

M O P P E L - M o n i t o r - B e s c h r e i b u n g

1. HEX-Monitor
(Kurzbeschreibung)
2. ASCII-Monitor
(Allgemeine Handhabung)
3. HEX-Monitor
(Basis-Symboltabelle)
4. HEX-Monitor
(Assembler-Listings)
5. ASCII-Monitor
(Basis-Symboltabelle)
6. ASCII-Monitor
(Assembler-Listings)

Copyright

(C) c o s m o s

1 9 8 5

MOPPEL-HEX-Monitor

Unter dem Begriff "HEX-Monitor" sind diejenigen Programmteile zusammengefaßt, die zur Programmeingabe im hexadezimalen Maschinencode erforderlich sind. Diese "minimale Intelligenz" des Systems ist in einem EPROM enthalten, das den Adreßbereich von 0000...0FFFh belegt (4 KBytes; im 87er-System ein 2732-EPROM, im 89er-System die untere Hälfte eines 2764-EPROMs).

Der HEX-Monitor unterscheidet drei Ebenen der Kommunikation, die anschaulich in einer Stelle der Siebensegmentanzeige dargestellt werden:

MEMory-Mode (Daten- oder Adreßeingabe): eine LED leuchtet
REGister-Mode (Register wählen/ändern): zwei LEDs leuchten
FUNCTion-Mode (Funktionseingabe): drei LEDs leuchten.

Im MEM-Mode gibt der leuchtende Dezimalpunkt im Daten- oder Adreßfeld an, wohin Eingaben von der HEX-Tastatur gelangen.

Die Tasten **DAT** und **ADR** dienen zum Umschalten zwischen Daten- und Adreß-Eingabe und umgekehrt.

Das Fortschalten zur nächsten Adresse (bzw. zum nächsten Register oder Parameter bei **FCT**) erfolgt mit der **NXT**-Taste (Next), das Zurückschalten mit der **BST**-Taste (Back Step).

RUN startet ein Programm von der im Adreßfeld angezeigten Adresse an, und die rote **RESET**-Taste bewirkt ein hardwaremäßiges Rücksetzen des Systems.

Mit der Funktions-Taste **FCT** bereiten Sie verschiedene Funktionen vor, wobei Sie mehrere Parameter durch **NXT** trennen und durch **RUN** die Ausführung der Funktion selbst starten (vgl. umseitige Aufstellung).

Funktionen des HEX-Monitors

FCT 0 aaaa >NXT< eeee >RUNK: Ausgabe über das Cassetten-Interface (vgl. Anweisung "O=Output Cassette" im ASCII-Monitor).

FCT 1 aaaa >NXT< eeee >RUNK: Eingabe über das Cassetten-Interface (vgl. Anweisung "I=Input Cassette" im ASCII-Monitor).

FCT 2 aaaa >NXT< eeee >RUNK: Ausgabe über die serielle CPU-Schnittstelle SOD (beim 8085); sonst wie FCT 0.

FCT 3 aaaa >NXT< eeee >RUNK: Eingabe über die serielle CPU-Schnittstelle SID (beim 8085); sonst wie FCT 1.

FCT 4 aaaa >NXT< eeee >RUNK: Speicherinhalt von aaaa bis eeee hexadezimal über den Thermodrucker listen.

FCT 5 aaaa >NXT< nn >RUNK: Ab Adresse aaaa ASCII-Texte über den Thermodrucker ausgeben; nn: Zeilenanzahl (hexadezimal).

FCT 6 aaaa >NXT< eeee >NXT< zzzz >RUNK: Identisch mit der Anweisung "X=Exchange" im ASCII-Monitor.

FCT 7 aaaa >NXT< nn >RUNK: Ab Adresse aaaa NOPs einfügen (=00h) und Speicherinhalt nach oben verschieben; nn: Anzahl der NOPs.

FCT 8 ssmm >NXT< wkk >NXT< MMJJ >RUNK: Identisch mit der Anweisung "U=Uhr stellen" im ASCII-Monitor.

FCT 9 aaaa >RUNK: Identisch mit der Anweisung "U=Uhr lesen" im ASCII-Monitor; ab Adresse aaaa zusätzlicher BCD-Uhrzeit-Buffer.

FCT A aaaa >NXT< eeee >NXT< kk >RUNK: Auffüllen; identisch mit der Anweisung "M=Speicher mit Konstante füllen" im ASCII-Monitor.

FCT B aaaa >NXT< eeee >RUNK: Breakpoint; identisch mit der Anweisung "Z=Haltepunkt setzen" im ASCII-Monitor.

FCT C aaaa >NXT< eeee >NXT< zzzz >RUNK: Identisch mit der Anweisung "C=Copy" im ASCII-Monitor.

FCT D aaaa >NXT< nn >RUNK: Delete; ab Adresse aaaa Bytes löschen und den Speicherinhalt nach unten verschieben; nn: Anzahl der Bytes.

FCT E aaaa >RUNK: Einzelschritt; identisch mit der Anweisung "S=Step" im ASCII-Monitor.

FCT F: Reserviert als internes Sprungziel; für den Anwender nicht zugänglich.

MOPPEL-ASCII-Monitor

Unter dem Begriff "ASCII-Monitor" sind diejenigen Programmteile zusammengefaßt, die zur Verwaltung des **Bildschirm-orientierten** Systems mit ASCII-Tastatur erforderlich sind. Diese "erweiterte Intelligenz" des Systems ist in einem EPROM enthalten, das den Adreßbereich 1000...1FFFh belegt (4 KBytes; im 87er-System ein 2732-EPROM, im 89er-System die obere Hälfte eines 2764-EPROMs). Der ASCII-Monitor verwendet zu einem wesentlichen Teil Routinen aus dem HEX-Monitor, so daß sich beide zur gebotenen Leistungsfähigkeit ergänzen.

Sämtliche **Anweisungen** an den Monitor (a...z) können wahlweise in Groß- oder Kleinbuchstaben erfolgen; ein anschließendes Leerzeichen ("Blank") ist zur Trennung nicht erforderlich. Die folgenden Parameter (maximal drei vierstellige) müssen voneinander durch Komma oder Punkt getrennt werden; führende Nullen brauchen bei den Parametern nicht mit eingegeben zu werden. Die Ausführung jeder Funktion wird durch die Return-Taste veranlaßt.

Zu jedem Zeitpunkt kann der parallel laufende **Protokolldruck** ein- oder ausgeschaltet werden, indem die CTRL-Taste zusammen mit einer der Zifferntasten 7...4 betätigt wird; resultierend daraus wird ein Bit im I/O-Byte umgeschaltet (aktiv HIGH), das den Protokolldruck bei allen Ausgabe-Operationen steuert: Bit 7 aktiviert die serielle Drucker-Schnittstelle, Bit 6 die Centronics-Parallel-Schnittstelle (89er-Hardware), Bit 5 die 8255-Parallel-Schnittstelle (87er-Hardware), und Bit 4 aktiviert den Thermodrucker (87er-Hardware). PORT A

Laufendes Listen (z.B. beim Listen von Speicherinhalten) läßt sich durch Betätigen einer beliebigen Taste **stoppen** und bei erneutem Druck **fortsetzen**; die gleichzeitige Betätigung von CTRL und C führt bei laufendem Listen zum Abbruch und Monitor-Rücksprung. In der Anweisungs-Ebene des Monitors führt die gleichzeitige Betätigung von CTRL und C zur Ausgabe der Monitor-Bereitmeldung.

Beim hardwaremäßigen **Rücksetzen** des Systems über die rote RESET-Taste werden die Inhalte aller CPU-Register in einen dafür reservierten RAM-Bereich gerettet, von wo aus sie bei jedem Programmstart wieder übernommen werden (vgl. Anweisung "G=Go"). Die 89er-ASCII-Tastatur hat zwei RESET-Tasten, die nur bei gleichzeitiger Betätigung zum Rücksetzen führen; wird nur die linke der beiden roten Tasten betätigt, führt dies (bei laufenden Editor- oder Assembler-Listings) zum zeitweiligen Anhalten.

Der Monitor gibt folgende **Fehlermeldungen** aus:

- Error # M00: Parameter-Eingabe fehlt
- Error # M01: Parameter-Eingabe zu lang
- Error # M02: Parameter fehlerhaft (kein HEX-Symbol)
- Error # M03: Parameter nicht im Menü enthalten
- Error # M04: Einschreiben ins RAM fehlerhaft

- Error # H00: Vorspann-Fehler (CAS-Input)
- Error # H01: Startzeichen-Fehler (CAS-Input)
- Error # H02: Sync-Fehler (CAS-Input)

Zur Vermeidung von Irrtümern ist allen HEX-Zahlen ein "h" nachgestellt; wo Verwechslungen möglich sind, ist an Dezimalzahlen ein "d" angehängt worden.

