

Allgemeines Do,12.01.89;00:35:52h  
Sicherungskopie + Installation der DRIVES  
Installation des Druckers  
Installation sonstiger Parameter  
Terminal-Sequenzen  
CT-Codes  
HILFE: ALLGEMEINES

HILFE IM ONLINE-BETRIEB FÜR DAS IK/NS-BIOS  
-----

Di,03.01.89;22:29:27h

Dieses Hilfe-File wird fortlaufend überarbeitet und so allmählich auf das richtige Niveau gebracht. Die einzelnen Abschnitte enthalten alle das jeweilige Bearbeitungsdatum, so daß der erfahrene Anwender später lediglich die "UPDATES" studieren muß.

Das IK-Bios und das neue NS-Bios sind zu einer Gesamt-Einheit verschmolzen worden:

Die Federführung ist wie folgt aufgeteilt:

Nicolaus Seibel <NS>: FLOPPY-BETRIEB & UTILITIES  
Ingolf Kühn <IK> : TERMINAL-BETRIEB & allgemeine Organisation

Die Einnahmen aus Neu-Auslieferungen und UPDATES werden intern zwischen uns aufgeteilt. Also leider nicht mehr ganz der alte Preis, aber mit mehr Leistung und mehr KNOW-HOW für Euch und halbe Einnahmen für uns. Mehr ist also wirklich nicht drin, wenn das ganze Projekt nicht scheitern soll. Die Koordinierung zwischen uns beiden ist äußerst zeit- und kostenintensiv (lange Ferngespräche zur Tageszeit).

Die Features dieses BIOS sind enorme Flexibilität in jeder Hinsicht:

TERMINAL-EMULATION: Standard-Sequenzen für TELEVIDEO und etliche andere Terminals. Damit kann endlich aus den Installations-Programmen der Software Hersteller eine Anpassung vorgenommen werden.

DRIVE-EINSTELLUNG: Es können nun alle möglichen Fremd-Formate eingestellt werden. Außerdem kann jedem logischen Laufwerk ein spezielles physikalisches Laufwerk zugeordnet werden. Besitzer von lediglich 2 Laufwerken können also trotzdem auf C> ... E> zugreifen, indem sie hierfür als physikalische Quelle Drive1 (=B>) angeben.

Im übrigen sind diverse leicht verborgene Eigenschaften implementiert worden:

1. User-Anzeige durch Prompt "0A>"
2. schnelles Bildschirm-Scrollen (Faktor 10), da dies für DF] erforderlich war.
3. Abfangen der BDOS-Fehler-Meldung "Bad Sector"!

Bevor Ihr nun Eure spezielle Anpassung vornehmt, sollte Ihr die bespielten 40 Spuren auf eine Kopie übertragen und die Original-Disk gut weglegen.

>>>>WARNUNG: Bitte unbedingt eine Sicherungskopie fertigen!

Damit nicht jedesmal bei einer Änderung der ganze Text geändert werden muß, wird hier mit den symbolischen Namen gearbeitet. Die zugehörigen Adressen könnt Ihr der Symbol-Tabelle entnehmen. Da mit könntet Ihr nun mit dem DEBUGGERN beginnen.

Die Aufteilung der Disk ist im ANHANG B beschrieben und das Muster eines DPH im sogenannten EXTENDED FORMAT (64 statt 15 Bytes) im Anhang A.

Wem das zu schwierig ist, der sollte <HILFE> unter <B> aufrufen.

An die Adresse der Experten aber folgender wichtiger Hinweis:

Ändert zunächst nicht das Format für 0A>, denn sonst könnt Ihr die Files ja nicht mehr lesen! Ja, Ihr könnt dann auch nicht ein mal mehr die dann unbrauchbaren DIR-Einträge löschen.

Rettet daher erst einmal die Files über ein anderes Laufwerk mit Euren neuen Drive-Parametern und löscht dann mit 0A>ERA \*.\*!

