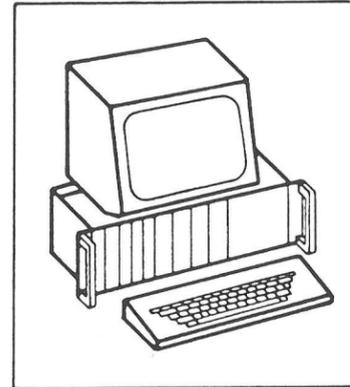


FACHPRAKTISCHE ÜBUNG MIKROCOMPUTER-TECHNIK



MAT 32 K

bzw.

MAT 32 K-Terminalversion



Wichtiger Hinweis

Im Zuge der Weiterentwicklung des MFA-Mikrocomputers sind einige wichtige Änderungen zu beachten, die in der vorliegenden Dokumentation zum Betriebssystem MAT32 K leider noch nicht berücksichtigt werden konnten. Die hier beschriebenen Änderungen beziehen auf das MAT32 K Vers.1.8/S sowie das MAT32 K Vers.1.0/T.

Schnellere Videoausgabe für Ausbaustufe MT

Mit dem vorliegenden MAT32 K-EPROM wurde eine "schnellere Videoausgabe" durch Änderung der seriellen Ein-/Ausgabe-Routinen im Betriebsprogramm MAT85 erreicht. Die Geschwindigkeit der Zeichenübertragung zwischen Mikroprozessor und Video-Interface wurde hierbei von 1200 Bit/s auf 9600 Bit/s erhöht (Baudrate). Je nach Art der verwendeten Video-Interface-Baugruppe sind hierzu noch geringfügige Hardwareänderungen vorzunehmen. Genauer Informationen stehen in der Beilage, die Teil der vorliegenden Dokumentation des Teil 1, Ausbaustufe MT ist.

Funktionsplanausgabe für Ausbaustufe MT und PC

Außer den SPS-Erweiterungen "Kontaktplanausgabe und Dokumentation" ist in der vorliegenden MAT32 K-Version auch eine Funktionsplanausgabe implementiert. Die Funktionsplanausgabe ist ebenso wie der SPS-Interpreter an die DIN angepaßt. Der Aufruf erfolgt aus dem SPS-Interpreter durch Eingabe des Anfangsbuchstaben des Kommandonamens, hier "V" für Verknüpfungsplan ("F" für Floppy).

Speicherbelegung für Ausbaustufe MT und PC

Durch die oben beschriebene Implementierung der Funktionsplanausgabe ist der in der Dokumentation als reserviert beschriebene Speicherbereich von 6000H bis 67FFH mit dem Programm zur Funktionsplanausgabe belegt.

BFZ/MFA-Editor für Ausbaustufe MT und PC

In der Beschreibung zum BFZ/MFA-Editor ist folgendes zu beachten: Der Hinweis auf S.2 trifft nur bei Verwendung des MFA-Video-Interface BFZ/MFA-8.2 zu. Wird das Video-Interface BFZ/MFA-8.4 verwendet, ist auch in der Bildschirmdarstellung eine Unterscheidung zwischen Klein- und Großbuchstaben gegeben. Die ab S.3 beschriebene Installation des Editors auf der Grundlage verschiedener Softwarekonfigurationen stellt sich für den MAT32 K-Benutzer nicht. Sollte der Aufruf des Editors mit dem GO-Kommando erfolgen, lautet die neue Startadresse 6800H. Da das EPROM mit dem MAT32 K den Adreßbereich von 0000H-7FFFH belegt, beginnt der Edit-Buffer erst ab Adresse 8000H (oder höher) und nicht ab 6000H wie in der Beschreibung. **Wichtig für Ausbaustufe PC.** Anstatt wie in der Beschreibung zur Cursor-Steuerung Control-C bitte Control-G benutzen. **Control-C** führt beim PC zum Ausstieg aus dem Terminalprogramm.

Inhaltsverzeichnis

1.	Allgemeines	2
2.	Beschreibung der Programmteile	2
2.1	Betriebsprogramm MAT85	3
2.2	Betriebsprogramm-Erweiterung SP1	3
2.2.1	Kommando-Erweiterung MAT85+	3
2.2.2	Steuer-BASIC-Interpreter	4
2.2.2.1	Anpassung alter BASIC-Programme	4
2.2.3	SPS-Interpreter	8
2.2.4	EPROM-Programmierer	8
2.3	Mini-DOS	8
2.4	SPS-Erweiterungen	9
2.4.1	Kontaktplanausgabe	9
2.4.2	SPS-Dokumentation	9
2.4.2.1	Programmtexter	10
2.4.2.2	Programmloader	14
2.4.2.3	Fehlerbehandlung	15
2.5	BFZ-Editor	17
3.	Inbetriebnahme-Hinweise MAT32	17
Anhang		
A1	MAT85 - und MAT85+ - Kommandos	18
A2	SPS - Kommandos	19
A3	Steuer-BASIC-Befehle	20
A4	Mini-DOS - Kommandos	24
A5	EDITOR - Kommandos	25
A6	SPS - Programmbeispiel	26
A7	Speicherbelegung	29
A8	Bestückungsplan 64-KByte-RAM	30

Funktionsbeschreibung

1. Allgemeines

Das hier beschriebene Betriebsprogramm MAT32 für das MFA-Mikrocomputersystem ist, bis auf eine Erweiterung zur Dokumentation von SPS-Programmen, lediglich eine Zusammenfassung aller verfügbaren Programmteile zum Betrieb des MFA-Computers in einem 32-KByte-EPROM vom Typ 27256 und nur für den Betrieb zusammen mit der 64-KByte-RAM-Baugruppe BFZ/MFA 3.3 vorgesehen. Daher ersetzt die hier vorliegende Unterlage nicht die zu den einzelnen Programmteilen gehörenden Funktionsbeschreibungen. Für die Handhabung und Bedienung des Betriebsprogramms mit allen Erweiterungen sind daher die folgenden Fachpraktischen Übungen und Unterlagen zusätzlich erforderlich:

- | | | |
|---|-------------|---------------------------------------|
| - MAT 85 (Betriebsprogramm) | BFZ/MFA 7.1 | } in den Fach-
theoret.
Übungen |
| - Software-Paket SP 1 (Betriebs-
programm-Erweiterung) | BFZ/MFA 7.2 | |
| - Fachpraktische Übungen Band 3
(Floppy-Disk-Controller und
Mini-DOS) | BFZ/MFA 4.7 | |

In dieser Beschreibung wird daher nur auf wesentliche Unterschiede oder einzelne Erweiterungen eingegangen. Ansonsten sind die Funktionen und die Bedienungen unverändert zu den bisher ausgelieferten Programmversionen. Aus diesem Grund ist diese Beschreibung sehr kurz gefaßt.

2. Beschreibung der Programmteile

Das Betriebsprogramm MAT32 enthält folgende Programm-Module:

- Betriebsprogramm MAT85
- Betriebsprogramm-Erweiterung SP1 mit
 - ...Kommando-Erweiterung MAT85+
 - ...Steuer-BASIC-Interpreter
 - ...SPS-Interpreter
 - ...EPROM-Programmierer
- Mini-DOS
- SPS-Erweiterungen
 - ...Kontaktplanausgabe
 - ...SPS-Dokumentation mit
 Programmloader und Programmtexter
- BFZ-Editor
 - ...Hilfsmenü

Funktionsbeschreibung

Diese sind in einem 32-KByte-EPROM vom Typ 27256 zusammengefaßt. In Verbindung mit der 64-KByte-RAM-Baugruppe besteht somit die Möglichkeit, den Systemspeicher mit nur einer Speicherbaugruppe voll auszubauen. Damit stehen dem Anwender alle Systemfunktionen sowie ein 32-KByte-RAM-Speicher zur Verfügung. Der 32-KByte-EPROM-Bereich mit allen Betriebsprogrammfunktionen bleibt stets aktiv. Damit ist die gleiche Betriebssicherheit wie bei der Nutzung der bisher verwendeten EPROM-Baugruppen gewährleistet.

Diese Konfiguration kann von Vorteil sein, wenn eine Reihe von Steckplätzen im Baugruppenträger beispielsweise für Interface-Baugruppen im Rahmen einer speziellen Anwendungen benötigt werden.

Es wird aber empfohlen, diese Ausbaustufe des MFA-Mikrocomputers lediglich im fortgeschrittenen Stadium der MC-Ausbildung oder im Rahmen der Unterrichtsvorbereitung zu nutzen und die didaktischen Vorteile der stufenweisen Ausbaubarkeit des Systems in der Vermittlung von MC-Kenntnissen für die Auszubildenden und Schüler nicht zu übersehen.

2.1 Betriebsprogramm MAT85

Das Betriebsprogramm MAT85 ist nur soweit verändert worden, daß die Baud-Raten-Erfassung nach dem Einschalten der Betriebsspannung zusammen mit der Betätigung der SPACE-Taste wegfällt. Daher meldet sich das Betriebssystem unmittelbar nach dem Einschalten mit dem Ausdruck der MAT85-Kommandos.

Die Zeitkonstante für die Baud-Rate der serielle Datenübertragung zum Video-Interface, die normal mit der Abfrage der SPACE-Taste ermittelt wird, wird nach einem Kaltstart des Gerätes direkt im RAM-Speicher unter der Adresse FCDFH und FCEOH (STIME) abgelegt.

2.2 Betriebsprogramm-Erweiterung SP1

Die Betriebsprogramm-Erweiterung SP1 besteht aus den vier Modulen Kommando-Erweiterung MAT85+, Steuer-BASIC-Interpreter, SPS-Interpreter und EPROM-Programmierer. Im wesentlichen wurden zwei Änderungen vorgenommen. Die erste Änderung bezieht sich auf das Steuer-BASIC. Um den Betrieb mit einem 32-KByte-EPROM im Systemspeicher zu ermöglichen, mußte der Arbeitsspeicher des BASIC-Interpreters von der Adresse 6000H nach 8000H verlegt werden. Die zweite Änderung bezieht sich auf den SPS-Interpreter, der an die Programmierung mit den DIN-Symbolen angepaßt wurde.

2.2.1 Kommando-Erweiterung MAT85+

In diesem Modul wurden keine Änderungen vorgenommen.

Funktionsbeschreibung

2.2.2 Steuer-BASIC-Interpreter

Der Steuer-BASIC-Interpreter wurde so verändert, daß der Arbeitsspeicher ab der Adresse 8000H, d.h. im zweiten 32-KByte-Block des Systemspeichers liegt. Hier beginnt bei Verwendung eines 32-KByte-EPROMs vom Typ 27256 mit der Betriebssoftware der RAM-Bereich auf der 64-KByte-Speicherbaugruppe. Der maximal zur Verfügung stehende Arbeitsspeicher ist somit gegenüber der normalen RAM-Startadresse von 6000H um 8-KByte kleiner geworden. Dies dürfte kein Problem darstellen, da das Steuer-BASIC nicht für extrem große Programme vorgesehen ist.

2.2.2.1 Anpassung alter BASIC-Programme

BASIC-Programme, die mittels der normalen Betriebsprogrammversion für die RAM-Startadresse 6000H entwickelt wurden und auf einer Diskette gespeichert sind, können an den neuen Arbeitsspeicher angepaßt werden. Die Schritte werden im folgenden beschrieben. Dazu sei das folgende Programmbeispiel betrachtet, welches unter der Normalversion für den RAM-Speicher ab 6000H erstellt wurde.

```
KMD+> BASIC
BFZ-STEUER-BASIC V2.4
```

```
READY
>FOR I=1 TO 10
>20 PRINT I
>30 NEXT I
>QUIT
```

Betrachtet man nach der Programmeingabe den BASIC-Arbeitsspeicher so erkennt man, daß das Programm ab der Adresse 606FH im ASCII-Code im Speicher abgelegt ist. Die einzelnen BASIC-Programmzeilen beginnen allerdings mit der jeweiligen Zeilennummer im Hexadezimal-Code. Die Zeilennummer "10" (hexadezimal 000A) der ersten Programmzeile ist in den Speicherstellen 606FH und 6070H abgelegt. Im folgenden ist der Ausdruck des BASIC-Arbeitsspeichers von 6060H bis 609FH sowohl im Hexadezimal- als auch im ASCII-Code abgebildet. Das letzte Zeichen im Programmtext ist ein CR-Steuerzeichen (ODH).

Funktionsbeschreibung

```
KMD > PRINT
START-ADR =0000 6060
STOP -ADR =0000 609F
FORMAT   =H      +----->+ Zeiger auf die nächste
                        !     ! freie Speicherstelle im
6060 00 00 00 00 92 60 4E DF      ! Arbeitsspeicher
6068 00 00 AF DF 00 E0 0D 0A      !
6070 00 46 4F 52 20 49 3D 31      !
6078 20 54 4F 20 31 30 0D 14      !
6080 00 50 52 49 4E 54 20 49      !
6088 0D 1E 00 4E 45 58 54 20      !
6090 49 0D>00 00 00 00 00 00      !
6098 00 00!00 00 00 00 00 00      !
                        !
                        +<-----+
```

```
KMD > PRINT
START-ADR =6060
STOP -ADR =609F
FORMAT   =H A

6060 00 00 00 00 92 .' .N DF
6068 00 00 AF DF 00 E0 0D 0A
6070 00 .F .O .R . .I .= .1
6078 . .T .O . .1 .0 0D 14
6080 00 .P .R .I .N .T . .I
6088 0D 1E 00 .N .E .X .T .
6090 .I 0D 00 00 00 00 00 00
6098 00 00 00 00 00 00 00
```

In den Speicherstellen 6064H und 6065H verwaltet der BASIC-Interpreter eine Adresse, die auf die nächste freie Speicherstelle im Arbeitsspeicher zeigt. In diesem Beispiel ist dies die Adresse 6092H. Beachten Sie, daß Adressen mit dem Low-Byte beginnend im Speicher abgelegt werden. Diese Adresse ist im neuen Arbeitsspeicher mit der Startadresse 8000H zu korrigieren.

Wird dieses Programm über den Aufruf der Floppy-Funktionen aus dem Steuer-BASIC heraus auf eine Diskette gespeichert, so legt das MiniDOS für den Programmtext eine Datei auf der Diskette an, in der auch die Adressen des Textanfangs und des Textendes gespeichert werden. Beim Zurückladen eines BASIC-Programms von der Diskette wird insbesondere die Zeigeradresse unter 6064H und 6065H auf die nächste freie Speicherstelle im BASIC-Arbeitsspeicher korrigiert.

Soll nun ein BASIC-Programm, welches im Arbeitsspeicher ab 6000H erstellt und auf der Diskette gespeichert wurde, an den neuen Arbeitsspeicher angepaßt werden, so kann das Programm nicht über den Aufruf der FLOPPY-Funktionen aus dem BASIC-Interpreter geladen werden. Folgende Schritte sind auszuführen:

Funktionsbeschreibung

1. Schritt

Start des BASIC-Interpreters und sofortiges verlassen über QUIT.

```
KMD+> BASIC
BFZ-STEUER-BASIC V2.4
```

```
READY
>QUIT
```

2. Schritt

Aufruf der Floppy-Funktionen aus dem MAT85 nach Rückkehr aus dem BASIC und laden des BASIC-Programms von der Diskette durch Eingabe des Programmnamens ohne die Erweiterung ".BAS". Anstelle der Vorschlagsadresse 606FH für das Zurückladen des BASIC-Programms wird hier die Adresse 806FH eingegeben.

```
KMD> F
BFZ-MINI-DOS V1.4
```

```
...
MENUE:
  DIRECTORY
  ERASE
  ...
  ...
BITTE BUCHSTABE EINGEBEN: LOAD
```

```
...
NAME: TEST
  START-ADRESSE: 606F 806F <----- neue Zieladresse für den
  ...          ---- BASIC-Arbeitsspeicher ab
  ...          8000H
```

```
BITTE BUCHSTABE EINGEBEN: QUIT
```

3. Schritt

Mittels des PRINT-Kommandos kann der Programmtext im BASIC-Arbeitsspeicher kontrolliert werden. Dabei erkennt man, daß der Zeiger für das Programmende auf die Speicherstelle 806FH zeigt. Das bedeutet für den BASIC-Interpreter, daß der Programmspeicher leer ist. Dies ist richtig, weil im ersten Schritt nach dem Aufruf des BASIC-Interpreters kein Programm eingegeben wurde. Diese Adresse muß mit Hilfe des MEMORY-Kommandos korrigiert werden. Dazu wird zunächst das Ende des Programmtextes ermittelt. Der folgende Speicherausdruck zeigt, daß die nächste freie Speicherstelle hinter dem Programmtext bei der Adresse 8092H zu finden ist.

Funktionsbeschreibung

```
KMD > PRINT
START-ADR =0000 8060
STOP -ADR =0000 809F
FORMAT   =H          +----->+
                        !
8060 00 00 00 00 00 6F 80 4E DF      !
8068 00 00 AF DF 00 E0 0D 0A <----+ zeigt auf den Anfang des
8070 00 46 4F 52 20 49 3D 31          Programmspeichers, das be-
8078 20 54 4F 20 31 30 0D 14          deutet, der Programmspei-
8080 00 50 52 49 4E 54 20 49          cher ist leer
8088 0D 1E 00 4E 45 58 54 20
8090 49 0D>00 00 00 00 00 00
8098 00 00!00 00 00 00 00 00
                        !
                        +<----- nächste freie Speicherstelle
                        hinter dem Programmtext ist 8092H
```

Bei langen BASIC-Programmen kann das Programmende mit Hilfe des FIND-Kommandos aus dem MAT85+ durch Aufsuchen einer Folge von 00-Bytes ab dem Speicherbereich 806FH gefunden werden.

4. Schritt

Die Adresse der ersten freien Speicherstelle hinter dem Programmtext muß unter den Adressen 8064H und 8065H mit dem Low-Byte beginnend abgelegt werden. Zur Sicherheit empfiehlt sich eine Kontrolle über das PRINT-Kommando.

```
KMD> MEMORY
START-ADR =0000 8064
FORMAT   =H
8064 6F 92      <----- Low-Byte
8065 80        <----- High-Byte (hier unverändert)
```

```
KMD > PRINT
START-ADR =8060
STOP -ADR =809F
FORMAT   =H          +----->+ korrigierter Zeiger auf
                        ! die nächste freie Spei-
8060 00 00 00 00 00 92 80 4E DF      ! cherstelle im Arbeits-
8068 00 00 AF DF 00 E0 0D 0A      ! speicher
8078 20 54 4F 20 31 30 0D 14      !
8080 00 50 52 49 4E 54 20 49      !
8088 0D 1E 00 4E 45 58 54 20      !
8090 49 0D>00 00 00 00 00 00      !
8098 00 00!00 00 00 00 00 00      !
                        !
                        +<----->+
```

Funktionsbeschreibung

5. Schritt

Wird nun der BASIC-Interpreter erneut aufgerufen, so meldet sich dieser mit einem RESTART. Nach Eingabe des LIST-Kommandos wird das BASIC-Programm auf dem Bildschirm ausgegeben und kann nun wieder zur Sicherung auf einer Diskette gespeichert werden.

```
KMD+> BASIC
BFZ-STEUER-BASIC V2.4 RESTART
```

```
READY
>LIST
10 FOR I=1 TO 10
20 PRINT I
30 NEXT I
```

```
READY
```

2.2.3 SPS-Interpreter

Im SPS-Interpreter wurden lediglich die Symbole für die UND- und ODER-Verknüpfung sowie für die Signalumkehr (NEGATION) an die DIN-Programmierung angepaßt. In der folgenden Tabelle sind die geänderten Zeichen zusammengefaßt:

! Operation	! altes Symbol	! neues Symbol (DIN)
! UND	! *	! U
! ODER	! +	! O
! NICHT	! /	! N

Die SPS-Erweiterungen zur Kontaktplanausgabe und zur Dokumentation von SPS-Programmen mit dem BFZ-Editor werden im Kapitel 2.4 beschrieben.

2.2.4 EPROM-Programmierer

In diesem Modul wurden keine Änderungen vorgenommen.

2.3 Mini-DOS

Das Mini-DOS zum Betrieb einer Floppy-Disk-Station wurde funktionsmäßig nicht verändert. Eine Anpassung erfolgte lediglich im Zusammenhang mit der Speicherung von BASIC-Programmen auf Diskette durch den geänderten BASIC-Arbeitsspeicher ab der Adresse 8000H.