Die **Cursor-Steuerung** ist über eigens dafür vorgesehene Tasten auf der Tastatur-Erweiterung möglich; bei der 89er-Hardware fungiert die an die ASCII-Tastatur angeschlossene HEX-Tastatur als Erweiterung mit den Funktionen Cursor-Steuerung, Zehner-Block und Rechentasten; ersatzweise kann der Cursor über folgende ASCII-Sequenzen bewegt werden:

```
CTRL + U : Cursor aufwärts
CTRL + N : Cursor abwärts
CTRL + H : Cursor nach links
CTRL + J : Cursor nach rechts
CTRL + Z : Cursor nach links oben (home)
```

Mit Hilfe der **Funktionstaste FCT** besteht die Möglichkeit, zu 16 frei wählbaren Sprungzielen zu verzweigen; der Sprung erfolgt bei gleichzeitiger Betätigung der CTRL-Taste mit der unten aufgeführten Zusatz Taste. Die zugehörigen Zieladressen müssen wie folgt im (gepufferten) CMOS-RAM abgelegt werden (erst untere, dann obere Adreßhälfte):

```
in 2E20/21h Sprungziel für FCT + 7
in 2E22/23h Sprungziel für FCT + 6
in 2E24/25h Sprungziel für FCT + 5
in 2E26/27h Sprungziel für FCT + 4
in 2E28/29h Sprungziel für FCT + 3
in 2E2A/2Bh Sprungziel für FCT + 2
in 2E2C/2Dh Sprungziel für FCT + 1
in 2E2E/2Fh Sprungziel für FCT + 9 (87er-Hardware)
in 2E30/31h Sprungziel für FCT + 7 (87er-Hardware)
in 2E32/33h Sprungziel für FCT + DEL
in 2E34/35h Sprungziel für FCT + "+"
in 2E36/37h Sprungziel für FCT + "0"
in 2E38/39h Sprungziel für FCT + 0
in 2E3A/3Bh Sprungziel für FCT + 9
in 2E3C/3Dh Sprungziel für FCT + 8
in 2E3E/3Fh Sprungziel für FCT + 8 (87er-Hardware)
```

Folgende **Parameter** sind im ROM voreingestellt:

Quarzfrequenz der CPU (in MHz)	in Adresse 0006h
Prellzeit der Tastatur (in ms)	in Adresse 0007h
Drucker-Baudrate (Voreinstellung: 4800 Bd)	in Adresse 000E/0Fh
Drucker-Betriebsart (8 Daten, ohne Parit.)	in Adresse 005Eh
Cassetten-Baudrate (Voreinstellg.: 1200 Bd)	in Adresse 0016/17h
Cassetten-Betriebsa. (8 Daten, ohne Parit.)	in Adresse 005Fh
V.24-Baudrate (Voreinstellung: 4800 Bd)	in Adresse 001E/1Fh
V.24-Betriebsart (8 Daten, ohne Parität)	in Adresse 0060h

Folgende **Laufzeit-Unterprogramme** sind verfügbar (sie verändern keinen Registerinhalt; aktuelle CPU-Quarzfrequenz in Adresse 0006h!):

DELY1	1 ms Laufzeit	Einsprungadresse 000Bh
DLY100	100 ms Laufzeit	Einsprungadresse 0013h
ONESEC	1 s Laufzeit	Einsprungadresse 001Bh

Bei Verwendung von Monitor-Unterprogrammen in Anwender-Software dürfen nur die Festadressen der **Sprungtabellen** am PROM-Anfang verwendet werden (ab 0040h bzw. ab 1000h)! Generell gilt für die ROM-residente Software, daß bei Adresse X000h der Sprung zum Kaltstart und bei Adresse X003h der Sprung zum Warmstart des jeweiligen PROMs steht.

Beispiel 1.1:

Um die Anweisung "K=Kill" zu demonstrieren (vgl. dort), soll zuvor der Speicherbereich von 1000...1073h ins RAM ab 2800h umkopiert werden:

MOPPEL-Video-Monitor V 7.5
Copyright (C) hms'85

Anweisung a...Z eingeben:

```
M>C 1000,1073,2800
2800 C3 80 10 C3 8C 10 C3 E3 15 C3 A0 13 C3 6B 14 C3
o.k.
```

C Copy

1. Speicherbereich umkopieren

Format: C aaaa,eeee,zzzz>Ret<

(die Anweisung kann ein Kleinbuchstabe sein,
das Leerzeichen kann entfallen, und anstelle
der Kommas können Sie auch Punkte eingeben)

Funktion: Umkopieren eines Speicherbereichs von der Anfangsadresse
aaaa bis zur Endadresse eeee (einschließlich) in einen
RAM-Bereich, der bei Zieladresse zzzz beginnt.

Nach erfolgtem Umkopieren erscheinen die ersten 16 Bytes
des Zieladreibereichs zur Kontrolle hexadezimal gelistet.

Beachten: Es sind keine Einschränkungen hinsichtlich der Parameter
aaaa und eeee zu beachten; es kann also auch überlappend
kopiert werden, d.h. zzzz kann ohne weiteres im Bereich
aaaa...eeee liegen. Lediglich der Zieladreibereich ab
zzzz muß im RAM liegen, weil sonst kein Einschreiben er-
folgen kann.

2. Programmausführung starten

Format 1: G aaaa>Ret<

(die Anweisung kann ein Kleinbuchstabe sein,
und das Leerzeichen kann entfallen)

Funktion: Programm bei der Adresse aaaa starten (Cursor dabei ausschalten).

Vor dem Start werden die CPU-Register einschließlich Stack-Pointer geladen, und zwar mit denjenigen Werten, die in den korrespondierenden RAM-Zellen abgelegt sind:

REG A mit dem Inhalt von REGA = 2FA7h
REG B mit dem Inhalt von REGB = 2FABh
REG C mit dem Inhalt von REGC = 2FA9h
REG D mit dem Inhalt von REGD = 2FAAh
REG E mit dem Inhalt von REGE = 2FABh
REG F mit dem Inhalt von REGF = 2FACH
REG H mit dem Inhalt von REGH = 2FADh
REG L mit dem Inhalt von REGL = 2FAEh
Stack-Pointer mit REGS & REGP = 2FAF/2FB0h

Beachten: Es sind keine Einschränkungen hinsichtlich des Parameters aaaa zu beachten; es können mit dem G-Kommando also auch Programme im EPROM gestartet werden.

Beim Laden des FLAG-Registers (REG F) ist zu beachten, daß nur die FLAGS CARRY (Bit 0), PARITY (Bit 2), AUX CARRY (Bit 4), ZERO (Bit 6) und SIGN (Bit 7) beeinflusst werden können.

Format 2: G>Ret<

(die Anweisung kann ein Kleinbuchstabe sein)

Funktion: Programmstart bei Adresse 2800h (unterste RAM-Adresse).

Beachten: Es gelten die unter Format 1 genannten Randbedingungen.

Beispiel 2.1:

Das im Zusammenhang mit der Anweisung "M=Speicher inspizieren und modifizieren" eingegebene Programm (vgl. dort) soll gestartet werden; da dieses Programm bei Adresse 2800h beginnt, können Sie hierzu die unter Format 2 angegebene Kurzform wählen.

Allerdings setzt das Programm voraus, daß das Registerpaar H&L mit derjenigen Adresse geladen ist, bei der der Text anfängt, den das Monitor-Unterprogramm STRING ausgeben soll; die betreffende Anfangsadresse 2806h laden Sie deshalb (per M-Anweisung) in die RAM-Adressen 2FADh und 2FAEh, von wo aus sie bei Ausführung der Anweisung G in die CPU-Register H&L übertragen wird:

MOPPEL-Video-Monitor V 7.5
Copyright (C) hms'85

Anweisung a...Z eingeben:

```
M>m 2fad
2FAD > 28 00101000
2FAE > 06 00000110
2FAF > 2F 00101111
```

```
M>g
MOPPEL
```

H Help

3. Hilfsfunktion: Monitor-Menü listen

Format: H>Ret<

(die Anweisung kann ein Kleinbuchstabe sein)

Funktion: Listen sämtlicher Monitor-Anweisungen.

Beachten: Während des Listens ist kein Stoppen möglich (wohl aber der Paralleldruck).

Die Anweisungen A (=Assembler), B (=BASIC), D (=Disassembler), E (=Editor) und F (=Floppy) verlassen die Monitor-Ebene und rufen eigene Menüs (mit eigener Kennung vor dem Eingabe-Pfeil ">") auf; der Rücksprung aus diesen Ebenen zurück in den Monitor erfolgt durch Eingabe eines "m" (oder "M") ohne anschließendes Return.

Zusammen mit dem 89er-EF-ROM-Programmiermodul führt auch die Anweisung P (=Prommer) zur Verzweigung in ein eigenes Menü (vgl. Anweisung "P=Prommer").

Beispiel 3.1:

Sie wissen nicht mehr genau, welche Monitor-Anweisung was bewirkt? Dann rufen Sie doch einfach die HELP-Funktion auf, und Sie bekommen alles gelistet! Wenn Sie vorher den Paralleldruck eingeschaltet haben (durch gleichzeitige Betätigung von CTRL+7...'), dann wird dieses Monitor-Menü auch noch ausgedruckt.

