MOPPEL
System-Software

Editor-
Handhabung

Modulares Prozessor-Programm zum Entwickeln und Lernen

® Eingetragenes Warenzeichen

hms

Inhaltsverzeichnis

hms

```
MOPPEL-Video-Monitor V 7.6
Copyright (C) hms'86

M: Memory
O: Outward
P: Peripheral
S: Start

M>o

A: Assembler
B: BASIC
D: Disassembler
E: Editor
F: Floppy
P: Prommer
X: Ext.2000h

Mo>e

MOPPEL-Editor V 6.6
Copyright (C) hms'86

 A: Assembler
 C: Clear
 F: Floppy
 G: Go to
 I: Insert
 L: List
 M: Monitor
 S: Setup RAM
 X: Extend

E>
```

Ø. Installation und Aufruf

Das ROM-residente Editor-Programm ist Bestandteil des EPROMs 'blau'; es wird wahlweise im 4-K-EPROM (2732) oder 8-K-EPROM (2764) geliefert und auf Platz #6 der 89er-Speicherkarte (Platz #7 der 87er-Speicherkarte) eingesetzt (Brücke 'Type 64/32' entsprechend setzen).

Erst der Editor versetzt Sie in die Lage, mit Ihrem Mikrocomputer auch Schreibmaschinen-Funktionen auszuführen, d.h. Klartexte erstellen und jederzeit beliebig verändern zu können. Diese Funktion geschieht mit drei unterschiedlichen Zielrichtungen:

1. Erstellen und Modifizieren von Assembler-Quellprogrammen.
Der Editor dient in diesem Fall dazu, als Handwerkszeug bei der Programmentwicklung die Texte (im Assembler-Code) zu erstellen, die anschließend vom Assembler-Programm in ein lauffähiges Mikrocomputer-Programm (im Maschinen-Code) umgesetzt werden. Dies ist die eigentliche Aufgabe des Editors, und dementsprechend sind seine Leistungsmerkmale organisiert (vgl. Fallbeispiel neben Abschnitt 3).

2. Erstellen und Modifizieren von BASIC-Programmen.
Sie können mit Hilfe des Editors komplette BASIC-Programme erstellen (einschließlich BASIC-Zeilennummern) und diese an den ROM-residenten BASIC-Interpreter zur Ausführung übergeben; auf diese Weise können Sie (im Editor) Ihr BASIC-Programm modifizieren, obwohl der BASIC-Interpreter selbst nicht über Editor-Funktionen verfügt (vgl. BASIC-Handhabung).

3. Erstellen von Texten für die Daten-Fernübertragung.
Sie können umfangreiche Texte vorbereiten, die Sie anschließend z.B. mit Hilfe der Monitor-Terminal-Funktion per Telefonleitung übertragen (vgl. System-Handbuch).

Vom Monitor-Grundmenü aus erfolgt der Editor-Aufruf über das OUTWARD-Menü (Eingabe von 'o' ohne 'Return'), gefolgt von 'e' (wiederum ohne 'Return'). Der Editor-Aufruf kann außerdem direkt aus dem Assembler- oder Floppy-Menü heraus erfolgen. Sämtliche Editor-Anweisungen können in Klein- oder Großschreibung erfolgen; die danach eingegebenen Parameter können, müssen aber nicht durch einen Leerschritt ('Blank') vom Kommando-Buchstaben getrennt sein. Mehrere Parameter sind wahlweise durch Komma oder Punkt voneinander zu trennen. Aus Gründen der Übersichtlichkeit ist das Kommando in den folgenden Beschreibungen stets ein Großbuchstabe, der von den Parametern durch einen Leerschritt abgesetzt ist.

Für den praktischen Einsatz ist es sehr vorteilhaft, daß der Drucker jederzeit (auch bei laufenden Listings!) zu- oder abgeschaltet werden kann; zur Drucker-Ansteuerung ist das Universal-Interface erforderlich.

```
E>i1

      0001  test:  lxi  h,1234h   ;Beispiel-Eingabezeile.........

E>m

M>m

C: Copy
F: Fill
K: Kill
L: List
M: Memory
R: Revise
T: Text
X: Exchange

Mm>l 9000,902c
9000   74 65 73 74   3A 20 20 6C     78 69 20 20   68 2C 31 32
9010   33 34 68 20   20 20 2A 42     65 69 73 70   69 65 6C 2D
9020   45 69 6E 67   61 62 65 7A     65 69 6C 65   0D

Mm>t 9000,0d
test:  lxi  h,1234h   ;Beispiel-Eingabezeile

Mm>
```

1. Speicher-Organisation

Der Editor benutzt ausschlieBlich den RAM-Bereich von 9000...EFFFh als Text-Buffer, d.h. in diesem 24-K-Speicherraum legt er seine Texte ab. Der hierfür reservierte Platz ist ausreichend für über 24000 ASCII-Zeichen, was etwa einem Dutzend vollgeschriebener DIN-A4-Seiten entspricht oder rund 1200 Assembler-Zeilen mit normaler Kommentierung; dies ist der Umfang für ein ca.2 KByte langes Maschinenprogramm (entspricht etwa dem Umfang der Floppy- oder EPROM-Utilities).

