

1.BASIC - Start

Beim Starten von BASIC mu~ der verf}gbare Anwenderbereich durch Eingabe von:

&8F00
eingeschr{nkt werden.

2.S V C - Routinen zur Arbeit mit der Ein- und der Ausgabetabelle

2.1. Setzen aller Werte in einer Tabelle:

```
SVC 0,0, eor_extern , eor_intern , kommando , anf.adr._kodetab. ,
      parity , eof , konvrout._adr , richtung , 0,70,0,&9004
```

+ 0	eor_extern	EOR - Zeichen auf Lochband
+ 1	eor_intern	EOR - Zeichen f}r BASIC (immer 0DH)
+ 4	kommando	Kommando - Byte f}r die Lochbandroutine
	Bit 0	... nicht benutzt
	1 = 0	... Transport r}ckw{rts bei Lesen
	1 = 1	... Transport vorw{rts bei Lesen
	2 = 0	... keine EOR-Umwandlung
	2 = 1	... EOR auf LB in 0DH bzw. ODH in EOR auf LB umwandeln
	3 = 0	... keine Kodekonvertierung
	3 = 1	... Kodekonvertierung
	4	... nicht benutzt
	5 = 0	... NUL und DEL lesen
	5 = 1	... NUL und DEL }berlesen
	6 = 0	... ungerade Parit{t (Bit 7=1)
	6 = 1	... gerade Parit{t (Bit 7=1)
	7 = 0	... ohne Parit{t
	7 = 1	... mit Parit{t
+ 5, 6	anf.adr._kodetab.	Anfangsadresse von Anwenderkodetabelle(n)
+11	parity	Parit{tsverkn}pfungszeichen Eingabe: eingelesenes_Zeichen AND parity Ausgabe: auszugebendes_Zeichen OR parity
+12	eof	EOF - Zeichen auf Lochband (bei Einlesen dises Zeichens wird ein Vermerk in der BASIC-Variablen "ER\$" eingetragen)
+19,20	konvrout._adr.	Adresse einer Anwenderroutine f}r die Lochband- kodekonvertierung
	richtung	Eingabe = &49 bzw. 73 Ausgabe = &4F bzw. 79

2.2 Setzen eines Wertes auf einen gezielten Tabellenplatz

```
SVC 0,0, rel._tabadr. , wert , richtung , 1,70,0,&9004
```

rel._tabadr.	relative Tabellenadresse die ver{ndert werden soll (siehe 2.2 erste Spalte)
wert	Wert, welcher in der Tabelle gesetzt werden soll

! 2.3. Standardeintrag f}r Ein- bzw. Ausgabe im bin{ren Format !
! ----- (Anwendung bei USR-Routine und bei BASIC-Programmein-/ausgabe) !

! SVC 0,0, richtung , 2,70,0,&9004 !

! (eor_extern=1EH, eor_intern=0DH, eof=19H, !
! kommando: Eingabe= 2 /Ausgabe= 0 , !
! parity : Eingabe=7FH/Ausgabe=80H, !
! anf.adr._kodetab. und konvrout._adr. gleich Null) !

! 2.4. Standardeintrag f}r Ein- bzw. Ausgabe im ASCII- Kode !
! ----- !

! SVC 0,0, richtung , 3,70,0,&9004 !

! (eor_extern=1EH, eor_intern=0DH, eof=19H, !
! kommando: Eingabe=E6H/Ausgabe=C4H, !
! parity : Eingabe=7FH/Ausgabe=80H, !
! anf.adr._kodetab. und konvrout._adr. gleich Null) !

! 2.5. Standardeintrag f}r Ein- bzw. Ausgabe im EIA- Kode (Anwenderkode) !
! ----- !

! SVC 0,0, richtung , 4,70,0,&9004 !

! (eor_extern=80H, eor_intern=0DH, eof=19H, !
! kommando=AEH, !
! parity : Eingabe=EFH/Ausgabe=10H !
! konvrout._adr.=Standardkonvertierung mit einer Tabelle, !
! anf.adr._kodetab.=Adresse der EIA-Kodetabelle bzw. einer !
! anderen generierten Standardtabelle) !

! 3. Routinen zur Byte- Ein-/Ausgabe !
! ----- !

! 3.1 Eingabe !
! ----- !

! SVC 0,0,75,0,&9004 : Z = USR(X) : SVC 0,44,0 !

! SVC 0,0,75,0,&9004 schaltet USR-Routine auf LB-Eingabe um !
! Z = USR(X) liest das n{chste Zeichen vom Lochband !
! auf die Variable "Z" . !
! (X ist eine Dummy-Variable) !

! SVC 0,44,0 schaltet USR-Routine auf Standard zur}ck !

! 3.2 Ausgabe !
! ----- !

! SVC 0,0,76,0,&9004 : X = USR(Z) : SVC 0,44,0 !

! SVC 0,0,76,0,&9004 schaltet USR-Routine auf LB-Ausgabe um !
! X = USR(Z) stanzt den Wert der Variablen "Z" auf !
! Lochband aus !

! SVC 0,44,0 schaltet USR-Routine auf Standard zur}ck !

4. Routinen zur Ein-/Ausgabe mit BASIC

4.1. Eingabe

```
SVC 0,0,71,0,&9004 : INPUT ... : SWITCH0
```

SVC 0,0,71,0,&9004 schaltet Konsoleninput auf Lochbandeingabe

4.2. Ausgabe

```
SVC 0,0,72,0,&9004 : PRINT ... : SWITCH0
```

SVC 0,0,72,0,&9004 schaltet Konsolenoutput auf Lochbandausgabe!

5. Routinen zur Ein-/Ausgabe von BASIC Programmen

5.1. Eingabe

SVC 0,0,73,0,&9004 : LOADA : SVC 0,65

SVC 0,0,73,0,&9004 schaltet Readerkanal auf Lochbandeingabe

SVC 0,65 schaltet Readerkanal zur}ck

5.2. Ausgabe

SVC 0,0,74,0,&9004 : SAVEA : SVC0,65

SVC 0,0,74,0,&9004 schaltet Stanzerkanal auf Lochbandausgabe

SVC 0,65 schaltet Stanzerkanal zur}ck

6. Fehler-Byte in der BASIC-Variablen "ER\$"

```
wenn MID$(ER$,7,1) = "1"      dann wurde als letztes Zeichen ein
                                EOF gelesen
```

wenn MID\$(ER\$,7,1) = "2" dann wurde als letztes Zeichen ein EOR gelesen

Bei allen anderen Zeichen wird `MID$(ER$,7,1) = "0"` gesetzt.