Sie beachten bitte, daß die Anweisungen a, b, d, e, f (und mit Einschränkungen p) zu einem anderen Menü verzweigen; sie sind deshalb mit einem Sternchen versehen. Aus einem anderen Menü kommen Sie durch Eingabe eines "m" (ohne Return) zurück zum Monitor. Wie dies beim Aufruf des Assemblers mit sofortigem Monitor-Rücksprung aussieht, zeigt Ihnen der folgende Protokolldruck:

```
MOPPEL-Video-Monitor V 7.5  
Copyright (C) hms'85
```

Anweisung a...Z eingeben:

```
M>h  
S:user  
a*Assembler  
b*BASIC  
c:copy  
d*Disassembler  
e*Editor  
f*Floppy  
g:go  
h:help  
i:casin  
j:jump back  
k:kill  
l:list  
m:mem/fill  
n:hardcopy  
o:casot  
p*Prommer  
q:sample  
r:revise  
s:step  
t:text  
u:uhr  
v:v24out  
w:v24in  
x:exchange  
y:byte  
z:break  
*
```

o.k.

```
MOPPEL-Video-Monitor V 7.5  
Copyright (C) hms'85
```

Anweisung a...Z eingeben:

```
M>a  
  
MOPPEL-Assembler V 8.5  
Copyright (C) hms'85  
  
0/1/2/3: Pass #  
D: Disassembler  
E: Editor  
F: Floppy  
H: HEX-Mode  
I: In/Out on/off  
M: Monitor  
O: OCT-Mode  
S: Symbol Table
```

A>m

M>

I Input Cassette

4. Einlesen von Magnetband-Cassette

Format: I aaaa,eeee>Ret<

(die Anweisung kann ein Kleinbuchstabe sein,
das Leerzeichen kann entfallen, und anstelle
des Kommas können Sie auch einen Punkt eingeben)

Funktion: Einlesen von Daten oder Programmen ins RAM, beginnend
bei der Anfangsadresse aaaa bis zur Endadresse eeee (ein-
schließlich).

Die Übertragungsrate steht in ROM-Adresse 0016/17h und
die Betriebsart in ROM-Adresse 005Fh.

Beachten: Das Einleseprogramm muß in dem ca. 5 s langen Vorspann
gestartet werden (kontinuierlicher Dauerton), der bei der
Aufnahme vor dem eigentlichen Datenblock aufgezeichnet
worden ist (vgl. Anweisung "O=Output Cassette").

Für den Cassetten-Betrieb ist das Serielle Interface er-
forderlich.

Bei der 89er-Hardware schaltet an den Stiften 4 und 5
der DIN-Buchse zum Bandgeräte-Anschluß ein Relais-Kon-
takt, um den Laufwerksmotor automatisch zuzuschalten.

Beispiel 4.1:

Zum Einlesen von Daten oder Programmen, die auf Magnetbandcassette gespeichert sind, geben Sie einfach die Anweisung "I=Input Cassette" ein; das auslösende Return darf allerdings erst dann folgen, wenn das Band bis zum Vorspann vorgespult ist, der jeder Datenaufzeichnung vorausgeht.

MOPPEL-Video-Monitor V 7.5
Copyright (C) hms'85

Anweisung a...Z eingeben:

M>I 9000,97ff
o.k.

M>

J Jump Back

5. Rücksprung zum HEX-Monitor

Format: J>Ret<
(die Anweisung kann ein Kleinbuchstabe sein)

Funktion: Betrieb der hexadezimalen Siebensegmentanzeige zusammen mit der ASCII-Tastatur (also ohne Datensichtgerät). Die Tasten der HEX-Tastatur haben ihre hexadezimale Grundbedeutung, und folgende ASCII-Tasten haben eine Sonderfunktion:

Pfeil rechts (neben TAB):	DAT
Pfeil links (neben TAB):	ADR
"r" oder "R":	REG
Return:	NXT
Pfeil hoch (neben CTRL):	BST
"g" oder "G":	RUN
Funktionstaste:	FCT

Beachten: Da diese Betriebsart nur zu Demonstrationszwecken vorgesehen ist, können hierüber keine Funktionsaufrufe erfolgen (Ausnahme: FCT 9 = Digitaluhr in der Siebensegmentanzeige).

Hierfür ist die 89er-Hardware Voraussetzung (ASCII-Tastatur und mit LED-Einheit).

Beispiel 5.1:

Nach Eingabe der Anweisung J und anschließendem Return verhält sich die LED-Anzeige so, als sei die HEX- und nicht die ASCII-Tastatur angeschlossen: Im Adreßfeld erscheint die unterste RAM-Adresse 2800h, und Sie können nun ohne Datensichtgerät im hexadezimalen Maschinencode programmieren, und zwar unter Zuhilfenahme der ASCII-Tastatur.

Wenn Sie genug davon haben, müssen Sie RESET betätigen, um zurück in den Monitor zu kommen.

K Kill

6. Bestimmte Bytes durch andere ersetzen

Format: K aaaa,eeee,xyy>Ret<

(die Anweisung kann ein Kleinbuchstabe sein,
das Leerzeichen kann entfallen, und anstelle
der Kommas können Sie auch Punkte eingeben)

Funktion: Im RAM-Bereich von Anfangsadresse aaaa bis Endadresse
eeee (einschließlich) werden alle Bytes xx durch yy
ersetzt.

Nach erfolgtem Ersetzen erscheinen die ersten 16 Bytes
des RAM-Bereichs zur Kontrolle hexadezimal gelistet.

Beachten: Der Speicherbereich aaaa...eeee muß vollständig im RAM
liegen, weil sonst keine Modifikation erfolgen kann.

Beispiel 6.1:

Mit Hilfe der Anweisung "C=Copy" ist der Speicherbereich von 1000...1073h ins RAM ab 2800h umkopiert worden (vgl. dort); um die Anweisung "K=Kill" zu demonstrieren, sollen im Zieladreibereich alle Bytes "10h" durch "F0h" ersetzt werden (z.B., um Adressen aus dem 1000er-Bereich nach F000h zu verlegen):

MOPPEL-Video-Monitor V 7.5
Copyright (C) hms'85

Anweisung a...Z eingeben:

M>K 2800,2860,10f0

2800 C3 80 F0 C3 8C F0 C3 E3 15 C3 A0 13 C3 6B 14 C3

o.k.

L List

7. Speicherbereich listen

Format: L aaaa,eeee>Ret<

(die Anweisung kann ein Kleinbuchstabe sein,
das Leerzeichen kann entfallen, und anstelle
des Kommas können Sie auch einen Punkt eingeben)

Funktion: Listen des Speicherinhalts von der Anfangsadresse aaaa
bis zur Endadresse eeee (einschließlich) mit hexadezima-
ler Darstellung der Bytes.

Das Listen kann durch Drücken einer beliebigen Taste ge-
stoppt (bzw. fortgesetzt) und durch CTRL+C abgebrochen
werden.

Beachten: Die eingegebene Anfangsadresse aaaa wird im niedrigstwer-
tigen Digit auf 0 abgerundet (zu aaa0), und die Endadres-
se eeee wird im niedrigstwertigen Digit aufgerundet (zu
eeeF), um jeweils volle Blöcke von je 16 Bytes darzu-
stellen.

Beispiel 7.1:

Nach dem Kopieren mit der Anweisung "C=Copy" wird zur Kontrolle immer nur eine Zeile mit 16 Bytes aus dem Zielbereich gelistet (vgl. dort); wenn Sie sich mehr ansehen wollen, hilft Ihnen dazu die LIST-Funktion:

MOPPEL-Video-Monitor V 7.5
Copyright (C) hms'85

Anweisung a...Z eingeben:

M>L 2800,2873

2800	C3	80	F0	C3	8C	F0	C3	E3	15	C3	A0	13	C3	6B	14	C3
2810	AE	18	C3	6B	11	C3	ED	11	C3	3B	1B	C3	72	12	C3	26
2820	1C	C3	5F	18	C3	2C	1C	C3	A1	12	C3	79	18	C3	1D	17
2830	C3	FC	16	C3	DF	17	C3	F3	17	C3	FC	17	C3	20	13	C3
2840	0D	13	C3	E5	12	C3	2C	13	00	00	00	F0	00	60	00	40
2850	46	19	00	6C	00	70	00	20	28	19	08	1F	8E	00	CD	00
2860	DA	18	5B	19	FC	19	AF	1B	8B	00	A0	1B	1B	10	18	10
2870	E1	1C	10	19	CE	28	CD	21	10	CD	36	10	CD	43	00	FE

o.k.

M Memory

Doppelfunktion; hier:

8.2 RAM mit Konstante füllen

Format: **M** *aaaa,eeee,kk*>Ret<

(die Anweisung kann ein Kleinbuchstabe sein,
das Leerzeichen kann entfallen, und anstelle
der Kommas können Sie auch Punkte eingeben)

Funktion: Auffüllen eines Speicherbereichs von der Anfangsadresse
aaaa bis zur Endadresse *eeee* (einschließlich) mit der
Konstanten *kk*.

Nach erfolgtem Einschreiben erscheinen die ersten 16 By-
tes des RAM-Bereichs zur Kontrolle hexadezimal gelistet.

Beachten: Der Adreßbereich von *aaaa...eeee* muß im RAM-Bereich
liegen, weil sonst kein Einschreiben erfolgen kann.

Beispiel 8.2.1:

Das Füllen von Speicherbereichen mit einer Konstanten kann u.a. aus zwei Gründen erfolgen: Man schreibt "blind" einen Bereich voll, der mit RAMs bestückt sein kann; anschließend führt man ein Kontroll-Lesen durch und sieht, wie weit der Speicher tatsächlich bestückt ist (Speicher-Test).

Eine andere Möglichkeit ist beispielsweise das Löschen des Bildschirm-Speichers; der muß dazu mit Leerzeichen ("Blanks=20h) gefüllt werden.