Funktionsbeschreibung

2.4 SPS-Erweiterungen

In diesem 32-KByte-EPROM sind zwei SPS-Erweiterungen implementiert, die Kontaktplanausgabe und eine Erweiterung zur Erstellung und Dokumentation von SPS-Programmen mit dem BFZ-Editor.

2.4.1 Kontaktplanausgabe

Die Kontaktplanausgabe ist ebenso wie der SPS-Interpreter an die DIN-Programmierung angepaßt. Der Aufruf erfolgt aus dem SPS-Interpreter heraus durch Eingabe des Anfangsbuchstabens des Kommandonamens, hier "K". Dieser Kommandoname wird mit dem Aufruf der HELP-Funktion im SPS nicht aufgelistet. Die Kontaktplanausgabe erstellt aus der Anweisungsliste eines SPS-Programms einen sogenannten Kontaktplan, der in seiner Symbolik an die Darstellung von Schützsaltungen angenähert ist. Im wesentlichen werden folgende Symbole verwendet:

! FUNKTION	! SPS-ANWEISUNG	! KONTAKTPLANDARSTELLUNG
! Teste auf H-Signal	! z.B. UE00	! -I I- Schließer !*)
		! E00
! Teste auf L-Signal	! z.B. ONM03	! -I/I- Öffner !*)
		! M03
! Ausgang auf H	! z.B. =A12	! -()- Schütz
		! A12
! Setze Ausgang auf H	! z.B. =SA00	! -(S)- Schütz mit
		! A00 Selbsthaltung !

*) Anstelle der eckigen Klammern für die Kontakte wird in dieser Beschreibung das Zeichen "I" verwendet. Die eckigen Klammern werden sonst auf einem Drucker, der auf den deutschen ASCII-Zeichensatz eingestellt ist, als Umlaute gedruckt.

Alle anderen Möglichkeiten werden ähnlich dargestellt. Handelt es sich in der Anweisungsliste um UND-Verknüpfungen, so wird eine Reihenschaltung der Kontakte erzeugt. Jede ODER-Verknüpfung erzeugt dagegen einen neuen parallelen Strompfad.

2.4.2 SPS-Dokumentation

Die Software-Erweiterung zur Dokumentation von SPS-Programmen ermöglicht die Erstellung bzw. Kommentierung von SPS-Programmen in Verbindung mit dem BFZ-Editor (siehe Abschnitt 2.5). Dabei können einerseits mit dem SPS-Interpreter erstellte Programme aus dem SPS-Arbeitsspeicher in den Textpuffer des Editors übertragen und beispielsweise mit Kommentar versehen werden, damit die Lesbarkeit der Programme verbessert wird. Andererseits können kommentierte SPS-Programme direkt mit dem Editor erstellt

Funktionsbeschreibung

werden. Danach kann das SPS-Programm aus dem Textpuffer des Editors in den SPS-Arbeitsspeicher geladen werden, wobei alle Kommentare und Trennzeichen bei der Übertragung unterdrückt werden.

Die Übertragung von SPS-Anweisungen aus dem SPS-Arbeitsspeicher in den Textpuffer des Editors erfolgt über den "Programmtexter". Das Laden von SPS-Anweisungen aus dem Textpuffer des Editors in den SPS-Arbeitsspeicher wird von dem "Programmlader" vorgenommen.

```
+-----+
! SPS-Erweiterung !      Übertragungsrichtung      !
!-----!-----!
! Programmtexter ! SPS-Arbeitsspeicher --> Editor-Textpuffer !
!-----!-----!
! Programmlader  ! Editor-Textpuffer  --> SPS-Arbeitsspeicher !
+-----+-----+
```

2.4.2.1 Programmtexter

Der Programmtexter übernimmt die Übertragung von SPS-Programmen in den Textpuffer des Editors. Dabei fügt er zwischen den Operationssymbolen (z.B. U, ON, =) und den Operanden (z.B. E, A, M) Leerzeichen zur besseren Lesbarkeit ein. Ebenso trennt er den Operanden von der Operandenadresse. Darüber hinaus ergänzt er jede Anweisung um ein Semikolon ";", damit Kommentare angefügt werden können. Das bedeutet, daß Kommentare grundsätzlich mit einem Semikolon beginnen müssen. Die einzelnen Strompfade bzw. Anweisungsfolgen werden außerdem im Kommentar nummeriert. Damit soll das Auffinden von Anweisungen erleichtert werden.

Der Aufruf der Dokumentations-Funktionen kann vom SPS-Interpreter wie auch vom Editor aus erfolgen. Hier wird zunächst der Aufruf über den SPS-Interpreter beschrieben. Die Dokumentations-Funktionen werden im SPS durch Eingabe des Buchstabens "D" für "Dokumentation" gestartet. Danach wird der Bediener über ein Menü geführt. Im folgenden ist ein einfaches Beispiel für den Umgang mit dem Programmtexter abgebildet. Im Gegensatz zum Bildschirmausdruck wurden hier Leerzeilen oder Auflistungen von Kommandonamen entfernt, wodurch die Verdeutlichung der Arbeitsschritte nicht beeinflußt wird.

Funktionsbeschreibung

1. Schritt

Start des SPS-Interpreters und Eingabe eines neuen SPS-Programms. Die Kontrolle des eingegebenen SPS-Programms erfolgt mit der LIST-Funktion.

```
KMD+> SPS          <---- Aufruf des SPS-Interpreters
BFZ-SPS-PROGRAMM V2.3
EDIT
...
...
SPS> EDIT          <---- Programm-Eingabe über die
ANF: <ENDE>        UE00      EDIT-Funktion
      <ENDE>        UE01
      <ENDE>        =A00
      <ENDE>        UE02
      <ENDE>        OE03
      <ENDE>        =A01
      <ENDE>
```

```
SPS> LIST          <---- Kontrolle der Eingabe über
UE00               die LIST-Funktion
UE01
=A00
```

```
UE02
OE03
=A01
```

2. Schritt

Nach der Eingabe und dem Test des Programms wird dieses zur Dokumentation zum Editors übertragen und mit Kommentar versehen. Dazu werden die Dokumentation-Funktionen aufgerufen und mit dem Programmtexter das SPS-Programm in den Textpuffer übertragen.

```
SPS> DOKUMENTATION SPS-PROGRAMM V1.8  <---- Aufruf der Dokumen-
                                         tations-Funktionen
```

```
TEXTPUFFER ==> SPS = 1
SPS ==> TEXTPUFFER = 2
```

```
START EDITOR = E
START SPS    = S
```

```
>2          <---- Wahl der Übertragungsrichtung
```

```
SPS ==> TEXTPUFFER
TEXTPUFFER-START-ADR = 8000
OK
==> SPACE
```

```
<---- Fertigmeldung der Übertragung
<---- Fortsetzung beim Auswahl-Menü
      durch Betätigen der SPACE-Taste
```

Funktionsbeschreibung

3. Schritt

Nach der Übertragung des SPS-Programms in den Textpuffer des Editors wird dieser durch die Eingabe von "E" über einen Warmstart gestartet. Danach kann die weitere Textbearbeitung vorgenommen werden.

DOKUMENTATION SPS-PROGRAMM V1.8

TEXTPUFFER ==> SPS = 1
SPS ==> TEXTPUFFER = 2

START EDITOR = E
START SPS = S

>E <---- Aufruf des Editors

*** BFZ-EDITOR V1.6 ***
(C) 1989 BFZ, ESSEN

MAT85 = ESC
HILFSMENUE = X
KALTSTART = K
WARMSTART = <CR>, <SPACE>

> <SPACE> <---- Warmstart des Editors durch
Betätigen der SPACE-Taste

EDIT-BUFFER:
START-ADR =8000
STOP -ADR =8077
==> <CR>, <SPACE>

U E 00 ;< <---- in den Textpuffer übertragenes
U E 01 ;< SPS-Programm
= A 00 ;1<

<
U E 02 ;<
O E 03 ;<
= A 01 ;2<

<
END <---- Ende-Kennung

Die letzten Zeilen im Protokoll zeigen das formatierte SPS-Programm mit Leerzeichen und der Einleitung von Kommentaren durch ein Semikolon nach jeder Anweisung sowie der Nummerierung der zusammengehörenden Anweisungen bzw. Strompfade. Darüber hinaus wird am Programmende ein "END" angehängt, welches im Zusammenhang mit dem Programmlader noch weiter erklärt wird.

4. Schritt

Das Ergebnis einer Kommentierung dieses Programmbeispiels mit dem Editor ist nachfolgend abgebildet. Ist diese Arbeit abgeschlossen, so kann das kommentierte SPS-Programm für den späteren Wiedergebrauch auf Diskette gesichert werden.

Funktionsbeschreibung

```
;*****< <---- überarbeitetes SPS-
;*** A U F G A B E ***< Programm im Textpuffer
;*****<
<
;EIN PUMPENMOTOR FUER EINEN FLUESSIGKEITS-<
;BEHAELTER SOLL UEBER ZWEI SCHALTER EIN-<
;GESCHALTET WERDEN<
;DARUEBER HINAUS SIND ZWEI NIVEAU-SCHALTER<
;FUER EIN UNTERES UND EIN OBERES NIVEAU ZU<
;UEBERWACHEN UND BEI BETAETIGUNG IST EINE<
;SIGNALLAMPE EINZUSCHALTEN<
<
;*****<
;*** BELEGUNGS-LISTE ***<
;*****<
<
; E 00 - SCHALTER 1 PUMPE EIN - SCHLIESSER<
; E 01 - SCHALTER 2 PUMPE EIN - SCHLIESSER<
; E 02 - NIVEAU-SCHALTER 1 - SCHLIESSER<
; E 03 - NIVEAU-SCHALTER 2 - SCHLIESSER<
<
; A 00 - SCHUETZ PUMPE EIN<
; A 01 - SIGNALLAMPE NIVEAU<
<
;*****<
;*** ANWEISUNGS-LISTE ***<
;*****<
<
U E 00 ; WENN SCHALTER 1 BETAETIGT IST<
U E 01 ; UND SCHALTER 2 BETAETIGT IST<
= A 00 ;1 DANN PUMPE EINSCHALTEN<
<
U E 02 ; WENN NIVEAU 1 BETAETIGT IST<
O E 03 ; ODER NIVEAU 2 BETAETIGT IST<
= A 01 ;2 DANN SIGNALLAMPE EINSCHALTEN<
<
;*****<
;*** PROGRAMM ENDE ***<
;*****<
<
END<
```

Wenn die Dokumentationsschritte abgeschlossen sind und das kommentierte SPS-Programm auf Diskette gesichert wurde, kann der Editor verlassen und der SPS-Interpreter wieder aufgerufen werden.

5. Schritt

Der Aufruf des SPS-Interpreters aus dem Editor-Betrieb erfolgt durch gleichzeitiges Betätigen der Tasten CONTROL und "Y" (nachfolgend durch CNTL-Y dargestellt). Danach erscheint das bekannte Menü für die Dokumentations-Funktionen. Mit der Eingabe von "S" wird zum SPS-Interpreter verzweigt.

Funktionsbeschreibung

```

DOKUMENTATION SPS-PROGRAMM V1.8 <---- Aufruf der Dokumentations-
                                Funktionen durch Eingabe
                                von CNTL-Y im Editor-Betrieb
TEXTPUFFER ==> SPS = 1
SPS ==> TEXTPUFFER = 2

START EDITOR = E
START SPS    = S

>S <---- Rückkehr zum SPS-Interpreter
                                nach Abschluß der Dokumenta-
                                tionsarbeiten
BFZ-SPS-PROGRAMM V2.3 RESTART
  EDIT
  GO
  ...
  ...
SPS>

```

2.4.2.2 Programmlader

Kommentierte SPS-Programme können später wieder über den Editor in den Textpuffer geladen und zum SPS-Arbeitsspeicher übertragen werden. Die Übertragung wird von einem Ladeprogramm übernommen, welches Trennzeichen sowie eingefügte Kommentare aus dem SPS-Programm entfernt. Der Start des Programmladers erfolgt im Editor-Betrieb über den Aufruf der Dokumentations-Funktionen, indem CNTL-Y eingegeben wird.

```

DOKUMENTATION SPS-PROGRAMM V1.8 <---- Aufruf der Dokumentations-
                                Funktionen durch Eingabe
                                von CNTL-Y im Editor-Betrieb
TEXTPUFFER ==> SPS = 1
SPS ==> TEXTPUFFER = 2

START EDITOR = E
START SPS    = S

>1 <---- Wahl der Übertragungs-
                                Richtung
TEXTPUFFER ==> SPS
TEXTPUFFER-START-ADR = 8000

OK <---- Meldung, daß Übertragung
                                fehlerfrei war

==> SPACE <---- Rückkehr zum Auswahl-Menü

```

Funktionsbeschreibung

Während der Übertragung werden die SPS-Anweisungen auf korrekte Schreibweise überprüft. Die Übertragung wird abgebrochen, falls ein Fehler auftritt. Fehlerhafte Programmzeilen werden mit der Zeilennummer aufgelistet und die Fehlerstellen werden im Textpuffer des Editors mit einem "?" zum leichteren Auffinden gekennzeichnet. Der SPS-Arbeitsspeicher wird beim Auftreten eines Fehlers sofort wieder gelöscht. War das SPS-Programm korrekt, so meldet der Programmlader "OK" und das Dokumentations-Menü erscheint wieder nach Eingabe von SPACE. Danach kann aus dem Editor durch Eingabe von "S" der SPS-Interpreter gestartet werden.

```
DOKUMENTATION SPS-PROGRAMM V1.8
```

```
TEXTPUFFER ==> SPS = 1
SPS ==> TEXTPUFFER = 2
```

```
START EDITOR = E
START SPS    = S
```

```

>S <---- Aufruf des SPS-Interpreters
                                aus dem Editor-Betrieb
BFZ-SPS-PROGRAMM V2.3 RESTART
  EDIT
  GO
  ...
  ...
SPS>

```

Die Überprüfung des Programms im SPS-Arbeitsspeicher erfolgt mit dem LIST-Kommando. Sofern es sich um das zuvor erstellte Programm handelt, erscheinen die nachfolgenden SPS-Anweisungen.

```

SPS> LIST <---- Überprüfung des übertragenen
                                SPS-Programms

UE00
UE01
=A00

UE02
OE03
=A01

SPS>

```

2.4.2.3 Fehlerbehandlung

Der Programmlader bricht die Übertragung eines SPS-Programms vom Textpuffer des Editors zum SPS-Arbeitsspeicher ab, wenn ein Fehler im SPS-Programm vorliegt. Eine fehlerhafte Programmzeile wird mit der Zeilennummer aufgelistet. Im Textpuffer wird die Fehlerstelle mit einem Fragezeichen "?" überschrieben und kann so leicht gefunden werden. Darüber hinaus wird der SPS-Arbeitsspeicher sofort wieder gelöscht, so daß keine fehlerhaften SPS-Programme gestartet werden können.

Funktionsbeschreibung

In der Testphase eines mit dem Editor erstellten SPS-Programms kann auch eine abschnittsweise Übertragung des SPS-Programms erfolgen, weil die Übertragung stets nur bis zur END-Anweisung im Textpuffer erfolgt. Diese END-Anweisung kann somit Schritt für Schritt im Programm verschoben werden, um den zu übertragenden Programmbereich zu erweitern.

Da mit dem Texteditor auch Assembler-Programme erstellt werden, wird es vorkommen, daß sich ein Assembler-Programm im Textpuffer befindet. Um in diesem Fall eine Übertragung mit dem SPS-Programmlader zu verhindern, untersucht dieser zunächst den Inhalt des Textpuffers. Falls hier ein Text vorliegt, der am Zeilenanfang nicht mindestens einmal die Operationssymbole "U" und "=" für eine einfache SPS-Anweisungsfolge enthält, so erfolgt keine Übertragung.

Als Trennzeichen im Text sind neben den Leerzeichen auch Tabulator-Steuerzeichen (09H) erlaubt.

Die Operationssymbole UN, ON, =N, =NL dürfen grundsätzlich nicht getrennt werden.

Da der Texteditor den Punkt "." als Steuerzeichen für die Druckerausgabe verwendet, darf auch dieses Zeichen am Anfang einer Zeile stehen. Der Punkt bewirkt beim Ausdruck eines Textes aus dem Textpuffer, daß der Drucker einen Seitenvorschub ausführt. Darüber hinaus lassen sich mit den Punkt-Zeilen sogenannte Kopfzeilen einschließlich einer Seitennummerierung definieren, die auf jeder neuen Seite ausgedruckt werden.

Zusammenfassend können noch folgende Fehlermeldungen auftreten:

Fehlermeldung	Fehlerursache
PROGRAMM ?	kein SPS-Programm im Textpuffer
END ?	SPS-Programm im Textpuffer nicht mit END abgeschlossen
SPS-RAM-ÜBERLAUF	SPS-Programm für den SPS-Arbeitspeicher zu groß
TEXT-RAM ?	kein RAM unter der angegebenen Textpuffer-Adresse
SPS-RAM ?	kein RAM im SPS-Arbeitspeicherbereich
TEXT-RAM ? GLEICH	Der Text im Textpuffer überschreibt den Inhalt des SPS-
SPS-RAM ?	Arbeitspeichers (Anf.-Adr. E000H)

Funktionsbeschreibung

2.5 BFZ-Editor

Der Editor, der speziell für die Assembler-Programmierung vorgesehen ist und durch die Dokumentations-Funktionen auch für die SPS-Programmierung verwendet werden kann, ist dahingehend geändert, daß er in einem neuen Speicherbereich liegt. Damit wird der gleichzeitige Betrieb mit der SPS-Kontaktplanausgabe möglich. Als Anfangsadresse des Textpuffers schlägt der Editor die RAM-Startadresse 8000H vor. Im EPROM befindet sich auch der Text des Hilfsmenus, der durch Eingabe von CNTL-X auf dem Bildschirm ausgegeben wird.

3. Inbetriebnahme-Hinweise MAT32

Für die Inbetriebnahme der Betriebssoftware im 32-KByte-EPROM vom Typ 27256, bestücken Sie eine 64-KByte-RAM-Baugruppe entsprechend dem Bestückungsplan im Anhang A8. Beachten Sie, daß die erforderlichen Lötbrücken hergestellt werden.

Entgegen der Betriebsweise der 64-KByte-RAM-Baugruppe im CP/M-System, werden die Lötbrücken so hergestellt, daß das 32-KByte-EPROM stets aktiv bleibt, d.h. nicht ausgeblendet wird. Damit soll ein Überschreiben des Speicherbereichs, in dem sich die Betriebssoftware befindet, bei einer Fehlbedienung des Systems verhindert werden. Daher braucht die 64-KByte-RAM-Baugruppe nur mit vier 8-KByte-RAM-Bausteinen bestückt werden.

Da mit der 64-KByte-Speicherbaugruppe der maximale Systemspeicher zur Verfügung steht, darf sich keine weitere Speicherbaugruppe im Baugruppenträger befinden.

Nach dem Einschalten der Betriebsspannung muß sich die Betriebssoftware mit dem Auflisten der MAT85-Kommandos melden. Sollten hier Schwierigkeiten auftreten, so überprüfen Sie die Funktion der Speicherbaugruppe mit Hilfe des BUS-Signalgebers und der BUS-Signalanzeige unter Zuhilfenahme der entsprechenden Fachpraktischen Übungen.

Die Funktion der Baugruppe läßt sich schnell überprüfen, indem die beiden Speicherbereiche von 0000H bis 7FFFH bzw. von 8000H bis FFFFH getestet werden. Im ersten Bereich befindet sich das 32-KByte-EPROM. Dieser Bereich kann lediglich gelesen werden. Ein Überschreiben darf nicht möglich sein. Der zweite Speicherbereich ist der RAM-Bereich. Hier müssen sich die Speicherinhalte verändern lassen. Überprüfen Sie insbesondere nach einem Schreib-/Leseversuch im RAM oberhalb der Adresse C000H erneut den EPROM-Bereich, ob dieser noch aktiv ist. Sollte sich in diesem Bereich der Speicherinhalt überschreiben lassen, so ist die EPROM-Ab-schaltung z.B. durch falsch hergestellte Lötbrücken nicht deaktiviert worden.