Ändert nun die System-Diskette hinsichtlich der Drive-Parameter für Drive A von dem anderen Drive aus und kopiert anschließend alle Files zurück nach 0A:!

Änderungen auf der System-Diskette sind dem Betriebssystem zunächst jedoch nicht bekannt. Daher ist erst einmal ein RESET mit neuem BOOTEN fällig!

Denkt einmal intensiv über die letzten Zeilen nach und wartet auf Euer geistiges "AHA!". Dann könnt Ihr loslegen. Nervt uns mit Rückfragen aber bitte nur, wenn es wirklich nicht mehr anders geht. Ihr blockiert uns nämlich für die Erledigung inzwischen längst überfällig gewordener Dinge. Habt Erbarmen mit uns. Ja, wenn es nur ein Einzelfall wäre, ließe es sich vielleicht noch ertragen! Versucht bitte möglichst Hilfe über andere USER zu erhalten, die 'no problems' hatten. Schaut einmal die Liste unter IK/NS-USER-GROUP durch, die extra hierfür vorgesehen ist.

DANKE! DANKE! DANKE!

P.S.: Dieser Text wurde mit WORDSTAR erstellt und kann daher nicht ohne weiteres per "TYPE" gelistet werden oder über den Drucker ausgegeben werden. Damit dies doch möglich ist, wurde der Text über ein "hauseigenes" Konvertierprogramm geschickt, daß alle WS-Codes umkonvertiert: z.B A0H→20H usw. Es kann nun leicht passieren, daß nach der letzten Konvertierung irgendetwas redigiert worden ist und dann die erneute Konvertierung vergessen worden ist. Versucht dann bitte mittels ED oder ähnlichem Editor (auch PIP ist möglich) alle MSB zu löschen. Meist reicht dies schon aus.

:HILFE: INSTALLATION DER DRIVES + SICHERUNGSKOPIE

SICHERUNGSKOPIE Di,10.01.89;23:48:06h

Leere Diskette in B: einlegen und folgenden Befehl von DRIVE A: aus eingeben:

0A>mformat

Nun die Diskette in B: mit 40 oder auch 80 Tracks SS formatieren. Aber bitte ohne DOUBLE STEP! Wählt bitte die richtigen Parameter: nämlich 5 Sektoren und 1024 Bytes! Auch müßt Ihr die intelligente Frage <System-Spuren mit 16x256> bejahen!

Entfernt nun auch die SYSTEM-Diskette aus DRIVE A: und legt auch hier eine leere Diskette ein, die Ihr jetzt formatiert. Nun muß wieder die SYSTEM-DISK in DRIVE A: eingelegt werden, damit MFORMAT jetzt über "E" wie EXIT verlassen werden kann. Fehlt nämlich jetzt die System-Diskette in DRIVE A:, kann der eingebaute Warmstart für die Rückkehr zum System (=Nachladen von CPM) nicht ausgeführt werden!

Jetzt die gesamte Diskette von A: nach B: umkopieren! Hierzu ruft Ihr MCOPY als 0A>mcopy b:=a: auf und wählt OPTION "D" wie ganze Disk! Anschließend verläßt Ihr das Programm am besten per ^C!

Nun loggt Ihr DRIVE B: ein und ruft MCOPY ohne Zusatz auf! Dann entfernt Ihr die

heilige SYSTEM-DISK und legt in A: die hierfür formatierte Diskette ein und gebt das Kommando "a:=b:" ein. Als OPTION wählt Ihr auch hier "D" für ganze Disk. Wenn Ihr jetzt per ^C aussteigt, habt Ihr in A: eine Sicherungskopie. Testet es aus! Also RESET und dann <FCT>+<. > sowie ein Test per HELP BIOS40!

Wenn Ihr wollt, könnt Ihr Euch ja gleich noch einige Kopien ziehen!

-----

INSTALLATION DER DRIVES Do,12.01.89;00:35:30h

Ihr solltet nun schon einmal eine vorhandene Diskette bereitlegen, die DDT oder etwas ähnliches sowie ein Kopier Programm zum file-weisen übertragen bereitlegen!