Natürlich können Sie auch jedes beliebige andere Zeichen ins RAM einschreiben, etwa die Konstante 2Ah, die bei Darstellung auf dem Bildschirm ein Sternchen ergibt:

MOPPEL-Video-Monitor V 7.5
Copyright (C) hms'85

Anweisung a...Z eingeben:

MM 2900,2944,2a
2900 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A
o.k.

M Memory

Doppelfunktion; hier:

8.1 Speicher inspizieren und modifizieren

Format 1: M aaaa>Ret<

(die Anweisung kann ein Kleinbuchstabe sein,
und das Leerzeichen kann entfallen)

Funktion: Inhalt der Speicherstelle aaaa hexadezimal und binär darstellen; soweit die Adresse aaaa im RAM liegt, kann der Inhalt modifiziert werden.

Das Weiterschalten zur nächstfolgenden Adresse aaaa+1 erfolgt mit der Return- oder Blank-Taste; das Zurückschalten zur vorherigen Adresse aaaa-1 erfolgt mit der Minus-Taste. Die Pfeiltasten "<" bzw. ">" schalten auf den Adreß- bzw. Daten-Mode um.

Beachten: Um einen eventuellen Protokolldruck nicht zu stören, erfolgt beim Modifizieren des Speicherinhalts die Eingabeprüfung erst beim Weiterschalten zur nächsten Adresse.

In jedem Fall wird der modifizierte Speicherinhalt erst mit dem Fortschalten zur nächsten Adresse ins RAM eingeschrieben.

Format 2: M>Ret<

(die Anweisung kann ein Kleinbuchstabe sein)

Funktion: Inspizieren der Adresse 2800h (unterste RAM-Adresse).

Beachten: Es gelten die unter Format 1 genannten Randbedingungen.

Beispiel 8.1.1:

Mit der Anweisung "M=Speicher inspizieren und modifizieren" können Sie unter Umständen abgeschlossene kleine Programme eingeben; um dies anschaulich darzustellen, vollziehen Sie die Eingabe der wenigen Befehle nach, die bei der Ausführung das Wort "MOPPEL" auf den Bildschirm bringen (vgl. Anweisung "G=Go"). Unten finden Sie, lediglich zur Erläuterung, das zugehörige Assembler-Listing zu diesem Programmbeispiel; Sie können daran nachvollziehen, wie dieses Mini-Programm aufgebaut ist.

Beachten Sie bitte den Nebeneffekt, daß das Monitor-Unterprogramm STRING beim Einsprung erwartet, daß im Registerpaar H&L die Startadresse des auszugebenden Textes steht; diese Adresse wird nicht im Programm vorgegeben, sondern manuell vor dem Start geladen (vgl. Anweisung "G=Go"). Damit wird der Effekt des Register-Ladens vor dem Programm-Start demonstriert.

MOPPEL-Video-Monitor V 7.5
Copyright (C) hms'85

Anweisung a...Z eingeben:

```
Mem
2800  >  CD  11001101
2801  >  21  00100001
2802  >  10  00010000
2803  >  03  11000011
2804  >  05  00000011
2805  >  10  00010000
2806  >  0D  00001101
2807  >  4D  01001101
2808  >  4F  01001111
2809  >  50  01010000
280A  >  50  01010000
280B  >  45  01000101
280C  >  4C  01001100
280D  >  00  00000000
```

b.k.

```
2800          MOVID EQU 1003H      *Monitor-Warmstart
2800          STRING EQU 1021H     *ASCII-Textausgabe
2800          *
2800  CD 21 10 TEST: CALL STRING    *H&L: Textanfang!
2803  C3 03 10      JMP MOVID      *zurück zum Monitor
2806          *
2806  0D          DE      0DH      *Carriage Return
2807  4D          DE      "M"
2808  4F          DE      "O"
2809  50          DE      "P"
280A  50          DE      "F"
280B  45          DE      "E"
280C  4C          DE      "L"
280D  00          DE      0H      *String-Ende
```

N Hardcopy

9. Bildschirminhalt ausdrucken

Format: N x>Ret<

(die Anweisung kann ein Kleinbuchstabe sein,
und das Leerzeichen kann entfallen)

Funktion: Ausgabe des Bildschirmspeichers ab 3000h über den mittels
x gewählten Drucker; Bedeutung des Parameters x:

x = 7: serielle Drucker-Schnittstelle
x = 6: Centronics-Parallel-Schnittstelle
x = 5: 8255-Parallel-Schnittstelle (87er-Hardware)
x = 4: Thermodrucker (87er-Hardware)

Beachten: Bei fehlender Angabe des Parameters x wird x=7 ange-
nommen (serielle Drucker-Schnittstelle).

Die unterste Bildschirmzeile, in der die Monitor-An-
weisung steht ("M>n"), wird nicht mit ausgedruckt.

Beispiel 9.1:

Es gibt immer wieder gute Gründe, bestimmte Vorgänge beim Dialog mit dem Computer zu protokollieren; wenn Sie den Protokolldruck einmal nicht aktiviert hatten und nachträglich etwas festhalten wollen, können Sie dies mit der N-Funktion veranlassen.

Das Beispiel unten zeigt den Ablauf, der den Beispielen 1.1 (C=Copy), 6.1 (K=Kill) und 7.1 (L=List) vorausgegangen ist:

Copyright (C) hms 85

Anweisung a...Z eingeben:

M>E 1000,1073,2800

2800 C3 80 10 C3 8C 10 C3 E3 15 C3 A0 13 C3 6B 14 C3
o.k.

M>K 2800,2860,1040

2800 C3 80 F0 C3 8C F0 C3 E3 15 C3 A0 13 C3 6B 14 C3
o.k.

M>L 2800,2873

2800 C3 80 F0 C3 8C F0 C3 E3 15 C3 A0 13 C3 6B 14 C3
2810 AE 18 C3 6E 11 C3 ED 11 C3 3B 1B C3 72 12 C3 26
2820 1C C3 5F 18 C3 2C 1C C3 A1 12 C3 79 18 C3 1D 17
2830 C3 FC 16 C3 DF 17 C3 F3 17 C3 FC 17 C3 20 13 C3
2840 0D 13 C3 E5 12 C3 2C 13 00 00 00 F0 00 60 00 40
2850 46 19 00 6C 00 70 00 20 28 19 08 1F 8E 00 CD 00
2860 DA 18 5B 19 FC 19 AF 1E 8B 00 A0 1B 1B 10 18 10
2870 E1 1C 10 19 CE 28 CD 21 10 CD 36 10 CD 43 00 FE
o.k.

0 Output Cassette

10. Ausgabe auf Magnetband-Cassette

Format 1: O aaaa,eeee>Ret<

(die Anweisung kann ein Kleinbuchstabe sein,
das Leerzeichen kann entfallen, und anstelle
des Kommas können Sie auch einen Punkt eingeben)

Funktion: Ausgabe von Daten oder Programmen, beginnend bei der
Anfangsadresse aaaa bis zur Endadresse eeee (einschließ-
lich).

Die Übertragungsrate steht in ROM-Adresse 0016/17h und
die Betriebsart in ROM-Adresse 005Fh.

Beachten: Vor Beginn der eigentlichen Datenübertragung wird ein
ca. 5 s langer Vorspann erzeugt (kontinuierlicher Dauert-
on), in dem das spätere Einlesen gestartet werden muß
(vgl. Anweisung "I=Input Cassette").

Für den Cassetten-Betrieb ist das Serielle Interface er-
forderlich.

Bei der 89er-Hardware schaltet an den Stiften 4 und 5
der DIN-Buchse zum Bandgeräte-Anschluß ein Relais-Kon-
takt, um den Laufwerksmotor automatisch zuzuschalten.

Format 2: O ad>Ret<

(die Anweisung kann ein Kleinbuchstabe sein,
und das Leerzeichen kann entfallen)

Funktion: Abgleich ("Adjust") des Cassetten-Interfaces; nach ent-
sprechender Hardware-Vorbereitung wird die vom Timer
gemessene Zeit auf dem Bildschirm dargestellt (vgl. Hard-
ware-Anwender Tip "CAS-Abgleich").

Beachten: Der Rücksprung in den Monitor ist nur über RESET möglich.

Beispiel 10.1:

Mit Hilfe der Magnetbandaufzeichnung können Sie sich ein preiswertes Programmarchiv zulegen. Zur Ermittlung der benötigten Aufzeichnungsdauer berücksichtigen Sie bitte, daß der eigentlichen Datenausgabe ein Vorspann von ca 5 s vorausgeht (zum definierten Starten beim Wieder-Einlesen). Bei einer Übertragungsrate von 1200 Baud (Optimum für die Aufzeichnung mit Phasen-Codierung) und jeweils einem Start- und Stop-Bit (also 10 Übertragungsbits pro Datenbyte) wird pro Byte eine Zeit von rund 8,3 ms benötigt.

Im Beispiel unten werden 100h Bytes aufgezeichnet (256 dezimal); das dauert demzufolge 2,1 s plus 5 s Vorspann, nimmt also zusammen gute 7 s in Anspruch:

MOPFEL-Video-Monitor V 7.5
Copyright (C) hms'85

Anweisung a...Z eingeben:

M00 9000,7100
o.k.

P Prommer

Mehrfachfunktion; hier:

11.1 Prommer-Menü aufrufen

Format: P>Ret<

(die Anweisung kann ein Kleinbuchstabe sein)

Funktion: Bei der 89er-Hardware Aufruf des Menüs, das das EPROM-Programmiermodul ansteuert. Damit lassen sich die Standard-Typen des Marktes programmieren oder lesen (2716, 2732, 2764, 27128 und 27256), wobei nicht nur eine abkürzende Eingabeform möglich ist, sondern beim Programmieren auch noch die verstreichende Zeit eingeblendet wird.