Sämtliche Eingaben werden Zeilen-orientiert abgelegt und verarbeitet, d.h. jede Zeile wird bei der Eingabe durch einen Wagenrücklauf (Carriage Return) abgeschlossen und intern mit dem entsprechenden ASCII-Code 0Dh beendet (Beispiel links: Eingabe eine Zeile im Editor und anschlieBendes Listen des Speicherinhalts -hexadezimal und als ASCII-Zeichenfolge- im Monitor).

Zur Orientierung numeriert der Editor sämtliche Zeilen fortlaufend durch; die (dezimalen) Zeilennummern werden nicht mit ins Listing eingetragen, sondern bei jeder Ein- und Ausgabe neu errechnet (durch Zählen der Zeilen-End-Codes 0Dh). Die Ausgabe der Zeilennummern ist abschaltbar (vgl. Abschnitt 12 'Extend').

Zur Aufbereitung der Zeilennummern sind zwei Unterprogramme vorgesehen, die Sie bei Bedarf in eigene Programme einbauen können:

700Ah BINOUT: Den dezimalen Inhalt des Registerpaars H&L hexadezimal umsetzen und ausgeben. Beispiel: H&L mit 1000 laden und 700Ah aufrufen führt zur Ausgabe von 03E8 (=1000d)

700Dh LINOUT: Den hexadezimalen Inhalt des Registerpaars H&L dezimal umsetzen und ausgeben. Beispiel: H&L mit 0064 laden und 700Dh aufrufen führt zur Ausgabe von 0100 (=0064h)

Hinweis: Keins der Unterprogramme führt einen Test auf zulässige Zeichen durch.

```
MOPPEL-Editor V 6.6
Copyright (C) hms'86

 A: Assembler
 C: Clear
 F: Floppy
 G: Go to
 I: Insert
 L: List
 M: Monitor
 S: Setup RAM
 X: Extend                                  Error # 79

E>c                                         Error # 70

E>y                                         Error # 71

E>g 1e                                      Error # 72

E>l 8888

                                            Error # 75

E>l 7,6

                                            Error # 76

E>x 0,22,88                                 Error # 77

E>s 123,124                                 Error # 78

E>
```

2. Fehlermeldungen

Der Editor erkennt und meldet folgende Fehler:

Error # 70: Nulleingabe unzulässig; beim Löschen muß mindestens eine Zeilennummer angegeben werden.

Error # 71: Unzulässige Eingabe (entstammt nicht dem Menü).

Error # 72: Parameter-Fehler (keine Dezimalzahl; sämtliche Zeilennummern werden dezimal verabeitet).

Error # 75: Zeilennummer existiert nicht; die angegebene (oder beim Löschen bzw. Listen errechnete) Zeilennummer ist zu groß.

Error # 76: Parameter-Fehler (der zweite Parameter ist größer als der erste).

Error # 77: Vorgaben für die Console-Einstellung fehlerhaft.

Error # 78: Parameter-Fehler (beide Parameter sind bei der RAM-Initialisierung ungleich).

Error # 79: RAM ist nicht initialisiert (dazu wird nur der Inhalt der ersten RAM-Zelle des Text-Buffers ausgewertet; die Fehlermeldung erfolgt, wenn in 9000h nicht der ASCII-Code 2Ah für '*' steht).

Hinweis: Die Fehlermeldung 'Error # 79' kommt jedesmal vor der Editor-Bereitmeldung (Prompt 'E>'), wenn am Anfang des Text-Buffers kein '*' steht; um dies zu unterdrücken, brauchen Sie vor Zeile 1 nur eine Leerzeile einzufügen, die intern mit einem Sternchen aufgefüllt wird. Nach Fertigstellen des Textes löschen Sie dann die Zeile 1 einfach wieder.

hms

Fallbeispiel: Anhand eines einfachen Programmbeispiels soll die interaktive Arbeitsweise veranschaulicht werden, die sich bei der Software-Entwicklung zwischen Editor und Assembler abspielt. Sie finden hier (gleichlautend in den Abschnitten 3 der Assembler- und der Editor-Handhabung) die schematische Vorgehensweise zusammengestellt. Wenn Sie mit dem Einsatz von Editor und Assembler noch nicht vollkommen vertraut sind, sollten Sie sich die Mühe machen, die einzelnen Schritte durch entsprechendes Nachschlagen in der Software-Beschreibung nachzuvollziehen!