Anhang A1: MAT85- und MAT85+ - Kommandos

Nach dem ersten Einschalten des Mikrocomputers ist das Betriebsprogramm MAT85 aktiv. Die erweiterten Funktionen unter MAT85+ können durch Eingabe der SPACE-Taste aufgerufen und auch wieder verlassen werden. Der Kommando-Aufruf erfolgt durch Eingabe des ersten Buchstabens des Kommandonamens gefolgt von der RETURN-Taste (CR). Die Auflistung aller verfügbaren Funktionen erhält man durch Aufruf des HELP-Kommandos.

KMD> HELP

```
ASSEMBLER
BREAKPOINT
DISASSEMBLER
GO
HELP
IN
LOAD TAPE
MEMORY
NEXT INSTRUCTION
OUT
PRINT
REGISTER
SAVE
TRACE INTERVAL
```

KMD> <SPACE-Taste> <---- Umschaltung nach MAT85+ durch Betätigen der SPACE-Taste

KMD+> HELP

```
BASIC
COPY
FIND
HELP
INSERT
PROMMER
RAM-TEST
SPS
VERIFY
WRITE CONSTANT
```

KMD+> <SPACE-Taste> <---- Rückkehr nach MAT85 durch erneutes Betätigen der SPACE-Taste

KMD>

Anhang A2: SPS - Kommandos

Der SPS-Interpreter wird aus dem MAT85+ gestartet. Die SPS-Erweiterungen für die Kontaktplanausgabe und die Dokumentation werden durch Eingabe des Buchstabens "K" bzw. "D" aufgerufen. Diese Funktionen werden in der HELP-Auflistung nicht mit angezeigt.

```
KMD+> SPS <---- Start des SPS-Interpreters
BFZ-SPS-PROGRAMM V2.3 im MAT85+ oder durch Eingabe
                     von CNTL-Y im Editor-Betrieb
EDIT
GO
HELP
LIST
NEW
READ
STEP
TRACE
WRITE
QUIT
<KONTAKTPLAN-AUSGABE> <---- diese Kommando-Namen werden
<DOKUMENTATION> <---- nicht angezeigt
```

Die folgenden Aufstellungen geben einen Überblick über die verfügbaren Verknüpfungsanweisungen (Operationen), die möglichen Operanden, sowie die erlaubten Operanden-Adressen.

OPERATION	SYMBOL
UND	U
UND NICHT	UN
ODER	O
ODER NICHT	ON
GLEICH	=
GLEICH NICHT	=N
SETZEN	=S
NICHT SETZEN	=NS
RÜCKSETZEN	=R
NICHT RÜCKSETZEN	=NR
LADEN	=L
NICHT LADEN	=NL

OPERAND	SYMBOL
EINGANG	E
AUSGANG	A
MERKER	M
TIMER (Hardware)	T
ZEITWERK (Software)	Z
COUNTER	C
erlaubte Operanden- adressen	00...07 10...17 20...27 30...37

Anhang A3: Steuer - BASIC - Befehle

Der BASIC-Interpreter wird aus dem MAT85+ gestartet. Im folgenden sind die verfügbaren BASIC-Anweisungen in einer Kurzschreibweise zusammengefaßt.

```
KMD+> BASIC          <---- Start des BASIC-Interpreters
BFZ-STEUER-BASIC V2.4      im MAT85+
```

```
READY
>
```

EDITIER-ANWEISUNGEN:

```
-----
Zeichen löschen: DEL oder BS oder -
Zeile eingeben: <zeilen-nummer> <anweisung> : <anweisung> :...CR
Zeile löschen: <zeilen-nummer> CR
Zeile einfügen: <zwischen-zeilen-nummer>..<anweisungen>...CR
Drucker ein/aus: Ctrl-P
```

SYSTEM-KOMMANDOS:

```
-----
Programm starten:      RUN
Programm listen:      LIST
                       LIST <zeilen-nummer>
Programm löschen:     NEW
Programm von Cassette laden: LOAD
Programm auf Cassette sichern: SAVE
Disketten-Funktionen aufrufen: FLOPPY
Rückkehr zum MAT85+:  QUIT
```

PROGRAMMTTEST-KOMMANDOS/ANWEISUNGEN:

```
-----
Einzelschritt ein:    STON
Einzelschritt aus:   STOFF
Tracelauf ein:       TRON
Tracelauf aus:       TROFF
```

KONSTANTE:

```
-----
dezimal:              -32768.....32767
hexadezimal:          DEC(0000).....DEC(FFFF)
```

```
VARIABLE:             A,B,C,D,E,.....,X,Y,Z
-----
```

```
FELD-VARIABLE:       @(0), @(1), @(2), ..., @(n)
-----
```

```
WERTZUWEISUNGEN:    LET <variable> = <ausdruck>
-----
                       <variable> = <ausdruck>
LET @( <ausdruck> ) = <ausdruck>
  @( <ausdruck> ) = <ausdruck>
```

Anhang A3: BASIC - Kommandos

ARITHMETISCHE OPERATIONEN:

```
-----
Addition:             +
Subtraktion:          -
Multiplikation:       *
Division:             /
Klammer auf           (
Klammer zu            )
```

VERGLEICHOPERATIONEN:

```
-----
gleich:               =
größer:               >
kleiner:              <
ungleich:             <>
kleiner gleich:      <=
größer gleich:       >=
logisch UND:         AND
logisch ODER:        OR
logisch NICHT:      NOT
```

BILDSCHIRM-AUSGABE:

```
-----
Bildschirm löschen:  CLS
```

```
Textausgabe:         PRINT "<text>"
Konstanten:          PRINT <konstante>;<konstante>...
Variablen:           PRINT <variable>;<variable>.....
Ausdrücke:           PRINT <ausdruck>;<ausdruck>.....
```

Wirkung der Steuerzeichen:

```
Kein Steuerzeichen: PRINT ... in die nächste Zeile gehen
Semikolon:          PRINT ...; in der Position stehenbleiben
Komma:              PRINT ..., in die nächste Zone gehen
```

TASTATUR-EINGABE:

```
-----
Dezimalzahl:        INPUT <variable>
Hexadezimal-Zahl:   INPUT #<variable>
Variablen-Liste:    INPUT <variable>,<variable>,...
Melde-Text:         INPUT "<melde-text>",<variable>,...
Einzelzeichen-
Eingabe:            <variable> = $
```

DRUCKER-STEUERUNG:

```
-----
Drucker ein:        LPON
Drucker aus:        LPOFF
```

Anhang A3: BASIC - Kommandos

PORT-EIN-/AUSGABE:

```

-----
Port-Eingabe:      <variable> = INP( <port-nummer> )
Port-Ausgabe:     OUT <port-nummer>, <ausgabe-wert>
Warteschleife:    WAIT <port-nummer>, <maske>, <vergleichswert>

```

SPEICHER-ZUGRIFF:

```

-----
Speicher lesen:   <variable> = PEEK( <adresse> )
Speicher schreiben: POKE <adresse>, <speicher-wert>

```

PROGRAMMVERZWEIGUNGEN:

```

-----
absolut:          GOTO <zeilen-nummer>
bedingt:          IF <bedingung> THEN GOTO <zeilen-nummer>
                                   GOSUB <zeilen-nummer>

```

UNTERPROGRAMME:

```

-----
Aufruf:           GOSUB <zeilen-nummer>
Rückkehr:         RETURN

```

PROGRAMMSCHLEIFEN:

```

-----
FOR <variable>=<start-wert> TO <stop-wert> STEP <inkrement>
...
<anweisung>
<anweisung>
...
NEXT <variable>

```

FUNKTIONEN:

```

-----
Betrag:           <variable> = ABS( <ausdruck> )
Zufallszahl:     <variable> = RND( <grenz-wert> )
Hexadezimal-/
Dezimal-Wandlung: <variable> = DEC( <hexadezimal-zahl> )
Freier Speicher-
bereich:         <variable> = FREE

```

DATEN-/KONSTANTEN-MENGEN

```

-----
Datenzeile:      DATA <konstante>, <konstante>, ...
Daten lesen:     READ <variable>, <variable>, ...
Datenzeiger
positionieren:   RESTORE <zeilen-nummer>
                 RESTORE <variable mit zeilen-nummer>

```

Anhang A3: BASIC - Kommandos

UNTERPROGRAMM IN MASCHINENSPRACHE:

```

-----
<variable> = USR( <adresse>, <übergabe-wert> ) <ausdruck>
-----
!!
liefert <rückgabe-wert>

```

```

vor Aufruf:      <übergabe-wert> im HL-Registerpaar
nach Rückkehr:   <rückgabe-wert> verknüpft mit <ausdruck>
                 in <variable>

```

SONSTIGE ANWEISUNGEN:

```

-----
Programm-Ende:   END
Programm-Stop:   STOP
Kommentar:       REM <kommentar-text>

```

Anhang A4: Mini-DOS-Kommandos

Die Disketten-Funktionen des MiniDOS können aus dem MAT85, dem Steuer-BASIC, dem SPS-Interpreter sowie aus dem Editor heraus aufgerufen werden. Davon abhängig werden den Datei-Namen des Benutzers Erweiterungen angehängt, die im Directory anzeigen, über welche Programmoption die Sicherung der Dateien erfolgte. Das MiniDOS-Menü wird mit der Eingabe von Q(UIT) verlassen.

Start des MiniDOS aus Programmteil	Eingabe
MAT85	KMD> F
Steuer-BASIC	>FLOPPY
SPS	SPS> F
Editor	CNTL-S

KMD > F <---- Start des Mini-DOS im MAT85

BFZ-MINI-DOS V1.4

(C) 1985 BY BFZ, ESSEN, W. GERMANY

- MENUE:
- DIRECTORY
 - ERASE
 - FORMAT
 - LOAD
 - SAVE
 - QUIT

BITTE BUCHSTABE EINGEBEN:

Anhang A5: EDITOR - Kommandos

Der Editor wird mit der Eingabe von "E" im MAT85-Betrieb gestartet. Darüber hinaus kann er über die Dokumentations-Funktionen im SPS-Betrieb aufgerufen werden.

KMD> E <---- Start des Editors im MAT85
 *** BFZ-EDITOR V1.6 ***
 (C) 1989 BFZ, ESSEN

MAT85 = ESC
 HILFSMENUE = X
 KALTSTART = K
 WARMSTART = <CR>, <SPACE>

Die für die Textverarbeitung zur Verfügung stehenden Editier-Kommandos sind im folgenden Hilf-Menü zusammengefaßt. Dieses Hilfs-Menü kann während der Textbearbeitung stets wieder aufgerufen werden, wenn eine bestimmte Funktion und deren Aufruf gesucht wird. In der Darstellung bedeutet die Zeichenkombination "^" gefolgt von einem Buchstaben, daß gleichzeitig mit der Buchstaben-Taste die CONTROL-Taste zu betätigen ist.

 *** HILFS-MENUE ***

----- VERARBEITUNG -----

- ^X = HILFS-MENUE
- ^N = NEUSTART
- ^S = SPEICHERN/LADEN
- ^D = DRUCKER-MENUE
- ^A = ASSEMBLIEREN
- ^Y = SPS-PROGRAMM-LADER/TEXTER

---- ZEICHEN EDITIEREN --- --- ZEILEN EDITIEREN ---

- ^R = ZEICHEN RADIEREN
- DEL = ZEICHEN LINKS LOESCHEN
- ^E = EINFUEGEN EIN/AUS
- ^ZL = ZEILE LOESCHEN
- ^ZE = ZEILE EINFUEGEN
- ^ZA = ZEILE ANHAENGEN

---- CURSOR-STEUERUNG --- --- SEITEN BLAETTERN ---

- PFEILE = CURSOR-STEUERUNG
- ^CO = AN OBEREN BILDRAND
- ^CU = AN UNTEREN BILDRAND
- ^CL = AN LINKEN BILDRAND
- ^CR = AN RECHTEN ZEILENRAND
- ^W = AUF NAECHSTES WORT
- ^U = SEITE UNTEN
- ^O = SEITE OBEN
- ^CA = CURSOR ZUM TEXTANFANG
- ^CE = CURSOR ZUM TEXTENDE
- ^T = TABULATOR

---- BLOCK-BEFEHLE ---- --- FINDE / ERSETZE ---

- ^BM = BLOCK MARKIEREN
- ^BR = B.-MARK. RUECKSETZEN
- ^BF = B.-MARK. FINDEN
- ^FS = SETZE SUCHWORT
- ^FF = FINDE SUCHWORT
- ^FE = ERSETZE SUCHWORT

- ^BL = BLOCK LOESCHEN
- ^BK = BLOCK KOPIEREN
- ^BV = BLOCK VERSCHIEBEN
- ^BS = BLOCK SPEICHERN (FLOPPY/CASSETTE)

Anhang A6: SPS - Programmbeispiel "Anweisungsliste"

```

;-----;
; B F Z   E S S E N ! AUTOMATISIEREN MIT MFA-SPS ;
;                               ! MIKROCOMPUTER   ;
;-----;
; BEISPIEL:           ! DATUM : 14.04.1989      ;
;                               ! AEND. :       ;
; WENDESCHUETZ-      ! BEARB.: N.N.           ;
; STEUERUNG          !                          ;
;-----;

```

```

;*****
;***   A U F G A B E   ***
;*****

```

```

;EIN DREHSTROMMOTOR, DER IN EINEM HYDRAULIK-
;SYSTEM EINGEBUNDEN IST, WIRD UEBER DREI
;HANDTASTER IM TIPP-BETRIEB IN DER DREH-
;RICHTUNG GESTEUERT.
;DREHRICHTUNGSUMKEHR IST NUR AUS DEM STILL-
;STAND ERLAUBT.
;DER MOTOR DARF NICHT EINGESCHALTET WERDEN,
;WENN DIE TEMPERATUR DES KUEHLSYSTEMS BZW.
;WENN DER GRENZDRUCK DES HYDRAULIK-SYSTEMS
;ERREICHT IST.

```

```

;*****
;***   BELEGUNGS-LISTE   ***
;*****

```

```

; E 00 - HANDTASTER RECHTSLAUF - SCHLIESSER
; E 01 - HANDTASTER LINKSLAUF - SCHLIESSER
; E 02 - HANDTASTER AUS      - OEFFNER
; E 03 - TEMPERATUR-FUEHLER  - OEFFNER
; E 04 - DRUCK-SCHALTER      - OEFFNER

; A 00 - SCHUETZ RECHTSLAUF
; A 01 - SCHUETZ LINKSLAUF

```

Anhang A6: SPS - Programmbeispiel "Anweisungsliste"

```

;*****
;***   PROGRAMM - ANFANG   ***
;*****

```

```

;*** AUSSCHALTBEDINGUNGEN ***

```

```

UN E 02 ; WENN HANDTASTER AUS BETAETIGT
ON E 03 ; ODER TEMPERATUR ZU GROSS
ON E 04 ; ODER GRENZDRUCK ERREICHT
= M 00 ; DANN EINSCHALTEN VERHINDERN

```

```

;*** RECHTSLAUF EIN ***

```

```

U E 00 ; WENN RECHTSLAUF BETAETIGT
UN M 00 ; UND KEINE AUSSCHALT-BEDINGUNG
UN A 01 ; UND NICHT IM LINKSLAUF
=S A 00 ; DANN MOTOR EIN

```

```

;*** RECHTSLAUF AUS ***

```

```

U M 00 ; WENN AUSSCHALT-BEDINGUNG
=R A 00 ; DANN MOTOR AUS

```

```

;*** LINKSLAUF EIN ***

```

```

U E 01 ; WENN LINKSLAUF BETAETIGT
UN M 00 ; UND KEINE AUSCHALT-BEDINGUNG
UN A 00 ; UND NICHT IM RECHTSLAUF
=S A 01 ; DANN MOTOR EIN

```

```

;*** LINKSLAUF AUS ***

```

```

U M 00 ; WENN AUSSCHALT-BEDINGUNG
=R A 01 ; DANN MOTOR AUS

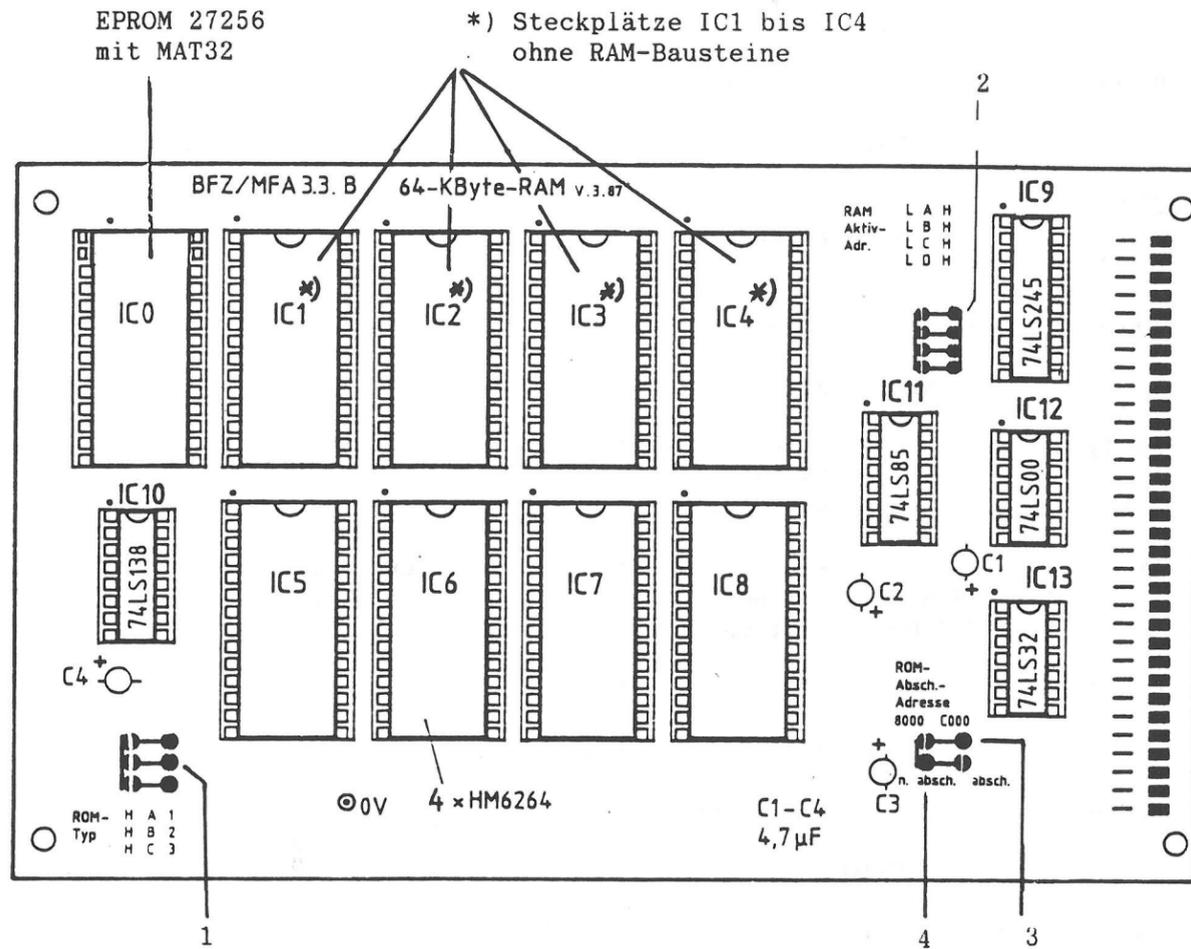
```

```

;*****
;***   PROGRAMM ENDE   ***
;*****

```


Anhang A8: Bestückungsplan 64-KByte-RAM



Hinweis:

Für den Betrieb des 32-KByte-EPROMs sind die in der nachfolgenden Tabelle aufgeführten Lötbrücken erforderlich.

Pos.	Funktion	geschlossene Brücken
1	ROM-Typ 27256	A - 1 B - 2 C - 3
2	RAM-Aktiv-Bereich ab 8000H	A - H B - H C - H D - H
3	ROM-Abschalt-Adresse (unbedeutend)	C000 oder (8000)
4	ROM nicht abschaltbar oder abschaltbar	n.absch.

