im Anmelderegister in den Voreinstellwert-Bytes eingetragen und die Zyklusähler-Bytes werden Null gesetzt.

1.3.3. Unveränderte Register : IY

1.4. Fehler : CY= 1 wenn VAP-Nr. > 59  
                                   S= 1 wenn Zykluszeit > 32767

1.5. Rechenzeit: 0,432 ms

2. Beispiel:

Anmeldung VAP 33 zum zyklischen Restart mit Zykluszeit 1 1/2 min.  
   LD        DE, 90  
   LD        B, 33  
   RST      10H  
   DEPB     5            ; APAMRS

AFABPG

UP-Nr.: 6

1.5:

UP Abmeldung von Verarbeitungsprogrammen

1.1. Programmname : AFABPG1.2. Funktion des Programmes:

Mit dem UP AFABPG kann aus einem beliebigen VAP heraus ein Neustart bzw. ein zyklischer Restart eines VAP abgemeldet werden. Auch bei einer Unterbrechung durch DUE oder E/A erfolgt keine Fortsetzung des abgemeldeten Programmes.

1.3. Schnittstellenbeschreibung :1.3.1. Eingangsgrößen : B := VAP-Nr.1.3.2. Ausgangsgrößen :

Das Statusbyte des abzumeldenden VAP wird Null (00H) gesetzt.

1.3.3. Unveränderte Register: IX, IY1.4. Fehler: CY=1 wenn VAP-Nr. >591.5. Rechenzeit: 0,304 ms2. Beispiel: Abmeldung VAP 5

```
LD      B, 5
RST    10H
DEFB   6      ; AFABPG
```

-----APLEWI-----UP-Nr.: 7-----1.6.-----  
 Lesen des Wiederanlaufbits

1.1. Programmname : APLEWI

1.2. Funktion des Programmes :

Mittels UP APLEWI können VAP's aus dem Statusbyte ihres Anmelderegisters das Wiederanlaufbit lesen. Dieses Bit wird mit dem Lesevorgang zurückgesetzt.

1.3. Schnittstellenbeschreibung :

1.3.1. Eingangsgrößen: keine

1.3.2. Ausgangsgrößen: CY=0 Wiederanlaufbit nicht gesetzt  
 CY=1 Wiederanlaufbit war gesetzt

1.3.3. Unveränderte Register : IX, IY

1.4. Fehler: Z=1 bei VAP-Nr. > 59

1.5. Rechenzeit: 0,272 ms

APRUVP

UF-Nr.: 8

1.7

-----  
Rücksprung aus dem VAP zum SPE

1.1. Programmname : APRUVP

1.2. Funktion des Programmes :

APRUVP stellt die zentrale Einsprungstelle ins Steuerprogramm dar. Alle VAP sind mit Ansprung über diese Marke zu beenden.

1.3. Schnittstellenbeschreibung :

1.3.1. Eingangsgrößen: keine

1.3.2. Ausgangsgrößen: Rücksprung ins Steuerprogramm

1.3.3. Unveränderte Register: IX, IY, BC, DE, AF, HL

1.4. Fehler: keine

1.5. Rechenseit : -

-----APLESP-----UP-Nr.: 9-----1.8.-----

UP Lesen des Steuerbytes für Parametersatz

1.1. Programmname: APLESP

1.2. Funktion des Programmes:

Das UP APLESP liest für das aufrufende VAP aus der Programm-  
adressenliste das Steuerbyte für den Parametersatz.

1.3. Schnittstellenbeschreibung

1.3.1. Eingangsgrößen : keine

1.3.2. Ausgangsgrößen : A := Steuerbyte des Parametersatzes

1.3.3. Unveränderte Register : BC, IX, IY,

1.4. Fehler: keine

1.5. Rechenzeit: 0,164 ms

APAMRE

UP-Nr.: 1

1.9:

## Anmeldung Restart zyklisch

1.1. Programmname : APAMRE1.2. Funktion des Programmes :

Mit dem UP APAMRE kann durch ein Verarbeitungsprogramm der zyklische Restart eines VAP gemäß einer Zykluszeit, die in Minuten oder Sekunden angebar ist, ab Zyklusende initialisiert werden. Auf diese Weise ist eine Änderung des Aktivierungsszyklus für ein VAP möglich, ohne dass die zum Aufrufzeitpunkt von APAMRE aktuelle (alte) Zykluszeit sofort unterbrochen wird. (Vgl. hierzu im Gegensatz APAMRS.)

1.3. Schnittstellenbeschreibung :

1.3.1. Eingangsgrößen : B := VAP-Nr.  
 DE := Zykluszeit (1...32767)  
 DE:Bit 15=0 Zykluszeit in Sekunden  
 Bit 15=1 Zykluszeit in Minuten

1.3.2. Ausgangsgrößen :

Im Statusbyte des anzumeldenden VAP wird das Bit 6 (Anmeldebit Restart) gesetzt, die restlichen Bits bleiben unbeeinflusst. Die Zyklusähler-Bytes des Anmelderegisters bleiben unverändert, während die Zykluszeit aus DE in die Voreinstellwert-Bytes eingetragen werden.

1.3.3. Unveränderte Register: IY1.4. Fehler: CY=1 wenn VAP-Nr. > 591.5. Rechenzeit: 0,442 ms2. Beispiel: Anmeldung VAP 15 nach 1 min zum Restart

```
LD B, 15
LD DE, 8001H
RST 10H
DEFB 1 ; APAMRE
```

-----APWART-----UP-Nr.: 0-----1.10-----

Warten eine Zeitscheibe

1.1. Programmname : APWART

1.2. Funktion des Programmes :

Das UP APWART ermöglicht in dem aufrufenden VAP die Unterbrechung der Abarbeitung des VAP's in einer Zeitscheibe (d. h. 330 ms). Das Steuerprogramm setzt das VAP hinter den UP-Aufruf dann fort.

1.3. Schnittstellenbeschreibung :

1.3.1. Eingangsgrößen : keine

1.3.2. Ausgangsgrößen : In Statusbyte des Anmelderegisters  
werden die Bits  
WA: B:5=0  
RU: B:4=1

gesetzt. Die übrige Belegung bleibt unverändert. Ausserdem wird für das Steuerprogramm die Fortsetzungsadresse in der Anmelde-  
registerliste abgelegt.

1.3.3. Unveränderte Register: IY

1.4. Fehler: keine

1.5. Rechenzeit: 0,3 ms

2. Beispiele: RST 10H  
DBFB 0 ; APWART

3. Bemerkung: Das UP APWART darf nicht in VAP's im Zyklus 3  
aufgerufen werden.

BSSTT

UP-Nr. 12

2.1

## Schlüsselschaltertest

1.1. Programmname : BSSTT1.2. Funktion des Programmes :

Dieses UP testet, ob Schlüsseltaster gesetzt oder nicht gesetzt ist. Diese Information wird durch das CY-Flag übergeben.

CY = 0 Schlüsseltaster nicht gesetzt  
CY = 1 Schlüsseltaster gesetzt

1.3. Schnittstellenbeschreibung1.3.1. Eingangsgröße : keine1.3.2. Ausgangsgröße : CY1.3.4. Unveränderte Register : HL, DE, BC, IX, IY1.4. Fehler : keine1.5. Rechenzeit: 0,15 ms

----- BSDAT

UP-Nr.: 13

2.2  
-----

## Ausgabe des Datums

1.1. Programmname : BSDAT1.2. Funktion des Programmes :

Dieses Programm ermöglicht die Ausgabe des Datums als SIF 1000- Zeichenkette ab angegebener Adresse. Zulässig ist der Bildspeicher von Adresse 0F040H - 0F73FH und der Speicherbereich des Anwenders. Wählt man als Ausgabeadresse eine Adresse des Bildspeichers, kann das Datum direkt auf Bildschirm abgebildet werden. (Voraussetzung : Aufruf UP KML0BB; d. h. Löschen Vordergrund und Setzen einer Hintergrundfarbe)

1.3. Schnittstellebeschreibung1.3.1. Eingangsgrossen : IX:=Adresse 1.SIF-Zeichen1.3.2. Ausgangsgrossen : Ausgabe der Uhrzeit als SIF 1000-  
Zeichenketten  
IX:=IX+81.3.3. Unveränderte Register : keine1.4. Fehler: CY=1; Ausgabeadresse nicht im Bereich Bildwieder-  
hol-speicher Adresse  
0F040H bis 0F73FH oder Anwenderbereich1.5. Rechenzeit: 1,5 ms bei Ausgabe auf Monitor2. Beispiel:

Ausgabe Datum ab Bildschirmadresse F200H

Datum 4.3.86

LD IX, 0F200H

RST 10H

DEFB 13

; BSDAT

ab Adresse 0F200H (30, 34, 2E, 30, 33, 2E, 36, 36)H

BSUHR

UP-Nr. 1 14

2.3

## Ausgabe der Uhrzeit

1.1. Programmname : BSUHR1.2. Funktion des Programmes :

Dieses Programm ermöglicht die Ausgabe der Uhrzeit als SIF 1000- Zeichenkette ab angegebener Adresse. Zulässig ist der Bildspeicher von Adresse 0F040H - 0F73FH und der Speicherbereich des Anwenders. Wählt man als Ausgabeadresse eine Adresse des Bildspeichers, kann die Uhrzeit direkt auf Bildschirm abgebildet werden. (Voraussetzung: Aufruf UP KIL0BB; d. h. Löschen Vordergrund und Setzen einer Hintergrundfarbe).

1.3. Schnittstellenbeschreibung1.3.1. Eingangsgrößen : IX : Adresse 1. SIF-Zeichen1.3.2. Ausgangsgrößen : Ausgabe der Uhrzeit als SIF 1000-  
Zeichenkette  
IX: IX+81.3.3. Unveränderte Register : keine1.4. Fehler: CY=1; Ausgabeadresse nicht im Bereich Bildwiederholungspeicher  
Adresse 0F040H bis 0F73FH oder Anwenderbereich1.5. Rechenzeit: 1,5 ms bei Ausgabe auf Monitor2. Beispiel:

Ausgabe Uhrzeit ab Bildschirmadresse F200H

Uhrzeit 2:04:10

LD IX, 0F200H

RST 10H

DEPB 14

; BSUHR

ab Adresse 0F200H (30, 32, 3A, 30 31, 3A, 31, 30)H

BSEULE

UP-Nr.: 15

2.4

## Lesen Uhrzeit

1.1. Programmname : BSEULE1.2. Funktion des Programmes :

Mit Hilfe dieses UP's ist es möglich, die Echtzeituhr zu lesen.

1.3. Schnittstellenbeschreibung1.3.1. Eingangsgrößen : keine1.3.2. Ausgangsgrößen : Werte als Integerzahl

H : Stunde

L : Minute

D : Sekunde

1.3.3. Unveränderte Register : IX, IY, A1.4. Fehler : keine1.5. Rechenzeit: 0,227 ms2. Beispiel

Lesen der aktuellen WRE-Uhrzeit : 10:12:14 Uhr

RST 10H

DEPB 15 ; BSEULE

| |

H:= OAH ; Stunden

L:= OCH ; Minuten

D:= ODH ; Sekunden

BSDALE

UP-Nr.: 16

2.5

-----  
Datum Lesen1.1. Programmname : BSDALE1.2. Funktion des Programmes :

Mit Hilfe dieses UP's ist es möglich, das Datum zu lesen.

1.3. Schnittstellenbeschreibung1.3.1. Eingangsgrößen : keine1.3.2. Ausgangsgrößen : Werte als Integerzahl

H : Tag

L : Monat

D : Jahr

1.3.3. Unveränderte Register : IX, IY, A1.4. Fehler: keine1.5. Rechenzeit: 0,227 ms2. Beispiel:

Lesen des aktuellen WRE-Datums: 1.02.86

RST 10H

DEFB 16 ; BSDALE

||

H:= 01 ; Tag

L:= 02 ; Monat

D:= 86 ; Jahr

BSEBEI

UP-Nr.: 17

2.6.

## Ebeneneinstellung

1.1. Programmname : BSEBEI

1.2. Funktion des Programmes :

Die Speicherbereiche mit den Adressen 1000H und 7000H sind in der WRE in Ebenen zu je 16 kByte RAM geschaltet, um die Gesamtspeicherkapazität zu erhöhen.

Speicherstruktur :    1000 - 4FFFH                    7000 - AFFFH

n 0	-----	}*)	m 0	-----	}*)
1	-----		1	-----	
2	-----		2	-----	
3	-----		3	-----	
			4	-----	
			5	-----	}*)

\*) durch Betriebssystem belegt

Der 1000H-Speicherbereich kann in 4 und der 7000H-Speicherbereich in 6 Ebenen geschaltet werden. Für den Anwender stehen nur die Ebenen 2 - 4 im 1000H-Bereich und 7 - 9 im 7000H-Bereich zur Verfügung. Ebeneneinstellungsforderungen auf betriebssystemverwalteten Ebenen werden mit Fehlermeldung abgewiesen.

Aufbau Steuerbyte EBCO : | m | n |

	-----	Ebenencodierung
	-----	1000H-Bereich $1 \leq n \leq 3$
	-----	Ebenencodierung
	-----	7000H-Bereich $2 \leq n \leq 4$

Der Anwender hat mit diesem UP die Möglichkeit, aus einem Verarbeitungsprogramm auf Datenbereiche zuzugreifen, die auf verschiedenen Ebenen stehen. Um dies zu ermöglichen, müssen die Ebenen der 2 Speicherbereiche getrennt voneinander schaltbar sein. Dies ist durch entsprechende Festlegung von m, n möglich.

n > 3 Ebeneneinstellung des 1000H-Bereiches wird nicht verändert

m > 5 Ebeneneinstellung des 7000H-Bereiches wird nicht verändert

1.3. Schnittstellenbeschreibung

1.3.1. Eingangsgrößen : A : &lt;EBCO&gt;

1.3.2. Ausgangsgrößen : Einstellung der Ebene durch Ausgabe auf KOPPEL-PIO der ZRE (Port A / B)

1.3.3. Unveränderte Register: IY

BSEBEI

UP-Nr. 17

BE der PLB Teil 9  
Blatt 2, 2.6

- 
- 1.4. Fehler : CY=0    korrekte Belegung von EBCO  
              CY=1    unkorrekte Belegung von EBCO liegt dann  
                          vor, wenn  $n=0$  bzw.  $n<2$  oder  $= 5$
- 1.5. Rechenzeit: 0,694 ms
2. Beispiel:

Eingestellte Ebene 2,9 nach UP-Aufruf soll Ebene 3,9 gestellt sein

LD    A, 72H    bzw.    A, 42H  
RST    10H  
DEFB    17        ;    BSEBEI

BSEBRE

UP-Nr.: 19

2.7.

## Ebene retten

1.1. Programmname : BSEBRE1.2. Funktion des Programmes :

Dieses Programm schreibt die Koppel-PIO Belegung der ZRE Port A und Port B auf eine Merzkelle. Die Belegung wird dabei nicht verändert.

1.3. Schnittstellenbeschreibung1.3.1. Eingangsschleifen : keine1.3.2. Ausgangsschleifen : Beschreiben Merzkelle zur Ebenenrettung1.3.3. Unveränderte Register : DE, BC, IX, IY1.4. Fehler: keine1.5. Rechenzeit: 0,135 ms2. Beispiel: -

3. Bemerkung: Bei mehrmaligem Aufruf hintereinander steht stets nur die zuletzt gerettete Belegung in der Merzkelle.

BSEBRU

UP-Nr.: 20

2.8.

## Ebene rückstellen

1.1. Programmname : BSEBRU1.2. Funktion des Programmes :

Dieses Programm liest, die durch das UP BSEBRE gespeicherte  
MZEB und stellt die alte Ebene wieder ein.

1.3. Schnittstellenbeschreibung1.3.1. Eingangsgrößen : keine1.3.2. Ausgangsgrößen : Rückstellen der PIO-Belegung Daten  
Port A und Port B (Einstellen der  
alten Ebene)1.3.3. Unveränderte Register : BC, DE, IX, IY1.4. Fehler : keine1.5. Rechenzeit: 0,135 ms2. Beispiel: -3. Bemerkung : siehe UP BSEBRE

BSREZ

UP-Nr.: 21

2.9.

## Bestimmung Relativzeit aus Echtzeit

1.1. Programmname : BSREZ

1.2. Funktion des Programms :

Das Unterprogramm BSREZ ermittelt zu einer vorgegebenen Zeit  $t_1$ , s. B. Startzeit eines VAP, die Relativzeit  $t_R$  von Stand der Echtzeit  $t_0$  zum Aufrufzeitpunkt von BSREZ mit  $t_R = t_1 - t_0$  (in Minuten).  $t_1$  darf maximal 23:59 Std. nach  $t_0$  liegen. Liegt  $t_1$  am Folgetag, so wird dies vom UP erkannt und berücksichtigt, wenn die Zahlenangabe von  $t_1 < t_0$  ist.

1.3. Schnittstellenbeschreibung

1.3.1. Eingangsgrößen: D: Stunden STD } Zielzeit  $t_1$   
 E: Minuten MIM }

1.3.2. Ausgangsgrößen:

DE : Zielzeit  $t_1$  (unverändert)  
 IX : Relativzeit  $t_R$  in Minuten

1.3.3. Unveränderte Register : IY,

1.4. Fehler : Z=1 bei falscher Zeitangabe mit STD > 23 oder  
 MIN > 59 (Dez.)

1.5. Rechenzeit: 0,392 ms

2. Beispiel:

Zielzeit: 12.15 Uhr  
 Echtzeit: 12.10 Uhr  
 LD D, 12  
 LD E, 15  
 RST 10H  
 DEFB 21 ; BSREZ

Berechnung der Relativzeit IX:= 5 min.