1. Aufgabenstellung:
Die fest im Monitor programmierte Funktionstaste FCT+V kopiert den Video-Speicher von 3000...377Fh um ans obere Ende des RAM-Bereichs (nach F800...FF7Fh) und schließt diese Daten mit der End-Kennzeichnung 00h ab. Es soll ein Programm PRINT geschrieben werden (ab 2C40h im Monitor-RAM), das den Speicher-Inhalt ab F800h wieder ausgibt (Darstellung der zuvor angefertigten Bildschirm-Momentaufnahme).

2. Programmentwurf (vgl. Editor, Abschnitt 9 'List'):
Nach dem Initialisieren des RAMs (Editor-Setup) werden ORG und OFS neu definiert; die verwendeten Label VIDBUF und CO müssen ebenfalls im Editor definiert werden, anschließend folgt die Programm-Eingabe.

3. Assembler-Testlauf (vgl. Assembler, Abschnitt 5 'Pass#1'):
Das Quellprogramm wird erstmals an den Assembler übergeben, um grobe Fehler aufzuspüren; der erste Durchlauf von Pass #1 liefert sofort die Fehlermeldung 'Error # 64 in Line 14', d.h. in Zeile 14 liegt ein Syntax-Fehler vor.

4. Editor-Korrekturlauf (vgl. Editor, Abschnitt 7 'Go to'):
Der Absolutadresse 0F800 fehlt das angehängte 'h'!

5. Assembler-Testlauf (vgl. Assembler, Abschnitt 5 'Pass#1'):
Das Quellprogramm wird erneut durchlaufen; wieder endeckt der Assembler im Pass #1 einen Fehler: Das Label PRTLOP ist nicht definiert worden (es wird in der Null-Liste am Ende von Pass #1 ausgeworfen).

6. Editor-Korrekturlauf (vgl. Editor, Abschnitt 7 'Go to'):
Sprungziel PRTLOP in Zeile 18 nachtragen (mit CTL+Q einleiten!).

7. Assembler-Durchläufe und Programmtest (vgl. Assembler, Abschn.5):
Pass #1 und Pass #2 liefern keine Fehlermeldung mehr; Programm im Monitor testen ('GO 2C40' im Monitor-START-Menü): Endabfrage fehlt!

8. Editor-Korrekturlauf (vgl. Editor, Abschnitt 8 'Insert'):
Endabfrage vor Zeile 20 einfügen (mit Aussprung aus PRINT -> 1003h).

9. Assembler-Durchläufe und Programmtest (vgl. Assembler, Abschn.5):
Pass #1 und Pass #2 liefern keine Fehlermeldung, und der Testlauf im Monitor funktioniert (Paralleldruck manuell einschalten).

10. Dokumentation (vgl. Editor, Abschn.5 und Assembler, Abschn.7):
Listing bei Bedarf äußerlich überarbeiten (z.B. überflüssige Zeilen löschen), Assembler-Durchlauf Pass #3 starten und Symboltabelle ausgeben (alles mit Paralleldruck!); Programm auf Diskette sichern!

3. Zeilen- und Zeichen-Eingabe

Bei der Neueingabe von Texten unterscheidet der Editor zwei verschiedene Betriebsarten:

Zeilen-Eingabe (Insert-Mode, vgl. Abschnitt 8): Es werden in den Text so lange neue Zeilen eingefügt (mit Verschieben der vorhandenen nach hinten), bis der Insert-Mode wieder verlassen wird (durch 'Escape' bzw. CTL+C oder durch Umschalten auf Zeichen-Eingabe per CTL+F). Bereits eingegebene Zeichen werden (nach Zurückbewegen des Cursors) überschrieben, wenn Sie an derselben Stelle eine Neueingabe vornehmen. Der Insert-Mode ist durch den vollen Cursor erkennbar (volle Zeichenhöhe).