Inhaltsverzeichnis

	Seite
1. Einleitung	1
1.1. Aufgaben und Funktionsmerkmale des BFZ-EDITORS	1
1.2. Installation und Inbetriebnahme	3
2. Überblick über die Steuerbefehle des EDITORS	7
3. Aufruf des BFZ-EDITORS	8
4. Beschreibung der Steuerbefehle des BFZ-EDITORS	11
4.1. Cursor-Steuerung mit Hilfe der "Pfeil"-Tasten und <CR>	11
4.2. Neustart: CTRL-N	12
4.3. Aufruf des Hilfsmenues: CTRL-X (Übung 1, Übung 2)	12
4.4. Zeichen radieren, Zeichen links löschen	16
4.5. Einfüge-Modus Ein-/Aus	17
4.6. Zeile löschen, Zeile einfügen, Zeile anhängen, (Übung 3)	17
4.7. Erweiterte Cursor-Bewegungen	20
4.8. Tabulator	20
4.9. Seiten blättern	23
4.10. Finden eines "Text-Strings", Ersetzen eines "Text-Strings"	24
4.11. Block-Befehle	26
4.11.1. Block markieren, Block-Markierungen rücksetzen, Block-Markierungen finden	26
4.11.2. Block löschen, Block kopieren, Block verschieben	27
4.12. Speichern und Laden mit Hilfe von Diskette oder Cassette	31
4.12.1. Speichern auf Diskette oder Cassette	31
4.12.2. Laden einer Textdatei von Diskette oder Cassette	34
4.13. Drucken des EDIT-Buffer-Inhalts	36
4.13.1. Drucker-Menue	36
4.13.2. Steuerung des Formularvorschubs mit ".", Ausdruck von Kopfzeilen	38
4.14. Assemblieren	40

Anhang A: Lösung der Übung 1

Anhang B: Anpassung des BFZ-EDITORS an andere Terminals

Anhang C: Aufbau des Hilfs-Menues

1. Einleitung

1.1. Aufgaben und Funktionsmerkmale des BFZ-EDITORS

Die Aufgabe eines EDITOR-Programms (EDITOR = Redakteur, Schriftleiter) besteht darin, mit Hilfe eines Computers Texte zu erstellen, zu bearbeiten und zu verarbeiten. Bei diesen Texten kann es sich z.B. um Briefe, Beschreibungen, Listen oder ähnliches handeln. Von besonderer Bedeutung für die Computertechnik sind Programm-Texte in Assemblersprache (der sogenannte "Source- oder Quellcode"). Lassen sich diese Programm-Texte mit Hilfe eines EDITORS bearbeiten, auf Diskette oder Cassette speichern und anschließend assemblieren, so sind spätere Änderungen oder Ergänzungen eines Programms leicht und komfortabel möglich. Aus diesem Grund eignet sich der BFZ-EDITOR besonders dazu, Assembler-Quell-Programme komfortabel zu erstellen, zu bearbeiten und anschließend durch den Assemblier-Vorgang in Maschinensprache umzusetzen (8085-Assembler-Code).

Die Arbeitsweise eines EDITOR-Programms besteht darin, die vom Bediener eingegebenen Textzeichen auf dem Bildschirm anzuzeigen und zusätzlich im Arbeitsspeicher des Computers zu speichern und zu verwalten. Der Speicherbereich, in dem die Textzeichen abgelegt werden, ist der sogenannte "EDIT-Buffer" (Buffer = Puffer, Zwischenspeicher). Die Bearbeitung des Textes mit Hilfe des BFZ-EDITORS erfolgt "Bildschirm-orientiert", d.h., auf dem Bildschirm wird das genaue Abbild des EDIT-Buffer-Inhalts zeilenweise dargestellt. Jede Veränderung von Textzeichen innerhalb des EDIT-Buffers wird direkt auf dem Bildschirm angezeigt. Dazu gehören z.B.:

- Positionierung des "Cursors" auf ein bestimmtes Textzeichen
- Positionierung des "Cursors" an den oberen, unteren, linken oder rechten Textrand, sowie auf das nächste "Wort"
- Löschen, Einfügen, Trennen oder Anhängen von Textzeilen
- "Blättern" auf die nächste oder vorherige Textseite, sowie an den Textanfang oder an das Textende
- Markieren, Kopieren, Verschieben oder Löschen von "Textblöcken"
- Suchen und Ersetzen bestimmter Zeichenketten ("Strings")

Weitere Funktionsmerkmale des BFZ-EDITORS sind:

- Bediener-freundliches Speichern und Laden des EDIT-Buffer-Inhalts mit Hilfe von Diskette oder Cassette
- Speichern und Laden von markierten Textblöcken
- Anbinden von gespeicherten Texten an den aktuellen EDIT-Buffer-Inhalt

- Drucken des bearbeiteten Textes mit Voreinstellung des linken Zeilenrandes, der maximalen Zeilenzahl pro Druckseite, sowie Steuerung des Formular-Vorschubs und wahlweisem Ausdruck einer Kopfzeile mit automatischer Seitennumerierung
- Assemblieren von 8085-Assembler-Programmen, wahlweise mit oder ohne Bildschirm-Ausdruck, bzw. Drucker-Listing
- Möglichkeit der Weiterverarbeitung des erstellten Textes mit Hilfe eines anderen Textverarbeitungs-Programms (z.B. "WORD-STAR")

Hinweis:

Das MFA-Video-Interface stellt alle verwendeten Buchstaben in Großschreibung dar. Im EDIT-Buffer werden aber alle Buchstaben so abgelegt, wie sie auch tatsächlich mit der Tastatur eingegeben wurden (z.B. Tastatur-Eingabe "a" ==> "a" wird als ASCII-Zeichen (= 61 hex) im EDIT-Buffer abgelegt ==> auf dem Bildschirm erscheint "A"). Beim Ausdrucken des EDIT-Buffer-Inhaltes werden aber alle als Kleinbuchstaben eingegebenen Zeichen auch als Kleinbuchstaben gedruckt, so daß der gedruckte Text u.U. bzgl. der Groß- und Kleinschreibung nicht mit dem auf dem Bildschirm dargestellten Text übereinstimmt. Soll auch die Ausgabe auf dem Drucker nur in Großschreibung erfolgen, so muß bei der Texteingabe die Taste

"ALPHA-LOCK"

aktiviert werden (Ausnahme: der Ausdruck von Programm-Listings beim Assemblieren erfolgt immer in Großschreibung, siehe 4.14).

1.2. Installation und Inbetriebnahme

Der BFZ-EDITOR ist ein Maschinenprogramm mit einer Länge von 4 KByte. Es ist in zwei 2-KByte-EPROMs vom Typ 2716 gespeichert und belegt den Adreßraum von Adresse 5000 bis 5FFF (hex). Das sogenannte "Hilfs-Menue" mit einer Länge von 1440 Byte (entspr. 05A0 hex) ist in einem weiteren 2-KByte-EPROM 2716 gespeichert. Es besteht im wesentlichen aus einem Text, der die Steuerzeichen des BFZ-EDITORS beschreibt, und kann wahlweise mitbenutzt werden, um das Arbeiten mit dem BFZ-EDITOR gerade in den Anfangsphasen zu erleichtern.

Je nach vorliegendem Software-Ausbau des vorhandenen Systems ergeben sich unterschiedliche Installationsmöglichkeiten für das EDITOR-Programm in Verbindung mit dem "Hilfs-Menue":

a) Vorhandener Software-Ausbau: MAT-85

Adresse (hex)	Software
0000 - 07FF	MAT-85
0800 - 0FFF	MAT-85
1000 - 17FF	MAT-85
1800 - 1FFF	MAT-85
2000 - 27FF	
2800 - 2FFF	
3000 - 37FF	
3800 - 3FFF	
4000 - 47FF	EDITOR-Hilfs-Menue (wahlweise)
4800 - 4FFF	
5000 - 57FF	BFZ-EDITOR (1)
5800 - 5FFF	BFZ-EDITOR (2)

Aufruf des BFZ-EDITORS von der Kommando-Ebene aus:

KMD> G(o)	G	<CR>	oder <SPACE>	eintippen
START-ADR = 5000	5000	<CR>	oder <SPACE>	eintippen

Hinweis: <CR> = Carriage-Return-Taste
<SPACE> = Leer-Taste

b) Vorhandener Software-Ausbau: MAT-85 und Software-Paket SP1

Adresse (hex)	Software
0000 - 07FF	MAT-85
0800 - 0FFF	MAT-85
1000 - 17FF	MAT-85
1800 - 1FFF	MAT-85
2000 - 27FF	SP1
2800 - 2FFF	SP1
3000 - 37FF	SP1
3800 - 3FFF	SP1
4000 - 47FF	EDITOR-Hilfs-Menue (wahlweise)
4800 - 4FFF	
5000 - 57FF	BFZ-EDITOR (1)
5800 - 5FFF	BFZ-EDITOR (2)

Aufruf des BFZ-EDITORS von der Kommando-Ebene aus:

mit EDITOR-Hilfs-Menue:

KMD> E E <CR> oder <SPACE> eintippen

ohne EDITOR-Hilfs-Menue:

KMD> G(o) G <CR> oder <SPACE> eintippen
START-ADR = 5000 5000 <CR> oder <SPACE> eintippen

c) Vorhandener Software-Ausbau: MAT-85, Software-Paket SP1 und BFZ-MINI-DOS

Adresse (hex)	Software
0000 - 07FF	MAT-85
0800 - 0FFF	MAT-85
1000 - 17FF	MAT-85
1800 - 1FFF	MAT-85
2000 - 27FF	SP1
2800 - 2FFF	SP1
3000 - 37FF	SP1
3800 - 3FFF	SP1
4000 - 47FF	BFZ-MINI-DOS
4800 - 4FFF	BFZ-MINI-DOS
5000 - 57FF	BFZ-EDITOR (1)
5800 - 5FFF	BFZ-EDITOR (2)

Aufruf des BFZ-EDITORS von der Kommando-Ebene aus:

KMD> E E <CR> oder <SPACE> eintippen

oder

KMD> G(o) G <CR> oder <SPACE> eintippen
START-ADR = 5000 5000 <CR> oder <SPACE> eintippen

Soll in dieser Konfiguration das EDITOR-Hilfs-Menue mitbenutzt werden, so muß der Inhalt des EPROMs "HELP.MEN" auf eine Diskette gespeichert werden. Beim Arbeiten mit dem BFZ-EDITOR läßt sich das Hilfs-Menue dann an jede beliebige freie Stelle in den Arbeitsspeicher laden und steht dort dem BFZ-EDITOR zur Verfügung.

Anleitung zum Kopieren des EDITOR-Hilfs-Menues auf Diskette:

1. Möglichkeit:

Laden Sie den Inhalt des EPROMs "HELP.MEN" mit Hilfe der EPROM-Programmier-Baugruppe in den Arbeitsspeicher (z.B. E000 - E7FF). Speichern Sie den notwendigen Inhalt des EPROMs mit Hilfe des BFZ-MINI-DCS auf die Diskette:

SAVE

START-ADR = E000
 STOP -ADR = E59F

NAME: HELP.MEN

2. Möglichkeit:

Stecken Sie das EPROM "HELP.MEN" in einen freien EPROM-Steckplatz (z.B. 5000 - 57FF). Speichern Sie den notwendigen Inhalt des EPROMs mit Hilfe des BFZ-MINI-DOS auf die Diskette:

SAVE

START-ADR = 5000
STOP -ADR = 559F

NAME: HELP.MEN

2. Überblick über die Steuerbefehle des EDITORS

Die nachfolgende Liste stellt alle Steuerbefehle des BFZ-EDITORS dar. Die Wirkung dieser Steuerbefehle wird dem Bediener im nachfolgenden erläutert.

```

*****
***  HILFS-MENUE  ***
*****

----- VERARBEITUNG -----

^X = HILFS-MENUE           ^S = SPEICHERN/LADEN
^N = NEUSTART              ^D = DRUCKER-MENUE
                           ^A = ASSEMBLIEREN

---- ZEICHEN EDITIEREN ----      --- ZEILEN EDITIEREN ---

^R = ZEICHEN RADIEREN          ^ZL = ZEILE LOESCHEN
DEL = ZEICHEN LINKS LOESCHEN  ^ZE = ZEILE EINFUEGEN
^E = EINFUEGEN EIN/AUS       ^ZA = ZEILE ANHAENGEN

---- CURSOR-STEUERUNG ----      --- SEITEN BLAETTERN ---

PFEILE = CURSOR-STEUERUNG      ^U = SEITE UNTEN
^CO = AN OBEREN BILDRAND       ^O = SEITE OBEN
^CU = AN UNTEREN BILDRAND     ^CA = CURSOR ZUM TEXTANFANG
^CL = AN LINKEN BILDRAND      ^CE = CURSOR ZUM TEXTENDE
^CR = AN RECHTEN ZEILENRAND
^W = AUF NAECHSTES WORT      ^T = TABULATOR

--- BLOCK-BEFEHLE ---          --- FINDE / ERSETZE ---

^BM = BLOCK MARKIEREN         ^FS = SETZE SUCHWORT
^BR = B.-MARK. RUECKSETZEN   ^FF = FINDE SUCHWORT
^BF = B.-MARK. FINDEN        ^FE = ERSETZE SUCHWORT

^BL = BLOCK LOESCHEN
^BK = BLOCK KOPIEREN
^BV = BLOCK VERSCHIEBEN
^BS = BLOCK SPEICHERN (FLOPPY/CASSETTE)

```

Erläuterungen:

^N = CTRL-N , d.h. CTRL-Taste drücken und festhalten, "N" eingeben, CTRL-Taste loslassen

^ZL = CTRL-ZL , d.h. CTRL-Taste drücken und festhalten, "Z" eingeben, "L" eingeben, CTRL-Taste loslassen

Wird bei einer Doppel-CTRL-Sequenz (z.B. ^ZL) als zweites Zeichen ein ungültiger Buchstabe eingegeben, so führt dieses zu einem Abbruch der eingeleiteten CTRL-Funktion mit einer gleichzeitigen akustischen Fehlermeldung. So lassen sich irrtümlich eingeleitete Doppel-CTRL-Sequenzen durch die Eingabe von <SPACE> als zweites Zeichen gezielt abbrechen (z.B. CTRL-Z <SPACE>).

3. Aufruf des BFZ-EDITORS

Je nach Installation des BFZ-EDITORS (siehe 1.2.) wird dieser von der Kommando-Ebene aus aufgerufen:

- a) E <CR> oder <SPACE> eintippen
- b) G <CR> oder <SPACE> eintippen
 5000 <CR> oder <SPACE> eintippen

Nach dem EDITOR-Aufruf wird der Bildschirm gelöscht, und es erscheint folgender Meldetext:

```
*** BFZ-EDITOR V1.3 ***
(C) 1987 BFZ, ESSEN

MAT85 = ESC
HILFSMENUE = X
KALTSTART = K
WARMSTART = <CR>, <SPACE>

> _
```

Durch Betätigen der Taste

(Escape = Flucht)

erfolgt eine Rückkehr zum Betriebsprogramm MAT-85. Während der Bearbeitung des Textes (Editieren) ist die "ESC"-Taste inaktiv, so daß die Rückkehr zum Betriebssystem immer mit Hilfe eines "Neustarts" (CTRL-N, siehe 4.2.) mit anschließendem "ESC" erfolgen sollte.

Eine unbeabsichtigte Rückkehr zum Betriebsprogramm MAT-85 durch versehentliche Betätigung der "ESC"-Taste führt nicht zu einem Verlust des bearbeiteten Textes. Durch einen erneuten Aufruf des EDITORS mit nachfolgendem "Warmstart" kann die Textbearbeitung fortgesetzt werden.

Das Hilfsmenue läßt sich aufrufen, indem die Taste

(X = Hilfsmenue)

gedrückt wird. Das EDITOR-Programm durchsucht in diesem Fall den Arbeitsspeicher nach den Identifizierungs-Bytes des Hilfsmenues (ED DD ED, hex) und erkennt so, ob das Hilfsmenue installiert wurde.

Wird die Hilfsmenue-Kennung gefunden, so erscheint nach dem Löschen des Bildschirms die erste Seite des Menues. Durch Betätigen der <SPACE>-Taste läßt sich die zweite und dritte Seite des Hilfsmenues auf dem Bildschirm darstellen. Durch nochmalige Betätigung der <SPACE>-Taste beginnt die Darstellung des Hilfsmenues erneut mit der ersten Hilfsmenue-Seite ("Endlos-Blättern"). Mit Hilfe der <CR>-Taste kann die Darstellung des Hilfsmenues unterbrochen werden, und es erscheint wieder der EDITOR-Meldetext auf dem Bildschirm.

Wird das Hilfsmenue nicht gefunden, so bleibt der Bildschirminhalt unverändert, und es ertönt als Bedienerhinweis ein Klingelzeichen ("Bell").

Soll der EDIT-Buffer "gelöscht" werden, um einen neuen Text zu bearbeiten oder um einen auf Diskette oder Cassette gespeicherten Text in den "leeren" EDIT-Buffer zu laden, so muß ein "Kaltstart" ausgeführt werden:

(K = Kaltstart)

Wird dagegen ein "Warmstart" ausgeführt, so bleibt der aktuelle Text des EDIT-Buffers erhalten:

(= Warmstart)

In beiden Fällen (Kaltstart oder Warstart) erscheint nun die Meldung:

Die Startadresse 6000 wird immer dann vorgeschlagen, wenn das MFA-System neu eingeschaltet wurde ("Power-On-Reset"). Diese Vorschlagadresse kann durch <CR> oder <SPACE> angenommen oder durch Eintippen einer neuen Startadresse mit anschließendem <CR> oder <SPACE> verändert werden. Wichtig ist auf jeden Fall, daß ab der angegebenen Startadresse ein freier Speicherbereich für den EDIT-Buffer vorhanden ist, der mit dem zu bearbeitenden Text gefüllt werden kann.

Das EDITOR-Programm führt nun für die angegebene Start-Adresse einen "RAM"-Test durch. Steht an dieser Stelle kein Schreib-Lese-Speicher zur Verfügung, so wird erneut die Eingabe einer neuen Start-Adresse durch den Bediener verlangt. Zusätzlich erfolgt eine akustische Fehlermeldung.

War der "RAM"-Test erfolgreich, so wird anschließend der EDIT-Buffer ab der angegebenen Start-Adresse nach dem Markierungsbyte für das EDIT-Buffer-Ende (00) durchsucht. Wird dieses gefunden, so wird die Meldung über die Länge des EDIT-Buffers durch die Angabe der Stop-Adresse vervollständigt, z.B. bei einem Kaltstart:

```
EDIT-BUFFER:
START-ADR =6000
STOP -ADR =6002

==> <CR>,<SPACE>
```

(Bemerkung: Bei einem Kaltstart werden an den Anfang des EDIT-Buffers die beiden Steuerbytes "0D 0A" (hex), sowie die Buffer-Ende-Markierung "00" eingeschrieben. Als EDIT-Buffer-Ende ergibt sich also für obiges Beispiel die "Stop-Adresse" 6002 (hex), siehe auch Übung 1 im Kapitel 4.)

Bei einem Warmstart wird je nach Länge des bearbeiteten Textes die entsprechend veränderte Stop-Adresse angegeben. Die Länge des EDIT-Buffers ist beliebig, sie wird nur durch den zur Verfügung stehenden freien Bereich im Arbeitsspeicher eingeschränkt. Hierbei muß der Anwender unbedingt darauf achten, daß keine wichtigen Speicherbereiche durch den eingegebenen Text überschrieben werden, z.B.:

```
!!!      ab FB00:  Daten-Puffer für BFZ-MINI-DOS      !!!
!!!      ab FB00:  Variablen-Bereich für BFZ-EDITOR  !!!
```

Nach einem Warmstart wird der Bildschirm gelöscht, und die erste Seite des EDIT-Buffer-Inhalts erscheint auf dem Bildschirm. Der im EDIT-Buffer befindliche Text kann weiter bearbeitet werden.

Nach einem Kaltstart ist der Bildschirm leer. Hier kann nun die Eingabe eines neuen Textes erfolgen.