BSWIAN

UP-Nr.: 10

2.10

Mittellung VAP-Nr. für Wiederanlauf der WRE

1.1. Programmname: BSWIAN1.2. Funktion des Programmes:

Das UP legt die mitgeteilte VAP-Nr. auf eine Merzkelle ab. Bei Rechnerwiederanlauf in die Betriebsart OH wird dieses VAP zum Neustart mit gesetztem Wiederanlaufbit vom Betriebssystem gestartet.

1.3. Schnittstellenbeschreibung:1.3.1. Eingangsgrößen: A:= VAP-Nr. (0 ... 59)1.3.2. Ausgangsgrößen: Merzkelle := A1.3.3. Unveränderte Register: alle

1.4. Fehler: CY=0 Fehler VAP-Nr. > 59  
CY=1 keine Fehler

2. Beispiel: -

PDPALÉ

UP-Nr.: 24

3.1.

## Prozessabbild lesen

1.1. Programmname : PDPALÉ1.2. Funktion des Programmes :

Das UP PDPALÉ realisiert den Zugriff auf das Prozessabbild. Über die Adresse werden die entsprechenden Daten aus dem Prozessabbild gelesen. Die Länge des auszuleseenden Datenfeldes ergibt sich aus der Codierung von A.

1.3. Schnittstellenbeschreibung :

1.3.1. Eingangsgrößen : DE := Anfangsadresse des Datensatzes aus Prozessabbild  
 HL := Anfangsadresse, von wo aus beginnend Datensatz abgespeichert werden soll

A := KOM-Typ    A:=0    Aggregat-KOM  
                   A:=1    Binär-KOM  
                   A:=2    Leit-KOM  
                   A:=4    Zähl-KOM  
                   A:=8    Analog-KOM

1.3.2. Ausgangsgrößen : <HL>:    1. Byte des Datensatzes aus dem Prozessabbild

<HL+n>: Byte des Datensatzes aus Prozessabbild  
 n=1 bei allen Binär-KOM  
 n=8 bei Analog-KOM  
 n=5 bei Zähl-KOM

Die Datensätze haben den in 7.3.2.2. beschriebenen Aufbau.

1.3.3. Unveränderte Register : IY, HL, A, BC, DE1.4. Fehler : CY := 1    Verkehrter KOM-Typ, falsche Adresse im Grundbereich1.5. Rechenzeit: 0,95 ms2. Beispiel:

Aus dem Prozessabbild ab 4200H soll ein Binär-KOM gelesen und der Inhalt soll ab 7000H abgespeichert werden.

```
LD   HL, 7000H
LD   DE, 4200H
LD   A, 1
RST  10H
DRFB 24      ; PDPALÉ
```

----- PDPAAD UP-Nr.: 25 ----- 3.2: -----

Prozessabbild Adressenberechnung

1.1. Programmname : PDPAAD

1.2. Funktion des Programmes:

Über die problemorientierte Mess-Stellenbezeichnung (numerischer Teil) ermittelt das UP PDPAAD die Adresse des zugehörigen Datensatzes innerhalb des Prozessabbildes.

1.3. Schnittstellenbeschreibung:

1.3.1. Eingangsgrößen:

ADE:= Problemorientierte Mess-Stellenbezeichnung, numerischer Teil  
 (0...99999)  
 A höchstwertiger Teil (nur Bit 7)  
 D höherwertiger Teil  
 E niederwertiger Teil

1.3.2. Ausgangsgrößen:

DE:= Anfangsadresse des zugehörigen Datensatzes im Prozessabbild  
 A:= KOM-Typ (siehe PDPALB)  
 B:= 0 MST im PAS  
 1 MST im PAL  
 C:= rel. Pos. POMA

1.3.3. Unveränderte Register: keine

1.4. Fehler:

CY= 0 kein Fehler  
 CY= 1 Mess-Stelle ist nicht im Prozessabbild strukturiert

1.5. Rechenzeit: 18 ms (ruft PDBINP auf)

2. Beispiel:

Berechnung der Anfangsadresse der Mess-Stelle T 10

```
LD    A,0
LD    D,0
LD    E,0AH
RST  10H
DEFB  25    ; PDPAAD
    ||
DE:= 4200H
A:= 2      ; Leit-KOM
B:= 0      ; MST im PAS
C:= 12     ; rel. Pos POMA für 'T'
```

PDPIAD

UP-Nr.: 26

2.2

Berechnung problemorientierter Information  
aus Adresse

1.1. Programmname : PDPIAD

1.2. Funktion des Programmes:

Anhand der Adresse errechnet das UP, ob es sich um das Prozessabbild oder das WR-Abbild handelt. Das UP ermittelt dann aus der Adresse im Prozessabbild die n-POM der zugehörigen problemorientierten Mess-Stellenbezeichnung und den Alpha-Teil der POM, POMA bzw. aus der Adresse im Wartenrechnerabbild die Position des Datensatzes innerhalb des analogen bzw. binären Teiles des WR-Abbildes.

1.3. Schnittstellenbeschreibung:

1.3.1. Eingangsgrößen:

DB =- Anfangsadresse des Datensatzes im Prozessabbild bzw. WR-Abbild

1.3.2. Ausgangsgrößen:

CY = 0 Datensatz im Prozessabbild

CY = 1 Datensatz im WR-Abbild

Bei CY = 0:

ADB =- Problemorientierte Mess-Stellenbezeichnung

A, Bit 6...0 relative Wörterbuchadresse

A, Bit 7 höchstwertiger Teil der n-POM

D höherwertiger Teil der n-POM

E niederwertiger Teil der n-POM

B =- KON-Typ siehe PDPALF

IX/IY =- SIP-Zeichen der Alpha-POM, max. 4 Zeichen

Bei CY = 1:

A =- Position

(0...199 für analogen Teil)

(0... 99 für binären Teil)

D = 0 analoger Teil

D = 1 binärer Teil

1.3.3. Unveränderte Register: keine

1.4. Fehler:

Z = 0 kein Fehler

Z = 1 Adresse weist nicht auf den Anfang eines Datensatzes des Prozessabbildes bzw. WR-Abbildes

1.5. Rechenzeit: 2,7 ms

----- PDPIAD ----- UF-Nr.: 26 ----- Blatt 2 3.3 -----

## 2. Beispiel:

Ab Adresse 4300H steht DS der Mess-Stelle TC20 (Anlaag -KOM)

```
LD DE, 4300H
RST 10H
DEPB 26 ; PDPIAD
||
CY =0
A, D=0
B := 14H
B := 0
IX:= 2020H
IY:= 5443H
```

PDWALE

UP-Nr.: 27

3.4.

## Lesen Wartenrechnerabbild

1.1. Programmname : PDWALE1.2. Funktion des Programmes :

Mit dem UP PDWALE wird das Lesen aus dem WR-Abbild realisiert. Der Zugriff erfolgt über die Anfangsadresse des Datensatzes. Mit Hilfe des CY-Flags wird erkannt, ob es sich um eine analoge oder binäre Information handelt.

1.3. Schnittstellenbeschreibung :1.3.1. Eingangsgrößen : HL:= Anfangsadresse des Datensatzes  
aus WR-Abbild

CY:= 0 analoger Wert

CY:= 1 binärer Wert

1.3.2. Ausgangsgrößen : CY:= 0 BCDB Wert aus WR-Abbild

A Status

CY:= 1 B Maske

A Status/Wert

1.3.3. Unveränderte Register : IX, IY, HL1.4. Fehler : Keine1.5. Rechenzeit: 0.294 ms2. Beispiel:

Ein analoger Wert des WR-Abbildes ab Adr. 4A20H ist zu lesen

```
LD    HL, 4A20H
XOR   A           ; CY=0
RST   10H
DEFB  27         ; PDWALE
||
CY:= 0
A:= Status
B:= 82H
C:= 10H
DE:= 0
```

PDWASR

UP-Nr.: 26

3.5:

## Schreiben Wartenrechnerabbild

1.1. Programmname : PDWASR1.2. Funktion des Programmes :

Mit dem UP PDWASR wird das Schreiben auf das WR-Abbild realisiert. Das Einschreiben erfolgt über die Anfangsadresse des Datensatzes im WR-Abbild. Mit Hilfe des CY-Flags, dass entweder bei analoger Größe = 0 und bei binärer Größe = 1 ist, werden die zu schreibenden Größen in den Registern übergeben.

1.3. Schnittstellenbeschreibung :1.3.1. Eingangsgrößen : HL:= Anfangsadresse im WR-Abbild,  
die beschrieben wird

CY= 0 BCDE Analoger Wert

A Status

CY= 1 B Maske

A Status/Binärwert

IY:= relative Position DINT

1.3.2. Ausgangsgrößen : Eintragen des Datensatzes in das WR-  
Abbild1.3.3. Unveränderte Register : keine1.4. Fehler : CY = 1 Adresse in HL nicht Adresse des WR-Ab-  
bildes1.5. Rechenzeit: 0,623 ms2. Beispiel:

Ein analoger Wert soll ins WR-Abbild ab Adr. 4250H mit der relativen Pos. 15 der DINT abgespeichert werden.

```

XOR  A
LD   HL, 4250H
LD   A, Status
LD   BC, 3080H
LD   DE, 0
LD   IY, 15
RST  10H
DEFB 28          ; PDWASR

```

PDWAAD

UP-Nr.: 29

3.6

## Berechnung Wartenrechnerabbild-Adresse

1.1. Programmname : PDWAAD1.2. Funktion des Programmes :

Mit dem UP PDWAAD wird aus der relativen Adresse (Position) innerhalb des entsprechenden Teiles des WR-Abbildes (analoger oder binärer Teil) die direkte Adresse des zur Position gehörenden Datensatzes bestimmt.

1.3. Schnittstellenbeschreibung :

1.3.1. Eingangsgrößen : A:= Position (0...199 für analogen Teil)  
(0...99 für binären Teil)

CY:= 0 analoger Teil  
CY:= 1 binärer Teil

1.3.2. Ausgangsgrößen : DE:= Anfangsadresse des Datensatzes im WR-Abbild

1.3.3. Unveränderliche Register : IX, IY1.4. Fehler : keine1.5. Rechenzeit: 0,68 ms2. Beispiel:

Die Anfangsadresse des 5. analogen Wertes des Wartenrechnerabbildes ist zu berechnen.

```
XOR  A      ; CY=0
LD   A, 5
RST  10H
DEFB 29      ; PDWAAD
```

-----PDBINP-----UP-Nr.: 30-----3.7.-----

Berechnung BSE, IMEN aus der numerischen POM

1.1. Programmname : PDBINP

1.2. Funktion des Programmes :

Das UP berechnet anhand der WITA und der ZLIN aus der numerischen POM die zugehörige BSE-Nummer und die IMEN. Eine eventuell erforderliche Speicherebenenumschaltung erfolgt durch das UP.

1.3. Schnittstellenbeschreibung :

1.3.1. Eingangsgrößen : ADE = n-POM (vom Akku nur 7 Bit als höchstwertigen Teil der n-POM)

1.3.2. Ausgangsgrößen : A:= relat. Position des DS WIN in WITA  
C:= IMEN  
B:= BSE-Nummer  
D:= relative Position POMA

1.3.3. Unveränderte Register : keine

1.4. Fehler : CY = 1 n-POM nicht in WITA gefunden

1.5. Rechenzeit: 9.2 ms (max. 16 ms)

2. Beispiel:

Aus der numerischen POM 15 soll die BSE-Nr. und IMEN ermittelt werden.

```
LD   A, 0
LD   D, 0
LD   E, 0EH
RST  10H
DEFB 30      ; PDBINP
```

PDZUDI

UP-Nr.: 31

3.8.

## Zugriff auf Dimensionstyp

1.1. Programmname : PDZUDI1.2. Funktion des Programmes :

Mit Eingabe der Position eines Dimensionstyps im Wörterbuch DINT werden die 6 SIP-1000-Zeichen des Dimensionstyps in 6 Bytes ab einer Adresse abgelegt, die bei UP-Aufruf zu übergeben ist.

1.3. Schnittstellenbeschreibung :1.3.1. Eingangsgrößen :

A := Position des DINT im Wörterbuch  
DE := Anfangsadresse für 6 SIP-Zeichen

1.3.2. Ausgangsgrößen :

(DE) = 1. SIP-Zeichen

.

(DE+5) = 6. SIP-Zeichen

1.3.3. Unveränderte Register : IX, IY1.4. Fehler : CY = 1, wenn A > 141 oder die Zieladresse im verbotenen Bereich1.5. Rechenzeit:2. Beispiel:

Speichere die Dimension mit rel. Position 12 (Meter) ab Adresse 7100H ab

LD A, 12

LD DE, 7100H

RST 10H

DEPB 31 ; PDZUDI

||

ab Adresse 7100H ... (4D, 45, 54, 45, 52) H

----- GKADD                    UP-Hr.: 37                    4.1. -----

Addition zweier GK-Zahlen

- 1.1. Programmname : GKADD
- 1.2. Funktion des Programmes : Dieses Programm realisiert die Addition zweier Gleitkommazahlen  $X:=X + Y$
- 1.3. Schnittstellenbeschreibung
- 1.3.1. Eingangsgrößen : BCDE := Summand X  
IXIY := Summand Y
- 1.3.2. Ausgangsgrößen : BCDE := Summe  $X + Y$
- 1.3.3. Unveränderte Register : IX, IY
- 1.4. Fehler : CY=1 Z=0 arithm. Überlauf  
CY=1 Z=1 arithm. Unterlauf
- 1.5. Rechenzeit: 0,25 ms (max. 0,42 ms)
2. Beispiel: Addiere die GK-Zahlen  $0.5E+1$  und  $0.25E+1$

```
LD      BC,8220H
LD      DE,0      ;5
LD      IX,8120H
LD      IY,0      ;2,5
RST     10H
DEFB   37         ;GKADD
      ||
      ||
      BC:=8270H
      DE:=0      ;0.75E+1
```

GKSUB

UP-Nr.: 36

4.2.

## Subtraktion zweier GK-Zahlen

1.1. Programmname : GKSUB1.2. Funktion des Programmes :

GKSUB ermöglicht die Subtraktion zweier Gleitkommazahlen  
 $X = X - Y$

1.3. Schnittstellenbeschreibung

1.3.1. Eingangsgrößen : BCDE:= Minuend X  
 IXIY:= Subtrahend Y

1.3.2. Ausgangsgrößen : BCDE:= Differenz X - Y1.3.3. Unveränderte Register : IXIY

1.4. Fehler : CY=1 ; Z=0 arithm. Überlauf  
 CY=1 ; Z=1 arithm. Unterlauf

1.5. Rechenzeit: 0,26 ms (max. 0,43 ms)2. Beispiel: Subtrahiere die GK-Zahlen 0.5E+1 und 0.25E+1

```

LD      BC,8220H
LD      DE,0      ;5
LD      IX,8120H
LD      IY,0      ;2,5
RST     10H
DEFB   36         ;GKSUB
||
||
BC:=8120H
DE:=0         ;2,5

```

GKMUL

UP-Nr.: 39

4.3.

-----  
Multiplikation zweier GK-Zahlen1.1. Programmname : GKMUL1.2. Funktion des Programmes:GKMUL realisiert die Multiplikation zweier Gleitkommazahlen  
X:=X \* Y1.3. Schnittstellenbeschreibung :1.3.1. Eingangsgrößen : BCDE:=Faktor X  
IXIY:=Faktor Y1.3.2. Ausgangsgrößen : BCDE:=Produkt X \* Y1.3.3. Unveränderte Register : IX, IY1.4. Fehler : CY=1 Z=0 arithmetischer Überlauf  
CY=1 Z=1 arithmetischer Unterlauf1.5. Rechenzeit: 0,7 ms (max. 0,76 ms)2. Beispiel : Multipliziere die GK-Zahlen 0.5E+1 und 0.25E+1

```

LD      BC,8220H
LD      DE,0           ;5
LD      IX,8120H
LD      IY,0           ;2,5
RST     10H
DEFB    39             ;GKMUL
      ||
      ||
BC:=8350H
DE:=0           ;12,5

```

GKDIV

UP-Nr.: 40

4.4.

## Division zweier GK-Zahlen

1.1. Programmname : GKDIV1.2. Funktion des Programmes :Das UP GKDIV ermöglicht die Division zweier GK-Zahlen  $X := X/Y$ 1.3. Schnittstellenbeschreibung1.3.1. Eingangsgrößen : BCDE:= Divident X  
IXIY:= Divisor Y1.3.2. Ausgangsgrößen : BCDE:= Quotient  $X := X/Y$ 1.3.3. Unveränderte Register : IXIY1.4. Fehler : CY=1 Z=0 arithmetischer Überlauf oder  
Division durch Null  
CY=1 Z=1 arithmetischer Unterlauf1.5. Rechenzeit: 0,96 ms (max. 1,15 ms)2. Beispiel: Dividiere die GK-Zahlen 0.1E+2 und 0.25E+1

```

LD      BC,8320H
LD      DE,0          ;10,0
LD      IX,8120H
LD      IY,0          ;2,5
RST     10H
DEFB    40            ;GKDIV
      ||
      ||
BC:=8200H
DE:=0          ;4,0

```

GKSQU

UP-Nr.: 41

4.5:

-----  
Quadratbildung einer GK-Zahl1.1. Programmname : GKSQU1.2. Funktion des Programmes :Dieses UP realisiert das Quadrat einer GK-Zahl  $X := X ** 2$ 1.3. Schnittstellenbeschreibung1.3.1. Eingangsgrößen : BCDE := Argument X1.3.2. Ausgangsgrößen : BCDE := Quadrat  $X := X ** 2$ 1.3.3. Unveränderte Register : IX, IY1.4. Fehler : CY=1 Z=0 arithmetischer Überlauf  
CY=1 Z=1 arithmetischer Unterlauf1.5. Rechenzeit: 0,68 ms (max. 0,74 ms)2. Beispiel: Bilde das Quadrat der GK-Zahl 0.4E+1

```

LD      BC,8200H
LD      DE,0      ;4
RST     10H
DEFB    41        ;GKSQU
||
||
BC:=8400H
DE:=0          ;16

```

GKRVS                    UP-Nr.: 42                    4.6.