Zeichen-Eingabe (Go-to-Mode, vgl. Abschnitt 7): Es werden in eine bestehende Textzeile so lange neue Zeichen eingefügt (mit Verschieben der vorhandenen nach rechts), bis der Go-to-Mode wieder verlassen wird (durch 'Escape' bzw. CTL+C oder durch Umschalten auf Zeilen-Eingabe per CTL+A). Bereits eingegebene Zeichen werden bei Neueingaben nach rechts verschoben. Der Go-to-Mode ist durch den halben Cursor erkennbar (halbe Zeichenhöhe).

In beiden Betriebsarten zeigen Ihnen die bis zum Zeilenende ausgegebenen Punkte (Leerzeichen) den in der jeweiligen Zeile zur Verfügung stehenden Platz an; diese Punkte werden im Text-Buffer nicht mit abgespeichert. Alle Eingaben, die Sie in der Zeilen- oder Zeichen-Eingabe vornehmen (und die auf dem Bildschirm erscheinen), werden unmittelbar in den Text-Buffer übertragen, d.h. hierzu brauchen Sie nicht explizit die Return-Taste zu betätigen.

Mit der Delete-Taste DEL können Sie Falscheingaben korrigieren: Steht der Cursor auf einem Textzeichen, wird dies durch DEL gelöscht, und die rechts daneben stehenden Zeichen werden nach links nachgerückt. Steht der Cursor auf einem Leerzeichen (Punkt), wird er durch DEL nur nach links bewegt.

Drei weitere Lösch-Funktionen sind außerdem vorgesehen: SHIFT+DEL löscht die gesamte Zeile und füllt sie mit Leerzeichen auf; CTL+Q löscht die sieben links vom Cursor stehenden Zeichen und rückt den Zeilenrest nach links nach (Label löschen mit anschließender Neueingabe); CTL+W löscht ebenfalls die sieben links vom Cursor stehenden Zeichen, läßt den Zeilenrest aber am alten Platz (Label löschen ohne anschließende Neueingabe).

Das Verschieben des Cursors erfolgt außer mit den Pfeiltasten durch CTL+D (rechts) bzw. CTL+S (links); CTL+Y bringt den Cursor an den linken Zeilenrand, während 'Cursor abwärts' (=CTL+X) eine Zeilen-Fortschaltung bewirkt (genauso wie 'Return'). 'Cursor aufwärts' (=CTL+E) veranlaßt eine Zeilen-Rückschaltung mit Übergang in den Go-to-Mode.

Die Tabulator-Taste TAB rückt auf die Spalten 7, 12 oder 22 vor bzw. rechts von Spalte 22 bis zum ersten Leerzeichen (Punkt); übersprungene Leerzeichen werden durch Blanks ersetzt, Textzeichen bleiben unverändert erhalten.

hms

```
MOPPEL-Editor V 6.6
Copyright (C) hms'86

 A: Assembler
 C: Clear
 F: Floppy
 G: Go to
 I: Insert
 L: List
 M: Monitor
 S: Setup RAM
 X: Extend

E>a

MOPPEL-Assembler V 8.6
Copyright (C) hms'86

 Ø/1/2/3: Pass #
 D: Disassembler
 E: Editor
 F: Floppy
 H: HEX-Mode
 I: In/Out on/off
 M: Monitor
 O: OCT-Mode
 S: Symbol Table

A>
```

4. A: Assembler-Aufruf

Format: E>A (ohne 'Return')

Funktion: Verlassen des Editor-Menüs und direkter Sprung zum Assembler (vgl. separate Beschreibung im Rahmen eines eigenen Handbuchs).

Zwischen dem Editor, Assembler und den Floppy-Routinen können Sie ohne den Umweg über den Monitor direkt hin- und herspringen, um die rekusiven Aufrufe dieser Programmteile bei der Software-Entwicklung zu vereinfachen.

```
E>l 17

      0017  print: lxi   h,vidbuf  *H&L: Text-Pointer
      0018  prtlop:mov   c,m       *Text-Zeichen ins Reg C holen
      0019         call  co        *und ausgeben
      0020         mov   a,c       *Text-Zeichen ins Reg A
      0021         cpi   0h        *Nullabfrage
      0022         jz    1003h     *ja: Monitor-Warmstart
      0023         inx   h         *Text-Pointer erhöhen
      0024         jmp   prtlop    *Schleifendurchlauf
      0025  *
      0026  *
      0027  *
      0028  *
      0029  *
      0030  *
      0031  *
      0032  end

E>c 25,30

      0024         jmp   prtlop    *Schleifendurchlauf
      0025  *

E>
```

5. C: Clear (Zeilen löschen)

Format: E>C aaaa,eeee<Ret> oder E>C aaaa<Ret>

Funktion: Löschen der Zeile(n) aaaa...eeee (jeweils einschließlich).