Ihr könnt nun entweder die System-Diskette direkt mit einem Utility (DU, SZAP o.ä.) bzw. von der MONITOR-Ebene aus ändern.

Ihr könnt aber auch lediglich BIOS.SYS ändern und dieses geänderte BIOS.SYS dann mittels PUTBIOS auf die System-Diskette übertragen. Das sieht nicht ganz so gefährlich aus. Dann solltet Ihr alle Files der Sicherungskopie auf eine entsprechend vorformatierte DISK in Drive B: übertragen. Vorher könnt Ihr und müßt Ihr auf A: natürlich erst einmal besagtes Kopier-Programm und DDT nach A: gebracht haben! Also mit SDPB auf B: das Format Eurer DDT-Diskette einstellen und dann die Diskette mit DDT einlegen und dieses Laufwerk per B: einloggen. Testen kann nicht schaden.

Daher solltet Ihr Euch davon überzeugen, daß die Formateinstellung per SDPB auch wirklich dieses Wunder vollbracht hat. Dann die erforderlichen Files nach A: auf die Sicherungskopie übertragen. Nun wieder A: einloggen und in B: eine vorformatierte Diskette einlegen. Dann per SDPB für B: ein passendes Format (am besten und sichersten wieder X140) anwählen. Jetzt sind alle Files von A: nach B: zu save bzw. zu kopieren. Nun alle FILES auf A: löschen. Jetzt könnt Ihr wieder B: einloggen und dann nach Herzenslust BIOS.SYS manipulieren. Wenn angeblich alles geändert ist, ruft Ihr PUTBIOS auf und fertig ist Eure Wunschdiskette, allerdings ohne Files. Also schnell noch alle Files von B: nach A: zurückkopieren! Ja, das muß gehen - selbst wenn Ihr für B: in der Datei BIOS.SYS ein ganz anderes Format für B: eingerichtet habt. Diese Änderung ist ja dem RAM und damit dem derzeit aktivem BIOS noch gar nicht bekannt. Theoretisch hättet Ihr die ganze Operation auch von DRIVE A: ausführen können. Nur aus Gründen doppelter Sicherheit und für Euch als kleine Übung wurde die beschriebene Verfahrensweise gewählt. Das wäre es dann! Aber ein Test per RESET und erneutes BOOTEN ist sicherlich zu empfehlen!

Es geht aber noch einfacher! Lest aber hierfür auf jeden Fall den ganzen folgenden Text sehr genau!

Zuerst müßt Ihr Euch eine System-Diskette für DRIVE mit der erforderlichen Endformatierung (also ggf. 2\*80 mit 5\*1024 inkl. System-Spur-Formatierung a 16\*256) herstellen. Auf diese DISK müßt Ihr eine Kopie der Systemspuren übertragen. Diese dann wirklich leere (also physikalisch oder klinisch saubere DISK) wird gleich benötigt!

Im Drive A: muß nun aber zunächst wieder eine System-Disk stecken, die auch DDT und ein File-Kopier-Programm (z.B. PIP) enthält. Stellt nun mittels SDPB einfach für alle DRIVES (B:, C:, D:, E:) die gewünschten Formate ein. Für A: könnt Ihr dies noch nicht tun, da ja dann die Files nicht mehr lesbar wären! Sie müssen also wieder gelöscht sein! Aber ganz einfach löschen reicht hier nicht aus! Rettet erst einmal die Files "fileweise" von A: nach B: Dafür muß aber zunächst auf B: eine Diskette für das neue eingestellte Format formatiert werden. Nach dieser Umkopiererei auf B: kann endlich B: eingeloggt werden. Für DRIVE A: ist jetzt per SDPB das gewünschte Format einzustellen und die "leere" System-Diskette in Drive A: einzulegen! Nun muß der aktuelle RAM-Inhalt auf die vorbereitete "klinisch saubere" DISK in DRIVE A: gebracht werden. Also ein

passendes BIOS.SYS per DDT und dem Befehl MOVE nach Adresse 100H bringen und dann per SAVE abspeichern (vgl. Näheres unter <UTILITIES> und hier insbesondere zu BIOS.SYS!). - Aber bitte alle Operationen von B: aus vornehmen, so daß jeder Zugriff auf A: unterbleibt (außer Nachladen von CPM)!