Kehrwert einer GK-Zahl

- 1.1. Programmname : GKRVS
- 1.2. Funktion des Programmes :  
Dieses UP realisiert die Kehrwertbildung einer GK-Zahl  $X:=-1/X$
- 1.3. Schnittstellenbeschreibung :
  - 1.3.1. Eingangsgrößen : BCDE:= Argument X
  - 1.3.2. Ausgangsgrößen : BCDE:= Kehrwert  $X:=-1/X$
  - 1.3.3. Unveränderte Register : IX, IY
- 1.4. Fehler : CY=1 Z=0 Division durch Null  
CY=1 Z=1 arithmetischer Unterlauf
- 1.5. Rechenzeit: 0,91 ms (max. 1,1 ms)

2. Beispiel: Bilde den Kehrwert der GK-Zahl  $0.4E+1$

```

LD      BC,7ECCH           ;0,4
LD      DE,00000H
RST     10H
DEFB   42                  ;GKRVS
||
||
DC:=-8120H
DE:=0                       ;2,5

```

----- GKABS -----

UP-Nr.: 43 -----

4.7. -----

Betragbildung für eine GK-Zahl

1.1. Programmname : GKABS

1.2. Funktion des Programmes :

Dieses UP realisiert die Betragbildung einer GK-Zahl  $X := |X|$ 

1.3. Schnittstellenbeschreibung :

1.3.1. Eingangsgrößen : BCDE := Argument X

1.3.2. Ausgangsgrößen : BCDE := Betrag  $X := |X|$ 

1.3.3. Unveränderte Register : A, HL, IX, IY

1.4. Fehler : keine

1.5. Rechenzeit: 0,16 ns

-----GKFLOAT-----UP-Ver: 44-----4.8.-----

Konvertierung 16 Bit - Zweierkomplement in  
Gleitkommadarstellung

1.1. Programmname : GKFLOAT

1.2. Funktion des Programmes :

Das UP GKFLOAT realisiert die Umwandlung einer ganzen Zahl  
im Register HL in eine äquivalente Gleitkommazahl im  
Register BCDE X:= FLOAT (I).

1.3. Schnittstellenbeschreibung :

1.3.1. Eingangsgrößen : HL:= 16 Bit - ganze Zahl, Zweier-  
komplement

1.3.2. Ausgangsgrößen : BCDE:= Zahl im Gleitkommaformat

1.3.3. Verwendete Register : HL, IX, IY

1.4. Fehler : keine

1.5. Rechenzeit: 0,19 ms

GKTRUNC  
GKROUND

UP-Nr.: 45/46

4.9.

Konvertierung Gleitkommaformat in 16 Bit.  
Zweierkomplement (ganzzahlig) rundend / ab-  
schneidend

1.1. Programmname: GKTRUNC, GKROUND

1.2. Funktion des Programmes :

Die UP GKTRUNC und GKROUND wandeln eine Gleitkommazahl im Register BCDE in eine 16 Bit-Zahl in Zweierkomplementdarstellung um. Nachkommastellen werden bei GKTRUNC abgeschnitten und bei GKROUND gerundet.

1.3. Schnittstellenbeschreibung :

1.3.1. Eingangsgrößen : BCDE := GK-Zahl X

1.3.2. Ausgangsgrößen : HL := konvertierte Zahl  
I := GKTRUNC (X) bzw.  
GKROUND (X)

1.3.3. Unveränderte Register: keine

1.4. Fehler : CY=1 Zahlenbereichsüberschreitungen

1.5. Rechenzeit: 0,2 ms

KOUMAG

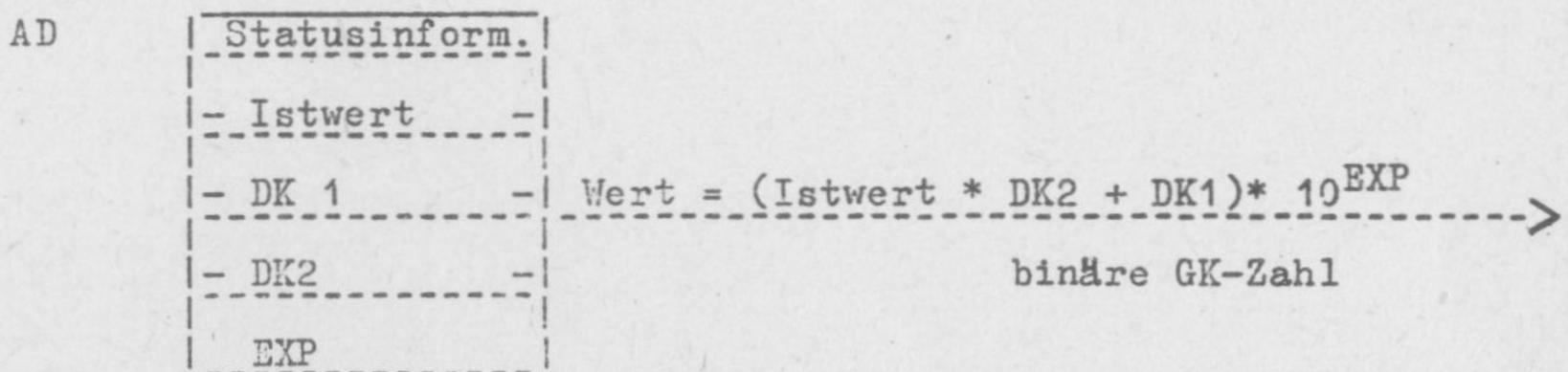
UP-Nr.: 49

4.10

## Zahlenkonvertierung 1

1.1. Programmname : KOUMAG1.2. Funktion des Programmes :

Mit Hilfe des UP KOUMAG ist es möglich, einen Analogwert der Form des Prozessabbildes in eine binäre GK-Zahl umzuwandeln. Darstellung des Analogwertes im PA:

1.3. Schnittstellenbeschreibung:1.3.1. Eingangsgrößen:

HL := AD  
 CY := 0, AD      Adresse im PA  
 CY := 1, AD      Adresse im Anwenderbereich

1.3.2. Ausgangsgrößen:

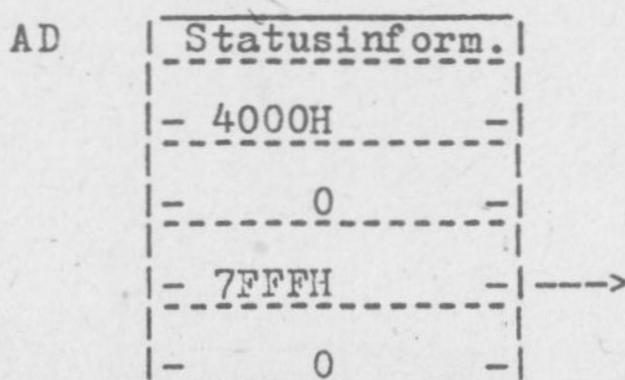
BCDE := binäre GK-Zahl  
 A := Status

1.3.3. unveränderte Register: keine1.4. Fehler:

CY = 1, Adresse nicht im entsprechenden Bereich

1.5. Rechenzeit: 1,7 ms2. Beispiel:

Ein analoger Wert (Aufbau des DS siehe AD) soll in eine GK-Zahl umgeformt werden.



----- KOUMAG -----

UP-Nr.: 49

Blatt 2 4.10 -----

LD HL, AD  
SCF ; CY=1 Adr. im Anwenderbereich  
RST 10H  
DEFB 49 ; KOUMAG  
||  
B: 7F  
C/D/E: OH  
CY= 0

KOUMGA

UP-Nr.: 50

4.11.

## Zahlenkonvertierung 2

1.1. Programmname : KOUMGA1.2. Funktion des Programmes:

Mit Hilfe dieses UP's ist es möglich, eine binäre GK-Zahl in eine normierte FK-Zahl mit Vorzeichen umzurechnen. Für diese Umrechnung werden zusätzlich die Dimensionierungskonstanten und der Exponent benötigt.

1.3. Schnittstellenbeschreibung:1.3.1. Eingangsgrößen:

BCDE := binäre GK-Zahl

HL := ADAT

ADAT	- DK 1 -
	- DK 2 -
	- EXP -

1.3.2. Ausgangsgrößen:

HL = FK-Zahl mit Vorzeichen

1.3.3. Unveränderte Register: keine1.4. Fehler:

CY = 1 arithmetischer Überlauf

1.5. Rechenzeit: 0,99 ms2. Beispiel:

Die normierte FK-Zahl für den Istwert soll aus einer GK-Zahl errechnet werden, wobei ein Datenfeld (siehe 1.3.1. Übergeben werden muss.

ADAT	- 0 -	BC := 8020H
	- 7FFF -	HL := ADAT
	- 1 -	

```
LD HL, ADAT
LD BC, 8000H
LD HL, ADAT
RST 10H
DEFB 50 ; KOUMGA
||
HL := 4000H
```

KOPUTI

UP-Nr.: 53

4.12.

Ausgabekontvertierung einer binären Gleitkommazahl

1.1. Programmname : KOPUTI1.2. Funktion des Programmes:

Dieses UP realisiert die Ausgabekontvertierung einer binären GK-Zahl mit 3 Byte-Mantisse und 1 Byte-Exponenten in eine dezimale GK-Zahl, die im Speicher als SIF 1000-Zeichenkette abgelegt wird. Es besteht z. B. die Möglichkeit, folgende Ausgabeformen zu vereinbaren:

- |                      |                        |
|----------------------|------------------------|
| 1. -.XXXX            | 4 Dezimalziffern       |
| 2. +.XXXXE + XX      | 4 Dezimalziffern + EXP |
| 3. -.XXXXXXXX        | 7 Dezimalziffern       |
| 4. +. XXXXXXXXE + XX | 7 Dezimalziffern + EXP |
| 5. -. XXXXX          | 5 Dezimalziffern       |
| 6. -. XXXXXE + XX    | 5 Dezimalziffern + EXP |

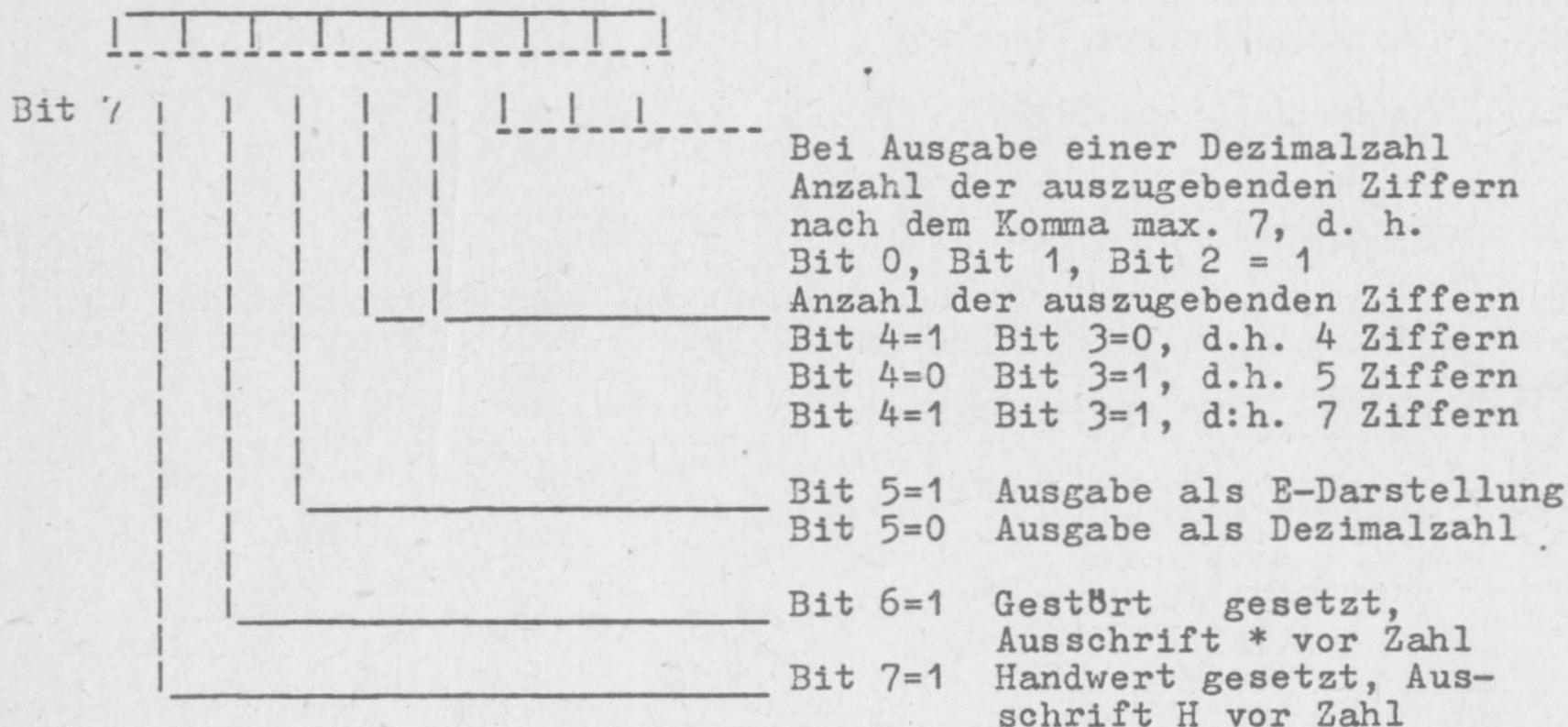
Zusätzlich kann jede Zahl mit einer Kennzeichnung versehen werden. Folgende Kennzeichnungen sind möglich:

1. H Hardwert
2. \* Gestört

1.3. Schnittstellenbeschreibung:1.3.1. Eingangsgrößen:

BCDE:= Binäre GK-Zahl  
 IX:= Zeiger auf erstem Zeichen des Ausgabefeldes  
 A:= Ausgabeverschlüsselung  
 A Verschlüsselung der Ausgabe

7 6 5 4 3 2 1 0

Beispiel einer Ausgabeverschlüsselung

Dezimalzahl mit max. 5 Ziffern, 3 Ziffern nach dem Komma und Aus-

schrift \*.  
Eingabe A: 4BH  
Ausgabe \*+12.304

### 1.3.2. Ausgangsgrößen:

Ausgabe der dezimalen GK-Zahl als SIF 1000-Zeichenkette ab Ausgabefeld entsprechend Verschlüsselung.

### 1.3.3. Unveränderte Register: keine

### 1.4. Fehler:

Z = 1 GK-Zahl nicht über gewähltes Ausgabeformat darstellbar. Auf der letzten Position der SIF 1000-Zeichenkette wird ein Fehlerkennzeichen (#) geschrieben.

CY= 1 Ausgabeadresse im verbotenen Bereich

1.5. Rechenzeit: 7,5 ms bei Zahlenwert 10E4 auf Speicher  
 14 ms bei Zahlenwert 10E12 auf Speicher  
 30,4 ms bei Zahlenwert 10E32 auf Speicher  
 7,7 ms bei Zahlenwert 10E4 auf Monitor  
 14,3 ms bei Zahlenwert 10E12 auf Monitor  
 32,1 ms bei Zahlenwert 10E32 auf Monitor

## 2. Beispiel:

```
LD BC DE, GK-Zahl
LD IX, Bildschirmadr.
LD A, Ausgabeverschlüsselung
RST 10H
DEFB 53 ; KOPUTI
  ||
```

### Registerbelegung

<A>	<B>	<C>	<D>	<E>	Ausschrift auf
33H	82	00	00	00	4,000
3FH	83	40	00	00	0,1200000E+2
30H	80	80	00	00	0,1000E+1
13H	00	FF	FF	00	0,00

KOAINT

UP-Nr.: 54

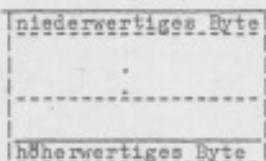
4.13.

Ausgabekonvertierung Integer in eine SIP  
1000-Zeichenkette1.1. Programmname : KOAINT1.2. Funktion des Programmes :

KOAINT realisiert die Ausgabekonvertierung einer positiven Integer-Zahl mit maximaler Länge von 4 Byte in eine SIP-1000-Zeichenkette.

1.3. Schnittstellenbeschreibung1.3.1. Eingangsgrößen : HL := ADR ADZAE  
IX := ADR 1. SIP Zeichen  
C := Anzahl Bytes  
C ≤ 4

Aufbau ADZAE

1.3.2. Ausgangsgrößen : SIP-Zeichen ab angegebener Adresse  
IX zeigt auf nächstfolgende Adresse  
des Ausgabefeldes1.3.3. Unveränderte Register : IX1.4. Fehler : CY = 1 Ausgabeadresse im verbotenen Bereich1.5. Rechenzeit: 4,5 ms bei Ausgabe von 3 Byte Integer2. Beispiel:

E: ADZAE |05|

Ab Adresse 4000H soll ein ZMhlwert von 5 angezeigt werden.

```
LD HL, ADZAE
LD IX, 4000H
LD C, 1
RST 10H
DEPB 54 ; KOAINT
```

3. Bemerkung: In Abhängigkeit der Eingangsgröße C wird jeweils die maximale Anzahl der möglichen SIP-Zeichen bei der Ausgabe reserviert.  
Beispiel: C=1 Wert= 4 Ausgabe\_4

KONORA

UP-Nr.: 55

4.14.