Das Programm erkennt automatisch, ob die 87er- oder 89er-Prommer-Hardware installiert ist.

Beachten: Das 89er-EPROM-Programmiermodul setzt den 89er-Floppy-Disk-Controller voraus (jedoch kein Laufwerk).

Beispiel 11.1.1:

Wenn Sie in Ihrem System das 89er-EEPROM-Programmiermodul (plus Floppy-Disk-Controller) installiert haben, verzweigt das Monitor-Programm bei Eingabe von "F" (und Abschluß durch Return) in ein eigenes, komfortables Menü zur Programmierung von EPROMs.

Zurück zum Monitor kommen Sie durch Eingabe eines "m" (ohne Return).

P Prommer

Mehrfachfunktion; hier:

11.2 EPROMs 2716/2732 lesen

Format 1: P aaaa,eeee,zzzz>Ret<
16 oder 32>Ret<
L

(die Anweisungen können Kleinbuchstaben sein, das Leerzeichen kann entfallen, und anstelle der Kommas können Sie auch Punkte eingeben)

Funktion: Mit dem 87er-EPROM-Programmierzusatz ein EPROM des angegebenen Typs einlesen, beginnend bei EPROM-Adresse aaaa bis eeee (einschließlich), ins RAM, beginnend bei RAM-Adresse zzzz.

Nach erfolgreichem Einlesen erscheinen die ersten 16 Bytes des Zieladreibereichs zur Kontrolle hexadezimal gelistet.

Beachten: Es sind keine Einschränkungen hinsichtlich der Parameter aaaa und eeee zu beachten; nur der bei Adresse zzzz beginnende Bereich muß im RAM liegen, weil sonst kein Einschreiben erfolgen kann.

Format 2: P>Ret<
16 oder 32
L

(die Anweisungen können Kleinbuchstaben sein)

Funktion: Mit dem 87er-EPROM-Programmierzusatz ein EPROM des angegebenen Typs komplett ins RAM einlesen, beginnend bei RAM-Adresse 8000h.

Beispiel 11.2.1:

Mit der abkürzenden Eingabeform können Sie auf kurzem Weg ein EPROM einlesen; wenn darin, wie im Beispiel unten, ein BASIC-Programm enthalten ist, steht es sofort am richtigen Platz (ab Adresse 8000h), um per Warmstart aktiviert zu werden:

MOPPEL-Video-Monitor V 7.5
Copyright (C) hms'85

Anweisung a...Z eingeben:

M>p

Typ (16/32)>16

lies/prog/vergl>1

8000 00 01 2C 9D 00 00 00 2C 31 39 38 33 00 00 B0 4A
o.k.

F Frommer

Mehrfachfunktion; hier:

11.3 EPROMs 2716/2732 programmieren

Format 1: P aaaa,eeee,zzzz>Ret<

16 oder 32>Ret<

P

(die Anweisungen können Kleinbuchstaben sein, das Leerzeichen kann entfallen, und anstelle der Kommas können Sie auch Punkte eingeben).

Funktion: Mit dem 87er-EPROM-Programmierzusatz ein EPROM des angegebenen Typs programmieren, beginnend mit Speicheradresse aa aa bis eeee (einschließlich), ab EPROM-Adresse zzzz.

Bei fehlerhaftem Programmieren eines Bytes bricht das Programm ab und meldet die Fehleradresse.

Beachten: Es sind keine Einschränkungen hinsichtlich der Parameter aa aa und eeee zu beachten; nur darf der von aa aa...eeee überdeckte Bereich nicht die Kapazität des angegebenen EPROMs überschreiten, und zzzz muß kleiner sein als die höchste EPROM-Adresse.

Format 2: P>Ret<

16 oder 32

P

(die Anweisungen können Kleinbuchstaben sein)

Funktion: Mit dem 87er-EPROM-Programmierzusatz ein EPROM des angegebenen Typs komplett programmieren, beginnend bei RAM-Adresse 8000h.

Beispiel 11.3.1:

Im Protokoll unten erkennen Sie bei genauerem Hinsehen eine Feinheit, die in der Praxis recht nützlich sein kann: Nach Abschluß des Programmierens werden Sie gefragt, ob die letzte Operation wiederholt werden soll; wenn ja, brauchen Sie dazu nur Return zu betätigen, was Ihnen bei Programmierfehlern eine Neueingabe der Parameter erspart.

Wenn Sie keine Wiederholung wünschen, kommen Sie mit CTRL+C zurück in den Monitor.

MOPPEL-Video-Monitor V 7.5
Copyright (C) hms'85

Anweisung a...Z eingeben:

M>P 8000,8200,0

Typ (16/32)>32
lies/prog/vergl>p
o.k.
wdh (=Ret)?

P Prommer

Mehrfachfunktion; hier:

11.4 EPROMs 2716/2732 vergleichen

Format 1: P aaaa,eeee,zzzz>Ret<
16 oder 32>Ret<
V

(die Anweisungen können Kleinbuchstaben sein, das Leerzeichen kann entfallen, und anstelle der Kommas können Sie auch Punkte eingeben)

Funktion: Mit dem 87er-EPROM-Programmierzusatz ein EPROM des angegebenen Typs vergleichen, ab EPROM-Adresse aaaa bis EPROM-Adresse eeee (einschließlich), mit dem Speicherbereich ab Adresse zzzz.

Bei Abweichungen erfolgt ein Listen wie bei der Anweisung "R=Revise".

Beachten: Der von aaaa...eeee überdeckte Bereich darf nicht die Kapazität des angegebenen EPROMs überschreiten; für die Zieladresse zzzz bestehen keine Einschränkungen.

Für den Vergleich wird der EPROM-Inhalt ins RAM eingelesen (beginnend bei Adresse 8000h); in diesem Bereich vorhandenen Daten werden hierbei überschrieben (bis maximal 8FFFh)!

Beispiel 11.4.1:

Der Vergleich eines EPROM-Inhalts mit dem Inhalt eines Speicherbereichs kann beispielsweise dazu dienen, EPROMs auf ihren Leerzustand hin zu überprüfen (alle Zellen haben FFh), oder sich bei einem unbekanntem EPROM von dessen Inhalt zu überzeugen.

Im Beispiel unten wird der Inhalt eines 2732-EPROMs (4-K-Kapazität von 0...FFFh) mit dem Speicherinhalt ab 1000h verglichen; beide sind bis auf die gelisteten elf Bytes identisch.

Auch hier haben Sie, wie im Beispiel 11.3.1, die Möglichkeit, die Operation durch Eingabe von Return zu wiederholen oder mittels CTRL+C zurück in den Monitor zu springen:

```
MOPPEL-Video-Monitor V 7.5  
Copyright (C) hms'85
```

```
Anweisung a...Z eingeben:
```

```
M>P 0,fff,1000
```

```
Typ (16/32)>32  
lies/prog/vergl>v  
9FD5=69 1FD5:6F  
9FD6=6E 1FD6:75  
9FD7=0D 1FD7:74  
9FD8=77 1FD8:0D  
9FD9=3A 1FD9:77  
9FDA=76 1FDA:3A  
9FDB=32 1FDB:76  
9FDC=34 1FDC:32  
9FDD=6F 1FDD:34  
9FDE=75 1FDE:69  
9FDF=74 1FDF:6E
```

```
o.k.  
wdh (=Ret)?
```

Q Sample

12. Zeichenvorrat darstellen

Format: Q>Ret<

(die Anweisung kann ein Kleinbuchstabe sein)

Funktion: Ausgabe des Video-Zeichensatzes auf dem Bildschirm (mit eventuellem Paralleldruck).

Beachten: Die HEX-Codes 08...0Dh werden hierbei unterdrückt, da es sich dabei um Steuerzeichen zur Beeinflussung des Bildschirm-Inhaltes bzw. der Drucker-Papiersteuerung handelt (u.a. Seitenvorschub).

Beispiel 12.1:

Die Q-Anweisung hat primär nicht den Sinn, daß Sie sich nur den Zeichengenerator des Video-Interfaces ansehen; vielmehr besteht hierüber die einfache Möglichkeit, bei aktiviertem Protokolldruck zu verfolgen, welchen Zeichenvorrat Ihr Drucker besitzt. In der Regel wird der übrigens alle Codes unterdrücken, die kleiner sind als 20h:

MOPPEL-Video-Monitor V 7.5
Copyright (C) hms'85

Anweisung a...Z eingeben:

M>q

```
! " # $ % & ' ( ) * + , - . / 0 1 2 3 4 5 6 7 8 9 : ; < = > ?  
$ A B C D E F G H I J K L M N O P Q R S T U V W X Y Z Ä Ö Ü ^ _  
' a b c d e f g h i j k l m n o p q r s t u v w x y z ä ö ü ß
```

```
! " # $ % & ' ( ) * + , - . / 0 1 2 3 4 5 6 7 8 9 : ; < = > ?  
$ A B C D E F G H I J K L M N O P Q R S T U V W X Y Z Ä Ö Ü ^ _  
' a b c d e f g h i j k l m n o p q r s t u v w x y z ä ö ü ß
```

```
! " # $ % & ' ( ) * + , - . / 0 1 2 3 4 5 6 7 8 9 : ; < = > ?
```

13. Speicherbereiche vergleichen und Abweichungen listen

Format: R aaaa,eeee,zzzz>Ret<

(die Anweisung kann ein Kleinbuchstabe sein,
das Leerzeichen kann entfallen, und anstelle
der Kommas können Sie auch Punkte eingeben)

Funktion: Vergleich des Speicherbereichs von Anfangsadresse aaaa
bis Endadresse eeee (einschließlich) mit dem bei der
Zieladresse zzzz beginnenden Speicherbereich.