4. Beschreibung der Steuerbefehle des BFZ-EDITORS

4.1. Cursor-Steuerung mit Hilfe der "Pfeil"-Tasten und <CR>

Nach dem Kalt- oder Warmstart des EDITOR-Programms kann der EDIT-Buffer mit Hilfe der Tastatureingabe des Bedieners bearbeitet werden. Der Text wird zeilenweise eingegeben und jede Zeile mit einem <CR> abgeschlossen. Als Kennzeichnung des Zeilenendes erscheint das Zeichen "<" auf dem Bildschirm. Die maximale Zeilenlänge für das MFA-Video-Interface liegt bei 61 Zeichen. Wird diese maximale Zeilenlänge erreicht, so können keine weiteren Textzeichen eingegeben werden. In diesem Fall ertönt als Hinweis für den Bediener ein Klingelzeichen. Um die Textbearbeitung in der nächsten Zeile fortsetzen zu können, muß ein <CR> eingegeben werden.

Die eingegebenen Text- und Satzzeichen werden vom EDITOR im EDIT-Buffer gespeichert und verwaltet. Als Kennzeichnung der Zeilentrennung stehen im EDIT-Buffer an der entsprechenden Stelle die beiden Steuer-Bytes "0D 0A" (hex). Diese Steuer-Bytes besitzen im internationalen ASCII-Code folgende Bedeutung:

0D = Carriage-Return (Wagen-Rücklauf, d.h. Cursor an den Zeilenanfang)

0A = Line-Feed (Zeilenvorschub, d.h. Cursor eine Zeile nach unten)

Wurde eine Zeile mit <CR> abgeschlossen und das Zeilenende mit "<" auf dem Bildschirm markiert, so kann dieses markierte Zeilenende nicht mehr überschrieben werden. Mit Hilfe des "Einfüge-Modus" läßt sich allerdings das markierte Zeilenende bis auf die maximale Zeilenlänge nach rechts verschieben (siehe "Einfüge-Modus" 4.5.).

Mit Hilfe der "Pfeil"-Tasten zur Cursor-Steuerung (auf der Tastatur des Video-Interface links und rechts neben der SPACE-Taste) läßt sich der Cursor über den erstellten Text auf dem Bildschirm bewegen.

Die Cursorbewegung nach links bzw. nach rechts wird jeweils gestoppt, wenn der linke bzw. rechte Textrand erreicht wird (Klingelzeichen als Hinweis für den Bediener).

Die Cursorbewegung nach oben bzw. unten erfolgt innerhalb der aktuellen Bildschirm-Spalte. Ist die neu erreichte Zeile allerdings kürzer als die "alte" Zeile, so wird der Cursor an das Ende der neuen Zeile geführt.

Eine Aufwärtsbewegung des Cursors endet am oberen Bildschirmrand (Klingelzeichen als Bedienerhinweis). Um den davor liegenden Text zu erreichen, muß um eine "Seite nach oben" geblättert werden (siehe auch 4.9.: "Seiten blättern").

Ist der erstellte Text länger als eine Bildschirmseite, so wird der Bildschirminhalt jeweils um eine Zeile nach oben verschoben ("Scrollen"), wenn eine Abwärtsbewegung des Cursors aus der untersten Zeile erfolgt (Cursor-Pfeil "unten"). Anschließend wird der Cursor auf den Anfang der neuen, untersten Bildschirmzeile geführt.

4.2. Neustart: CTRL-N

Alle in 2. angegebenen Steuerzeichen des EDITORS können vom Bediener eingegeben werden, während der EDIT-Buffer bearbeitet wird, der Inhalt des Buffers also auf dem Bildschirm dargestellt wird. Um die Eingabe von Steuerzeichen von der Eingabe von Text- und Satzzeichen zu unterscheiden, müssen alle Steuerzeichen mit Hilfe der CTRL-Taste eingegeben werden (Control). Dazu wird vom Bediener die CTRL-Taste gedrückt und festgehalten, dann das entsprechende Steuerzeichen eingegeben und beide Tasten wieder losgelassen. Die Eingabe mit Hilfe der CTRL-Taste wird vielfach auch wie folgt dargestellt, z.B.:

CTRL-N = ^N (N = Neustart)

Wird während der Textbearbeitung "CTRL-N" eingegeben, so wird der Bildschirm gelöscht, und es erscheint der EDITOR-Meldetext. Der Bediener hat nun die Wahl, einen Kalt- oder einen Warmstart auszuführen und die Startadresse des EDIT-Buffers unter Umständen zu verändern.

Wird ein Warmstart mit derselben Startadresse durchgeführt, erhält der Anwender auf diesem Wege eine Information über die derzeitige Länge des EDIT-Buffers. Dieses ist unter Umständen wichtig, um ein versehentliches Überschreiben anderer wichtiger Speicherinhalte durch den EDIT-Buffer zu vermeiden.

Mit Hilfe eines Kaltstarts mit derselben Startadresse läßt sich der aktuelle Buffer-Inhalt "löschen". Mit einem Kaltstart unter einer anderen Startadresse kann der Bediener einen neuen EDIT-Buffer "eröffnen" und unabhängig vom ersten Buffer bearbeiten. Die Umschaltung zwischen den einzelnen EDIT-Buffern läßt sich dann durch einen Warmstart mit der jeweiligen Buffer-Startadresse durchführen.

4.3. Aufruf des Hilfsmenues: CTRL-X

Auch während der Textbearbeitung läßt sich das Hilfsmenue aufrufen (siehe auch 3.):

CTRL-X (X = Hilfsmenue)

Nach der Rückkehr aus dem Hilfsmenue wird der alte Bildschirminhalt wieder hergestellt, und die Textbearbeitung kann in der alten Bildschirmzeile fortgeführt werden.

Übung 1

EDITOR-Aufruf, einfache Texteingabe, Cursor-Steuerung
Untersuchung des EDIT-Buffers, Neustart, Hilfsmenue-Aufruf

Hinweis: Die Lösungen zur Übung 1 befinden sich im Anhang A.

1. Rufen Sie das EDITOR-Programm von der Kommando-Ebene des Betriebssystems MAT 85 aus auf (siehe auch 3.):

a) KMD> E <CR> oder <SPACE>

bzw.

b) KMD> G(o)
START-ADR =5000 <CR> oder <SPACE>
<CR> oder <SPACE>

2. Drücken Sie nach dem Erscheinen des EDITOR-Meldetextes die ESC-Taste.
3. Starten Sie den BFZ-EDITOR erneut, und rufen Sie das "Hilfs-Menue" auf (Taste "X"). Wurde das Hilfs-Menue installiert (siehe auch 1.2. und 3.), so erscheint jetzt die erste Seite des Menues auf dem Bildschirm. "Blättern" Sie das Hilfs-Menue durch (<SPACE>), und kehren Sie schließlich zum EDITOR-Meldetext zurück (<CR>).
4. Führen Sie einen "Kaltstart" durch (Taste "K"). Welche Startadresse wird vorgeschlagen ?

START-ADR =

Ändern Sie die Startadresse auf 6200 und geben Sie <CR> oder <SPACE> ein. Wie lautet die Stopadresse ?

START-ADR =6000 6200

STOP -ADR =

Begründung:

Geben Sie <CR> oder <SPACE> ein. Tippen Sie nach dem Löschen des Bildschirms folgenden Text ein:

DIESES IST EINE TESTZEILE.
AAA BBB CCC <CR>
<CR>

5. Geben Sie ein: CTRL-N (CTRL-Taste drücken und festhalten, anschließend "N" eingeben, CTRL-Taste loslassen). Führen Sie nach dem Erscheinen des EDITOR-Meldetextes einen "Warmstart" aus (<CR> oder <SPACE>). Wie lauten jetzt Start- und Stopadresse ?

START-ADR =

STOP -ADR =

Begründung:

6. Nach der Rückkehr zum Monitorprogramm MAT-85 (Neustart: CTRL-N, anschließend ESC-Taste) können Sie den Inhalt des EDIT-Buffers ab Adresse 6200 mit Hilfe des PRINT-Modus darstellen (FORMAT = A).

```
KMD> P          <CR> oder <SPACE>
  START-ADR =6200
  STOP -ADR =6230
  FORMAT   =A
```

Durch welche Steuer-Bytes werden die einzelnen Zeilen des eingegebenen Textes getrennt ?

Zeilentrennung:

Welches Markierungs-Byte steht am Ende des EDIT-Buffers an der Stopadresse ?

Ende-Markierung:

7. Starten Sie den BFZ-EDITOR erneut und führen Sie einen Warmstart mit der vorgeschlagenen Adresse 6200 durch.

Wirkung:

Anschließend mit CTRL-N wieder zurück zum EDITOR-Meldetext und Kaltstart mit Adresse 6200.

Wirkung:

Nochmals zurück mit CTRL-N und Warmstart mit Adresse 6200.

Wirkung:

8. Sehen Sie sich nach der Rückkehr nach MAT-85 den Inhalt des EDIT-Buffers an (wie unter 6.)

6200

Ändern Sie mit Hilfe des MEMORY-Modus (FORMAT = H) das 00-Byte unter der Adresse 6202 um in 20:

```
KMD> M          <CR>
  START-ADR =0000 6202 <CR>
  FORMAT   =A H      <CR>

 6202 00 20        <CR>
```

Führen Sie nun einen Warmstart mit der EDIT-Buffer-Start-Adresse 6200 aus.

Wirkung:

9. Schalten Sie das MFA-System aus und nach ca. 5 Sekunden wieder ein. Führen Sie jetzt einen Kaltstart des BFZ-EDITORS durch. Wie lautet die angebotene Startadresse ?

START-ADR =

Begründung:

Übung 2:

 Cursor-Steuerung und Carriage-Return:

1. Geben Sie nach einem Kaltstart des BFZ-EDITORS folgende Zeichen in den EDIT-Buffer ein:

- Zeile 1: sovielen "A", bis Klingelzeichen ertönt, anschließend <CR>
- Zeile 2: 5 x <SPACE> , 3 x "B" , 5 x <SPACE> , <CR>
- Zeile 3: <CR>
- Zeile 4: <CR>
- Zeile 5: 3 x "C" , 5 x <SPACE> , kein <CR>

2. Bringen Sie den Cursor mit Hilfe der "Pfeil"-Tasten auf alle möglichen Positionen des erstellten Textes. Bewegen Sie den Cursor zum Schluß auf das zweite "C" der letzten Zeile, und geben Sie ein <CR> ein. Untersuchen Sie anschließend die Cursor-Bewegung in den Zeilen 5 und 6.

4.4. Zeichen radieren , Zeichen links löschen

Durch die Eingabe von

CTRL-R

(R = Radieren)

läßt sich das Text-Zeichen, auf das der Cursor positioniert ist, aus dem bestehenden Text entfernen. Um die Wirkung dieses Steuerbefehls auf dem Bildschirm sichtbar zu machen, wird das entsprechende Zeichen auf dem Bildschirm gelöscht und der um ein Zeichen nach links verschobene Zeilenrest erneut dargestellt. Je nach Länge des Zeilenrests und der Daten-Übertragungsrate zur Datensichtstation (Video-Interface) wird für die Ausgabe des Zeilenrests eine bestimmte Zeit benötigt. So ergibt sich z.B. für die Ausgabe einer Zeile mit 60 Zeichen bei einer Übertragungsrate von 1200 Bd (MFA-Video-Interface) eine Zeit von:

$$(60 \text{ Zeichen} \times 11 \text{ Bit/Zeichen}) / 1200 \text{ Bit/s} = 0,55 \text{ s}$$

11 Bit: 1 Start-Bit, 7 Daten-Bits, 1 Paritäts-Bit, 2 Stop-Bits

Für den Anwender bedeutet dieses, daß er beim Radieren eines oder mehrerer Zeichen die Bildschirmausgabe beobachten muß, um seine Tastatureingabe der Bildschirmausgabe anzupassen.

Zeilentrennungs-Zeichen "<" lassen sich durch "CTRL-R" nicht radieren. Hier muß entweder die gesamte Zeile gelöscht, bzw. die nachfolgende Zeile "angehängt" werden (siehe 4.6.). Beim Versuch, ein Zeilentrennungs-Zeichen zu radieren, ertönt als Bedienerhinweis ein Klingelzeichen.

Durch Drücken der

DEL-Taste

(Delete = Löschen)

läßt sich das Zeichen direkt links neben der Cursor-Position löschen. Auch hier wird nach dem Löschen des Textzeichens der um ein Zeichen nach links verschobene Zeilenrest erneut auf dem Bildschirm dargestellt (siehe oben "CTRL-R"). Wird mit "DEL" der linke Bildschirmrand erreicht, so ertönt als Bedienerhinweis ein Klingelzeichen. Die "DEL"-Funktion wird häufig bei der Eingabe eines Textes benötigt, um ein falsch eingegebenes Zeichen wieder zu löschen und durch das korrekte Zeichen zu ersetzen.

4.5. Einfüge-Modus Ein-/Aus

Wird beim BFZ-EDITOR ein Kaltstart ausgeführt, so befindet sich dieser während der Textbearbeitung im sogenannten "Einfüge-Modus". In dieser Betriebsart wird ein eingegebenes Textzeichen an der Cursor-Position in den Text eingeschoben und der Cursor um eine Stelle nach rechts verschoben. Es lassen sich auf diese Weise einzelne Zeichen oder Wörter auf einfache Art in den schon vorhandenen Text einfügen. Wird im Einfüge-Modus die maximale Zeilenlänge erreicht, so können keine weiteren Zeichen eingeschoben werden, und es ertönt ein Klingelzeichen als Bedienerhinweis.

Mit Hilfe des Einfüge-Modus lassen sich also auch Zeilen, die mit dem Zeilentrennungs-Zeichen "<" abgeschlossen wurden, auf die maximale Zeichenanzahl verlängern. Das Zeilentrennungs-Zeichen "<" selbst läßt sich nicht einfügen. Hierzu dient als besondere Funktion das "Einfügen einer Zeile" (siehe 4.6.).

Der "Einfüge-Modus" wird ausgeschaltet durch die Eingabe von

CTRL-E

(E = Einfügen)

Der BFZ-EDITOR befindet sich damit im sogenannten "Überschreib-Modus". Wird der Cursor in dieser Betriebsart auf ein "altes" Textzeichen positioniert und ein "neues" Zeichen eingegeben, so wird das "alte" durch das "neue" Zeichen überschrieben.

Durch erneute Eingabe von "CTRL-E" läßt sich der EDITOR wieder auf die Betriebsart "Einfügen" umschalten, und es ertönt als Hinweis ein Klingelzeichen. Der Bediener hat hierdurch eine Kontrolle über die aktuelle Betriebsart:

CTRL-E und	Klingelzeichen :	--> Einfüge-Modus EIN
CTRL-E und	kein Klingelzeichen :	--> Einfüge-Modus AUS

Bei einem Warmstart bleibt die zuletzt eingestellte Betriebsart erhalten.

4.6. Zeile löschen, Zeile einfügen, Zeile anhängen

Durch die Eingabe von

CTRL-ZL

(ZL = Zeile löschen)

(CTRL-Taste drücken und festhalten, "Z" eingeben, anschließend "L" eingeben, CTRL-Taste loslassen)

wird die Zeile, in der sich der Cursor befindet, aus dem Text entfernt. Um die Bildschirmanzeige zu aktualisieren, wird der "Bildschirmrest" ab der Cursor-Position erneut ausgegeben. Die dafür benötigte Zeit hängt von der Zeichenanzahl des "Bildschirmrests", sowie von der Daten-Übertragungsrate zur Datensichtstation ab (siehe auch 4.4.: "Zeichen radieren").

Mit Hilfe der Eingabe

CTRL-ZE

(ZE = Zeile einfügen)

wird an der Stelle der Cursor-Position ein Zeilentrennungs-Zeichen "<" eingefügt. Befindet sich der Cursor am Zeilenanfang, wird auf diese Weise eine neue Zeile, die nur aus dem Zeilentrennungs-Zeichen "<" besteht, vor der aktuellen Zeile eingefügt und der neue "Bildschirmrest" ausgegeben. Die Cursor-Position auf dem Bildschirm bleibt erhalten, er zeigt nunmehr auf den Anfang der neuen Zeile. Um diese neue Zeile beschreiben zu können, muß der Einfüge-Modus eingeschaltet sein (siehe 4.5.).

Steht der Cursor bei der Eingabe von "CTRL-ZE" nicht am Anfang einer Textzeile, so wird das Zeilentrennungs-Zeichen genau in dieser Cursor-Position eingefügt und der "Bildschirmrest" neu aufgebaut. Auf diese Weise lassen sich Textzeilen an der Cursor-Position trennen. Der Cursor wird anschließend auf das Ende der alten Zeile positioniert.

Ein Zusammenbinden von nacheinander liegenden Textzeilen ist möglich durch die Eingabe von

CTRL-ZA

(ZA = Zeile anhängen)

Dabei wird das Zeilentrennungs-Zeichen "<" aus der Zeile gelöscht, in der sich der Cursor befindet. Das hat die Wirkung, daß die nachfolgende Zeile an die aktuelle Zeile angehängt wird. Der Cursor wird dabei auf die "Verbindungsstelle" zwischen alter und angehängter Zeile geführt.

Wird beim Anhängen die maximale Zeichenanzahl für die neu entstehende Zeile überschritten, so wird diese automatisch auf die maximale Länge begrenzt, und der Rest der Zeile "rutscht" eine Zeile tiefer. Nach dem Anhängen einer Zeile wird der Bildschirm-inhalt aktualisiert.

Sollte nach dem Trennen einer Zeile mit CTRL-ZE der Bedarf bestehen, doch wieder die alte Zeile herzustellen, so läßt sich die Trennung durch die anschließende Eingabe von CTRL-ZA wieder rückgängig machen. Ebenso kann eine angehängte Zeile (CTRL-ZA) mit Hilfe einer nachfolgenden Trennung (CTRL-ZE) sofort wieder an ihre alte Position gebracht werden.

Übung 3

Zeichen löschen und einfügen,
Zeilen löschen, einfügen, trennen und anhängen

1. Geben Sie nach einem Kaltstart des EDITORs folgenden Text ein:

```
MIT HILFE DES EDITOR-<
<
PROGRAMMS LASSEN SICH      TEXTE ERSTELLEN <
<
      UND AUF DISKETTE ODER CASSETTE SPEICHERN. <
```

2. Ändern Sie den eingegebenen Text um in:

```
MIT HILFE DES BFZ-EDITORS LASSEN SICH TEXTE <
ERSTELLEN, BEARBEITEN UND AUF DISKETTE ODER <
CASSETTE SPEICHERN.<
```

Verwenden Sie dazu die Steuerfunktionen:

Cursor-Positionierung:	"Pfeil"-Tasten
Radieren einzelner Zeichen	(CTRL-R)
Ein- und Ausschalten des Einfüge-Modus	(CTRL-E)
Zeile löschen	(CTRL-ZL)
Zeile anhängen	(CTRL-ZA)
Zeile einfügen, bzw. Zeile trennen	(CTRL-ZE)

3. Ergänzen Sie den Text wie folgt:

```
AUFGABE DES EDITOR-PROGRAMMS: <
<
MIT HILFE DES BFZ-EDITORS LASSEN SICH TEXTE <
ERSTELLEN, BEARBEITEN UND AUF DISKETTE ODER <
CASSETTE SPEICHERN.<
```

4.7. Erweiterte Cursor-Bewegungen

Die erweiterten Cursor-Bewegungen dienen zum schnelleren Positionieren des Cursors innerhalb einer dargestellten Bildschirmseite:

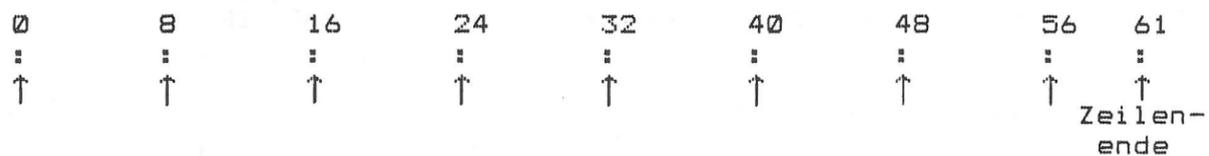
- CTRL-CO** (CO = Cursor nach oben)
Cursor-Positionierung auf die oberste Zeile
- CTRL-CU** (CU = Cursor nach unten)
Cursor-Positionierung auf die unterste Zeile
- CTRL-CL** (CL = Cursor nach links)
Cursor-Positionierung auf den Zeilenanfang
- CTRL-CR** (CR = Cursor nach rechts)
Cursor-Positionierung auf das Zeilenende
- CTRL-W** (W = nächstes Wort)
Cursor-Positionierung auf das nächste Wort

4.8. Tabulator

Mit Hilfe der Eingabe

CTRL-T (T = Tabulator)

wird der Cursor innerhalb einer Zeile auf die nächste Tabulatorposition bewegt. Die Tabulatorpositionen liegen jeweils 8 Zeichen voneinander entfernt, beginnend in der Spalte 0:



Mit Hilfe der Tabulator-Funktion läßt sich der zu erstellende Text auf einfache Weise in tabellarischer Form anordnen, hier am Beispiel einer Stückliste gezeigt:

Beispiel:	Stückliste	
Kennz.	Benennung/Daten	Bemerkung
R1 ... R9	Widerstand 4k7	
R10... R17	Widerstand 560	
R18... R25	Widerstand 4k7	
C1, C2	Tantal-Elko 4,7 uF/25 V	Tropfenform
IC 1, IC 6	Sechs Inverter 74 LS 04	
IC 3, IC 4	4-Bit-Vergleicher 74LS85	
IC 5	8-Bit-D-Register 74LS273	
IC 2, IC 7	Sechs Inverter 7406	offener Kollektor
↑	↑	↑
0	16	48

Die Tabulator-Funktion kann im "Überschreib-Modus" oder auch im "Einfüge-Modus" benutzt werden:

Im "Überschreib-Modus" bleibt beim Einsatz des Tabulators der alte Text erhalten, der Cursor wird nur auf die nächste Tabulator-Spalte positioniert, z.B.:

```
L1: CALL TABELLE           Cursor Spalte 0
↑
Eingabe: CTRL-T

L1: CALL TABELLE           Cursor Spalte 8
↑
```

Erreicht der Cursor die maximale Zeilenlänge, so ertönt als Bedienerhinweis das Klingelzeichen.