Ausgabekontvertierung von Analogwerten in  
eine SIP-1000-Zeichenkette

1.1. Programmname : KONORA

1.2. Funktion des Programmes :

Das UP KONORA konvertiert Analogwerte der Form des PA in eine  
SIP 1000-Zeichenkette ab angegebener Adresse IX .

1.3. Schnittstellenbeschreibung

1.3.1. Eingangsgrößen : IX=>1.SIF-Zeichen<  
HL:= AD Aufbau AD siehe 4.10  
CY:=0 ; AD Adresse im PA  
CY:=1 ; AD Adresse im Anwenderbereich

1.3.2. Ausgangsgrößen : SIP-Zeichen ab angegebener Adresse  
IX:=IX + Anzahl Zeichen + 1

1.3.3. Unveränderte Register : IY

1.4. Fehler : CY=1 ; Ausgabeadresse im verbotenen Bereich  
Z=1 ; AD Adresse nicht im PA 1000-4FFFF

1.5. Rechenzeit: 2.5 ms

2. Beispiel:

Ab Adr. 7000H soll der Analogwert aus dem Anwenderbereich ausge-  
geben werden

AD	Status
-	4000
-	0
-	7FFF
-	0

CY=1 AD im Anwenderbereich

```

SCP
LD HL, AD
LD IX, 7000H
RST 10H
DEPB 55 ; KONORA
||
ab Adresse 7000H ... 0,5

```

-----KOAKBI

UP-Nr.: 56

4.15

-----  
Ausgabe-Konvertierung eines maskierten  
Binärwertes in eine SIF 1000-Zeichenkette

- 1.1. Programmname : KOAKBI  
 1.2. Funktion des Programmes:

KOAKBI realisiert die Ausgabe eines maskierten Binärwertes in eine SIF 1000-Zeichenkette ab angegebener Adresse. Dieser maskierte Binärwert kann auch im Prozess- bzw. WR-Abbild stehen. Folgende Abapfeicherung des Binärwertes muss eingehalten werden:

ADBI	1   1   0   1   0   1   0   1	Binärwert
	1   1   0   0   1   1   0   0	Maske

Ist in der Maske ein Bit = 0, erscheint auf dem Bildschirm an entsprechender Stelle als SIF-1000 Zeichen ein Strich (5PH). Bei gesetztem Bit erscheint der entsprechende Wert 1 oder 0 (31H oder 30H) des Binärwertes.

### 1.3. Schnittstellenbeschreibung

#### 1.3.1. Eingangssgrößen: HL:= ADBI

IX:=>1.SIF-Zeichen<  
 CY:=0 ADBI im PA bzw. WR-Abbild  
 CY:=1 ADBI im Anwenderbereich

#### 1.3.2. Ausgangssgrößen: SIF-1000-Zeichenkette ab angegebener Adresse IX:=IX + 9

#### 1.3.3. Unveränderte Register : IY

#### 1.4. Fehler : CY:=1 Ausgabeadresse im verbotenen Bereich Z:=1 ADBI nicht korrekt

#### 1.5. Rechenzeit: 1,5 ms

### 2. Beispiel:

Ab Adresse F200H ist folgender Binärwert des PA abzubilden

Binärwert	1   1   0   0   0   1   1   1   0   0	<--- Adr. ADBI
Maske	1   1   1   1   1   1   1   0   0	

```
LD HL, ADBI
LD IX, 0200H
XOR A ; CY=0
RST 10H
DEPB 56 ; KOAKBI
||
ab Adresse F200H 100111__
```



KOPUTWA

UP-Nr.: 58

A.17.

Ausgabekontvertierung einer binären GK-Zahl  
des WR-Abbildes in eine SIF 1000-Zeichenkette

1.1. Programmname : KOPUTWA1.2. Funktion des Programmes:

Das UP KOPUTWA realisiert die Ausgabekontvertierung einer binären GK-Zahl der Form des WR-Abbildes in eine dezimale GK-Zahl, die im Speicher als SIF 1000-Zeichenkette abgelegt wird. Folgende Ausgabeformen sind zu vereinbaren:

- |                   |                        |
|-------------------|------------------------|
| 1. -.XXXX         | 4 Dezimalziffern       |
| 2. +.XXXXE + XX   | 4 Dezimalziffern + EXP |
| 3. -.XXXX         | 5 Dezimalziffern       |
| 4. +.XXXXE + XX   | 5 Dezimalziffern + EXP |
| 5. -.XXXXXX       | 7 Dezimalziffern       |
| 6. +.XXXXXXE + XX | 7 Dezimalziffern + EXP |

Zusätzlich kann jede Zahl mit einer Kennung versehen werden. Folgende Kennzeichnungen sind möglich:

1. H Handwert
2. \* gestört

ADWA

Statusinform.	} GK-Zahl
-----	
-----	
-----	
-----	

1.3. Schnittstellenbeschreibung:1.3.1. Eingangsgrößen:

HL := ADWA

IX := Zeiger auf erstem Zeichen des Ausgabefeldes

A := Ausgabeverschlüsselung (siehe Anlage 1/4.12)

1.3.2. Ausgangsgrößen:

Ausgabe der dezimalen GK-Zahl als SIF 1000-Zeichenkette ab  
Ausgabefeld

1.3.3. Unveränderte Register : keine1.4. Fehler:

Z = 1, CY = 0 GK-Zahl nicht über gewähltes Ausgabeformat  
darstellbar. Auf der letzten Position der SIF  
1000-Zeichenkette wird ein Fehlerkennzeichen  
(#) geschrieben

Z = 1, CY = 1 ADWA nicht korrekt

Z = 0, CY = 1 Ausgabeadresse im verbotenen Bereich

1.5. Rechenzeit: 7,5 ns bei Zahlenwert 10E4 auf Speicher  
14 ns bei Zahlenwert 10E12 auf Speicher  
30,4 ns bei Zahlenwert 10E32 auf Speicher

----- KOPFZEILE UT-Nr.: 58 ----- Blatt 2 ----- 4.17 -----

7,7 ms bei Zahlenwert 10E4 auf Monitor  
 14,3 ms bei Zahlenwert 10E12 auf Monitor  
 32,1 ms bei Zahlenwert 10E32 auf Monitor

## 2. Beispiel:

ADWA

```

| Statusinform. |
| -           - |
| - Mantisse  - |
| Charakterist. |

```

```

| Statusinform. |
| - 00          - |
| - 00          - |
| - 00          - |
| - 82H         - |
| -----      - |

```

&lt;A&gt;:=50H

Aussschrift := 4.000

KOGET

UP-Nr.: 59

4.18

## Eingabekontvertierung einer dezimalen GK-Zahl

1.1. Programmname : KOGET1.2. Funktion des Programmes :

Das UP konvertiert eine als Zeichen im SIP-1000-Code im Speicher abgelegte GK-Zahl in die interne Gleitkommadarstellung. Für die GK-Zahl sind alle gebräuchlichen Darstellungsarten zulässig, wobei lediglich zu beachten ist, dass der Dezimalpunkt verwendet und der Exponent durch den Buchstaben E gekennzeichnet wird. Jedes Zeichen, das syntaktisch nicht zur Zahl gehören kann, wird als Endekennzeichnung akzeptiert. In die Konvertierung werden 8 gültige Stellen einbezogen. Vornullen sind zulässig.

1.3. Schnittstellenbeschreibung1.3.1. Eingangsgrößen : IX := Zeiger auf erstes Zeichen der dezimalen GK-Zahl1.3.2. Ausgangsgrößen : IX := zeigt auf das der Endekennzeichnung folgende Byte  
BCDB := interne Form der GK-Zahl1.3.3. Unveränderte Register: keine1.4. Fehler : CY = 1 Konvertierungsfehler (entweder Zahl leer oder Zahl nicht darstellbar (zu gross / zu klein))1.5. Rechenzeit: 8,7 ms pos. Mantisse, pos. Exponent  
34 ms neg. Mantisse, neg. Exponent2. Beispiel:

Ab Adresse F200H steht die dezimale GK-Zahl 0,1E+2 und soll mit KOGET in eine interne GK-Zahl umgewandelt werden.

```
LD IX, 0F200H
RST 10H
DEFE 59 ; KOGET
II
E:= 83H
C:= 20H
D:= 0
E:= 0
```

KOSINT

UP-Nr.: 60

4.19

Eingabekontvertierung einer SIF 1000-Zeichen-  
kette in eine Integer-Zahl

1.1. Programmname : KOSINT

1.2. Funktion des Programmes :

KOSINT realisiert die Eingabekontvertierung von SIF 1000-Zeichen der GrÖÙe 30H bis 39H in eine Integerdarstellung (positive Integer-Zahl). Die Anzahl der eingegebenen SIF-Zeichen darf 9 nicht überschreiten, da für die Ausgabe nur 4 Byte zur Verfügung stehen. Die Eingabe hat linksbündig zu erfolgen.

1.3. Schnittstellenbeschreibung

1.3.1. Eingangsgrößen : IX:=>1.SIF Zeichen<  
A := Anzahl SIF-Zeichen

1.3.2. Ausgangsgrößen : BC HL:= Ergebnis  
L := niederwertiges Byte

1.3.3. Unveränderte Register: IY

1.4. Fehler: CY = 1 Fehler

1.5. Rechenzeit: 1,2 ms (bei 9 SIF-Zeichen 2,8 ms)

2. Beispiel:

Ab Adresse 0F100H steht Integersahl 12 (31H 32H)

```
LD IX, 0F100H
LD A, 2
RST 10H
DEPB 60 ; KOSINT
||
B:= 0
C:= 0
H:= 0
L:= 0CH
```

KOZACK

UP-Gr.: 51

4.20:

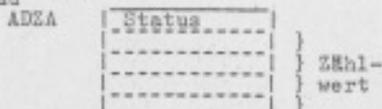
Konvertierung eines Zählwertes in eine  
binäre GK-Zahl

1.1. Programmname : KOZACK

1.2. Funktion des Programmes :

Das UP KOZACK konvertiert eine positive Integerzahl (4 Byte)  
in eine binäre GK-Zahl.

Aufbau



1.3. Schnittstellenbeschreibung :

1.3.1. Eingangsgrößen : HL:= ADZA

CY:= 0 Adr. aus PA

CY:= 1 Adr. im Anwenderbereich

1.3.2. Ausgangsgrößen : BCDE binäre GK-Zahl

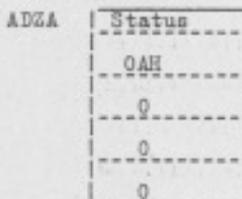
1.3.3. Unveränderte Register: keine

1.4. Fehler : CY=1 ADZA in verbotenen Bereich

1.5. Rechenzeit: 1,9 ms

2. Beispiel:

Ein Zählwert von 10 aus dem PA soll in eine binäre GK-Zahl umgewandelt werden.



LD HL, ADZA

XOR A

BST 10H

DEPB 51 ; KOZACK

||

B := 03H

C := 20H

DE:= 0

-----INMU11-----

UP-Nr.: 62

4.21

-----  
Intergermultiplikation

1.1. Programmname : INMU11

1.2. Funktion des Programmes :

Das UP INMU11 realisiert die Intergermultiplikation 1 Byte x 1 Byte  
zu 2 Byte

X := X \* Y

1.3. Schnittstellenbeschreibung :

1.3.1. Eingangsgrößen : A := Faktor X  
L := Faktor Y

1.3.2. Ausgangsgrößen : HL := Produkt X := X \* Y

1.3.3. Unveränderte Register: IV, IY, BC, DE

1.4 Fehler: keine

1.5. Rechenzeit: 0,42 ns

2. Beispiel:

Multiplikation zweier Integerzahlen 5, 7

LD A, 5

LD L, 7

RST 10H

DEFB 62 ; INMU11

||

H := 0

L := 23H

INDI21

UP-Nr.: 63

4.22

## Integerdivision

1.1. Programmname : INDI211.2. Funktion des Programmes :

Das UP INDI21 realisiert die Integerdivision 2 Byte durch 1 Byte zu 1 Byte und bei vorhandenem Rest dessen Grösse.  
 $X := X / Y$

1.3. Schnittstellenbeschreibung :1.3.1. Eingangsgrössen : DE := Divident X  
C := Divisor Y1.3.2. Ausgangsgrössen : E := Quotient  $X := X / Y$   
D := Restgrösse  
CY := 1 bei vorhandenem Rest1.3.3. Unveränderte Register: IY1.4. Fehler: Z = 11.5. Rechenzeit: 0,9 ms2. Beispiel:

Integerdivision von 12 : 5

LD DE, OCH

LD C, 5

RST 10H

DEFB 63

||

E := 2

D := 2

CY := 1 Rest vorhanden

KMAUTA

UP-Nr.: 65

5.1

Textausschrift aus Textspeicher auf Monitor  
bzw. anderen Speicherbereich

1.1. Programmname : KMAUTA

1.2. Funktion des Programmes:

KMAUTA gibt einen Einzeltext aus einem Textspeicher

- auf dem Farbmonitor aus, wobei die Hintergrundfarbe der Textausschrift über Parameter bestimmbar ist. Die Farbinformation wird 800H hinter der Zeicheninformation abgespeichert.
- auf anderen Speicherbereich aus, wobei keine Farbinformation gespeichert wird.

Aus dem zentralen Textspeicher (SPEICHER) wird über eine Relativadresse (0...N) derjenige Einzeltext bestimmt, dessen Transfer organisiert werden soll.

Mit dem UP sind z. B. aus einem zentralen Textspeicher alle Einzeltexte eines statischen Monitorbildes problemlos auszuschreiben.

Der zentrale Textspeicher muss folgenden Aufbau besitzen:

SPEICHER: Teil 1: Adressen                      Teil 2: Textinhalt      Beispiel

0	- Adr. Text 0 (TX0)-	TX0	46	F	
1	- Adr. Text 1 (TX1)-		45	E	
2	- (TX2)-		48	H	
3	- (TX3)-		4C	L	
N	- : -		45	E	
			D2	R	---> Bit 7 gesetzt
		TX1	45	E	
			49	I	
			4E	N	
			47	G	
			41	A	
			42	B	
			C5	E	---> Bit 7 gesetzt
		TX2	4D	M	
			45	E	
			4E	N	

	55	U	
	05	E	--> Bit 7 ge-
	46	F	setzt
TX3	55	U	
	4E	N	
	4B	K	
	54	T	
	49	I	
	4F	O	
	CE	N	--> Bit 7 ge-
			setzt

Beim letzten Zeichen eines jeden Textbereiches muss das Bit 7 = 1 gesetzt sein. Es wird vom UP als Endekennzeichen interpretiert. Über ein Steuerbyte FAC0 in Register C können Farbe des Zeichens und des Hintergrundes und Sonderfunktionen (Blinken) codiert werden, sofern die Ausschrift auf Monitor erfolgt. Ist der Ausgabespeicherbereich nicht Monitorbereich, so wird der Inhalt C (FAC0) nicht verwertet.

Aufbau Steuerbyte FAC0 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |

D7	D6	D5	D4	D3	D2	D1	D0	Hexa-Code	
							1	01	Vordergrund rot
							1	02	Vordergrund grün
							1	03	Vordergrund gelb (rot/-grün)
					1			04	Vordergrund blau
					1		1	05	Vordergrund purpur
					1	1		06	Vordergrund cyan
					1	1	1	07	Vordergrund weiss
								00	Vordergrund schwarz/ Hintergrund schwarz
							1	08	Hintergrund rot
							1	10	Hintergrund grün
							1	18	Hintergrund gelb
							1	20	Hintergrund blau
							1	28	Hintergrund purpur
							1	30	Hintergrund cyan
							1		Hintergrund weiss
							1	19	Hintergrund gelb Vordergrund rot
									Sonderfunktion
							1		Blinken

Der vom Anwender aufgebaute zentrale Textspeicher kann auf einer Ebene stehen. Es ist zu beachten, dass

- bei Ausgaben auf Monitorbereiche (>DE<:F040...F73F) die Ebene mit dem zentralen Textspeicher SPEICHER vorher eingestellt sein muss (Ebenenschaltung!),
- bei Ausgaben auf andere Speicherbereiche des Anwenders (BS-Speicherbereiche sind gesperrt) sowohl die Ebene des zentralen Textspeichers SPEICHER als auch die Ebene des Ausgabebereiches eingestellt sein müssen (Ebenenschaltung!); sofern vom Anwender als Ausgabebereich ein BS-Speicherbereich in DE angegeben wird, erfolgt durch KNAUTA keine Bearbeitung

**ACHTUNG:** Wird KNAUTA aufgerufen und die Ebene, auf der SPEICHER steht, ist nicht geschaltet, erfolgt ein falscher Zugriff. Damit können Speicherzellen, die mit DE adressiert sind, zerstört werden, da der falsche Zugriffsbereich als organisierter SPEICHER gewertet und "ausgeschrieben" wird.  
Die Angabe von FACO in <C> ist für doppelte Zeichenbreite (D7=1) nicht zulässig.