Bei Differenzen werden die betreffenden Adressen aus dem
Quell- und Zielbereich mit den zugehörigen Speicherin-
halten gelistet.

Das Listen kann durch Drücken einer beliebigen Taste ge-
stoppt (bzw. fortgesetzt) und durch CTRL+C abgebrochen
werden.

Beachten: Es sind keine Einschränkungen hinsichtlich der Parameter
aaaa, eeee und zzzz zu beachten; es können also beliebige
Speicherbereiche aus RAM und/oder ROM miteinander vergli-
chen werden.

Beispiel 13.1:

Eine für die Fehlersuche sehr hilfreiche Funktion ist das Auflisten von Abweichungen zwischen zwei Speicherbereichen (beim Vergleich von EPROM-Inhalten wird diese Funktion ebenfalls benutzt; vgl. Beispiel 11.4.1).

Beim der Anweisung "X=Exchange" werden nach erfolgtem Umkopieren Absolutadressen umgerechnet (vgl. dort); wenn Sie sich davon überzeugen wollen, lassen Sie sich die Abweichungen zwischen Quell- und Zielbereich auflisten. Wie Sie sehen, bestehen die Unterschiede im Beispiel unten nur in den beiden (richtig umgerechneten) Sprungadressen:

```
M>I/O-FLAGS: 10000000
```

```
M>  
MOPPEL-Video-Monitor V 7.5  
Copyright (C) hms'85
```

Anweisung a...Z eingeben:

```
M>R 4af,4be,2800  
04BB=B9 280C:0A  
04BC=04 280D:28
```

o.k.

14. Einzelschritt-Befehlsausführung

Format: S aaaa>Ret<

(die Anweisung kann ein Kleinbuchstabe sein,
und das Leerzeichen kann entfallen)

Funktion: Programmstart bei Adresse aaaa mit vorherigem Laden der CPU-Register (vgl. Anweisung "G=Go"); Ausführung von jeweils nur einem Befehl mit anschließender Darstellung von RAM-Adresse, RAM-Inhalt (aktueller OpCode) und Inhalt sämtlicher CPU-Register (einschließlich der FLAGS).

Das Weiterhalten zum nächsten Befehl erfolgt mit der Taste Return oder dem Leerschritt ("Blank").

Die CPU-Register können jederzeit modifiziert werden, wobei über die Cursor-Steuertasten die Möglichkeit besteht, den Cursor nach links und rechts (aber nicht nach oben oder unten) zu bewegen. Fehlerhafte Eingaben werden erst beim Weiterschalten bemängelt; im FLAG-Register wird bei Falscheingaben nur das niedrigstwertige Bit des eingegebenen ASCII-Codes berücksichtigt.

Beachten: Die Anwender-Programme müssen komplett im RAM stehen, weil der Monitor in dieser Betriebsart Software-Interrupts einfügt und anschließend wieder ersetzt (anstelle des jeweils nächstfolgenden OpCodes).

Aus diesem Grund dürfen die Anwender-Programme nicht die Befehle RST0, RST1 und RST2 enthalten.

Beispiel 14.1:

Bei der Inbetriebnahme und Fehlersuche in eigenen Programmen ist die Einzelschritt-Funktion von unschätzbarem Wert, und das umso mehr, wenn man dabei die Registerinhalte nicht nur verfolgen, sondern auch noch modifizieren kann.

Im Beispiel unten finden Sie die schrittweise Ausführung des umkopierten Monitor-Unterprogramms DELY1 protokolliert (vgl. Anweisung "X=Exchange"); die Zeilennummern stammen nicht vom Protokolldruck, sondern sind zur Erläuterung nachträglich hinzugefügt worden.

In Zeile 2 sehen Sie u.a. die Auswirkungen des PUSH-PSW-Befehls, der in Adresse 2800h steht: Nach dem Einschreiben in den Stack ist der Stack-Pointer um Zwei erniedrigt worden; der POP-PSW-Befehl in Adresse 280Eh macht dies wieder rückgängig (Zeile 14).

Die die Zeitverzögerung bestimmende Programmschleife (280A/0Bh) soll an sich so oft durchlaufen werden, wie es der Faktor C8h (=200d) beim Einsprung vorgibt; um dies beim schrittweisen Nachvollziehen abzukürzen, ist in Zeile 11 der Wert C7h manuell auf 01h geändert worden (im Ausdruck nicht sichtbar, weil erst nach der Befehlsausführung protokolliert wird). In Zeile 12 sehen Sie die Auswirkungen: Das Leerzählen von REG A setzt das ZERO-FLAG im FLAG-Register, und der bedingte Sprung in 280Bh wird diesmal nicht ausgeführt; stattdessen geht es weiter bei dem in 280Eh stehenden Befehl. - Ganz unten finden Sie, zu Ihrer Orientierung, ein Assembler-Listing dieses kleinen Beispielprogramms.

MOPPEL-Video-Monitor V 7.5
Copyright (C) hms'85

Anweisung a...Z eingeben:

M>S 2800

	Adr.	Inh.	A	B&C	D&E	H&L	SP	SZxcxPxC
1	2800	F5	BC	520B	BE08	2FCE	2F70	00000000
2	2801	3A	BC	520B	BE08	2FCE	2F6E	00000000
3	2804	87	06	520B	BE08	2FCE	2F6E	00000000
4	2805	87	0C	520B	BE08	2FCE	2F6E	00001000
5	2806	3C	18	520B	BE08	2FCE	2F6E	00011000
6	2807	87	19	520B	BE08	2FCE	2F6E	00001000
7	2808	87	32	520B	BE08	2FCE	2F6E	00110000
8	2809	87	64	520B	BE08	2FCE	2F6E	00100000
9	280A	3D	C8	520B	BE08	2FCE	2F6E	10001100
10	280B	C2	C7	520B	BE08	2FCE	2F6E	10000010
11	280A	3D	C7	520B	BE08	2FCE	2F6E	10000010
12	280B	C2	00	520B	BE08	2FCE	2F6E	01000010
13	280E	F1	00	520B	BE08	2FCE	2F6E	01000010
14	280F	C9	BC	520B	BE08	2FCE	2F70	00000000

2800	F5			DELY1:	PUSH	PSW		* 1 ms
2801	3A	06	00		LDA	FQUARZ		*4/6/8 MHz!
2804	87				ADD	A		
2805	87				ADD	A		
2806	3C				INR	A		*adjust to 99%
2807	87				ADD	A		
2808	87				ADD	A		
2809	87				ADD	A		*Zeitfaktor *32
280A	3D			DELOP:	DCR	A		
280B	C2	0A	28		JNZ	DELOP		
280E	F1				POP	PSW		
280F	C9				RET			

T Text

15. ASCII-Zeichenketten ausgeben

Format: T aaaa,ee>Ret<

(die Anweisung kann ein Kleinbuchstabe sein,
das Leerzeichen kann entfallen, und anstelle
des Kommas können Sie auch einen Punkt eingeben)

Funktion: Den bei Adresse aaaa beginnenden Speicherinhalt als
ASCII-Zeichenfolge ausgeben, bis das Endzeichen ee auf-
tritt (das selbst nicht mehr ausgegeben wird).

Beachten: Die Ausgabe kann nicht wie beim Listen gestoppt oder ab-
gebrochen werden; wird das Endzeichen ee nicht erreicht,
kann der Rücksprung in den Monitor nur mittels RESET er-
folgen.

Beispiel 15.1:

Die T-Funktion hat den Sinn, in HEX-Listings enthaltene ASCII-Zeichenfolgen auszugeben; dies erleichtert häufig die Orientierung, wenn man (beispielsweise in umfangreichen CP/M-Programmen) anhand von Text-Überschriften erkennen will, um welches Programm es sich handelt.

Zur Demonstration lassen Sie sich einfach den Text ausgeben, den das Beispielprogramm 8.1.1 erzeugt (vgl. dort); er beginnt bei 2806h und endet mit dem Zeichen 00h:

MOPPEL-Video-Monitor V 7.5
Copyright (C) hms'85

Anweisung a...Z eingeben:

M>T 2806,00
MOPPEL
o.k.

U Uhr

Doppelfunktion; hier:

16.1 Echtzeit-Uhr lesen

Format: U>Ret<

(die Anweisung kann ein Kleinbuchstabe sein)

Funktion: Auslesen der Echtzeit-Uhr und Überschreiben der Datums/-Uhrzeit-Information in die Uhrzeit-Buffer UHRBUF (ab 2FDCh; BCD-Format) bzw. ASCBUF (ab 2FEAh; ASCII-Zeichenkette). Außerdem Darstellung von Datum und Uhrzeit auf dem Bildschirm und Freigabe des RST6-Interrupts zur ständigen Einblendung von Datum und Uhrzeit in die rechte obere Bildschirm-Ecke:

Di, 10. 12. 85; 16: 38: 49h

Beachten: Die Hardware der 87er-Echtzeit-Uhr ist hierfür geringfügig zu modifizieren (vgl. Hardware-Anwender-Tip "EZU-Interrupt"); bei der 89er-CPU muß die Brücke RST6 eingesetzt sein.