Im "Einfüge-Modus" läßt sich mit Hilfe der erweiterten Cursor-Bewegung "CTRL-W" (Cursor auf das nächste Wort) und der Tabulator-Eingabe "CTRL-T" ein schon erstellter Text spaltenweise formatieren:

```
L1: CALL TABELLE           Cursor in Spalte 0
↑
Eingabe: CTRL-W ==>

L1: CALL TABELLE           Cursor nächstes Wort (Spalte 4)
↑
Eingabe: CTRL-T ==>

L1: CALL TABELLE           Cursor mit Zeilenrest in Spalte 8
↑
Eingabe: CTRL-W ==>

L1: CALL TABELLE           Cursor nächstes Wort (Spalte 13)
↑
Eingabe: CTRL-T ==>

L1: CALL TABELLE           Cursor mit Zeilenrest Spalte 16
↑
: : :
0 8 16
```


4.10. Finden eines "Text-Strings", Ersetzen eines "Text-Strings"

Der sogenannte "FIND"-Modus dient zum Auffinden von Zeichenketten ("Strings") innerhalb des bearbeiteten Textes. Je nach Bedarf kann die gesuchte Zeichenkette durch eine neue ersetzt werden.

Durch die Eingabe von

CTRL-FS (FS = "FIND"-String setzen)

wird der Bildschirm gelöscht, und in der linken oberen Bildschirmcke erscheint der Meldetext:

FINDE : <

Der Anwender hat nun die Möglichkeit, eine beliebige Zeichenkette mit einer maximalen Länge von 31 Zeichen als "Find"-String einzugeben (auch Leerzeichen, Ziffern oder Satzzeichen). Eine Korrektur der Eingabe ist mit Hilfe der "DEL"-Taste möglich. Mit Hilfe von <CR> wird die Definition des "Find"-String abgeschlossen.

Durch die Ausgabe des Meldetextes

ERSETZE: <

erfolgt die Aufforderung an den Bediener, bei Bedarf eine "Ersatz"-Zeichenkette einzugeben (maximal 31 Zeichen, Eingabekorrektur wiederum mit Hilfe der "DEL"-Taste). Die Eingabe wird mit <CR> beendet.

Wird "CTRL-FS" zu einem späteren Zeitpunkt erneut aufgerufen, so erfolgt nach der Ausgabe des Meldetextes die Darstellung des alten "Find"- bzw. "Ersatz"-String. Diese einmal definierten Zeichenketten bleiben erhalten, wenn sie durch die Eingabe von <CR> vom Anwender quittiert werden. Wird ein anderes Zeichen als <CR> angegeben, können jeweils neue Zeichenketten eingegeben werden.

Beispiel:

- a) **FINDE : SUCHWORT-ALT<** **<CR>** : SUCHWORT-ALT bleibt erhalten
- b) **FINDE : N** **N** : neues Suchwort wird definiert, N ist erstes neues Zeichen
- FINDE : Neues Suchwort<** **Neues Suchwort <CR>**

Sind "Find"- und auch "Ersatz"-String definiert worden, so erfolgt anschließend die Suche nach der gewünschten Zeichenkette innerhalb des EDIT-Buffers, und zwar ab der aktuellen Cursor-Position.

Dabei bleiben Groß- oder Kleinschreibung innerhalb des "Find"-String unberücksichtigt. D.h., mit dem Suchwort

"ERGEBNIS"

wird z.B. auch gefunden:

"Ergebnis"
"Zwischenergebnis"
"ergebnislos"

War die Suche erfolgreich, so wird die gesuchte Bildschirmseite dargestellt. Die gefundene Zeichenkette befindet sich in der obersten Bildschirmzeile, der Cursor zeigt auf den Anfang dieser Zeichenkette. Wurde die gesuchte Zeichenkette nicht gefunden, so wird der "alte" Bildschirminhalt wieder dargestellt, und es ertönt das Klingelzeichen.

Durch die Eingabe von

CTRL-FE (FE = "Find"-String ersetzen)

läßt sich die gefundene Zeichenkette durch die definierte "Ersatz"-Zeichenkette austauschen. Um Fehlbedienungen zu vermeiden, ist ein Austausch nur dann möglich, wenn der Cursor bei der Eingabe von "CTRL-FE" auf den Anfang des Such-String positioniert ist. Ist der "Ersatz"-String länger als der "Such"-String, so kann die neu entstehende Zeile unter Umständen die maximale Zeilenlänge überschreiten. In diesem Fall wird die Zeile auf ihre maximale Länge beschnitten, der Zeilenrest "rutscht" dann eine Zeile tiefer. Ist der Ersatz-String "leer", so wird der "Such-string" beim Ersetzen lediglich aus dem Text gelöscht.

Mit Hilfe von

CTRL-FF (FF = "Find"-String finden)

läßt sich jederzeit eine einmal definierte Zeichenkette erneut im EDIT-Buffer suchen und je nach Bedarf austauschen. Wird das gesuchte Wort nicht gefunden, so ertönt als Bedienerhinweis das Klingelzeichen.

Bei einem Warmstart des EDITOR-Programms bleiben einmal definierte Such- oder Ersatz-Zeichenketten erhalten. Wird ein Kaltstart ausgeführt, so werden die "alten" Zeichenketten gelöscht.

4.11. Block-Befehle

4.11.1 Block markieren, Block-Markierungen rücksetzen, Block-Markierungen finden

Mit Hilfe des "BLOCK"-Modus lassen sich beliebige zusammenhängende Text-Blöcke markieren, löschen, kopieren, verschieben oder aus dem EDIT-Buffer heraus als Text-Datei auf die Diskette oder Cassette speichern.

Das Markieren von Text-Blöcken ist nur zeilenweise möglich. Dazu wird der Cursor auf die erste Zeile positioniert, die noch zu dem Text-Block gehören soll (siehe Beispiel unten: 1 -->). Durch die Eingabe von

CTRL-BM

(BM = Block markieren)

wird die sogenannte Block-Marke "--B--" vor den zu markierenden Text-Block gesetzt.

Das Ende des Text-Blocks wird auf die gleiche Art festgelegt: der Cursor wird auf die erste Zeile positioniert, die nicht mehr zu dem Text-Block gehören soll (siehe Beispiel unten: 2 -->). Mit "CTRL-BM" wird die Block-Marke "--B--" nun direkt hinter dem zu markierenden Text-Block eingefügt. Dieser ist nun also durch zwei Block-Marken eingeschlossen:

```

1 --> Diese Zeile gehört noch nicht zu dem Text-Block.<
      --B--<
      Hier beginnt der markierte Text.<
      Die Länge des markierten Textes<
      ist beliebig.<
      Das ist die letzte Zeile des Text-Blocks.<
      --B--<
2 --> Diese Zeile gehört nicht mehr zu dem Text-Block.<

```

Die Zeichenkette "--B--" wird vom EDITOR-Programm nur dann als Blockmarke gewertet, wenn sie völlig allein in einer Zeile steht. Werden vom Anwender in der markierten Zeile zusätzliche Zeichen vor oder hinter die Block-Marke geschrieben, so verliert diese Block-Markierung ihre Funktion (Sicherung vor Fehlbedienung).

Alle im Text befindlichen Block-Marken "--B--" lassen sich durch die Eingabe von

CTRL-BR

(BR = Block-Markierung rücksetzen)

zurücksetzen.

Mit Hilfe von

CTRL-BF

(BF = Block-Markierung finden)

werden Block-Marken "--B--" ab der aktuellen Cursor-Position gesucht und die gefundene Seite auf dem Bildschirm dargestellt.

4.11.2 Block löschen, Block kopieren, Block verschieben

Die nachfolgenden Block-Befehle dienen dazu, einen markierten Block weiter zu verarbeiten. Ein Text-Block kann nur dann eindeutig in seinem Anfang und seinem Ende definiert sein, wenn innerhalb des gesamten Textes genau zwei Block-Marken "--B--" gesetzt sind. Ist dieses nicht der Fall (z.B. keine Block-Marke oder nur eine oder mehr als zwei Block-Marken), so erfolgt beim Aufruf aller folgenden Block-Befehle die Fehlermeldung:

--- FEHLER: BLOCK-MARKIERUNG

==> <CR>,<SPACE>

Durch die Eingabe von

CTRL-BL

(BL = Block löschen)

wird ein markierter Block (inklusive seiner beiden Block-Marken) aus dem EDIT-Buffer gelöscht. Dieser EDITOR-Steuerbefehl ist mit Vorsicht zu verwenden, da der gelöschte Text-Block in den meisten Fällen unwiederbringlich aus dem Arbeitsspeicher des Computers entfernt wird.

Beispiel:

Vor CTRL-BL:

```

Dieser Text gehört nicht zum markierten <
Block und wird nicht gelöscht.<
--B--<
Markierter Block<
--B--<
Ab hier soll nicht mehr gelöscht werden.<

```

Nach CTRL-BL:

```

Dieser Text gehört nicht zum markierten <
Block und wird nicht gelöscht.<
Ab hier soll nicht mehr gelöscht werden.<

```

Mit Hilfe von

CTRL-BK

(BK = Block kopieren)

läßt sich ein markierter Text-Block an eine beliebige Stelle innerhalb des Textes duplizieren. Der kopierte Block wird dabei vor die Zeile der aktuellen Cursor-Position eingeschoben. Die Block-Marken des "alten" Blocks werden gelöscht, der kopierte Block ist erneut durch die beiden Block-Marken gekennzeichnet:

Beispiel:

Vor CTRL-BK:

Cursor -->

```
Dieser Text wird nicht kopiert.<
--B--<
Markierter Block<
--B--<
Diese Zeile wird ebenfalls nicht kopiert.<
```

Nach CTRL-BK:

Cursor -->

```
--B--<
Markierter Block<
--B--<
Dieser Text wird nicht kopiert.<
Markierter Block<
Diese Zeile wird ebenfalls nicht kopiert.<
```

Hinweise:

1. Der Kopierversuch eines Blocks in die eigenen Block-Grenzen wird durch das EDITOR-Programm abgefangen, und es ertönt ein Klingelzeichen als Fehlermeldung.
2. Je nach Länge des gesamten EDIT-Buffers und Länge des zu kopierenden Blocks wird für das Duplizieren des Text-Blockes eine bestimmte Zeit benötigt. Diese Wartezeit macht sich allerdings erst bei Buffer-Längen von einigen 10 KByte bemerkbar.

Der Block-Befehl

CTRL-BV

(BV = Block verschieben)

dient dazu, einen markierten Textblock (einschließlich der Block-Markierung "--B--") innerhalb des EDIT-Buffers zu verschieben. Der verschobene Block wird dabei vor die Zeile der aktuellen Cursor-Position eingesetzt.

Beispiel:

Vor CTRL-BV:

```
Dieser Text wird nicht verschoben.<
--B--<
Markierter Block<
--B--<
Der markierte Block soll <
hinter diese Zeile verschoben werden.<
In dieser Zeile muß der Cursor stehen.<
```

Cursor -->

Nach CTRL-BV:

```
Dieser Text wird nicht verschoben.<
Der markierte Block soll <
hinter diese Zeile verschoben werden.<
--B--<
Markierter Block<
--B--<
In dieser Zeile muß der Cursor stehen.<
```

Cursor -->

Hinweise:

1. Auch hier gilt, daß das Verschieben eines Blocks in die eigenen Block-Grenzen nicht möglich ist (Klingelzeichen).
2. Um einen Text-Block innerhalb des EDIT-Buffers zu verschieben, lassen sich verschiedene Verfahren anwenden, z.B.:
 - a) Der gesamte Block wird auf einmal verschoben, d.h. zuerst erfolgt ein Duplizieren des "Quellblocks" an die gewünschte Stelle, anschließend wird der "Quellblock" gelöscht. Diese Verschiebetechnik ist verhältnismäßig schnell, benötigt aber unter Umständen je nach Länge des zu verschiebenden Blocks zusätzlichen Speicherplatz (nach dem Kopieren des "Quellblocks" ist dieser zweimal innerhalb des EDIT-Buffers enthalten).
 - b) Der Block wird Zeichen für Zeichen verschoben, d.h. nach dem Kopieren eines Zeichens des Quellblocks in den Zielbereich wird dieses Zeichen erst aus dem Quellblock gelöscht, bevor das nächste Zeichen verschoben wird. Der EDIT-Buffer verlängert sich während des Verschiebevorgangs also nur immer um ein Zeichen, anschließend wird die alte EDIT-Buffer-Länge wieder hergestellt. Der Nachteil dieser Speicherplatz-sparenden Verschiebetechnik ist allerdings ein gehöriger Zeitaufwand (Platz machen für ein Zeichen im Zielbereich, kopieren des Zeichens, löschen des Zeichens aus dem Quellbereich).

Das im EDITOR-Programm verwendete Verschiebep Verfahren stellt einen Kompromiß beider dargestellten Verfahren dar:

Das Verschieben erfolgt Zeile für Zeile, d.h. zuerst wird im Zielbereich des EDIT-Buffers Platz geschaffen für die zu kopierende Zeile, dann wird die Zeile aus dem Quellbereich in den Zielbereich dupliziert und anschließend kann die übertragene Zeile aus dem Quellbereich gelöscht werden. Dieser Vorgang wird für jede Zeile des zu verschiebenden Blocks durchgeführt.

Während des Verschiebevorgangs wird der EDIT-Buffer also maximal um die Länge einer Zeile vergrößert. Dadurch wird zum Verschieben eines Blocks kein wesentlich größerer Speicherplatz benötigt. Der Zeitbedarf für das "Auf- und Ab-Pumpen" des EDIT-Buffers (Kopieren und Löschen einer Zeile) hält sich in vertretbaren Grenzen und macht sich erst bei großen zu verschiebenden Textblöcken und EDIT-Buffer-Längen von einigen 10 KByte bemerkbar.

Ein markierter Block läßt sich mit Hilfe von

CTRL-BS

(BS = Block sichern)

auf die Diskette oder Cassette speichern. Dabei wird der Textblock ohne die Block-Markierungen "--B--" zum angeschlossenen Diskettenlaufwerk oder zum MFA-Cassetten-Interface übertragen. Der gesicherte Textblock wird dabei als eigene Textdatei abgespeichert und läßt sich später wieder in den EDIT-Buffer laden (siehe auch 4.12. "Speichern und Laden mit Hilfe von Diskette oder Cassette").

4.12. Speichern und Laden mit Hilfe von Diskette oder Cassette

4.12.1 Speichern auf Diskette oder Cassette

Mit Hilfe von

CTRL-S

(S = Speichern/Laden)

läßt sich der Inhalt des aktuellen EDIT-Buffers auf die Diskette oder die Cassette speichern.

Nach dem Aufruf der Steuerfunktion CTRL-S wird der Bildschirm gelöscht und folgender Meldetext ausgegeben:

--- SPEICHERN/LADEN

F = FLOPPY
C = CASSETTE
E = EDITOR

> _

Wie in 4.11.2 beschrieben, läßt sich auch ein markierter Block mit Hilfe des Block-Befehls "CTRL-BS" auf Diskette oder Cassette speichern. Zur Unterscheidung erfolgt beim "Block"-Speichern die veränderte Meldetext-Ausgabe:

--- BLOCK SPEICHERN

F = FLOPPY
C = CASSETTE
E = EDITOR

> _

Durch die Eingabe des Zeichens "F" erfolgt ein Aufruf des BFZ-MINI-DOS (DOS = Disk Operating System, Disketten-Verwaltungsprogramm). Ist das BFZ-MINI-DOS nicht installiert, so wird eine akustische Fehlermeldung ausgegeben, und obiger Meldetext erscheint erneut.

Zum Speichern einer Textdatei auf die Cassette muß der Bediener nach der Ausgabe des oben angegebenen Meldetextes die Taste "C" betätigen. Der Meldetext wird daraufhin wie folgt erweitert:

```

--- CASSETTE
S = SAVE
L = LOAD
E = EDITOR
> _

```

Durch die Eingabe von "S" wird die SAVE-Routine für das MFA-Cassetten-Interface aufgerufen, und der EDIT-Buffer-Inhalt läßt sich als Textdatei auf die Cassette speichern (siehe auch "Fachtheoretische Übungen, Beschreibung MAT-85"). Auch hier erfolgt die Übergabe der Start- und Stop-Adressen an die Cassetten-SAVE-Routine automatisch durch das EDITOR-Programm.

4.12.2 Laden einer Textdatei von Diskette oder Cassette

Das Laden einer Textdatei von der Diskette oder der Cassette in den EDIT-Buffer erfolgt ebenfalls durch die Eingabe der Steuerfunktion

```

[CTRL-S]      (S = Speichern/Laden)

```

Wie schon in 4.12.1 beschrieben, erfolgt nach der Ausgabe des Meldetextes die Auswahl durch den Bediener ("F" = FLOPPY, "C" = CASSETTE).

Wurde das BFZ-MINI-DOS aufgerufen ("F" = FLOPPY), so läßt sich mit Hilfe von

```

"LOAD"          [L]      <CR>  eingeben
"Laufwerkauswahl" [A] oder [B] <CR>  eingeben
"NAME:"         [NAME.EXT] eingeben

```

eine Textdatei (Directory-Typ: "??") von der Diskette in den Arbeitsspeicher des Computers übertragen. Dabei wird diese Textdatei immer an das Ende des aktuellen EDIT-Buffers geladen. Dadurch lassen sich Texte, die mit Hilfe des EDITORS auf die Diskette gespeichert worden sind, aneinanderbinden (engl.: "Merge", dtsh. "Vereinigung"). Ist der EDIT-Buffer "leer" (nach einem Kaltstart), so befindet sich nach dem Ladevorgang nur der von der Diskette übertragene Text im EDIT-Buffer.

Nach dem Rücksprung zum BFZ-EDITOR mit Hilfe von "Q" (Quit) steht der geladene Text zur weiteren Bearbeitung im EDIT-Buffer bereit.

Ebenso erfolgt die Übertragung einer Textdatei von der Cassette an das Ende des aktuellen EDIT-Buffers, wenn der Bediener nach der Auswahl von "C" (CASSETTE) die "LOAD"-Routine für das MFA-Cassetten-Interface aufruft.

4.13. Drucken des EDIT-Buffer-Inhalts

4.13.1 Drucker-Menue

Das Drucker-Menue wird aufgerufen mit Hilfe von

CTRL-D (D = Drucker)

Der Bildschirm wird gelöscht, und es erscheint der Meldetext des Drucker-Menues:

```
--- DRUCKER-MENUE
ZEILENRAND:  8 _
```

Der vorgegebene Zeilenrand für den Drucker kann mit Hilfe von <CR> oder <SPACE> angenommen oder durch die Eingabe eines neuen Zeilenrandes (maximal 99 dez.) mit anschließendem <CR> oder <SPACE> vom Bediener verändert werden.