### 1.3. Schnittstellenbeschreibung

#### 1.3.1. Eingangsgrößen:

- A: Relativadresse aus SPEICHER für den Einzeltext, der ausgeschrieben werden soll. 0 ... N
- HL: Enthält die Adresse vom zentralen Textspeicher SPEICHER, wobei die Ebene mit SPEICHER vom Nutzer geschaltet sein muss.
- DE: Enthält die Ausgabeadresse
- Monitorausgabe F040 - F73F (Monitorzeilen 1, 30-32 besetzt vom BS)
  - Speicherausgabe 1000 ... 4FFF  
7000 ... AFFF
- zulässig für die geschaltete Anwenderenebene
- Speicherausgabe 0000 - 0FFF  
5000 - 6FFF  
B000 - EFFF
- unzulässige Bereiche, werden mit Fehler (CY-Flag) abgewiesen
- C: FACO Steuerbyte für Farbausgabe auf Farbmonitor. Ist nur wirksam, wenn DE eine Monitoradresse enthält.  
Bit 7 =1 für doppelte Zeichenbreite ist nicht zulässig.

#### 1.3.2. Ausgabegrößen:

Der Einzeltext wird auf der angegebenen DE Adresse ausgeschrieben. Das Bit 7 im letzten Zeichen des Einzeltextes ist rückgesetzt, so dass ein unverfälschtes Zeichen ausgegeben wird. Eine Monitorausgabe wird in der durch <C> beschriebenen Farbcodierung/ Sonderfunktion organisiert.

- <DE> letzte Text-Position +1 (letztes ausgegebenes Zeichen F000 ... F7FF) nach Ausgabe
- <HL> letzte Farbposition +1 (F800 ... FFFF) nach Ausgabe

KMAUTA

UP-Nr.: 65

Blatt 3 5.1

1.3.3. Unveränderte Register: keine

1.4. Fehler:

CY = 0 Fehlerfreiheit

CY = 1 Ausgabebereich (DE) liegt nicht im zulässigen Bereich

1.5. Rechenzeit: 0,318 ms  
max. 1,4 ms

2. Beispiel:

Das Wort 'Eingabe' soll auf Monitor ab Adr. 0F200H ausgegeben werden.

Textspeicher und Textinhalt siehe Beispiel in Abschnitt 1.2.

```
LD  A, 1          ; Textausgabe TX1
LD  HL, SPEICHER
LD  DE, 0F200H
LD  C, 2          ; Zeichen grün, Hintergrund schwarz
RST 10H
DEFB 65           ; KMAUTA
||
ab Adresse 0F200H 'EINGABE'
```

KMLOEM

UP-Nr.: 66

5.2.

Löschung des Bildbereiches von 2.- 29. Zeile  
des Farbmonitors

1.1. Programmname : KMLOEM

1.2. Funktion des Programmes:

Das UP KMLOEM realisiert das Löschen des Bildbereiches von der 2. bis 29. Zeile des Farbmonitors. Der Bildbereich wird mit Leerzeichen, der Hintergrundspeicher mit grün (FACO =02H) beschrieben. Bei Textausgaben erscheint damit eine grüne Schrift auf schwarzem Hintergrund.

1.3. Schnittstellenbeschreibung:

1.3.1. Eingangsgrößen : keine

1.3.2. Ausgangsgrößen : Löschung des o. g. Bereiches  
Zeilen 2 - 29

1.3.3. Unveränderte Register: IY

1.4. Fehler: keine

1.5. Rechenzeit: 180 ms

2. Beispiel:

Löschen des Bildbereiches von der 2 bis 29. Zeile des Farbmonitors mit Leerzeichen und Hintergrund grün.

```
RST 10H
DEFB 66 ; KMLOEM
```

----- KMLOBB -----

UP-Nr.: 67 -----

5.3. -----

## Löschung von Bereichen auf dem Farbmonitor

1.1. Programmname : KMLOBB1.2. Funktion des Programmes :

KMLOBB realisiert Bereichslöschungen. Löszeichen und Farbwert FACO sind frei wählbar.

Die Funktion wird nur ausgeführt, wenn sich Anfangs- und Endadresse des Bildspeichers im Bereich der 2. bis 29. Monitorzeile bewegen (F040 ... F73F).

Das Zeichen, mit dem gelöscht werden soll, wird ab Anfangsadresse Bildspeicher (HL) abgespeichert. Das Farbcode-Zeichen wird ab Bildspeicheradresse (HL) plus 800H abgespeichert.

1.3. Schnittstellenbeschreibung:1.3.1. Eingangsgrößen:

HL := Anfangsadresse Bildspeicher für Zeichenbereich  
(F040 ... F73F)

BC := Anzahl zu löschender Bytes

D Zeichen entsprechend Monitor-Zeichengenerator

E Farbwert FACO (Steuerbyte für Vordergrund/Hintergrundfarbe)

1.3.2. Ausgangsgrößen: Gelöschter Bereich1.3.3. Unveränderte Register: IY1.4. Fehler:

CY = 1 - UP wird nicht abgearbeitet  
- zu löschender Bereich im verbotenen Bildschirmbereich

1.5. Rechenzeit: 3,8 ms je Zeile zu 40H Zeichen mit Leerzeichen2. Beispiel:

Monitorzeile 2 soll gelöscht werden (Hintergrundfarbe grün)

LD HL, 0F040H

LD BC, 40H

LD D, 20H

LD E, 02

RST 10H

DEFB 67 ; KMLOBB

-----KMTAPL-----

UP-Nr.: 68-----

5.4-----

Übergabe des Tastaturpuffers aus dem System-  
RAM (Tastaturpuffer lesen)

1.1. Programmname : KMTAPL

1.2. Funktion des Programmes :

KMLAPL realisiert die Übergabe des Tastaturpuffers, beginnend auf angegebener Adresse der angegebenen Ebene, in den Anwender-RAM. Der Tastaturpuffer umfasst eine maximale Speicherkapazität von 8 Tastencodes. Sie werden in zeitlich einlaufender Reihenfolge ab Adresse TACO abgespeichert. Sind weniger als 8 Tastencodes im Puffer abgelegt, wird der Rest mit 0 aufgefüllt. Dem UP muss ein Steuerbyte EBCO zur Ebeneneinstellung übergeben werden. Aufbau Steuerbyte EBCO siehe UP Ebeneneinstellung BSEBEI.

TACO:

TAQUIT
1. Tastencode
7. Tastencode
8. Tastencode

### Bedeutung TAQUIT

TAQUIT = 01H Anmeldung VAP zum ersten Mal nach Aufruf im Grundmenü der WRE. TAQUIT wird vom Betriebssystem (BS) gesetzt.

TAQUIT = 10H Voranmeldung für eine durch Tastaturbedienung erkannte bevorstehende Abmeldung des Verarbeitungsprogrammes. Die Tastencodes, die noch vom VAP verarbeitet werden müssen, werden in dieser anstehenden 1/3 s übergeben.

### Bedeutung EBCO

Im Register A wird vom Anwender vor Aufruf KMTAPL die Ebene verschlüsselt, in der sich der vereinbarte TACO-Speicher des Anwenders befindet. Sie wird vom UP eingestellt und anschliessend auf die Ursprungsebene rückgestellt.

1.3. Schnittstellenbeschreibung :

1.3.1. Eingangsgrossen :

HL := >TACO<

A := <EBCO> für TACO

----- KMTAPL ----- UP-Nr.: 68 ----- Blatt 2 ----- 5.4 -----

### 1.3.2. Ausgangsgrößen :

Tastencodes ab Adresse TACO + 1 als hexaverschlüsselte Byteinformationen gemäß Tastencodes der Tastatur (siehe Tastaturbeschreibung TAS 651.03).

### 1.3.3. Unveränderte Register : IY

1.4. Fehler : keine

1.5. Rechenzeit: 0,654 ms

### 2. Beispiel:

Auf Ebene 2 ab Adresse 1000H soll der Tastaturpuffer des System-RAM abgespeichert werden

```
LD HL, 1000H
LD A, 81H
RST 10H
DEFB 68 ; KMTAPL
||
```

1000H | 01 | VAP zum ersten Mal nach Aufruf im Grundmenü der  
| 41H | WRE angemeldet

9DH
0
0
0
0
0
0
0

in den Tastaturpuffer des Anwenders sind die Tasten  
| A | | ENTER | abgespeichert worden

KMTAPS

UP-Nr.: 69

5.5

Tastaturpuffer schreiben (siehe auch KMTAPL)

1.1. Programmname : KMTAPS

1.2. Funktion des Programmes :

Einschreiben der Rückmeldung des VAP des Anwenders in den Parameter TAQUIT des Feldes TACO, sofern vom Betriebssystem mit KMTAPL ein TAQUIT = 10H (Vor anmeldung VAP) gemeldet wurde.

Das Anwender-VAP hat alle Abmeldehandlungen interner Art, z. B. Abmeldung angemeldeter Verarbeitungsprogramme, zu tätigen in der Gewissheit, dass in der folgenden 1/3 s keine Kommunikations handlungen über Bildschirm/ Tastatur der WRE mehr stattfinden kann.

Die Fertigmeldung hat das VAP über Aufruf KMTAPS an das BS zu übergeben mit TAQUIT = 00H.

Das BS wertet diese Information als Quittung. Erfolgt sie nicht, setzt das BS eine Fehlermeldung über das Info-System der WRE ab.

1.3. Schnittstellenbeschreibung :

1.3.1. Eingangsgrößen :

HL := &gt;TACO&lt;

Beschreibung siehe UP KMTAPL

A := &lt;EBCO&gt; Ebenencodierung für &gt;TACO&lt;

Beschreibung siehe UP KMTAPL

Inhalt des 1. Speicherplatzes von TACO muss TAQUIT = 00H sein bei Aufruf KMTAPS.

1.3.2. Ausgangsgrößen : keine

1.3.3. Unveränderte Register : IY

1.4. Fehler :

Erfolgt nach Übergabe von BS mit TAQUIT = 10H (VAP-Abmeldung) keine Quittung durch das VAP mit TAQUIT = 00H (in der gleichen Zeitscheibe), wird das VAP von der Kommunikation getrennt und eine INFO-Fehlermeldung auf WRE abgesetzt.

1.5. Rechenzeit: 0,58 ms

2. Beispiel:

```

XOR  A
LD   HL, TACO      ; Adr. Tastaturpuffer
LD   (HL), 0      ; TAQUIT:=0
LD   A, 84H       ; EBCO
RST  10H
DEFS 69           ; KMTAPS

```

KMINFO

UP-Nr.70

5.6

## Ausgabe Informations- und Meldepuffer

1.1. Programmname : KMINFO1.2. Funktion des Programmes :

Das UP ermöglicht die Ausgabe einer Information bzw. Meldung in der Zeile 30 des Displaybildes für den WRE-Operator. Die Informationszeile kann als schwarze Schrift jeweils dem Interpretationszweck entsprechend auf farbigem Hintergrund (rot, grün, cyan) ausgegeben werden.

Wird die Funktion KMINFO von BS-Programmen benutzt, haben diese Programme grundsätzlich die Ausschriften auf weißem Hintergrund zu tätigen (gesperrt für Anwender-VAP's).

Die Monitor-Ausgabezeile besteht aus

- der aktuellen Uhrzeit in h : min : s,
- einem blinkenden Pfeil bei Informationseinlauf, bis die Informations/Meldung vom Operator quittiert wurde. Pfeil geht dann in ein Ruhelicht über,
- einer Textinformation, Länge 50 Zeichen, mit der von jedem VAP eine Kennung und Information bzw. Meldung ausgegeben werden kann.

Zusätzlich ertönt ein Hupenton bei Informationseingang (Hupe Prozessalarm bei VAP-Meldungen, Hupe Systemalarm bei BS-Meldungen).

Die Informationszeile 30 (einlaufende Meldung) wird in Zeile 31 und 32 weitergeschoben, sofern neue Informationen von anderen VAP's einlaufen. Die auszugebende Information ist in einem Speicher ZIMP gemeinsam mit einem Steuerbyte STEUBY bereitzustellen. Im Steuerbyte STEUBY kann verschlüsselt werden,

- in welcher Farbe die Ausgabe erfolgen soll,
- ob die Ausgabe auf Monitor oder Drucker (Bedien- und Meldeprotokoll) bzw. auf beiden Geräten gleichzeitig erfolgen soll.

Ist das Bedien- und Meldeprotokoll nicht angemeldet, so wird

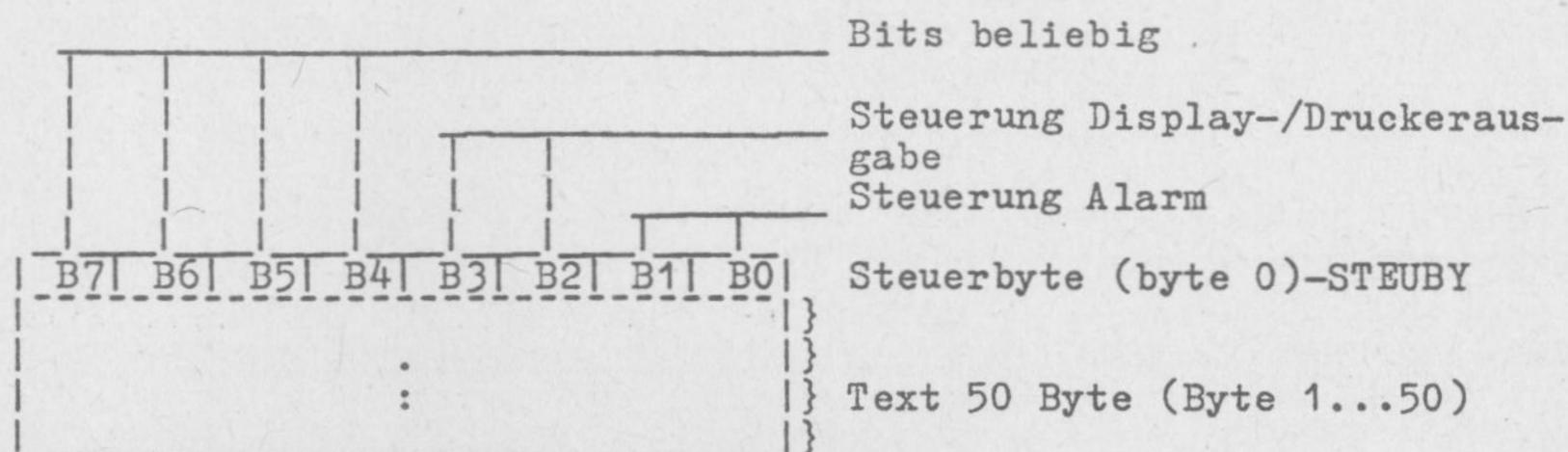
- eine Zwangsausgabe auf Monitorzeile 30 mit rotem 'D' ausgeführt, sofern vom VAP eine Druckerausgabe vorgesehen war.

Quittierungen und Löschhandlungen sind im Teil 7 der BE der PLE beschrieben.

1.3. Schnittstellenbeschreibung

Durch das aufrufende VAP ist ein Speicher ZIMP mit nachfolgender Struktur aufzubauen, in dem neben einem Steuerbyte die auszugebende Information gespeichert wird.

## Struktur ZIMP



Länge ZIMP: 51 Byte

1.3.1 Eingangsgrößen:

DE := Anfangsadresse ZIMP

ZIMP -Struktur mit Byte 0 (Steuerbyte STEUBY)

- Bit 3 DIS Bit wird gesetzt, wenn die Ausschrift auf Display (Zeile 30 bis 32) erscheinen soll  
 B 3 =1 Displayausgabe mit Hupenton  
 B 3 =0 keine Ausgabe auf Display
- Bit 2 DRU Bit wird gesetzt, wenn die Ausschrift im Bedien- und Meldeprotokoll erscheinen soll  
 B 2 =1 Ausgabe im Bedien- und Meldeprotokoll (BMP) auf Drucker  
 B 2 =0 keine Druckerausgabe im BMP

Kombination Bit 3/Bit 2

Bit 3	Bit 2	Wirkung
0	0	Keine Ausgabe. Es erfolgt eine Fehlermeldung an das angemeldete Programm (CY=1)
1	0	Ausgabe erfolgt nur auf Display-Zeilen 30 bis 32, keine Ausgabe in Bedien- und Meldeprotokoll Drucker
0	1	Ausgabe auf Display erfolgt nicht. Information erscheint sofort auf dem Drucker (Bedien- und Meldeprotokoll BMP), sofern es angemeldet ist. Sofern das BMP abgemeldet ist, erfolgt Zwangsausgabe der Informationszeile auf Monitor
1	1	Ausgabe auf Display Zeile 30 erfolgt zeitgleich mit Ausgabe auf Bedien- und Meldeprotokoll Drucker, sofern das BMP angemeldet ist

KNINFO

UP-Nr.: 70

Blatt 3 3.6

Bit 1 ) PACO Farbcode für anzugebende Informationen auf  
 ) Display,  
 Bit 0 ) Die Information erscheint in schwarzer Schrift  
 (Vordergrund) auf farbigen Hintergrund (HG)

Bit 1	0	1	0	1
Bit 0	0	0	1	1
Farbe HG	weiss	gruen	rot	cyan
Hupenton fuer	Systemalarm	Prozessalarm	Prozessalarm	Prozessalarm

Beispiel: STEUBY = OPH, Ausgabe Monitor und Drucker, schwarze Schrift, Hintergrund cyan

Byte 1 bis 50 (Textbyte): 50 Zeichen aus den Zeichengenerator des Wartenrechners. In diesen Zeichen sind bei Notwendigkeit auch Programmname, aktuelle Uhrzeit oder andere dynamische Informationen aus den Anwender-VAP's für den Operator zu verschlüsseln.