Beim Betrieb eines Floppy-Laufwerks wird der Uhrzeit-Interrupt inaktiviert, da er anderweitig verwendet wird; bei Bedarf muß nach der Floppy-Ansprache die Anweisung "u" erneut aufgerufen werden.

Beispiel 16.1.1:

Mit der U-Funktion können Sie sich die Uhrzeit auf dem Bildschirm darstellen lassen; zur Verarbeitung dieser Datums/Uhrzeit-Information besitzt BASIC genau passende Befehle.

Um diese Daten auch in Assembler-Programmen auswerten zu können, ist das ASCII-Format weniger geeignet; deshalb stellt diese Anweisung Datum und Uhrzeit auch im BCD-Format im RAM zur Verfügung:

MOPPEL-Video-Monitor V 7.5
Copyright (C) hms'85

Anweisung a...Z eingeben:

M>u
Di,10.12.85;11:34:09h

M>m2fdc

2FDC	>	09	00001001	Sekunden-Einer
2FDD	>	00	00000000	Sekunden-Zehner
2FDE	>	04	00000100	Minuten-Einer
2FDF	>	03	00000011	Minuten-Zehner
2FE0	>	01	00000001	Stunden-Einer
2FE1	>	01	00000001	Stunden-Zehner
2FE2	>	00	00000000	Kalender-Tages-Einer
2FE3	>	01	00000001	Kalender-Tages-Zehner
2FE4	>	02	00000010	Wochentag (=Dienstag)
2FE5	>	00	00000000	Dummy-Zelle (ohne Bedeutung)
2FE6	>	02	00000010	Monats-Einer
2FE7	>	01	00000001	Monats-Zehner
2FE8	>	05	00000101	Jahres-Einer
2FE9	>	08	00001000	Jahres-Zehner

o.k.

U Uhr

Doppelfunktion; hier:
16.2 Echtzeit-Uhr stellen

Format: U ssmm,wkk,MMJJ >Ret<

(die Anweisung kann ein Kleinbuchstabe sein,
das Leerzeichen kann entfallen, und anstelle
des Kommas können Sie Punkte eingeben)

Funktion: Stellen der Echtzeit-Uhr mit folgenden Vorgaben:

ss: Stunden
mm: Minuten (Sekunden automatisch Null)
w: Wochentag mit folgender Codierung: 1=Montag
2=Dienstag
3=Mittwoch
4=Donnerstag
5=Freitag
6=Sonnabend
7=Sonntag

kk: Kalendertag
MM: Monat
JJ: Jahres-Zehner und -Einer

Anschließend erfolgt zur Kontrolle das Auslesen der Echtzeit-Uhr, wie unter 16.1 beschrieben.

Beachten: Eine Plausibilitätsprüfung der eingegebenen Werte erfolgt nicht; es wird auch nicht abgefragt, ob überhaupt eine Echtzeit-Uhr im System installiert ist.

Zur Pufferung der Uhr bei abgeschaltetem Gerät muß ein entsprechender Akku vorhanden sein: Bei der 89er-Hardware auf der CPU-Platine 8902x, und bei der 87er-Hardware auf der EZU-Platine 8707x-.

Beispiel 16.2.1:

Um die Echtzeit-Uhr einzustellen, müssen Sie die Uhrzeit und das Datum in drei Parametern vorgeben: Im ersten sind Stunden und Minuten zusammengefaßt (2030 für 20 Uhr, 30 Minuten); im zweiten stehen Wochentags-Codierung und Kalendertag (318 für Mittwoch, den 18.), und die dritte Eingabe nennt schließlich Monat und Jahr (1285 für Dezember '85):

MOPPEL-Video-Monitor V 7.5
Copyright (C) hms'85

Anweisung a...Z eingeben:

M>U 2030,318,1285
Mi,18.12.85;20:30:00h
o.k.

V V.24 Output

17. Ausgabe über die V.24-Schnittstelle

Format: V aaaa,eeee>Ret<

(die Anweisung kann ein Kleinbuchstabe sein,
das Leerzeichen kann entfallen, und anstelle
des Kommas können Sie einen Punkt eingeben)

Funktion: Speicherbereich von Anfangsadresse aaaa bis Endadresse
eeee (einschließlich) über die V.24-Schnittstelle aus-
geben.

Die Übertragungsrate steht in ROM-Adresse 001E/1Fh
und die Betriebsart in ROM-Adresse 0060h.

Beachten: Es sind keine Einschränkungen hinsichtlich der Parameter
aaaa und eeee zu beachten.

Hardwaremäßig ist das serielle Interface erforderlich;
bei der 87er-Hardware erfolgt der Peripherie-Anschluß
über die frontseitige 25polige Buchsenleiste, bei der
89er-Hardware wird die Peripherie über die 25polige Buch-
senleiste auf der Bus-Platine angeschlossen.

Beispiel 17.1:

Zur seriellen Datenübertragung, beispielsweise zwischen zwei Computern, können Sie die V.24-Schnittstelle benutzen. Die Übertragungsrandbedingungen werden im EPROM vorgegeben und können bei Bedarf modifiziert werden (durch Umprogrammieren der entsprechenden ROM-Zellen).

Ein im Bereich 9000...AFFH mit dem Editor erstelltes Quellprogramm können Sie beispielsweise wie folgt ausgeben:

MOPPEL-Video-Monitor V 7.5
Copyright (C) hms'85

Anweisung a...Z eingeben:

M>V 9000,afff
o.k.

M>

W V.24 Input

18. Eingabe über die V.24-Schnittstelle

Format: **W aaaa,eeee**>Ret<

(die Anweisung kann ein Kleinbuchstabe sein,
das Leerzeichen kann entfallen, und anstelle
des Kommas können Sie einen Punkt eingeben)

Funktion: Eingabe über die V.24-Schnittstelle ins RAM, beginnend
bei Anfangsadresse aaaa bis Endadresse eeee (einschließ-
lich).

Die Übertragungsrate steht in ROM-Adresse 001E/1Fh
und die Betriebsart in ROM-Adresse 0060h.

Beachten: Der angegebene Speicherbereich von aaaa...eeee muß im
RAM liegen, weil sonst kein Einschreiben erfolgen kann.

Hardwaremäßig ist das serielle Interface erforderlich;
bei der 87er-Hardware erfolgt der Peripherie-Anschluß
über die frontseitige 25polige Buchsenleiste, bei der
89er-Hardware wird die Peripherie über die 25polige Buch-
senleiste auf der Bus-Platine angeschlossen.

Beispiel 18.1:

Zur seriellen Datenübertragung, beispielsweise zwischen zwei Computern, können Sie die V.24-Schnittstelle benutzen. Die Übertragungsrandbedingungen werden im EPROM vorgegeben und können bei Bedarf modifiziert werden (durch Umprogrammieren der entsprechenden ROM-Zellen).

Ein Quellprogramm, das mit dem Editor weiterbearbeitet werden soll, können Sie beispielsweise wie folgt einlesen:

```
MOPPEL-Video-Monitor V 7.5  
Copyright (C) hms'85
```

Anweisung a...Z eingeben:

```
M>W 9000,afff  
o.k.
```

```
M>
```

X Exchange

19. Speicherbereich umkopieren und Adressen umrechnen

Format: X aaaa,eeee,zzzz>Ret<

(die Anweisung kann ein Kleinbuchstabe sein,
das Leerzeichen kann entfallen, und anstelle
der Kommas können Sie auch Punkte eingeben)

Funktion: Umkopieren eines Speicherbereichs von der Anfangsadresse
aaaa bis zur Endadresse eeee (einschließlich) in einen
RAM-Bereich, der bei Zieladresse zzzz beginnt.

Dabei werden diejenigen Adressen an den Zielbereich an-
gepaßt, die vor dem Umkopieren im Bereich aaaa...eeee
lagen.

Nach erfolgtem Umkopieren und Umrechnen der Adressen er-
scheinen die ersten 16 Bytes des Zieladreibereichs zur
Kontrolle hexadezimal gelistet.

Beachten: Es sind keine Einschränkungen hinsichtlich der Parameter
aaaa und eeee zu beachten; Sie können also auch Monitor-
Programme aus dem EPROM ins RAM umkopieren, um sie dort
laufen zu lassen und zu analysieren. Der Ziel-Adreibere-
ich ab zzzz muß allerdings im RAM liegen, weil sonst
keine Modifikation der Absolutadressen erfolgen kann.

Beispiel 19.1:

Wenn Sie Maschinenprogramme im RAM verschieben wollen, etwa beim Einfügen oder Löschen von Bytes, ändern sich die darin enthaltenen Absolutadressen; beim Umkopieren mit der X-Funktion erfolgt dabei die automatische Anpassung.