Zur Kontrolle werden nach erfolgtem Löschen die Zeilen vor und hinter dem gelöschten Block gelistet (Zeilen-Nummern aaaa-1 und eeee+1). Beispiel links: Nach dem Löschen der Zeilen 25...30 wird die alte Zeile 31 mit 25 neu numeriert (durch Nachschieben) und zusammen mit der unveränderten Nummer 24 gelistet.

Hinweis: Beim Löschen wird der Text-Buffer ab Adresse EFFFh nach unten verschoben, und der Speicherplatz vom Textende bis zur Buffer-Obergrenze EFFFh wird mit 00h vollgeschrieben.

Beim Löschen am Textende (bei der jeweils höchsten Zeilennummer) kann keine nachfolgende Zeile mehr gelistet werden; unabhängig vom korrekten Löschvorgang erfolgt in diesem Fall die Fehlermeldung 'Error # 75' (vgl. Abschnitt 2).

```
MOPPEL-Editor V 6.6
Copyright (C) hms'86

 A: Assembler
 C: Clear
 F: Floppy
 G: Go to
 I: Insert
 L: List
 M: Monitor
 S: Setup RAM
 X: Extend

E>f

Drv.A:xx     Stp/Sid:xx       Dens:xD
Drv.B:xx     Tracks: xxd      Secs:xxd
Drv.C:xx     RAM-Beg:xxh      Byts:xxh
Drv.D:xx     RAM-End:xxh      Cmd:xxh
-------------------------------------------------------------------------------

RAMb,RAMe,Tracks>90,EF,40

MOPPEL-FDC-Utilities V 10.6
Copyright (C) hms'86

 A/E: Assembler/Editor
 B/D: Batch out/Disk in
 C/F: Copy/Format
 R/W: Read/Write
  M:  Monitor
```

6. F: Floppy-Aufruf

Format: E>F (ohne 'Return')

Funktion: Verlassen des Editor-Menüs und direkter Sprung zu den Floppy-Disk-Utilities (vgl. separate Beschreibung im Rahmen eines eigenen Handbuchs).

Zwischen dem Editor, Assembler und den Floppy-Routinen können Sie ohne den Umweg über den Monitor direkt hin- und herspringen, um die rekusiven Aufrufe dieser Programmteile bei der Software-Entwicklung zu vereinfachen.

hms

```
E>g 14

     0011  org 2c40h   *freier RAM-Bereich hinter FCT-Buffer
     0012  ofs 3c40h   *Objekt-Code nach F000+3c40=2c40h
     0013  *
     0014  vidbuf 0f800h   *Video-Buffer für FCT+V.............

E>g 18

     0015  co     49h      *Monitor-Ausgabe-Routine
     0016  *
     0017  print: lxi  h,vidbuf  *H&L: Text-Pointer
     0018  mov  c,m       *Text-Zeichen ins Reg C holen.........

E>g 18

     0015  co     49h      *Monitor-Ausgabe-Routine
     0016  *
     0017  print: lxi  h,vidbuf  *H&L: Text-Pointer
     0018  prtlop:mov  c,m       *Text-Zeichen ins Reg C holen..
```

7. G: Go to (Zeichen einfügen)

Format: E>G aaaa<Ret>

Funktion: Sprung zur Zeilennummer aaaa und Einstellen der Betriebsart 'Zeichen einfügen'; die Kennzeichnung erfolgt durch einen halbhohen Cursor (halbe Zeichenhöhe).

Folgende Editier-Funktionen sind hierbei möglich (vorhandene Textzeichen werden bei jeder Neueingabe nach rechts verschoben; vgl. Abschnitt 3 'Zeilen- und Zeichen-Eingabe'):

CTL+S: Cursor links
CTL+D: Cursor rechts
CTL+Y: Cursor an linken Zeilenrand
TAB: Cursor auf Spalte 7, 12 oder 22 bzw. bis zum nächsten Leerzeichen (=Punkt) weiterbewegen

CTL+Q: Label löschen (sieben Zeichen links vom Cursor), Zeilenrest nach links nachrücken; Beispiel links: Mit CTL+Q wurde die gesamte Zeile um sieben Zeichen nach links verschoben (Mitte); nach Eingabe des neuen Labels 'PRTLOP' steht der Zeilenrest wieder am alten Platz (unten).
CTL+W: Label löschen (sieben Zeichen links vom Cursor), Zeilenrest am alten Platz belassen
DEL: Zeichen löschen, Zeilenrest nachrücken (bei Textzeichen) oder Cursor links (bei Leerzeichen = Punkt)
SHIFT+DEL: Zeile löschen und mit Leerzeichen (=Punkten) auffüllen

CTL+E: Zeile zurückschalten
CTL+X oder Return: Zeile weiterschalten

CTL+C oder Escape: Got-to-Mode verlassen und zum Editor-Grundmenü zurückkehren

Bei Zurückweisung einer Eingabe ertönt das akustische Signal ('Bell'), sofern Ihre Mikrocomputer-Hardware entsprechend ausgebaut ist; Zurückweisungen erfolgen z.B., wenn der Cursor bereits linksbündig steht und Sie die Taste 'Cursor links' betätigen.