### 1.3.2 Ausgangsgrößen:

Monitorauschriften Zeile 30, BHP-Ausgabe Drucker, Hupenton

1.3.3 Unveränderte Register: keine

1.4 Fehler: keine

1.5 Rechenzeit : 3,6 ms  
 max 29 ms

2. Beispiel:

Displayzeile:

11:28:16 --->KENNWERT SILO: ACHTUNG/KEIN ZULAUF SEIT 10:00 UHR

Programname                      Detailinformation

Textinformation bis 50 Zeichen

-----KMCODE----- UP-Nr.: 71 ----- 5.7 -----

### Codewortvergleich

1.1. Programmname: KMCODE

1.2. Funktion des Programmes:

Dieses UP ermöglicht den Vergleich zwischen dem intern abgelegten Codewort. (Codepos. 4...7) und dem auf Monitor geschriebenen Codewort.

1.3. Schnittstellenbeschreibung

1.3.1. Eingangsgrößen:

B := Codeposition (4...7)

DE := Anfangsadresse Codewort auf Monitor

1.3.2. Ausgangsgrößen:

Z:= 0 Codewort des Monitors + internen Codewort

Z:= 1 Codeworte identisch

1.3.3. Unveränderte Register: IX, IY

1.4. Fehler: keine

1.5. Rechenzeit: 0,67 ms

2. Beispiel:

```
LD B, 7      Codeposition 7
LD DE, 0F200H  Monitoradresse
RST 10H
DEFB 71      ; KMCODE
JR Z, 10     ; Codeworte identisch
```

KMINFH

UP-Nr.: 22

5.8

-----  
 Informations- und Meldepuffer mit Löschung  
 ZIMP

1.1. Programmname: KMINFH

1.2. Funktion des Programmes:

Die Funktion ist identisch mit der des Programmes KMINF0, UP-Kennblatt 5.6. Zusätzlich wird der von Nutzer zur Verfügung gestellte Speicherinhalt des Feldes ZIMP (51 Byte, Übergabedresse in <DE>) nach Ausgabe der Meldung mit Leerzeichen (20) gelöscht.  
 Näheres siehe KMINF0, Kennblatt 5.6, Punkt 1.2

1.3. Schnittstellenbeschreibung

Identisch zu KMINF0, Kennblatt 5.6, Punkt 1.3

>DE< Adresse des 51 Byte Lagen Textspeichers ZIMP, dessen 1.Byte das Steuerbyte STEUBY ist

1.4. Fehler

Vom Programm KMINFH werden keine Fehler ausgegeben.

1.5. Rechenzeit: 29,3 ms

2. Beispiel:

Siehe KMINF0, Kennblatt 5.6, Punkt 2.

DUROG

UP-Nr. 10

5.9

Organisation des Datenaustausches zwischen  
Wartrechnereinheit und einem Fremdrechner

1.1. Programmname : DUROG

1.2. Funktion des Programmes:

Das Programm übernimmt die Anmeldung von Verarbeitungsprogrammen zur Datenübertragung mit einem Fremdrechner, die Quittierung des Empfanges und leitet das Senden eines Telegrammes ein. Durch das Lesen des Übergabebyte kann die Ausführung der DUE kontrolliert werden.

1.3. Schnittstellenbeschreibung

1.3.1. Einganggrößen:

A := SBY (Steuerbyte)  
 0 Programm Anmeldung zu DUE mit FR  
 HL := Adresse Übergabepuffer für Textteil des  
 Telegrammes zum Senden und für den Empfang.  
 Der Übergabepuffer muss auf der gleichen Ebene  
 wie das VAP selbst liegen.  
 B := VAP-Nr.  
 1 Es wird nur das Übergabebyte gelesen  
 2 Rücksetzen des Empfangs- und Sendebits  
 3 Rücksetzen des Empfangsbits und Setzen des Sendebits  
 4 Setzen des Empfangsbit

1.3.2. Ausganggrößen:

A := UGB (Übergabebyte)  
 C := Steuerbyte  
 B := VAP-Nr.  
 HL := Datenpufferadresse (wenn VAP in DUAMT angemeldet )

DUROG

UP-Nr.10

Blatt 2 5.9

Übergabebyte			
Bit	Bedeutung	gesetzt durch	rueckgesetzt durch
0	Empfangsbit	WRE-BS bei fehlerfreiem Empfang	VAP nach abgeschlossener Ausw.
1	Sendebit	VAP nach Bereitstellung der Antwort	WRE-BS nach abgeschlossener DUE
2	empfangenes Telegramm zu lang	WRE-BS	WRE-BS
3	Antwort zu lang		
4	Empfangsfehler		
5	Sendefehler		
7	DÜ schon gelaufen		Neuanlauf der WRE

### 1.3.3. Unveränderte Register : B, IY

### 1.4. Fehler

Wird mit Hilfe von DUROG ein Fehler festgestellt, so wird CY=1 gesetzt.

Folgende Fehler können auftreten:

- Datenpufferadresse liegt ausserhalb des vorgesehenen Anwenderbereiches. Der Fehler wird beim Anmelden des VAP's im DUAMT gemeldet.
- DUAMT ist mit 8 VAP's voll belegt
- Das VAP ist nicht mehr in der PADL angemeldet, gleichzeitig wird das gesuchte VAP in der DUAMT gestrichen d.h. das VAP muss wieder neu angemeldet werden.
- Der Wert des Steuerbyte lag ausserhalb der festgelegten Werte.

### 1.5. Rechenzeit : 1,5 ms

### 2. Beispiel : Empfang quittieren

```
LD   A,2      ; Rücksetzen Empfangs- und Sendebit
RST  10H
DEFB 10      ; DUROG
```

DUSA

UP-Nr.: 74

6.1

Anmeldung und Steuerung der Datenübertragung  
von Prozessdaten (KOM-Daten)

1.1. Programmname: DUSA

1.2. Funktion des Programmes:

Mit diesem UP können mehrere KOM-Daten einer oder mehrerer IMEN gelesen oder eingeschrieben werden. Die Daten dürfen auf max. 3 BSE verteilt sein.

1.3. Schnittstellenbeschreibung

1.3.1. Eingangsgrößen:

Der DUE -Ablauf für Anwender ist dadurch gekennzeichnet, dass vor dem Aufruf von DUSA Datenfelder mit dem in den Bildern 6.1.-1 bis 6.1.-4 dargestellten Aufbau bereitzustellen sind. Der Aufruf zur DUE erfolgt mit:

HL:= >Adr.DSF<  
IY:= >Adr.FEMI<  
DE:= >Adr.Kontrolltelegramm<  
RST 10H  
DEFB 74 (vgl.Tab.8.-1.)

Die benötigten DUE-Felder müssen auf den Ebenen liegen, die im jeweiligen VAP-Kopf definiert sind.

Der Aufbau des DSF ist wie folgt anzulegen:

Byte- Nr.	Struktur	
DSF+0	PRI0	
+1	für DUSA freih.	
+2	ANZA	
+3	ADEM 1	1.DS
+4	FEZD 1	
+5	TELEX 1	
+6	I	
+7	LDA 1	
+8	ANTW 1	
+9		
	u.s.w.	2.DS
	OFFH	Abschlussbyte DSF

Bild 6.1.-1 : Struktur DSF

----- DUSA                      UP-Nr.: 74                      Blatt 3    6.1 -----

Erläuterungen :

- DSP+3 :            BSE Nr. mit 02,07.... eintragen (nicht DUE Stationsadresse)
- DSP+4 :            bei Fehlern wiederholt die DUE die Datenübertragung und bei Erreichung eines Maximalwertes oder bei Fehlern im DSP (BSE nicht betriebsbereit) wird dem VAP mit Hilfe des Fehlerzählers FEED#0 mitgeteilt. Als Anfangswert ist durch das VAP 00 einzutragen.
- DSP+7 :            Durch das VAP ist die Länge des erwarteten Antworttelegrammes (Anzahl Lesesbytes) hier einzutragen. Maximallänge beträgt 250 Bytes. Bei reinen Schreibtelegrammen muss in jedem Falle die eingetragene Antwortlänge (auch bei PRIO = 4) = 0 sein.
- DSP+8 ...9 :        Hier ist bei Lesetelegrammen die Adresse des Antworttelegramms abzulegen, bei Schreibtelegrammen mit PRIO=4 die Adresse für das Kontrollteile und bei Schreibtelegrammen mit PRIO=5 ist eine Adresse im Ebenenbereich (z.Bsp. Teleadresse) einzutragen (für Sinnfälligkeitstest in DUSA).
- DSP+10 ...PPH :     Für jede in der aktuellen DUE-Anmeldung anzusprechenden BSE ist ein weiterer Datensatz zu eröffnen mit der Datenstruktur der Bytes DSP+3 bis DSP+9. Maximal zugelassen sind 3 Datensätze. Ein DSP ist grundsätzlich mit 0PPH nach dem letzten Informationsbyte abzuschliessen.

Das UP DUSA lässt 2 PRIORITÄTEN zu:

PRIO = 4 : es erfolgt bei Schreiboperationen in BSE

-----  
 (Bitschreiben, KON-Daten-Schreiben) ein Rücklesen der geschriebenen Bytes und ggf. wird in FEHI bei Rücklesefehlern eine Information an den Anwender übergeben.

In >DE< ist dazu die Adresse eines Speicherraumes in dem Ebenenbereich aus welchen das UP aufgerufen wird zu übergeben. In diesem kann das UP das Kontroll-(Rücklese-)telegramm aufbauen.

Die zu reservierende Länge lässt sich wie folgt berechnen:

$$L = \sum_{i=1}^M (k_i * 3 + M_i * 3 + 5)$$

N=Anzahl Teleg.in DSP  
 k=Ans.DS (KON-Schr.)in Teleg.  
 M=Ans.DS (Bit-Schr.)in Teleg.

In DSP+8 (u.f.), des jeweiligen DS ist in diesem Falle (auch bei Schreibtelegrammen) die Adresse des Speicherraumes zu übergeben, ab der die zurückgelesenen Bytes abgelegt werden. (entsp. Antworttelegrammaufbau bei Leseregine) Dabei ist pro zu beschreibenden Byte ein BYTE im Antwortpuffer freizuhalten.

PRIO = 5 : Es erfolgt keine Rücklesekontrolle

-----  
>DE< ist bei Aufruf beliebig

Die Fehlermitteilungen erfordern die Bereitstellung eines Pufferbereiches der Länge 7 Byte (auch wenn DSP nur 1 oder 2 Datensätze enthält).

Aufbau des Merkerfeldes FEMI :

```
FEMI+0 : DUFEL
+1 : DSFFE 1
+2 : TELFE 1
+3 : DSFFE 2
+4 : TELFE 2
+5 : DSFFE 3
+6 : TELFE 3
```

Ein DSP darf nur Telegramme mit Schreib- (KOM, oder Bit-schreiben) oder Leseoperationen enthalten. Bei gemischten Telegrammen siehe Fehlermitteilung in FEMI+0, Bit 7. Als weitere Eingangsgrösse ist für jede in einem DUSA-Ruf anzusprechende BSE eine TELEX-Liste nach Bild 6.1.-3 bereitzustellen.

Telex+0+2	frei lassen f.DUE	
Telex+3 :	Datenlänge (DALE)	
Telex+4	frei lassen f.DUSA	
Telex+5	Steuerbyte 1	usw. bis Datensatz 1 Aufbau lt.7.4.2.
	.....	
	.....	
	OFF H	Abschlussbyte des Telegramms

Bild 6.1.-3: Struktur für Telexaufbau

Die ersten 3 Byte und das 4. Byte des Sendetelegramms TELEX werden von der DUE-Routine belegt und sind dafür freizuhalten. Auf Byte TELEX+3 ist die Länge n der zu übertragenden Daten (zählt ab Telex +4) einzutragen. (<250) Im Datenteil des Telegramms ab TELEX+5 schliessen sich Datensätze an, die jeweils durch ein Steuerbyte eröffnet werden.

Mit dem Steuerbyte wird eine der 3 DUE-Routinen:

- KOM-DATEN-Lesen
- KOM-DATEN-Schreiben
- BIT-SCHREIBEN in den KOMs

für den jeweiligen Datensatz festgelegt. Der Aufbau dieser Datensätze ist in 7.4.2.2. beschrieben.

1.3.2. Ausgangsgrössen:

Wenn im TELEX Steuerbytes zum KOM-DATEN-Lesen verschlüsselt sind, so werden bei korrekter DUE in ANTW die Antwortdaten dichtgespeichert in Reihenfolge gemäss Datensätzen, aus TELEX abgelegt. Bei fehlerhafter DUE (FEZD # 0) keine Ausgangsdaten. In KOM Blöcke einzuschreibende Daten (KOM-DATEN-Schreiben, Bit-SCHREIBEN) sind in die BSE richtig eingeschrieben, wenn FEMI Bit 2,3,4 = 0 sind. (Dieser Rücklesetest erfolgt nur bei PRIO = 4.)

DUSA

UP-Nr.: 74

Blatt 4 6.1

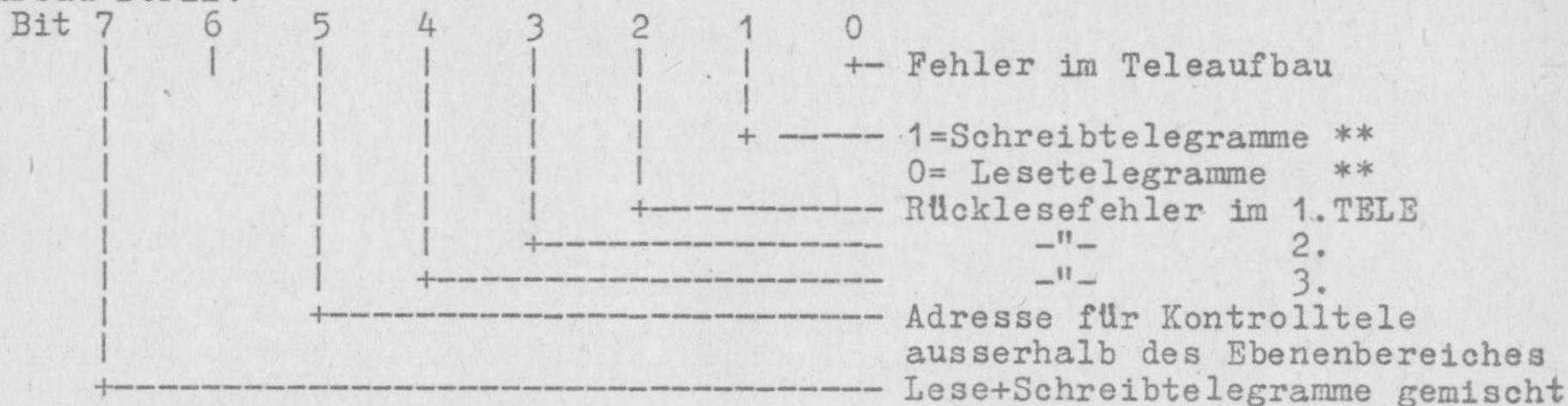
1.3.3. Unveränderte Register:

Nach UP-Aufruf sind sämtliche Register und der Stack verändert.

1.4. Fehler:

Nach Rückkehr aus dem UP-DUSA in das VAP muss der Fehlerzähler FEZD =0 und FEMI Bit 0=0 sein. Ist diese Bedingung nicht erfüllt, ist die DUE nur teilweise oder gar nicht ausgeführt worden. In dem Falle ist eine Auswertung der möglichen Fehlerursachen mit Hilfe des Fehlermerkerfeldes FEMI durch den Programmierer vorzunehmen. Zur Fehlerselektierung stehen im FEMI folgende Informationen zur Verfügung.

Aufbau DUFEL:

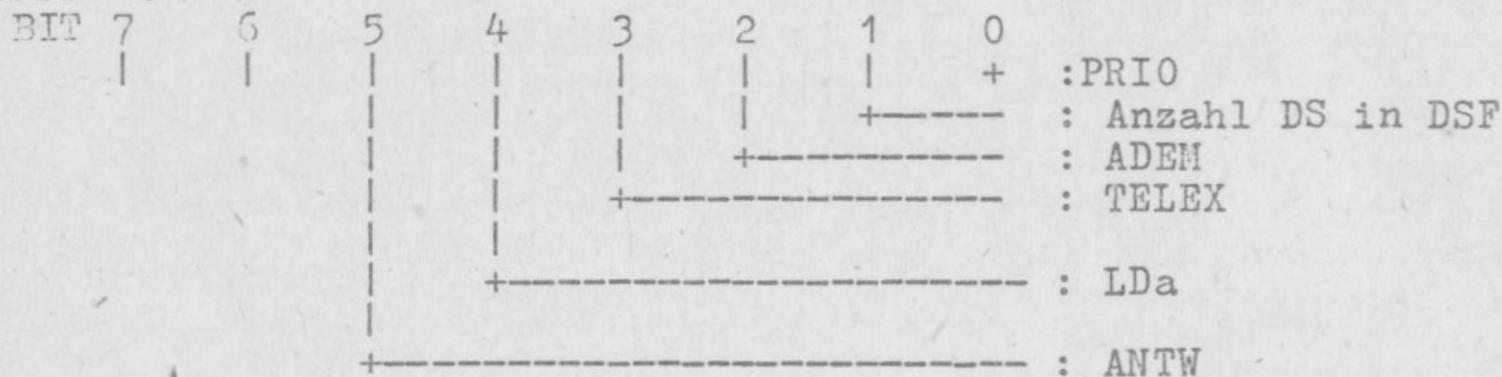


\*\* = kein Fehler sondern nur internes Merkbit !!!