Mit der Ergänzung des Meldetextes

```
ZEILEN/SEITE (0 = ENDLOSDRUCK):  0 _
```

wird der Bediener aufgefordert, die maximale Zeilenzahl für eine Druck-Seite anzugeben (maximal 99 dez.). Wird beim Ausdrucken diese maximale Zeilenzahl erreicht, so erfolgt automatisch ein Formularvorschub auf das nächste Blatt. Die Länge des Formularvorschubs ist Drucker-spezifisch und muß daher am Drucker eingestellt werden. Wird eine "0" eingegeben, so erfolgt der Ausdruck ohne Überprüfung der maximalen Zeilenzahl pro Druck-Seite (Endlosdruck).

Mit Hilfe von

```
SEITEN-NUMERIERUNG:  1 _
```

wird der Ausdruck der ersten Seiten-Nummer (maximal 99 dez.) innerhalb einer "Kopfzeile" festgelegt (siehe weiter unten).

Nach der Vervollständigung des Meldetextes

```
D = DRUCKEN
E = EDITOR
> _
```

kann der Bediener das Ausdrucken des EDIT-Buffer-Inhalts durch die Eingabe von "D" einleiten.

Ist kein Drucker angeschlossen oder dieser nicht eingeschaltet, so erfolgt eine akustische und optische Fehlermeldung:

```
*** PRINTER NOT READY ***
==> <CR>, <SPACE>
```

Bei angeschlossenem Drucker in Bereitschaftsstellung wird nach der Eingabe von "D" der Inhalt des EDIT-Buffers zum Drucker übertragen.

Hinweis: Das MFA-Video-Interface stellt alle verwendeten Buchstaben in Großschreibung dar. Im EDIT-Buffer werden aber alle Buchstaben so abgelegt, wie sie auch tatsächlich mit der Tastatur eingegeben wurden (z.B. Tastatur-Eingabe "a" ==> "a" wird als ASCII-Zeichen (= 61 hex) im EDIT-Buffer abgelegt ==> auf dem Bildschirm erscheint "A"). Beim Ausdrucken des EDIT-Buffer-Inhaltes werden aber alle als Kleinbuchstaben eingegebenen Zeichen auch als Kleinbuchstaben gedruckt, so daß der gedruckte Text u.U. bzgl. der Groß- und Kleinschreibung nicht mit dem auf dem Bildschirm dargestellten Text übereinstimmt. Soll auch die Ausgabe auf dem Drucker nur in Großschreibung erfolgen, so muß bei der Texteingabe die Taste

```
"ALPHA-LOCK"
```

aktiviert werden (Ausnahme: der Ausdruck von Programm-Listings beim Assemblieren erfolgt immer in Großschreibung, siehe 4.14).

4.13.2 Steuerung des Formularvorschubs mit ".", Ausdruck von Kopfzeilen

Wie unter 4.13.1 erläutert, läßt sich der Formularvorschub durch die Eingabe der maximalen Zeilenzahl pro Druck-Seite in festen Abständen einstellen. Eine weitere Möglichkeit zur Steuerung des Blatt-Vorschubs bieten die sogenannten "Punkt-Zeilen":

- Beginnt eine Zeile des Textes mit einem Punkt ".", so wird an dieser Stelle ein Formularvorschub beim Drucker ausgelöst. Der Rest der Zeile wird beim Ausdrucken ignoriert.

Eine zusätzliche Bedeutung haben "Punkt-Zeilen" am Anfang des Textes:

- Alle "Punkt-Zeilen", die direkt am Anfang des EDIT-Buffers liegen, werden als sogenannte Kopfzeilen definiert. Diese Kopfzeilen werden nach jedem Formularvorschub komplett ausgedruckt.
- Mit Hilfe des "Numerus"-Zeichens "#" läßt sich der Ausdruck einer fortlaufenden Seitenzahl innerhalb einer Kopfzeile festlegen. Der Beginn der Seiten-Numerierung kann mit Hilfe des Drucker-Menues eingestellt werden.

Beispiel:

.Dieses ist die 1. Zeile des EDIT-Buffers<	(Kopfzeile 1)
.<	(Kopfzeile 2)
. --- Seite# ---<	(Kopfzeile 3)
. Hier beginnt die erste Druckseite.<	(Kopfzeile 4)
Anschließend soll ein Formular-<	
vorschub erfolgen.<	
.<	
Der Text wird an dieser Stelle mit<	
der zweiten Druckseite fortgesetzt.<	
Anschließend soll ein Formular-<	
vorschub erfolgen.<	
.Der Rest der Zeile wird ignoriert.<	
Hier beginnt die dritte Druckseite.<	

← (Formularvorschub)

← (Formularvorschub)

Durch den Aufruf des Drucker-Menues wird z.B. gewählt:

```
ZEILENRAND:      8
ZEILEN/SEITE (0 = ENDLOSDRUCK):  0
SEITEN-NUMERIERUNG:  1
```

Das Druck-Ergebnis hätte dann folgende Form:

Dieses ist die 1. Zeile des EDIT-Buffers

 (Seite 1)

--- Seite 1 ---

Hier beginnt die erste Druckseite.
Anschließend soll ein Formular-
vorschub erfolgen.

Dieses ist die 1. Zeile des EDIT-Buffers

 (Seite 2)

--- Seite 2 ---

Der Text wird an dieser Stelle mit
der zweiten Druckseite fortgesetzt.
Anschließend soll ein Formular-
vorschub erfolgen.

Dieses ist die 1. Zeile des EDIT-Buffers

 (Seite 3)

--- Seite 3 ---

Hier beginnt die dritte Druckseite.

4.14. Assemblieren

Wie in der Einleitung schon beschrieben, eignet sich der BFZ-EDITOR besonders dazu, Assembler-Quell-Programme zu erstellen, zu bearbeiten und anschließend durch den Assembliervorgang in ein Maschinen-Programm umzusetzen. Die Bearbeitung des Assembler-Quelltextes erfolgt mit Hilfe des EDITORS im EDIT-Buffer (Vorschlags-Start-Adresse z.B.: 6000 hex.). Je nach Länge des erstellten Assembler-Programms verschiebt sich die EDIT-Buffer-Stop-Adresse entsprechend. Dabei ist zu beachten, daß ein voll kommentierter Assembler-Text von etwa einer DIN-A4-Seite (das entspricht ca. 60 Zeilen, bzw. ca. 4 Bildschirmseiten) einen EDIT-Buffer von ca. 1,5 bis 2 KByte Länge benötigt (d.h., die EDIT-Buffer-Stop-Adresse liegt dann bei ca. 6600 bis 6800 hex.).

Durch den Assembliervorgang läßt sich der erstellte Quell-Text in ein zugehöriges Maschinen-Programm umsetzen (Vorschlags-Adresse z.B.: E000 hex.). Für ein 60-zeiliges Maschinen-Programm mit durchschnittlich 2 Byte Maschinen-Code pro Zeile ergibt sich somit ein Speicherbedarf von ca. 120 Byte (d.h., E000 bis ca. E078 hex.).

Für den Assemblier-Vorgang wird der "Line-By-Line-Assembler" des Betriebssystems MAT 85 benutzt. Aus diesem Grund muß der erstellte Assembler-Text den Regeln des MAT 85 - Assemblers genügen (siehe MAT 85 - Beschreibung).

Ein mit dem EDITOR erstelltes Programm im 8085-Assembler-Code kann assembliert werden durch die Eingabe von

CTRL-A

(A = Assemblieren)

Nach dieser Eingabe wird der Bildschirm gelöscht, und es erscheint folgender Meldetext:

```

--- ASSEMBLIEREN
1 = MIT BILDSCHIRM-LISTING
2 = OHNE BILDSCHIRM-LISTING
3 = MIT DRUCKER-LISTING
E = EDITOR
> _

```

Mit der Eingabe der Ziffern 1, 2 oder 3 wird der Assemblier-Vorgang gestartet. Dabei ist folgendes zu beachten:

- a) Die "Start-Adresse" für den erzeugten Maschinen-Code wird bei jedem Assemblier-Vorgang auf "0000" eingestellt (EPROM-Bereich des Betriebs-Systems, daher kein unbeabsichtigtes Überschreiben eines wichtigen RAM-Bereichs möglich). Die eigentliche Wahl der Start-Adresse des erzeugten Maschinen-Programms erfolgt mit Hilfe der "ORG"-Anweisung, z.B.:

```
ORG 0E000 ;Programm-Anfang bei 0E000
```

Bei der Wahl der Anfangsadresse des Maschinenprogramms mit Hilfe der "ORG"-Anweisung ist unbedingt darauf zu achten, daß durch den erzeugten Maschinencode kein wichtiger RAM-Bereich (z.B. der Quelltext im EDIT-Buffer) überschrieben wird.

- b) Vor jedem Assemblier-Vorgang wird die "Label-Tabelle" des MAT 85 - Assemblers gelöscht. Alle im Assembler-Text verwendeten Labels werden also bei einem Assemblier-Vorgang neu erzeugt. Für die Verwendung von Labels gilt folgende wichtige Vereinbarung:

```

!!! Label-Definitionen müssen immer in der ersten   !!!
!!! Spalte einer Programm-Zeile stehen               !!!

!!! Alle anderen Assembler-Kommandos dürfen erst ab  !!!
!!! der zweiten Spalte einer Programm-Zeile stehen  !!!

!!! Kommentare (Kennzeichnung durch vorangestelltes ";" !!!
!!! dürfen in jeder Spalte einer Programm-Zeile beginnen !!!

```

Label-Definitionen können auf zwei Arten vorgenommen werden:

- i) Definition symbolischer Adressen mit dem Doppelpunkt, z.B.
- ANF:
NEXT:
LOOP:
- ii) Definition von Hex-Konstanten, bzw. festliegenden Adressen mit Hilfe der "EQU"-Anweisung, z.B.

```

EING EQU 01 ;8-BIT-PARALLEL-EINGABE
AUSG EQU 02 ;8-BIT-PARALLEL-AUSGABE
TAST EQU 0043 ;TASTATUR-ABFRAGE
KONST EQU 55 ;KONSTANTE = 55H

```

Die Definitionen mit Hilfe der "EQU"-Anweisung müssen am Anfang eines Assembler-Programms vorgenommen werden. Dadurch hat der Anwender die Möglichkeit, diese im Bedarfsfall einfach und schnell im gesamten Programm abzuändern.

Wird die beschriebene Vereinbarung bei der Definition von Labels nicht eingehalten, so führt dieses beim Assemblieren mit Bildschirm-Listing zu einer Fehlermeldung. Aus diesem Grund empfiehlt es sich, ein Assembler-Programm bei der Eingabe mit Hilfe der "Tabulator"-Funktion zu formatieren. Dieses erhöht außerdem die Übersichtlichkeit des erstellten Quelltextes (siehe Beispiel am Ende des Kapitels).

Zur Kontrolle, ob alle verwendeten Labels auch definiert wurden, sollte am Ende des Assembler-Textes die Anweisung

```
LP
```

stehen ("Label-Print"). Bei der Darstellung der Label-Tabelle werden dann alle nicht definierten Labels mit einem "*" gekennzeichnet.

- c) Die "END"-Anweisung zur Beendigung der Programm-Eingabe ist beim Assemblieren mit Hilfe des BFZ-EDITORS überflüssig, da das Ende des Assembler-Programms durch das Text-Ende des EDIT-Buffers definiert ist. Wird die "END"-Anweisung dennoch benutzt, so wird der Assemblier-Vorgang in dieser Zeile beendet, und es erfolgt eine Rückkehr zum BFZ-EDITOR.

Der "RESTART"-Befehl zur Rückkehr eines Maschinenprogramms in die "USER"-Ebene des MAT 85 läßt sich durch die Assembler-Anweisung

```
RST 1 ;*** USER ***
```

direkt erzeugen.

- d) Zur besseren Dokumentation eines Programms sollten entsprechende Kommentare benutzt werden. Die Kommentare werden durch ein vorangestelltes ";" gekennzeichnet, z.B.:

```
ZK EQU 0FFFF ;ZEITKONSTANTE = FFFF(HEX)
;--- UNTERPROGRAMM: ZEITSCHLEIFE ---
ZEIT: PUSH B ;REGISTER RETTEN
      PUSH PSW
      LXI B,ZK ;BC MIT ZEITKONSTANTE LADEN
Z1:   DCX B ;ABWAERTS ZAEHLEN
      MOV A,B
      ORA C ;BC = 0000 ?
      JNZ Z1 ;NEIN: WEITER ABWAERTS ZAEHLEN

      POP PSW ;REGISTER ZURUECK
      POP B
      RET ;RUECKSPRUNG
```

Bei der Verwendung von Kommentaren ist zu beachten, daß beim Assemblier-Vorgang vor jede bearbeitete Zeile die entsprechende Adresse mit dem zugehörigen Maschinen-Code geschrieben wird. Damit rückt der Assembler-Code in die Bildschirm-Spalte 17. Wird dadurch die mit einem Kommentar versehene Assembler-Zeile zu lang, so führt dieses zu einem Fehler beim Assemblier-Vorgang mit der entsprechenden Fehlermeldung (siehe auch nächster Abschnitt).

- e) Wurde beim Erstellen des Assembler-Textes gegen die Regeln des MAT 85 - Assemblers verstoßen ("Syntax-Fehler", z.B. MOV AB anstelle von MOV A,B oder doppelt definierte Labels oder ähnliches), so führt dieses beim Assemblieren zu einer Fehlermeldung:

```
--- FEHLER BEIM ASSEMBLIEREN
==> <CR>,<SPACE>
```

Nach der Betätigung von <CR> oder <SPACE> wird die entsprechende Seite aus dem EDIT-Buffer auf dem Bildschirm dargestellt und der Cursor auf den Anfang der fehlerhaften Zeile positioniert. Diese Zeile kann nun korrigiert und der nächste Assemblier-Vorgang gestartet werden. War der Assemblier-Vorgang fehlerfrei, so erfolgt die Meldung:

```
--- ASSEMBLIEREN OK
==> <CR>,<SPACE>
```

Mit der Eingabe von <CR> oder <SPACE> erfolgt dann eine Rückkehr zum BFZ-EDITOR (Warmstart).

- f) Beim Assemblieren mit Bildschirm-Listing wird der Assemblier-Vorgang nach der Ausgabe einer Bildschirmseite unterbrochen. Durch die Eingabe von <CR> oder <SPACE> läßt sich der Assemblier-Vorgang weiter fortführen.
- g) Beim Assemblieren ohne Bildschirm-Listing wird die Bildschirm-Ausgabe unterdrückt. Dadurch lassen sich längere Assembler-Programme wesentlich schneller in den zugehörigen Maschinen-Code umsetzen. Fehler beim Assemblieren führen auch in diesem Fall zu einer Fehlermeldung, die fehlerhafte Zeile wird durch die entsprechende Cursor-Position nach der Rückkehr zum BFZ-EDITOR lokalisiert.
- h) Ein Assemblier-Vorgang mit Drucker-Listing sollte verständlicherweise erst nach der Beseitigung aller Syntax-Fehler erfolgen, da sonst im Fehlerfall der Druckvorgang unterbrochen wird.

```
!!! Der Ausdruck während des Assemblierens erfolgt immer !!!
!!! in Großschreibung. !!!
```

Für die Steuerung des Formularvorschubs (maximale Zeilenzahl und/oder "Punktzeilen"), sowie für das Ausdrucken von Kopfzeilen ("Punktzeilen" am Anfang des Assembler-Textes) gelten dieselben Regeln wie für das einfache Ausdrucken eines Textes ohne gleichzeitiges Assemblieren. Die verwendeten "Punktzeilen" werden beim Assemblieren unterdrückt, sie dienen nur zur Drucker-Steuerung. Die Einstellung von Zeilenrand und maximaler Zeilenzahl pro Druck-Seite, sowie die Wahl der Seiten-Numerierung erfolgt mit Hilfe des "Drucker-Menues" (CTRL-D) mit anschließendem EDITOR-Rücksprung.

Beispiel: EDIT-Buffer-Start-Adresse: 6000
 EDIT-Buffer-Stop -Adresse: 6308

```

;-----
;-      BEISPIEL-PROGRAMM:      -
;-      UMWANDLUNG VON KLEIN-   -
;-      IN GROSS-BUCHSTABEN    -
;-----
BUFANF EQU 06000      ; BUFFER-ANFANG
GROSS  EQU 00EE9     ; MAT-85-ROUTINE

      ORG 0E000      ; <-- PROGRAMM-ANFANG

NEXT: LXI H,BUFANF   ; HL AUF BUFFER-ANFANG
      MOV A,M        ; ZEICHEN HOLEN

      CALL GROSS     ; UMWANDELN IN GROSS-
                   ; BUCHSTABEN

      MOV M,A        ; UND WIEDER SPEICHERN
      INX H          ; ZEIGER ERHOEHEN
      ORA A          ; TESTE AUF 00
      JNZ NEXT       ; NAECHSTES ZEICHEN, WENN
                   ; NICHT 00
      RST 1          ; BUFFER-ENDE: "USER"

      LP
  
```

Beim Assemblieren des obigen Programm-Beispiels wird der zugehörige Maschinen-Code im Speicherbereich von E000 bis E00D (hex.) abgelegt:

```

0000      ;-----
0000      ; -      BEISPIEL-PROGRAMM:      -
0000      ; -      UMWANDLUNG VON KLEIN-   -
0000      ; -      IN GROSS-BUCHSTABEN    -
0000      ;-----
0000
0000      BUFANF EQU 06000      ; BUFFER-ANFANG
0000      GROSS  EQU 00EE9     ; MAT-85-ROUTINE
0000
0000      ORG 0E000      ; <-- PROGRAMM-ANFANG
E000
E000 21 0060      NEXT: LXI H,BUFANF   ; HL AUF BUFFER-ANFANG
E003 7E          MOV A,M        ; ZEICHEN HOLEN
E004
E004 CD E90E     CALL GROSS     ; UMWANDELN IN GROSS-
E007            ; BUCHSTABEN
E007
E007 77          MOV M,A        ; UND WIEDER SPEICHERN
E008 23          INX H          ; ZEIGER ERHOEHEN
E009 B7          ORA A          ; TESTE AUF 00
E00A C2 03E0     JNZ NEXT       ; NAECHSTES ZEICHEN, WENN
E00D            ; NICHT 00
E00D CF          RST 1          ; BUFFER-ENDE: "USER"
E00E
E00E            LP
BUFANF 6000
GROSS  0EE9
NEXT   E003
E00E
  
```

Lösung der Übung 1

EDITOR-Aufruf, einfache Texteingabe, Cursor-Steuerung
 Untersuchung des EDIT-Buffers, Neustart, Hilfsmenue-Aufruf

- Zu 1.: Nach dem Aufruf des EDITOR-Programms erscheint der EDITOR-Meldetext.
- Zu 2.: Durch Betätigung der ESC-Taste erfolgt eine Rückkehr zum Betriebssystem MAT 85.
- Zu 3.: Durch die Eingabe von "X" läßt sich das Hilfs-Menue aufrufen und durchblättern (Taste <SPACE>). Mit <CR> erfolgt die Rückkehr zum EDITOR-Meldetext. Wurde das Hilfs-Menue nicht installiert, so erfolgt eine akustische Fehlermeldung. Der EDITOR-Meldetext bleibt auf dabei auf dem Bildschirm erhalten.
- Zu 4.: Nach dem "Kaltstart" wird als Start-Adresse vorgeschlagen:

```
START-ADR = 6000
```

Nach Änderung der Start-Adresse auf 6200 mit anschließendem <CR> oder <SPACE> lautet die Stop-Adresse:

```
START-ADR = 6000 6200 <CR>
```

```
STOP -ADR = 6202
```

Begründung: Bei einem Kaltstart werden an den Anfang des EDIT-Buffers die beiden Steuerbytes "0D 0A", sowie die Ende-Markierung "00" eingeschrieben ==> Stop-Adresse = 6202

- Zu 5.: Wird nach der Eingabe des angegebenen Textes ein "Warmstart" ausgeführt (CTRL-N mit anschließendem <CR> oder <SPACE>), so liegen die Grenzen des EDIT-Buffers bei:

```
START-ADR = 6200
```

```
STOP -ADR = 622F
```

Begründung: Durch die Text-Eingabe hat sich das EDIT-Buffer-Ende entsprechend verschoben (0D 0A, 26 Zeichen der ersten Zeile, 0D 0A, 15 Zeichen der zweiten Zeile, 0D 0A, 00 = 48 (dez.) Zeichen = 30 (hex.) Zeichen ==> Stop-Adresse = 622F)

Zu 6.: Nach der Rückkehr zum Monitorprogramm MAT-85 stellt sich der Inhalt des EDIT-Buffers mit Hilfe des PRINT-KModus wie folgt dar:

```
6200 0D 0A .D .I .E .S .E .S
6208 . .I .S .T . .E .I .N
6210 .E . .T .E .S .T .Z .E
6218 .I .L .E .. 0D 0A .A .A
6220 .A . . . .B .B .B .
6228 . . .C .C .C 0D 0A 00
6230 .
```

Als Steuerbytes zur Zeilentrennung werden benutzt:

Zeilentrennung:

0D

0A

Als Markierungsbyte am Ende des EDIT-Buffers steht:

Ende-Markierung:

00

Zu 7.: Nach einem erneuten Warmstart mit der vorgeschlagenen Start-Adresse 6200 erscheint der vorher eingegebene Text wieder auf dem Bildschirm.