Hinweis: Wenn Sie nach dem Löschen mit SHIFT+DEL ohne weitere Texteingabe weiterschalten zur nächsten Zeile, füllt der Editor die so entstandene Leerzeile mit einem Sternchen auf. Soll die Leerzeile als solche bestehen bleiben, müssen Sie nach dem Löschen mindestens ein Leerzeichen ('Blank') eingeben, ehe Sie zur folgenden Zeile weiterschalten.

hms

```
E>l 17,21

0017  print: lxi  h,vidbuf  *H&L: Text-Pointer
0018  prtlop:mov  c,m       *Text-Zeichen ins Reg C holen
0019         call co         *und ausgeben
0020         inx  h          *Text-Pointer erhöhen
0021         jmp  prtlop     *Schleifendurchlauf

E>i 20

0017  print: lxi  h,vidbuf  *H&L: Text-Pointer
0018  prtlop:mov  c,m       *Text-Zeichen ins Reg C holen
0019         call co         *und ausgeben
0020         mov  a,c        *Text-Zeichen ins Reg A  <<<neu
0021         cpi  0h         *Nullabfrage             <<<neu
0022         jz   1003h      *ja: Monitor-Warmstart   <<<neu

E>
```

8. I: Insert (Zeilen einfügen)

Format: E>I aaaa<Ret>

Funktion: Vor der aktuellen Zeilennummer aaaa neue Zeilen einfügen (Einstellen der Betriebsart 'Zeilen einfügen'). Die Kennzeichnung erfolgt durch einen hohen Cursor (volle Zeichenhöhe).

Beispiel links: Durch 'I 20' wurde das Einfügen der drei mit 'neu' gekennzeichneten Zeilen eingeleitet; alle folgenden Zeilen werden automatisch neu numeriert.

Folgende Editier-Funktionen sind im Insert-Mode möglich (vorhandene Textzeichen werden ohne Verschieben überschrieben; vgl. Abschnitt 3 'Zeilen- und Zeichen-Eingabe'):

CTL+S: Cursor links
CTL+D: Cursor rechts
CTL+Y: Cursor an linken Zeilenrand
TAB: Cursor auf Spalte 7, 12 oder 22 bzw. bis zum nächsten Leerzeichen (=Punkt) weiterbewegen

CTL+Q: Label löschen (sieben Zeichen links vom Cursor), Zeilenrest nach links nachrücken
CTL+W: Label löschen (sieben Zeichen links vom Cursor), Zeilenrest am alten Platz belassen
DEL: Zeichen löschen, Zeilenrest nachrücken (bei Textzeichen) oder Cursor links (bei Leerzeichen = Punkt)
SHIFT+DEL: Zeile löschen und mit Leerzeichen (=Punkten) auffüllen

CTL+E: Zeile zurückschalten und Umschalten zur Betriebsart 'Zeichen einfügen' (vgl. Abschnitt 7)
CTL+X = Return: Zeile weiterschalten und Fortsetzen der Betriebsart 'Zeilen einfügen'

CTL+C = Escape: Insert-Mode verlassen und zum Editor-Grundmenü zurückkehren

Bei Zurückweisung einer Eingabe ertönt das akustische Signal ('Bell'), sofern Ihre Mikrocomputer-Hardware entsprechend ausgebaut ist; Zurückweisungen erfolgen z.B., wenn der Cursor bereits linksbündig steht und Sie die Taste 'Cursor links' betätigen.