Bit 0 =1 weist auf Fehler Bit 7=1 oder Fehler in FEMI+1 bis +6 hin

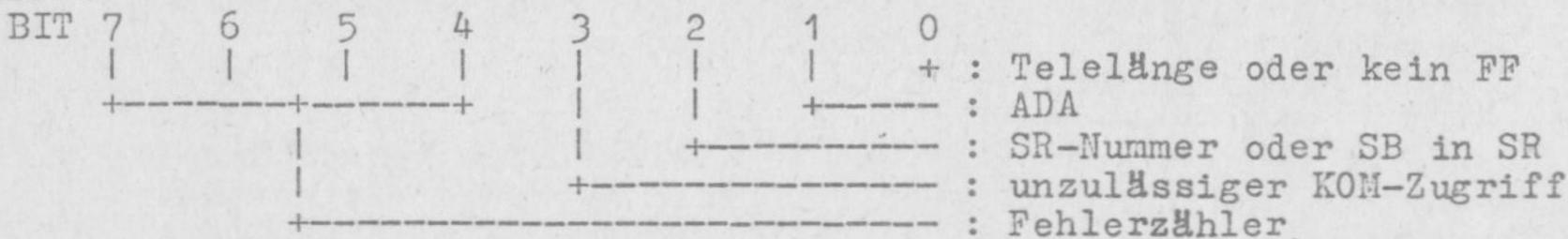
Fehler beim Aufbau des DSF:

DSFFE :



Fehler beim Telegammaaufbau:

TELF:



Fehler beim Sinnfälligkeitstests :

PRIO :# 04,05

ANZA : wenn nach angegebener Anzahl DS kein OFFH in SF oder wenn keine Übereinstimmung mit Anzahl der DS in DSF

DUSA

UP-Nr.: 74

Blatt 5

6.1

-----

ADEM : max.19  
 Telex,  
 ANTW : Fehler wenn Adresse ausserhalb Ebenenbereich  
 LDA wenn >250  
 SB : SR-Nummer ≠ 6CH  
 Zugriff nur auf zugelassene Bereiche

Aufbau SB: Bit 7 =1 = Schreiben  
                   =0 = Lesen  
           Bit 6,5 ADA  
           Bit 4-0 Anzahl Byte (0=1 Byte,...)

1.5. Rechenzeit: 20 ms Maximalwert (ohne Gesamtfunktion)

-zulässige Adressierungsarten:

ADA = 10 indirektes Lesen/Schreiben  
       11 Sonderregime

2. Beispiel:

In der BSE 3 soll mittels KOM-Schreiben in IMEN 0 (Analoger stetiger KOM) der "untere Grenzwert" UW2 und in IMEN 2 (Binärer Aggregate KOM) mittels Bitsschreiben das Betriebszustandsbyte geändert werden. Die DUE soll mit dem Regime mit Rücklesekontrolle (PRIO=4) erfolgen.

Folgende Datenfelder sind bereitzustellen:

DSF:    DEFB 4       ; PRIO  
           DEFS 1  
           DEFB 1       ; ANZAHL TELE  
           DEFB 2       ; BSE 3  
           DEFB 0       ; FEZD  
           DEFW TELEX  
           DEFB 0  
           DEFW ANTW  
           DEFB OFFH   ; Endekennung

FEMME:   DEFS 7       ; PUFFER für Fehlermitteilungen

TELEX:    DEFS 3       ; Telegrammkopf  
           DEFB 12      ; DL  
           DEFS 1  
           DEFB 0C1H   ; 2 Byte Schreiben  
           DEFB 0       ; IMEN  
           DEFB 6       ; STLG  
           DEFW 0010H  ; neuer Wert  
           DEFB 6C      ; SR  
           DEFB 0C1H  
           DEFB 1       ; IMEN  
           DEFB 6       ; STLG  
           DEFB 7H      ; Maske  
           DEFB 01H  
           DEFB OFFH   ; Endekennung

ANTW      DEFS 2+1    ; für Antwortdaten

TELEX 2:  DEFS 15

DUSA

UP-Nr.: 74

Blatt 6

6.1

Aufruf im UAP: LD HL, DSF  
LD DE, TELEX 2  
LD IY, FEMME  
RST 10H  
DEFB 74; DUSA

wenn <FEMME+0>=2 und <FEZD>=0 so ist DUE und Einschreibevorgang  
ordnungsgemäss abgelaufen.

DULAPI

UP-Nr.: 75

6.2

Laden Informationspuffer PIPU durch PDVP

1.1. Programmname : DULAPI

1.2. Funktion des Programms :

Die Dialog-VAP's mit dem PSR (PDVP) müssen zur Vorbereitung der Informationsübertragung an den PSR den PSR-Informationspuffer über das UP DULAPI laden. Das PDVP hat die Informationen in einem programmeigenen Datenfeld aufzubereiten und bereitzustellen, dessen Anfangsadresse beim UP Anspruch im HL-Register zu übergeben ist.

1.3. Schnittstellenbeschreibung

1.3.1. Eingangsgrößen :

In HL Anfangsadresse des in den Systemeigenen PIPU zu übertragenden VAP-eigenen Speicherbereiches (im folgenden auch als PIPU bezeichnet!). Die Länge beträgt max. 0F6H+1 Bytes.

Die Strukturierung der Informationen hat gemäß der Beschreibung in Abschnitt 5.3.4.2. zu erfolgen.

Aufbau eines Puffers zur Dialogsteuerung :

PIPU+0	I	Datenlänge	I	zählt ab PIPU+1 !!!
PIPU+1	I	...	I	Datensätze mit KSD (A0-A7)+
	I	...	I	Bildadressen +
	I	...	I	Daten bzw. Info
PIPU+0F6H	+		-	

Die übertragbaren Datensätze enthalten nach einem Kommunikationssteuerbyte (KSD) Bilddaten und (oder) Informationen zum Bildaufbau bzw. seiner Aktualisierung.

1.3.2. Ausgangsgrößen:

Im zentralen PIPU des WR ist der Speicherinhalt des PDVP-eigenen Informationspuffers abgelegt und wird bei der nächsten Anforderung des PSR zum dialogführenden PSR übertragen.

1.3.3. Unveränderte Register : IX, IY

1.4. Fehler

Bei fehlerhaften KSD (Überschreitung des zulässigen Darstellungsfensters auf dem PSR-Bildschirm, falsche KSD, falsche Datenlänge in PIPU) wird im Puffer PAPU eine Fehlermitteilung mit dem Kommando byte = 91H gesendet, und mit dem UP DULOPA dem PDVP in PAPU+1 zur Verfügung gestellt.

Z = 1 => PIPU zu lang, Nettoinfo > 0F6H

CY = 1 => letzte Info wurde noch nicht durch PSR übernommen

Abmeldung des PDVP's nach Vorschrift überprüfen !!!

DULAPI

UP-Nr.: 75

Blatt 2 6.2

1.5. Rechenzeit: 0,2 ms2. Beispiel:

Mit Hilfe des UP-DULAPF soll der Text "TEST" am dialogführenden Pult auf die Bildschirmrelativposition 0 mit Farbcode 10H ausgeschrieben werden.

Bereitzustellender Puffer:

```

PIPU: DEFB 9 ; DL
      DEFB 0A1H ; KSD siehe Beschreibung BE der PLE Teil 9A
      DEFW 0 ; Position
      DEFB 10H ; Farbe
      DEFM 'TEST'
      DEFB 0 ; Abschlusskennzeichen

```

```

LD HL, PIPU
RST 10H
DEFB 75 ; DULAPI

```

DULOPA

UP-Nr.: 76

6.3

## Bereitstellen Anforderungspuffer PSR- Dialog

1.1. Programmname: DULOPA

1.2. Funktion des Programms :

Über das UP DULOPA wird für Dialog- VAP's (PDVP) die Bereitstellung des PSR- Anforderungspuffers (PAPU) realisiert und ggf. die Sollzeitüberwachung (SZU) bei Dialogbetrieb mit einem PSR gestartet, die Sollzeit eingetragen bzw. gestoppt.

Im Register DE wird die Adresse des programmeigenen Papu's, auf welche der PSR- Anforderungspuffer zu laden ist, übergeben.

Wenn eine SZU erfolgen soll, muss zur Eintragung in >A< die Sollzeit in -min- übergeben werden. Möglichkeiten der Differenzierung s.1.3.1.

Aufbau des PAPU's:

PAPU+0	I Datenlänge	I
PAPU+1	I Steuerbyte	I *
PAPU+2	I Kdo.-byte	I
PAPU+3	I Telenummer	I
PAPU+4	I 28 SIF 1000	I
	I Zeichen	I
	I aus Kdo.Zeile	I
	I des PSR	I

\* : mögliche Zustände : 00 = keine Info von PSR da  
 77H= Info von PSR ab PAPU+4 da  
 91H= letzte Info an PSR war fehlerhaft  
 ( falsche KSD ... o.a. )

1.3. Schnittstellenbeschreibung

1.3.1. Eingangsgrößen :

(DE):= Adresse des pgr.eigenen PAPU's  
 A= OFFH , Z=0 => Sperren SZU  
 A= Zeit in min , Z=0 => aktivieren SZU  
 A= beliebig , Z=1 => keine Veränderung

1.3.2. Ausgangsgrößen:

- Sollzeitüberwachung entsp. Eingangsdaten
- PAPU geladen bzw. Steuerbyte entsprechend gesetzt
- EI = interrupt enable

1.3.3. Unveränderte Register : IX, IY

DULOPA

UP-Nr.: 76

Blatt 2 6.3

1.4. Fehler:

CY = 1 =&gt; DE im unzulässigen Bereich

1.5. Rechenzeit:

2. Beispiel :

LD	A,3	; SollzeitÜb. nach 3 min
AND	A	; Z = 0 => eintragen
LD	DE,PAPU	
RST	10H	
DEFB	76	; DULOPA
JP	C,FEHLER	

DUABPD

UP-Nr.: 77

6.4

Abmelden des dialogführenden Rahmenprogramms (PDR)  
im WR

1.1. Programmname : DUABPD

1.2. Funktion des Programms :

Das UP DUABPD ist zur Abmeldung des dialogführenden Rahmenprogramms PDR bestimmt. Es erfolgt dessen Austragung aus der Anmelderegisterliste AMR und die Freigabe der Dialogpuffer PAPU und PIPU.

Das dialogberechtigte Programm PDVP muss danach, wie in jedem Restartzyklus sich selber mit UP AFABPG abmelden. Bei der nächsten Abforderung von Information durch den PSR wird in Zeile 30 am PSR die Ausschrift "DIALOGENDE" zur Anzeige gebracht.

Aufruf erfolgt mit Restart- Befehl.

1.3. Schnittstellenbeschreibung

1.3.1. Eingangsgrößen: keine

1.3.2. Ausgangsgrößen:

- dialogführendes Pgr. PDR abgemeldet
- Telegramm "DIALOGENDE" fuer PSR bereitgestellt

1.3.3. Unveränderte Register : IX, IY

1.4. Fehler : keine

1.5. Rechenzeit:

2. Beispiel :

```

PDVPENDE:  RST  10H
            DB   77
            LD   B,VAPNR.
            RST  10H
            DEFB 6          ; PDVP abgemeldet AFABPG
            RST  10H
            DEFB 8          ; Rücksprung Steuerprogramm AFRUVP
  
```

EAST

UP-Nr.: 78

6.5

Steuerung der Ein-/Ausgabeanforderungen von Programmen der WRE an die Ein-/Ausgabegeräte Kassettenmagnetband bzw. Floppy-Disk und Drucker

1.1. Programmname: EAST

1.2. Funktion des Programmes:

Das UP übernimmt die Verteilung der E/A-Anforderungen der Verarbeitungsprogramme (VAP) der WRE auf die E/A-Geräte. Das Ein-/Ausgabeprogramm arbeitet mit dem zentralen Steuerprogramm STEU zusammen. Es wird von den Verarbeitungsprogrammen als Unterprogramm aufgerufen. Bei freiem Gerät wird die E/A-Operation eingeleitet und in das Steuerprogramm zur Bearbeitung des nächsten VAP zurückgesprungen.

Nach beendeter E/A-Operation wird das aufrufende VAP auf der Adresse nach E/A-Aufruf fortgesetzt. Dem VAP wird im Status des E/A-Steuerfeldes (EASF) bei aufgetretenem Fehler eine Fehlernummer übergeben. Bei fehlerfreier Ausführung ist das Statusbyte Null.

Ist das E/A-Gerät durch ein anderes VAP belegt, wird durch das Steuerprogramm ein zyklischer Test auf Gerätefreigabe organisiert. Die Dauer des Testes wird durch das VAP beim E/A-Aufruf im Register A festgelegt. Ist nach abgelaufener Wartezeit das Gerät noch belegt, wird das VAP mit Fehlermeldung fortgesetzt. Erfolgt innerhalb der Wartezeit die Gerätefreigabe, wird die E/A-Operation eingeleitet.

1.3. Schnittstellenbeschreibung

1.3.1. Eingangsgrößen:

A-Register:      Wartezeit auf Ein/Ausgabe  
                   Bit 0-5 Betrag 0....63  
                   Bit 7    0 Zeit in Sekunden  
                               1 Zeit in Minuten

Aufruf aus VAP: RST 10H  
                   DEFB 78           ; EAST  
                   DEFW >EASF<

Die benötigten E/A-Felder müssen auf den Ebenen liegen, die im jeweiligen VAP-Kopf definiert sind.

Das EASF hat einen festen Aufbau. Das EASF, sowie die bei den einzelnen E/A-Operationen beschriebenen Felder sind vor dem Aufruf einer E/A-Operation mit dem entsprechenden Inhalt bereitzustellen. Wird nach Floppy-Disk-Aufruf das VAP mit Fehler im Statusbyte fortgesetzt, kann nicht garantiert werden, dass die eingetragene Adresse ADR im EASF noch vorhanden ist.

Aufbau des EASF	+-----+	
	I SGN I	Symbol.Geräte-Nr.
	-----	
	I- ADR -I	Adresse
	-----	
	I SBY I	Steuerbyte
	-----	
	I STA I	Status
	+-----+	

EAST

UP-Nr.: 78

Blatt 2 6.5

Der Inhalt des EAST ist abhängig von der gewünschten E/A-Operation.

a) FUNKTIONEN DES EAST (WRE BS-Version A für WRE-Typ 621.33)

-Einlesen von Dateien vom Magnetband (MB)

SGN: 81H MB1  
82H MB2  
83H MB3  
84H MB4

ADR: Adresse der Dateinamensliste (DANAL)

Aufbau DANAL	+-----+	
	I DANA 1 I	DANA 1-n:Dateinamen,
	-----	der zu ladenden Dateien
	I DANA 2 I	(je DANA 8 Byte)
	-----	
	I : I	
	-----	
	I DANA n I	
	-----	
	I OFFH I	Endekennzeichen
	+-----+	

SBY: 0

- Schreiben von Dateien auf Magnetband

SGN: 01 MB1  
02 MB2  
03 MB3  
04 MB4

ADR: Adresse der Speicherbereichsliste (SPBEL)

Aufbau SPBEL	+-----+	Dateiname
	I I I	max.8 Alpha-n Zeichen im
	I DANA I	ASCII-Code, freie Zeichen
	I I I	mit 20H auffüllen,
	-----	1.Zeichen Buchstabe
	I EBCO I	Ebenencodierung (siehe
	-----	UP Anlage 1/2.6.)
	I ANF I	
	-----	Anfangs- und Endadresse
	I END I	Speicherbereich
	-----	
	I LEBCO I	
	-----	Für Listenkopf
	I LADR I	
	-----	

SBY: 04

- Fortschreiben von Dateien auf Magnetband

Dateien, die schon auf MB existieren, können mit diesem Kommando erweitert werden. Dabei wird die schon bestehende Datei in ihrer Länge auf ein Vielfaches von 100H aufgerundet und anschliessend um 100H Byte erweitert.

SGN: 11H MB1  
12H MB2  
13H MB3  
14H MB4

ADR: Adresse der Speicherbereichsliste (SPBEL)

EAST

UP-Nr.: 78

Blatt 3 6.5

```

Aufbau SPBEL  +-----+
                I  DANA  I      Dateiname der zu erwei-
                -----      ternden Datei (8 Byte)
                I  EBCO  I      Ebenencodierung
                -----
                I   ANF  I      Anfangs-und
                -----
                I   END  I      Endadresse (100H)
                -----
                I  LEBCO I      wie bei Datei schreiben
                -----
                I  LADR  I      "
                +-----+

```

SBY: 04

-Ausgabe auf Drucker (DR)

SGN: 05 DR1

06 DR2

07 DR3

ADR: Adresse Druckpuffer (DRUPU)

Aufbau DRUPU Byte 0: Steuerzeichen

00H keine Operation

0AH Zeilenvorschub

0CH Seitenvorschub

Durch den Druckerhändler erfolgt vor der Ausgabe einer Zeile das Senden des Steuerzeichens CR (Wagenrücklauf). Zeilen- und Seitenvorschub müssen durch das VAP (Byte 0) selbst gesteuert werden.

Folgebytes: Textzeichen im ASCII-Code.

Sind in der Druckzeile Steuerzeichen enthalten, kann es zu Fehlfunktionen im Drucker kommen. Nichtdruckbare Zeichen führen zur Fehlermeldung (Ablauffehler).