Nach einem "Neustart" (CTRL-N) mit nachfolgendem "Kaltstart" ab Adresse 6200 ist der alte Text "gelöscht". Die Stop-Adresse des EDIT-Buffers liegt bei 6202.

Auch ein nachfolgender "Warmstart" liefert das gleiche Ergebnis.

Zu 8.: Nach der Rückkehr zum MAT-85 läßt sich der Inhalt des EDIT-Buffers mit Hilfe des PRINT-Modus auf dem Bildschirm darstellen:

```
6200 0D 0A 00 .I .E .S .E .S
```

Die EDIT-Buffer-Endemarkierung 00 ist in den Speicherplatz 6202 eingeschrieben worden.

Mit Hilfe des MEMORY-Modus des MAT-85 läßt sich das 00-Byte unter der Adresse 6202 ändern in 20 (hex.). Wird nun erneut ein Warmstart des EDITOR-Programms ausgeführt, so erscheint der alte Text wieder, mit Ausnahme des ersten Zeichens:

```
IESES IST EINE TESTZEILE.<
AAA   BBB   CCC<
```

Das erste Zeichen wurde durch den Hex-Wert 20 (entspricht ASCII-Leerzeichen) überschrieben. Auf diese Weise läßt sich ein schon vorhandener Text im EDIT-Buffer, der versehentlich durch einen Kaltstart "gelöscht" wurde, wieder restaurieren.

Zu 9.: Nach dem Ausschalten des MFA-Systems und nachfolgendem Wieder-Einschalten wird ein "Power-On-Reset" durchgeführt. Als Start-Adresse für den EDIT-Buffer wird daher vorgeschlagen:

```
START-ADR = 6000
```

B. Anpassung des BFZ-EDITORS an andere Terminals

B.1. Terminal-Anpassungs-Tabellen

Sollte bei dem verwendeten MFA-System ein Terminal als Datensicht-, bzw. Eingabestation angeschlossen sein, so ist zusätzlich zur schon vorhandenen Änderung des Betriebsprogramms MAT 85 eine Anpassung des BFZ-EDITOR-Programms an dieses Terminal erforderlich.

Zur einfachen Anpassbarkeit sind innerhalb des EDITOR-Programms Tabellen angelegt, mit deren Hilfe sich die Terminal-spezifischen Parameter verändern lassen. Auf der folgenden Seite sind die Anpassungs-Tabellen für das MFA-Video-Interface dargestellt. Im Anschluß daran finden sich als Beispiel für ein anderes Terminal die Anpassungs-Tabellen für das Terminal "TELEVIDEO".

Um die Terminal-spezifischen Parameter zu verändern, muß das BFZ-EDITOR-Programm in einen freien RAM-Bereich des Arbeitsspeichers kopiert werden. Mit Hilfe des "MEMORY"-Kommandos des Betriebssystems lassen sich die entsprechenden Parameter des verwendeten Terminals in die zugehörigen Tabellenplätze einsetzen. Anschließend kann das EDITOR-Programm mit den veränderten Tabellenwerten in ein neues EPROM programmiert werden.

Die Ermittlung der Terminal-spezifischen Parameter wird im folgenden erläutert.

```
*****
* Tabellen der Steuerbytes für "MFA-Video-Interface" *
*****
```

----- TABELLE 1: BILDSCHIRMFORMAT -----

ADR.	HEX	DEZ	
5004	3D	61	; SPALTEN (= 64 - 3 dez.)
5005	0F	15	; ZEILEN (= 16 - 1 dez.)
5006	20	20000	; ZEITKONSTANTE FUER HOME
5007	4E		

----- TABELLE 2: AUSGABE-WERTE -----

ADRESSE	HEX	
5008	08	; (OUT)-CURSOR LINKS
5009	09	; " " RECHTS
500A	0A	; " " UNTEN
500B	0B	; " " OBEN
500C	0C	; LOESCHE BILDSCHIRM (HOME)
500D	0D	; LOESCHEN ZEILENREST + CARRIAGE RETURN (00 HIER: OHNE BEDEUTUNG)
500E	00	
500F	1D	; (OUT)-CARRIAGE-RETURN OHNE LOESCHEN DES ZEILENRESTS

----- TABELLE 3: EINGABE-WERTE -----

ADRESSE	HEX	
5010	08	; (IN)-CURSOR LINKS
5011	09	; " " RECHTS
5012	0A	; " " UNTEN
5013	0B	; " " OBEN
5014	0D	; (IN)-CARRIAGE-RETURN
5015	15	; CTRL-U = SEITE UNTEN
5016	0F	; CTRL-O = SEITE OBEN
5017	05	; CTRL-E = EINFUEGEN EIN/AUS
5018	12	; CTRL-R = RADIERE ZEICHEN
5019	7F	; DEL = DELETE ZEICHEN
501A	1A	; CTRL-Z = ZEILEN-BEFEHLE
501B	14	; CTRL-T = TABULATOR
501C	03	; CTRL-C = ERWEITERTE CURSOR-BEFEHLE
501D	17	; CTRL-W = CURSOR NAECHSTES WORT
501E	0E	; CTRL-N = NEUSTART
501F	18	; CTRL-X = HILFS-MENUE
5020	01	; CTRL-A = ASSEMBLIEREN
5021	04	; CTRL-D = DRUCKER-MENUE
5022	06	; CTRL-F = FIND-MODUS
5023	02	; CTRL-B = BLOCK-BEFEHLE
5024	13	; CTRL-S = SICHERN
5025	00	; MARKIERUNG: TABELLEN-ENDE

```
*****
* Tabellen der Steuerbytes für Terminal "TELEVIDEO" *
*****
```

----- TABELLE 1: BILDSCHIRMFORMAT -----

ADRESSE	HEX	DEZ	
5004	4D	77	; SPALTEN (= 80 - 3 dez.)
5005	17	23	; ZEILEN (= 24 - 1 dez.)
5006	00	0	; ZEITKONSTANTE FUER HOME
5007	00		

----- TABELLE 2: AUSGABE-WERTE -----

ADRESSE	HEX	
5008	08	; (OUT)-CURSOR LINKS
5009	0C	; " " RECHTS
500A	0A	; " " UNTEN
500B	0B	; " " OBEN
500C	1A	; LOESCHE BILDSCHIRM (HOME)
500D	1B	; LOESCHE ZEILENREST (HIER: ESCAPE- SEQUENCE: "ESC t")
500E	74	
500F	0D	; (OUT)-CARRIAGE-RETURN OHNE LOESCHEN DES ZEILENRESTS

----- TABELLE 3: EINGABE-WERTE -----

ADRESSE	HEX	
5010	08	; (IN)-CURSOR LINKS
5011	0C	; " " RECHTS
5012	16	; " " UNTEN
5013	0B	; " " OBEN
5014	0D	; (IN)-CARRIAGE-RETURN
5015	15	; CTRL-U = SEITE UNTEN
5016	0F	; CTRL-O = SEITE OBEN
5017	05	; CTRL-E = EINFUEGEN EIN/AUS
5018	12	; CTRL-R = RADIERE ZEICHEN
5019	7F	; DEL = DELETE ZEICHEN
501A	1A	; CTRL-Z = ZEILEN-BEFEHLE
501B	09	; TAB = TABULATOR
501C	03	; CTRL-C = ERWEITERTE CURSOR-BEFEHLE
501D	17	; CTRL-W = NAECHSTES WORT
501E	0E	; CTRL-N = NEUSTART
501F	18	; CTRL-X = HILFS-MENUE
5020	01	; CTRL-A = ASSEMBLIEREN
5021	04	; CTRL-D = DRUCKER-MENUE
5022	06	; CTRL-F = FIND-MODUS
5023	02	; CTRL-B = BLOCK-BEFEHLE
5024	13	; CTRL-S = SICHERN
5025	00	; MARKIERUNG: TABELLEN-ENDE

B.2. Ermittlung der Terminal-spezifischen Parameter:

B.2.1. Tabelle 1: Bildschirmformat

Mit Hilfe dieser Tabelle wird das EDITOR-Programm auf das Bildschirmformat des verwendeten Terminals eingestellt (Bildschirmformat: Spalten- und Zeilenzahl).

Die Berechnung der Spalten- und Zeilenzahl für das BFZ-EDITOR-Programm erfolgt nach folgender Vorschrift:

5004 Spalten = max. Bildschirmspalten - 3
 5005 Zeilen = max. Bildschirmzeilen - 1

Für das MFA-Video-Interface ergibt sich somit:

5004	Spalten	=	64	-	3	=	61 (Dez)	=	3D (Hex)
5005	Zeilen	=	16	-	1	=	15 (Dez)	=	0F (HEX)

Für das verwendete Terminal mit einer Bildschirm-Darstellung von

.... Spalten x Zeilen

ergibt sich damit:

5004	Spalten	=	-	3	=	=
5005	Zeilen	=	-	1	=	=

Das MFA-Video-Interface benötigt zur Ausführung des Steuerbefehls "HOME" (Bildschirm löschen, Cursor an linken oberen Bildrand) eine Zeit von ca. 0,21 s. Aus diesem Grund muß nach der Ausgabe des "HOME"-Befehls eine entsprechende Zeitverzögerung gewährleistet werden, bevor das nächste Zeichen ausgegeben wird. Diese Zeitverzögerung, die bei anderen Terminals unter Umständen nicht notwendig ist, läßt sich ebenfalls mit Hilfe der Tabelle 1 ändern.

Die erforderliche Zeitkonstante für die Ausführung von "HOME" muß für das MFA-Video-Interface wie folgt gewählt werden:

ZK-HOME = 20000 (DEZ) = 4E20 (HEX)

Wird für das verwendete Terminal keine Zeitverzögerung benötigt, so kann als Zeitkonstante verwendet werden:

ZK-HOME = 0 (DEZ) = 0000 (HEX)

Für das verwendete Terminal wird als Zeitkonstante benötigt (experimentell ermittelt oder aus Terminal-Handbuch):

5006 (L-Byte) (DEZ)
5007 (H-Byte)	

B.2.2. Tabelle 2: Ausgabewerte

Zur Steuerung der Cursor-Bewegung auf dem Bildschirm werden vom EDITOR-Programm bestimmte Steuer-Bytes zur Datensichtstation ausgegeben. Diese Terminal-spezifischen Ausgabewerte sind in der Tabelle 2 abgelegt.

Als erstes muß ermittelt werden, welche Tastatureingaben für die Steuerung der Cursor-Bewegung, für das Löschen des Zeilenrests, sowie für die "HOME"-Funktion benötigt werden. Dieses kann mit Hilfe des Terminals-Handbuches oder mit dem nachfolgenden Programm 1 geschehen:

```

;---- PROGRAMM 1 ----
                                ORG 0E000
L1: CALL 0FC80                   ;TASTATUR ABFRAGEN
    ANI 7F                       ;BIT 7 WEGMASKIEREN
    CALL 0052                     ;ZEICHEN ZUR DATENSICHTSTATION
    JMP L1
    
```

Als Tastatureingaben können unter Umständen auch "CTRL"-Eingaben oder auch "ESC-Sequenzen" in Frage kommen (z.B. CTRL-L für "HOME" oder "ESC t" (= 1B 74 hex.) für "Löschen des Zeilenrests").

Bei der Ermittlung der erforderlichen Tastatureingabe für die Cursor-Bewegung nach unten ist zu beachten, daß diese Eingabe zu einem "Scrollen" des Bildschirminhalts nach oben führt, wenn sich der Cursor in der untersten Bildschirmzeile befindet.

Tragen Sie in nachfolgende Liste ein, welche Tastatureingabe für die entsprechende Steuerfunktion erforderlich ist:

	Funktion	MFA-Video-Interface		Terminal	
		Tastatur-Eingabe	Steuer-byte	Tastatur-Eingabe	Steuer-byte
5008	Cursor links	: Pfeil links	08
5009	Cursor rechts	: Pfeil rechts	09
500A	Cursor unten (mit "Scrollen")	: Pfeil unten	0A
500B	Cursor oben	: Pfeil oben	0B
500C	HOME	: CTRL-L	0C
500D	Löschen des Zeilenrests, evtl. mit Carriage-Return	: <CR>	0D
500E			NULL	00
500F	Carriage-Return ohne Löschen Zeilenrest	: CTRL-]	1D

Die zu den Tastatureingaben zugehörigen Steuer-Bytes lassen sich anschließend mit Hilfe des folgenden Programms 2 ermitteln und in obige Liste eintragen.

```

;----- PROGRAMM 2 -----
                ORG 0E100

L2:             CALL 0FC80      ; TASTATUR ABFRAGEN
                ANI 7F         ; BIT 7 WEGMASKIEREN
                CALL 0058      ; HEX-WERT AUSGEBEN
                CALL 0070      ; CR-LF AUSGEBEN
                JMP L2
    
```

B.2.3. Tabelle 3: Eingabewerte

Die von der Tastatur bei der Betätigung einer Taste ausgegebenen Steuerbytes (Eingabewerte) sind ebenfalls Terminal-spezifisch. Mit Hilfe des obigen Programms 2 läßt sich ermitteln, welche Steuerbytes bei einer Tastenbetätigung von der Tastatur erzeugt und zum MFA-Computer übertragen werden.

Ermitteln Sie die zu den Tastatureingaben zugehörigen Steuerbytes und tragen Sie diese in nachfolgende Liste ein (dabei darf kein Steuerbyte doppelt vorhanden sein).

Adr.	EDITOR-Funktion	MFA-Video-Interface		Terminal	
		Tastatur-Eingabe	Steuer-byte	Tastatur-Eingabe	Steuer-byte
5010	Cursor links	Pfeil links	08
5011	Cursor rechts	Pfeil rechts	09
5012	Cursor unten	Pfeil unten	0A
5013	Cursor oben	Pfeil oben	0B
5014	Cursor an Zeilenanfang	<CR>	0D
5015	Seite unten	CTRL-U	15
5016	Seite oben	CTRL-O	0F
5017	Einfügen EIN/AUS	CTRL-E	05
5018	Radiere Zeichen	CTRL-R	12
5019	Delete Zeichen	DEL	7F
501A	Zeilen-Befehle	CTRL-Z	1A
501B	Tabulator	CTRL-T	14
501C	Erw. Cursorbew.	CTRL-C	03
501D	Nächstes Wort	CTRL-W	17
501E	Neustart	CTRL-N	0E
501F	Hilfs-Menue	CTRL-X	18
5020	Assemblieren	CTRL-A	01
5021	Drucker-Menue	CTRL-D	04
5022	Find-Modus	CTRL-F	06
5023	Block-Befehle	CTRL-B	02
5024	Sichern	CTRL-S	13

C. Aufbau des Hilfs-Menues

Um spätere Änderungen des Hilfs-Menues durch den Anwender zu ermöglichen, wird im folgenden der Aufbau des Hilfs-Menues erläutert.

Das Hilfs-Menue kann an jede beliebige freie Stelle des Arbeitsspeichers geladen werden. Ist das verwendete MFA-System nicht mit dem "BFZ-MINI-DOS" bestückt, so läßt sich das Hilfs-Menue als EPROM-Version ab der Speicherplatzadresse 4000 (HEX) installieren, und der BFZ-EDITOR kann aus der Kommando-Ebene des MAT 85 heraus mit Hilfe der Tastatur-Eingabe "E" (= EDITOR) gestartet werden.

Dazu ist in den ersten drei Speicherplätzen des Hilfs-Menues ein Sprungbefehl zur Anfangsadresse des EDITOR-Programms (5000 HEX) abgelegt:

4000	C3	
4001	00	JMP 5000
4002	50	

(Wird innerhalb der Kommando-Ebene des Betriebssystems MAT 85 eine Tastatureingabe vorgenommen, die nicht in der Kommando-Liste des MAT 85 enthalten ist, so überprüft das Betriebsprogramm, ob in dem Speicherplatz 4000 (HEX) ein Sprungbefehl ("C3") enthalten ist. Ist dies der Fall, so wird der Sprungbefehl ausgeführt, das EDITOR-Programm also gestartet. Dieses wird nur dann weiter abgearbeitet, wenn die Tastatureingabe ein "E" war. Ansonsten erfolgt ein Rücksprung zum Betriebssystem.)

In den nachfolgenden drei Speicherplätzen des Hilfs-Menues steht die sogenannte "Hilfs-Menue-Kennung":

4003	DD
4004	ED
4005	DD

Aufgrund dieser Kennzeichnung findet das EDITOR-Programm das Hilfs-Menue an jeder beliebigen Stelle des Arbeitsspeichers. Bei einem Aufruf des Hilfs-Menues vom EDITOR aus wird nach dem Auffinden der Kenn-Bytes der ab der nachfolgenden Adresse gespeicherte Text auf dem Bildschirm dargestellt.

Das Hilfs-Menue hat demnach folgenden Aufbau:

```

C3
00
50

DD
ED
DD
-----
-   Hilfs-Menue-Text   -       1.Seite
-   -----
00
-----
-   Hilfs-Menue-Text   -       2.Seite
-   -----
00
-----
-   Hilfs-Menue-Text   -       3.Seite
-   -----
00
00

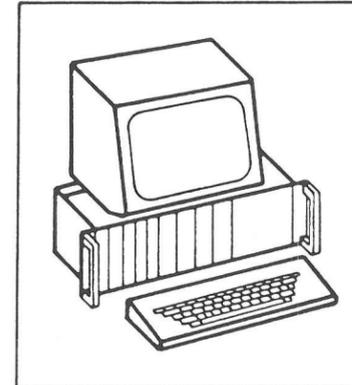
```

Findet das EDITOR-Programm bei der Darstellung einer Hilfs-Menue-Seite ein "00"-Byte, so wird die Bildschirmausgabe unterbrochen und die Tastatur abgefragt. Durch die Eingabe von <SPACE> läßt sich die nachfolgende Seite darstellen, bei der Eingabe von <CR> erfolgt eine Rückkehr zum EDITOR. Das Ende des Hilfs-Menues ist mit zwei "00"-Bytes gekennzeichnet. Hier erfolgt bei der Eingabe von <SPACE> eine erneute Darstellung des Hilfsmenues ab der ersten Hilfs-Menue-Seite, die Eingabe von <CR> führt auch hier zu einem Rücksprung in den EDIT-Modus.

Die Veränderung der einzelnen Hilfs-Menue-Seiten kann mit Hilfe des BFZ-EDITORS erfolgen. Die einzelnen Seiten des veränderten Hilfs-Menues müssen anschließend gemäß dem oben beschriebenen Aufbau zusammengefügt werden.

Es bleibt dem Anwender überlassen, unter Umständen auch andere Informationen innerhalb des Hilfs-Menues zu speichern, die dann bei Bedarf jederzeit vom Bediener abrufbar sind.

FACHPRAKTISCHE ÜBUNG MIKROCOMPUTER-TECHNIK



64-KByte-RAM

BFZ/MFA 3.3.