Auf diese Weise können Sie natürlich auch Programme aus dem Monitor ins RAM verlagern, um sie dort beispielsweise im Einzelschritt-Betrieb zu verfolgen. Das Unterprogramm DELY1, das eine Zeitverzögerung von 1 ms erzeugt, befindet sich im Adreßbereich 04AF...04BEh (vgl. ganz unten). Nach dem Umkopieren mit der X-Funktion ist es im RAM lauffähig, nachdem die Sprungadresse angepaßt worden ist (vgl. auch Beispiele 13.1 und 14.1):

MOPPEL-Video-Monitor V 7.5
Copyright (C) hms'85

Anweisung a...Z eingeben:

```
M>X 4af,4be,2800
2800 F5 3A 06 00 87 87 3C 87 87 87 3D C2 0A 28 F1 C9
o.k.
```

```
04AF F5          DELY1: PUSH PSW          * 1 ms
04B0 3A 06 00    LDA FQUARZ          *4/6/8 MHz!
04B3 87          ADD A
04B4 87          ADD A
04B5 3C          INR A              *adjust to 99%
04B6 87          ADD A
04B7 87          ADD A
04B8 87          ADD A              *Zeitfaktor *32
04B9 3D          DELOP: DCR A
04BA C2 B9 04    JNZ DELOP
04BD F1          POP PSW
04BE C9          RET
```

Y Byte

20. Einzelzyklus-Betrieb

Format: Y aaaa>Ret<

(die Anweisung kann ein Kleinbuchstabe sein,
und das Leerzeichen kann entfallen)

Funktion: Byteweise Programmausführung ab Adresse aaaa mit Status-
anzeige von Adreß-, Daten- und Steuerbus.

Das Weiterschalten zum nächstfolgenden Byte geschieht mit
dem dafür vorgesehenen Taster auf der CPU (89er-Hardware)
bzw. auf dem Einzelschritt-Modul (87er-Hardware).

Beachten: Es sind keine Einschränkungen hinsichtlich des Parameters
aaaa zu beachten; es können auf diese Weise also Program-
me im RAM oder ROM abgearbeitet und analysiert werden.

Hardwaremäßig sind hierfür die CPU plus LED-Einheit er-
forderlich (bei der 89er-Hardware) bzw. das Einzel-
schritt-Modul (bei der 87er-Hardware).

Beispiel 20.1:

Um eine Programmausführung in allen Einzelheiten nachzuvollziehen, bedienen Sie sich der Byte-weisen Befehlsausführung. Dabei können Sie genauestens verfolgen, was sich auf dem Daten- und Adreßbus im einzelnen vollzieht.

Da dies eine hardwaremäßig erzeugte Betriebsart ist, kann hierbei kein paralleler Protokolldruck ablaufen.

Die Ausführung des Programmbeispiels aus 19.1 vollzieht sich im Y-Mode wie folgt (Darstellung auf der LED-Einheit):

MOPPEL-Video-Monitor V 7.5
Copyright (C) hms'85

Anweisung a...Z eingeben:

M>Y 2800

1CA9	E9	PCHL	Monitor-Befehl; Sprung ins Anwenderprogramm
2800	F5	PUSH PSW	OpCode "F5" holen und decodieren
2EFD	xx		Ausführung (Teil 1): REG A in den Stack
2EFC	xx		Ausführung (Teil 2): REG F in den Stack
2801	3A	LDA 0006	OpCode "3A" holen und dec. (Drei-Wort-Befehl)
2802	06		2. Byte des Befehls holen (untere Adreßhälfte)
2803	00		3. Byte des Befehls holen (obere Adreßhälfte)
0006	06		Adresse auf Adreßbus und Inhalt auslesen
2804	87	ADD A	nächsten OpCode holen, decodieren
...			und so fort
...			

Z Break

=====

21. Haltepunkt setzen

Format: Z aaaa,eeee>Ret<

(die Anweisung kann ein Kleinbuchstabe sein,
das Leerzeichen kann entfallen, und anstelle
des Kommas können Sie einen Punkt eingeben)

Funktion: Programm bei Adresse aaaa starten und vorher die CPU-Register laden (vgl. Anweisung "G=Go"); es folgt die Programmausführung unter Echtzeit-Bedingungen.

Bei Adresse eeee stoppt die Programmausführung und wird im Einzelschritt-Betrieb fortgesetzt (vgl. Anweisung "S = Step").

Beachten: Es sind keine Einschränkungen hinsichtlich des Parameters aaaa zu beachten; der Programmstart mit anschließender Echtzeit-Programmausführung kann also auch im ROM erfolgen.

Lediglich die Haltepunkt-Adresse eeee muß im RAM liegen, weil hierfür dieselben Randbedingungen gelten wie für den Einzelschritt-Betrieb (vgl. Anweisung "S=Step").

Beispiel 21.1:

Eine Möglichkeit, die Programmausführung im Einzelschritt-Betrieb abzukürzen, bestand in der Modifikation von Registerinhalten (vgl. Beispiel 14.1).

Eine andere ist das Setzen eines Haltepunktes ("Breakpoint"), bei dem die Programmausführung stoppt; dies ist insbesondere dann von Bedeutung, wenn bestimmte Programmteile unter realen Zeitbedingungen durchlaufen werden müssen, ehe man die folgenden Operationen analysieren kann.

Im Beispiel 14.1 wird die Adresse 280Eh erst erreicht, wenn die Programmschleife entsprechend oft durchlaufen wurde. Wenn Sie diese stupide Tätigkeit überspringen wollen, setzen Sie einfach dort den Haltepunkt ein und starten wie zuvor bei 2800h:

MOPPEL-Video-Monitor V 7.5
Copyright (C) hms'85

Anweisung a...Z eingeben:

M>Z 2800,280e

Adr.	Inh.	A	B&C	D&E	H&L	SP	SZxcxPxC
280E	F1	00	5900	0008	2FCE	2F6E	01000010
280F	C9	00	5900	0008	2FCE	2F70	00000000

=====
22. Anwenderfunktion starten

Format: § aaaa,eeee,zzzz>Ret<

(die Anweisung kann ein Kleinbuchstabe sein,
das Leerzeichen kann entfallen, und anstelle
der Kommas können Sie auch Punkte eingeben)

Funktion: Programm ab Adresse F000h starten und die Parameter aaaa,
eeee und zzzz im RAM übergeben; dabei stehen

aaaa in den RAM-Adressen SRCBEG = 2FBF/C0h
eeee in den RAM-Adressen SRCEND = 2FC1/C3h und
zzzz in den RAM-Adressen DSTBEG = 2FC3/C4h

Beachten: Es sind keine Einschränkungen hinsichtlich der Parameter
aaaa, eeee und zzzz zu beachten.

Die Zieladresse F000h wurde deshalb gewählt, weil hier-
hin der Objektcode eines assemblierten Programms ge-
schrieben wird (sofern nicht mittels "OFS" etwas anderes
bestimmt wird).

Mit Hilfe der User-Anweisung kann ein assembliertes Pro-
gramm also unmittelbar bei F000h gestartet werden, und es
kann bei Bedarf die über den Bildschirm vorgegebenen Pa-
rameter aus den entsprechenden RAM-Zellen übernehmen.

Beispiel 22.1:

Zum Testen eigener Programme kann es sehr nützlich sein, diese mit bestimmten Randbedingungen auszuführen, um die Reaktion im einzelnen zu überprüfen.

Das ganz unten gelistete Assembler-Programm übernimmt beispielsweise die drei Parameter, die per Tastatur (und Bildschirm) vorgegeben worden sind; nach Durchlaufen des Befehls in Zeile 12 sind die Register (-Paare) wie folgt geladen:

B&C mit 0100h,
D&E mit 2E7Fh und
H&L mit 9000h.

MOPPEL-Video-Monitor V 7.5
Copyright (C) hms'85

Anweisung a...Z eingeben:

M>S 100,2e7f,9000

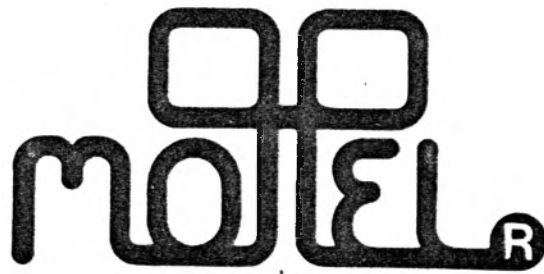
SRCBEG SRCEND DSTBEG

MOPPEL-Assembler V 8.5
Copyright (C) hms'85

0/1/2/3: Pass #
D: Disassembler
E: Editor
F: Floppy
H: HEX-Mode
I: In/Out on/off
M: Monitor
O: OCT-Mode
S: Symbol Table

A>3

0001	0000		ORG	0F000H	*Programmumfang bei F000h
0002	0000		*		
0003	F000		SRCBEG	EQU 2FBFH	*lt. Basis-Symboltabelle
0004	F000		SRCEND	EQU 2FC1H	
0005	F000		DSTBEG	EQU 2FC3H	
0006	F000		*		
0007	F000	2A BF 2F	TEST:	LHLD SRCBEG	*2FBF/C0h nach H&L
0008	F003	4D		MOV C,L	
0009	F004	44		MOV B,H	*B&H: (SRCBEG)
0010	F005	2A C1 2F		LHLD SRCEND	*2FC1/C3h nach H&L
0011	F008	EB		XCHG	*D&E: (SRCEND)
0012	F009	2A C3 2F		LHLD DSTBEG	*H&L: (DSTBEG)
0013	F00C		...		



MOPPEL ist ein
eingetragenes Warenzeichen
der Firma **hms**, Bremen

Entwickeln und Lernen

modulares Prozessor-Programm zum

MOPPEL ist ein