SBY: Länge Druckpuffer einschliesslich Steuerzeichen  
Der Druckpuffer ist max. 81 Byte lang.

b) FUNKTIONEN DES EAST (WRE BS-Version B und C für WRE-Typ  
621.33/1 und 621.33/2)

- Einlesen von Dateien von Floppy-Disk (FD)

SGN: 81H FD1

82H FD2

83H FD3

84H FD4

ADR: Adresse der Dateinamensliste (DANAL)

```

Aufbau DANAL  +-----+      DANA 1....n: Dateinamen,
                I DANA 1 I      der zu ladenden Dateien.
                -----      (je DANA 12 Byte)
                I DANA 2 I
                -----
                I   :   I
                -----
                I DANA n I
                -----
                I OFFH I      Endekennzeichen
                +-----+

```

SBY: 00

## - Schreiben von Dateien auf Floppy-Disk

SGN: 01 FD1  
 02 FD2  
 03 FD3  
 04 FD4

## ADR: Adresse Speicherbereichsliste (SPBEL)

Aufbau SPBEL	+-----+		
	I DANA I	DANA:	Dateiname (12 Byte)
	-----		
	I EBCO I		Speicherebene
	-----		
	I ANF I		Anfangsadresse Speicherb.
	-----		
	I END I		Endadresse Speicherber.
	-----		
	I LEBCO I		Rückladeebene
	-----		
	I LADR I		Rückladeadresse
	+-----+		

Der Dateiname muss den SCP-Festlegungen entsprechen. Er besteht aus max. 8 Zeichen Namen, einem Punkt und 3 Zeichen Dateityp. Freie Positionen sind mit Blank (20H) aufzufüllen.

## Einschränkungen WRE:

- Zeichenvorrat GRW-Bedientastatur (Techn. Dok. Teil 7)
- 1. Zeichen muss ein Buchstabe sein
- Dateityp nicht mit B beginnend

## - Fortschreiben von Dateien auf Floppy-Disk

Dateien, die schon auf FD existieren, können mit dieser Operation erweitert werden. Dabei wird die schon bestehende Datei in ihrer Länge auf ein Vielfaches von 80H aufgerundet und anschliessend um 80H erweitert.

SGN: 11H FD1  
 12H FD2  
 13H FD3  
 14H FD4

## ADR: Adresse Speicherbereichsliste (SPBEL)

Aufbau SPBEL	+-----+		
	I DANA I	DANA:	siehe "Datei schreiben"
	-----		
	I EBCO I		Speicherebene
	-----		
	I ANF I		Anfangsadresse Speicherbereich
	+-----+		

SBY: 04

## - Schreiben von Dateien mit BAK-Datei auf Floppy-Disk

In der SPBEL werden zwei Dateinamen angegeben.

DANA1- Dateiname der zueröffnenden Datei

DANA2- Name der Datei, die in .Bxx umbenannt werden soll. B als erstes Zeichen des Dateityps kennzeichnet bei "audatec" eine BAK-Datei. Es können "?" für Positionen im Namen angegeben werden.

Folgende Fälle sind möglich

EAST

Up-Nr.: 78

Blatt 5 6.5

1. Es existiert keine Datei, deren Name auf den signifikanten Positionen mit DANA2 übereinstimmt.  
Reaktion: - Datei DANA1 wird geschrieben
2. Es existiert eine Datei deren Name (DATNAME.nnn) auf den signifikanten Positionen mit DANA2 übereinstimmt.  
Reaktion: - Datei DANA1 wird geschrieben  
- Datei DATNAME.nnn wird in DATNAME.Bnn umgewandelt
3. Zusätzlich zu der Datei wie in 2. existiert noch eine BAK-Datei DATNAME.Bnn, deren Name auf den signifikanten Positionen mit DANA2 übereinstimmt.  
Reaktion: - Datei DANA1 wird geschrieben  
- BAK-Datei DATNAME.Bnn wird gelöscht  
- Datei DATNAME.nnn wird in DATNAME.Bnn umgewandelt
4. Es existiert nur eine BAK-Datei, deren Name auf den signifikanten Positionen mit DANA2 übereinstimmt.  
Reaktion: - Datei DANA1 wird geschrieben

SGN: 21H FD1  
22H FD2  
23H FD3  
24H FD4

ADR: Adresse Speicherbereichsliste (SPBEL)

```
Aufbau SPBEL  +-----+
                I  DANA1 I
                -----
                I  DANA2 I
                -----
                I  EBCO  I
                -----
                I   ANF  I
                -----
                I   END  I
                -----
                I  LEBCO I
                -----
                I  LADR  I
                +-----+
```

Erläuterungen:  
siehe "Datei schreiben"

SBY: 04

- Fortschreiben von Dateien mit BAK-Datei auf Floppy-Disk  
Dateien, die schon auf FD existieren, können mit dieser Operation erweitert werden, wobei gleichzeitig eine BAK-Datei angefertigt wird. Dabei werden die Dateien vor jedem Datensatzschreiben zum Fortschreiben eröffnet und danach sofort wieder geschlossen, bevor die andere Datei eröffnet wird. Falls nur eine Datei existiert (DANA oder DANA.Bnn), wird die nicht existierende Datei durch kopieren der anderen erzeugt.

SGN: 31H FD1  
32H FD2  
33H FD3  
34H FD4

ADR: Adresse Speicherbereichsliste (SPBEL)

EAST

UP-Nr.: 78

Blatt 6

6.5

Aufbau SPBEL

```

+-----+
I  DANA  I
-----
I  EBCO  I
-----
I   ANF  I
-----
I   END  I
-----
I  LEBCO I
-----
I  LADR  I
+-----+

```

Erläuterungen:  
siehe "Datei schreiben"

SBY: 04

- Einlesen eines 80H-Datensatzes (ab BS-Version C )  
Mit dieser Funktion kann der angegebene 80H Datenblock einer Datei in den Anwender-RAM-Bereich eingelesen werden. Blocknummer 1 ist der erste Datensatz der Datei. Durch die Angabe der Ladeebene und der Ladeadresse ist es möglich, den Datenblock auf einen beliebigen Bereich innerhalb des Anwenderbereiches zu laden. Die Angaben im Dateikopf werden bei dieser Operation nicht berücksichtigt.

SGN: 91H FD1  
92H FD2  
93H FD3  
94H FD4

ADR: Adresse der Dateinamensliste (DANAL)

```

Aufbau DANAL  +-----+  Dateiname der zu ladenden
                I  DANA  I  Datei (12 Byte)
                -----
                I  BLNR  I  2 Byte Blocknummer
                -----
                I  LEBCO  I  Ladeebene
                -----
                I  LADR  I  Ladeadresse
                +-----+

```

SBY: 0

- Ausgabe auf Drucker (DR)  
siehe Funktionen des EAST

BEACHTEN ! Bei Betriebssystemversion 2 C (mit Fremdrechner-schnittstelle) entfällt DR 3

### 1.3.2. Ausgangsgrößen:

-----  
Statusbyte (STA) im Ein-/Ausgabesteuerfeld

### 1.3.3. Unveränderte Register: keine

### 1.4. Fehler

#### a) EAST

#### Gerätefehler

21H -ISI Zeitüberschreitung  
22H -Gerätefehler

-----EAST-----UP-Nr.:78-----Blatt 7-----6.5-----

23H -Ablauffehler  
 Zuweisungs-/Übergabefehler  
 41H -SGN nicht belegt  
 42H -SGN fehlerhaft  
 43H -Gerät belegt  
 46H -Adresse verboten  
 47H -Einlesen im ON nicht möglich  
 48H -Fehler im VAP-Kopf

## Dateifehler

61H -Datei nicht auf MB  
 63H -MB voll  
 Die Fehler 46H....63H sind nur bei MB-Arbeit möglich.

## b) FDEAST

## Gerätefehler

21H -Zeitüberschreitung  
 22H -Gerätefehler  
 23H -Ablauffehler  
 Zuweisungs-/Übergabefehler  
 61H -SGN nicht belegt  
 62H -SGN fehlerhaft  
 63H -Gerät belegt  
 64H -Dateiname beginnt nicht mit Buchstabe  
 66H -Adresse verboten  
 67H -Einlesen im ON nicht möglich  
 68H -Fehler im VAP-Kopf

## Datei-/Diskettenfehler

40H -Sektor nicht gefunden  
 42H -Schreiben verboten  
 43H -Datei schreibgeschützt  
 4DH -Dateityp mit B  
 4EH -Dateiname zu lang  
 4FH -Diskette voll  
 50H -Dateiverzeichnis voll  
 51H -Datei existiert schon  
 52H -Datei existiert nicht  
 5AH -Fehler beim formatieren  
 Die Fehler 64H....5AH sind nur bei FD-Arbeit möglich.

1.5. Rechenzeit: 5,0 ms Maximalwert (ohne Gesamtfunktion)

2. Beispiel:

Aufgabe : Ausgabe des Textes "TEST" auf Drucker 2

LD A, 45 ; 45 sec. Wartezeit  
 RST 10H  
 DEFB 78  
 DEFW EASF

EASF: DEFB 6 ; SGN  
 DEFW DRUPU  
 DEFB 5 ; SBY  
 DEFB 0 ; Status

DRUPU: DEFB OAH ; LF  
 DEFM 'TEST'

EAGFG

UP-Nr.: 79

6.6.

## Freigabe Ein-/Ausgabe-Geräte

1.1. Programmname: EAGFG1.2. Funktion des Programmes:

Das UP hebt die Reservierung eines Peripheriegerätes für ein VAP wieder auf nach beendeter E/A-Operation.

1.3. Schnittstellenbeschreibung:1.3.1. Eingangsgrößen:

HL-Register: Adresse Ein-/Ausgabe Steuerfeld (EASF)  
 Aufbau EASF im UP-Kennblatt: Steuerung der Ein-/Ausgabe-Anforderungen an die E/A-Geräte

1.3.2. Ausgangsgrößen: keine1.3.3. Unveränderte Register: IY1.4. Fehler:

CY=1 Fehler SGN

1.5. Rechenzeit: 0,2 ms2. Beispiel:

```
LD HL, EASF
RST 10H
DEFB 79 ; EAGFG
```

DUSZYK

UP-Nr.: 73

6.7

Zeigerlesen zu Mitteilungen aus der  
Systemkommunikation1.1. Programmname: DUSZYK1.2. Funktion des Programmes:

Vor oder nach ereignisorientiertem Zugriff auf Prozessdaten über UP DUSA kann mit diesem UP der Zeiger (SYZ) auf die letzte Änderungsmitteilung des Systems zu Änderungen in einer BSE durch Systemkommunikation abgefragt werden. Das nutzende VAP muss hierzu in einem internen Merker die Zeiger-Nr. (ASYK) seiner letzten vom VAP ausgewerteten Änderung zum Vergleich ablegen. Der Zeiger wird mit 8 Bit Integer als Rundpuffer betrieben. Wenn der Vergleich positiv ausfällt, ist UP DUSYKO anzuwenden (siehe dort).

1.3. Schnittstellenbeschreibung1.3.1. Eingangsgrößen: keine1.3.2. Ausgangsgrößen: A:=SYK Zeiger-Nr(Integer 0...0FEH)1.3.3. Unveränderte Register: HL,BC,DE,IY,IX1.4. Fehler: -1.5. Rechenzeit: 0,15 ms

DUSYKO

UP-Nr.: 80

6.8

Lesen des Änderungspuffers zu Mitteilungen  
aus der Systemkommunikation an BSE'n

1.1. Programmname: DUSYKO

1.2. Funktion des Programms:

1.2.1. Allgemeine Beschreibung

Das UP dient speziell bei VAP mit ereignisorientiertem Prozessdatenzugriff dazu, durch Systemkommunikation eingetretene Änderungen in einer BSE für eine Testung im VAP bereitzustellen. Eine Änderung umfasst einen Datensatz von 3 Bytes an Änderungsinformationen.

Aus dem Änderungspuffer der Systemkommunikation (SYIN) wird jeweils der Datensatz aus dem Puffer gelesen, der sich aus der Differenz zum aktuellen Zeiger bei UP-Ansprung ableitet. Der aktuelle Zeigerstand (SYZ), der die letzte Änderung vermerkt, ist mittels UP DUSZYK zu ermitteln. (vgl. Anl. 1/6.7.)

Der Änderungspuffer SYIN ist als Rundpuffer für 20 Datensätze organisiert, ebenso der Zeiger SYZ für 255 Schritte. Im VAP sollte in einem internen Merker die alte Zeigernummer (ASYZ) der letzten vom VAP ausgewerteten Änderung des Puffers abgelegt werden und zum UP-Aufruf übergeben werden. Ausserdem ist zur Kontrolle der zugehörige Stand von SYZ zu übergeben.

Wenn die Differenz grösser 20 ist, so ist der Änderungspuffer SYIN mindestens einmal vollständig neu aufgefüllt worden, d.h. das betreffende VAP ist mit der Aktualisierung nicht nachgekommen. In diesem Falle ist ggf. der Restartzyklus des VAP's zu ueberprüfen.

1.2.2. Anwendung des UP

Bei UP-Aufruf ist in <C> der mit UP DUSZYK abgefragte Zeigerstand SYZ und in <B> der Wert des Zeigers (ASYZ) des zuletzt ausgewerteten Datensatzes zu übergeben. Bei C-B=1 wird die letzte Änderung in SYIN, bei C-B=2 die vorletzte Änderung in SYIN usw. bereitgestellt.

Bei mehrmaligem fortlaufenden Anspruch von DUSYKO kann der gesamte Änderungspuffer zur Auswertung gelesen werden. Allerdings ist dabei zu beachten, dass zwischen 2 UP-Aufrufen bereits neue Änderungsmitteilungen eingelaufen sein können. Je nach Restartzyklus des VAP's und Änderungsrate durch die Systemkommunikation kann diese Situation eintreten. Im allgemeinen sollte der Restartzyklus eines VAP zur möglichen Änderungsrate von IMEN durch Systemkomm. in einem solchen Verhältnis stehen, dass der Fall der mehrfachen Änderung als Ausnahme gilt.

Der Datensatz zu einer Änderungsmitteilung aus dem Puffer SYIN wird vom UP jeweils in den Registern A,D,E bereitgestellt. Nachfolgende Änderungsinformationen können mitgeteilt werden:

DUSYKO

UP-Nr. 180

Blatt 2

6.8

Var.	Art der Änderung	Register		
		A	D	E
1	Änderung eines MAB	01	BSE-Nr.	IMEN
2	Änderung einer KOMS	02	BSE-Nr.	IMEN
3	Neustrukturierung IMEN	03	BSE-Nr.	IMEN
4	Ende Listenladen BSE	04	BSE-Nr.	OPPH

### 1.3. Schnittstellenbeschreibung

#### 1.3.1. Eingangsgrößen:

<B> = absoluter Zeiger (ASYZ) der zuletzt zur Auswertung zu übergebende Änderungsmitteilung  
 <C> = abgefragter Zeiger (SYZ) auf Änderungspuffer

#### 1.3.2. Ausgangsgrößen:

<A> = Kennbyte für Art der Änderung  
 <D> = BSE-Nummer  
 <E> = Spezifikation

#### 1.3.3. Unveränderte Register: BC, IY, IX

#### 1.4. Fehler:

Z = 0 ^ CY = 0 : kein Fehler  
 Z = 1 ^ CY = 0 : Eingangsparameter <C> nicht mehr aktuell  
 Z = 0 ^ CY = 1 : Anzahl der auszuwertenden Datensätze > 20  
 Z = 1 ^ CY = 1 : Anzahl Datensätze = 0, d.h. keine Mitteilungen eingelaufen (<B>=<C>)

Bei Fehlermitteilung Z = 1 oder C = 1 keine Datenbereitstellung !!!

#### 1.5. Rechenzeit: 0,2 ms

#### 2. Beispiel:

Mit UP-DUSYK sei SYZ=4 ermittelt worden. Im Beispiel steht der Merker ASYZ auf 252. Für den Algorithmus des Beispiel-VAP's sei es notwendig alle Änderungen zu erfassen. Aus den Werten SYZ und ASYZ kann die Differenz für die Anzahl der Änderungen (8) ermittelt werden. (Nr.4,3,2,1,0,255,254,253). Das UP-DUSYKO muss demzufolge 8 mal aufgerufen werden um alle Änderungsmitteilungen zu lesen. Dabei ist der Zeiger ASYZ, nach jeder Wertebereitstellung durch DUSYKO, zu inkrementieren, bis SYZ = ASYZ.

Technische Dokumentation BE der PLE Teil 9B

Schnittstellenbeschreibung für Anwenderprogramme  
der Wartenrechnereinheit WRE/1 und WRE/2.

Während des Druckes auftretende Änderungen

Seite 147/ 148 UP KNTAPL

Punkt 1.2 (S. 147)

Der Satz : "Sind weniger als 8 Tastencodes ..."  
muss richtig heißen:

"Sind weniger als 8 Tastencodes im Puffer abgelegt, wird der  
Rest mit FFH aufgefüllt."

Punkt 2 (S. 148)

Die Abbildung des Feldes TACO wird ersetzt durch:

1000H	_____	VAP zum ersten Mal nach Aufruf im Grundmenü der WRE angemeldet (TAQUIT)
	01	
	41H	
	9DH	
	FF	
	FF	in den Tastaturpuffer des Anwenders sind die Tasten
	FF	A     ENTER   abgespeichert worden
	FF	
	FF	
	FF	

**ACHTUNG :** Diese Tastencodedefolge in TACO steht nur eine Zeitscheibe lang ( 1/3 Sekunde) zur Verfügung. Danach erfolgt vom Betriebssystem eine Löschung auf FFH und das Einschreiben der Tastenfolge, die in der nachfolgenden 1/3 Sek. über die Tastatur eingegeben wurde.  
Wurde in einer Zeitscheibe keine Tastatureingabe getätigt, so ist der Inhalt von TACO+1 bis TACO+8 mit FFH in dieser Zeitscheibe gefüllt.  
Der Nutzer hat damit den Puffer TACO innerhalb der Zeitscheibe zu verarbeiten bzw. zur Verarbeitung in einer anderen Zeitscheibe zwischenspeichern. Der TABE-Teil des VAP'a (siehe VAP-Kopf Adresse TABE) wird durch das Betriebssystem jede 1/3 Sekunde abgearbeitet, so dass TACO jede Zeitscheibe im VAP (Teil TABE) ausgewertet werden kann.

---

## **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