hms

```
MOPPEL-Editor V 6.6
Copyright (C) hms'86

 A: Assembler
 C: Clear
 F: Floppy
 G: Go to
 I: Insert
 L: List
 M: Monitor
 S: Setup RAM
 X: Extend

E>l 17,19

      0017  print: lxi  h,vidbuf  *H&L: Text-Pointer
      0018         mov  c,m       *Text-Zeichen ins Reg C holen
      0019         call co        *und ausgeben


E>l:0029/A77F

      0001  ***************************************************
      0002  *                                                 *
      0003  *      PRINT  Video-Buffer ab F800h ausdrucken    *
      0004  *                                                 *
      0005  ***************************************************
      0006  *
      0007  *   Stand:   Fr,01.05.87;12:28:28h
      0008  *
      0009  *   Druck:   Mo,04.05.87;16:17:17h
      0010  *
      0011  org 2c40h   *freier RAM-Bereich hinter FCT-Buffer
      0012  ofs 3c40h   *Objekt-Code nach F000+3c40=2c40h
      0013  *
      0014  vidbuf 0f800    *Video-Buffer für FCT+V
      0015  co     49h       *Monitor-Ausgabe-Routine
      0016  *
      0017  print: lxi  h,vidbuf  *H&L: Text-Pointer
      0018         mov  c,m       *Text-Zeichen ins Reg C holen
      0019         call co        *und ausgeben
      0020         inx  h         *Text-Pointer erhöhen
      0021         jmp  prtlop    *Schleifendurchlauf
      0022  *
      0023  *
      0024  *
      0025  *
      0026  *
      0027  *
      0028  *
      0029  end


E>
```

9. L: List (Zeilen listen)

Format: E>L<Ret> oder E>L aaaa<Ret> oder E>L aaaa,eeee<Ret>

Funktion: 1. Gesamten Text-Buffer listen (mit Angabe des Füllgrades und der höchsten Zeilennummer (Beispiel links, unten: Die höchste vorkommende Zeilennummer ist 0029, und der Text-Buffer ist bis zur Adresse A77Fh gefüllt).

2. Text-Buffer ab Zeilennummer aaaa listen (bis zur höchsten Zeilennummer, wenn das Listen vorher nicht gestoppt und abgebrochen wird; s.u.).

3. Zeilennummern aaaa...eeee listen (Beispiel links, Mitte)

Das Listen kann jederzeit gestoppt und wieder fortgesetzt werden, indem die überbreite Leertaste ('Blank') betätigt wird. Bei angehaltenem Listing läßt sich der Drucker zu- oder wieder abschalten (durch CTL+7...CTL+4). Mit 'Escape' bzw. CTL+C wird das Listen abgebrochen, und der Editor meldet sich mit seinem Bereitzeichen (Prompt).

hms

```
MOPPEL-Editor V 6.6
Copyright (C) hms'86

 A: Assembler
 C: Clear
 F: Floppy
 G: Go to
 I: Insert
 L: List
 M: Monitor
 S: Setup RAM
 X: Extend

E>m

M>
```

10. M: Monitor-Rücksprung

Format: E>M (ohne 'Return')

Funktion: Verlassen des Editor-Menüs und direkter Rücksprung zum Monitor (vgl. separate Beschreibung im Rahmen des System-Handbuchs).

Vom Monitor aus können Sie den Editor, den Assembler und die Floppy-Routinen über das OUTWARD-Menü aufrufen. Zwischen Editor, Assembler und Floppy-Utilities können Sie ohne den Umweg über den Monitor hin- und herspringen, um bei rekursiven Aufrufen dieser Programmteile die Software-Entwicklung zu vereinfachen.

hms

```
MOPPEL-Editor V 6.6
Copyright (C) hms'86

 A: Assembler
 C: Clear
 F: Floppy
 G: Go to
 I: Insert
 L: List
 M: Monitor
 S: Setup RAM
 X: Extend                                  Error # 79

E>s 456,456


E>s:0015/914C

      0001  ************************************************************
      0002  *                                                          *
      0003  *                                                          *
      0004  *                                                          *
      0005  ************************************************************
      0006  *
      0007  *   Stand:
      0008  *
      0009  *   Druck:   Fr,01.05.87;11:45:38h
      0010  *
      0011  org 0f000h
      0012  ofs 0h
      0013  *
      0014  *
      0015  end


E>
```

11. S: Setup RAM (Text-Buffer initialisieren)

Format: E>S aaaa,aaaa<Ret>

Funktion: Löschen des gesamten Text-Buffers im Bereich 9000... EFFFh und Einfügen der Assembler-Kopfzeilen; am Ende dieses Rumpf-Textes wird ein Byte 03h (=ETX) eingefügt, das bei der Verarbeitung von Klartexten das Textende markiert.

Die in Zeile 3 vorgesehene Programm-Kennzeichnung (Name und Versionsnummer) wird bei der Dokumentation von Assembler-Programmen (im Pass # 3) auf jeder neuen Druckseite mit ausgedruckt.

Wichtiger Hinweis: Vor Eingabe eines neuen Textes, gleichgültig, ob es sich um einen Klartext, ein BASIC-Listing oder ein Assembler-Quellprogramm handelt, soll die Setup-Funktion aufgerufen werden; nur so ist sichergestellt, daß alle folgenden Editor-Anweisungen auch ordnungsgemäß ausgeführt werden.

Wenn Sie die Assembler-Kopfzeilen nicht benötigen, löschen Sie sie einfach wieder (vgl. Abschnitt 5 'Clear'). Ansonsten beginnen Sie die Eingabe von Assembler-Quellprogrammen mit 'Insert 14', nachdem Sie ORG und OFS definiert und hinter 'Stand' das Datum eingetragen haben (durch FCT+U). An der Stelle 'Druck' wird bei jedem Listen automatisch die Echtzeit-Uhr ausgelesen und angezeigt; auf diese Weise erhalten Sie eine stets aktuelle Dokumentation des letzten Entwicklungsstandes und Zeitpunkt des Ausdrucks.

Zur Sicherung gegen unbeabsichtigtes Löschen müssen zusammen mit dem Kommando-Buchstaben die beiden (durch Komma oder Punkt getrennten) Parameter aaaa eingegeben werden; aaaa kann eine ein- bis vierstellige Dezimalzahl sein, wichtig ist nur, daß sie zweimal gleichlautend eingegeben wird.

hms

```
E>g10

      0007  *
      0008  *
      0009  *
      0010  Standard-Consolen-Einstellung:.....
      0011  6 Leerstellen am linken Rand,.....
      0012  Zeilennummer ausgeben,.....
      0013  Consolenbreite 59d Stellen, d.h. 59-6 Anschläge.....
      0014  *.....


E>x 0,0,79

E>i 16

Consolenbreite 59d Stellen, d.h. 59-6 Anschläge
*
*
Modifizierte Einstellung:.....
keine Leerstellen am linken Rand (erste Null),.....
Zeilennummer abschalten (zweite Null),.....
Consolenbreite 79d, d.h. 79-0=79 Anschläge.....


E>
```

12. X: Extend (Console einstellen)

Format: E>X ll,n,cc<Ret>

Funktion: Linken Rand einstellen (ll: Leerzeichen links),
Zeilennummer unterdrücken (n=Ø) oder darstellen (n<>Ø) und
Consolen-(Zeilen-)Breite auf cc Zeichen einstellen.

Diese Funktion ist dazu vorgesehen, die vom Editor gebotenen Textverarbeitungs-Möglichkeiten zu nutzen; Sie können damit die Zeilenbreite an ein bestimmtes Formular anpassen und die Zeilennummern (z.B. für Text-Ausdrucke) unterdrücken.

Die maximale Consolenbreite darf 79d Zeichen nicht überschreiten (einschließlich der führenden Leerzeichen am linken Rand). Die beim Kaltstart vorgenommene Standard-Einstellung ist an das MOPPEL-Assembler-Format angepaßt.

hms

```
MOPPEL-Editor V 6.6
Copyright (C) hms'86

 A: Assembler
 C: Clear
 F: Floppy
 G: Go to
 I: Insert
 L: List
 M: Monitor
 S: Setup RAM
 X: Extend


E>g5

0002  *
0003  *
0004  *
0005  h0c ...<<< h: HEX-Zeichen; 0C: Form Feed; Blank: Ende HEX-Zeichen.....


E>
```

13. Hexadezimale Steuerzeichen

Format: CTL+8 (einleiten) und Blank (beenden)

Funktion: Einfügen von hexadezimalen Steuerzeichen in den Text und Übergabe des entsprechenden HEX-Codes an die Ausgabe-Routine CO bei allen späteren Listings.

Mit dieser Funktion haben Sie die Möglichkeit, auch solche Codes in einen Text einzufügen, die vom Editor normalerweise zurückgewiesen werden. Auf diese Weise können Sie z.B. einen expliziten Seitenvorschub einfügen (=ØCh) oder auch Escape-Sequenzen zum Hervorheben bestimmter Textpassagen (Inversdarstellung ein- und ausschalten; vgl. Abschnitt 6.5 des System-Handbuchs).

Mit CTL+8 leiten Sie eine Folge von hexadezimalen Steuerzeichen ein; sie wird optisch duch das halbfette 'h' kenntlich gemacht. Jeder HEX-Code muß danach zweistellig angegeben werden (das gilt auch für eine führende Null). Sie können beliebig viele Code-Folgen nacheinander eingeben und verlassen den HEX-Mode einfach dadurch wieder, daß Sie ein Leerzeichen ('Blank') eingeben.

hms