Nun erst einmal eine kleine Pause und Besinnung auf das bisher Durchgeführte! Durch SDPB wurde der RAM-Inhalt mit dem aktuellen BIOS so verändert, daß dort die Parameter für die ausgewählten DISK-Formate stehen. Dieses aktuelle BIOS wurde dann als BIOS.SYS auf Diskette gespeichert und kann von Euch als persönliches oder privates BIOS betrachtet werden. Es sollte auch gut geschützt werden. Falls doch noch etwas nachgebessert werden soll, braucht Ihr nicht jedes Mal von vorne anfangen: es sind ja nur in aller Regel ein paar "Kleinigkeiten" vergessen oder falsch gemacht worden. Schaut Euch Euer Produkt doch einfach mittels DDT nochmals gründlich an!

Irgendwann habt Ihr dann (es ist noch immer DRIVE B: aktiv!) das Finale einzuleiten, nämlich PUTBIOS aufzurufen. Euer BIOS steht jetzt auf der Diskette in DRIVE A:

Vertrauen ist gut, aber Kontrolle ist noch besser. Auch hier muß daher ein TEST über RESET durchgeführt werden. Falls nach dem erneuten BOOTEN alles in Ordnung ist, werdet Ihr vielleicht sagen, daß alles ganz einfach war. Kompliziert war alles eigentlich nur, weil ich praktisch jeden Handgriff pingelig genau beschrieben habe. Die Menge des geschriebenen Textes mag daher zunächst Furcht und Panik ausgelöst haben. Es sollte aber - gerade für weniger Geübte - erreicht werden, daß nicht wegen einer kleinen "Dummheit" oder Unachtsamkeit plötzlich "nichts mehr geht".

Doch noch seid Ihr nicht ganz fertig. Wahrscheinlich wird niemand per SDPB oder sonstwie für ein 40-Spur-Laufwerk ein 80-Spur-Format wählen. Aber umgekehrt sollte auf einem 80er durchaus ein 40-Spur-Format gefahren werden können - natürlich ohne Umschalter. Das Laufwerk muß nur einen sogenannten DOUBLE STEP ausführen, aber nur wenn die Bedingung erfüllt ist. Dazu müßt Ihr dem BIOS schon irgendwie mitteilen, daß ein 80er eingesetzt ist.

Aus diesem Grund wurde der automatische DOUBLE-STEP-BETRIEB für alle vorhandenen DRIVES bewußt unterdrückt, damit nicht plötzlich beim ein unplanmäßiger DOUBLE STEP auftritt und alles noch mehr kompliziert hätte, zumindest hinsichtlich Auslieferung und einheitlicher DOKUMENTATION. Daher erst hier an allerletzter Stelle der erforderliche Hinweis. Ich habe lange über die Art der geeigneten Darstellung gegrübelt.

Zu den sogenannten hardwareabhängigen Parametern des USERS gehören auch sonstige physikalische Eigenschaften der DRIVES wie z.B. STEPRATE! Diese Parameter ändern sich ja nicht in Abhängigkeit vom eingestellten Format und werden deshalb auch nicht von SDPB überschrieben. Die Parameter stehen jeweils in den letzten 8 Bytes des 64 Bytes langen DPBs. Studiert nun einmal in aller Ruhe ANHANG A!

Hier noch ein paar kleine Tips. Wenn Ihr Änderungen im File BIOS.SYS vornehmt, solltet Ihr meinen Vordruck zur Unterstützung nehmen und Euch eine kleine Umrechnungstabelle fertigen:

```
100 - EA00
200 - EB00
300 - EC00
... - ....
1300 - FC00
```

Bei der üblichen Darstellung per DDT oder anderer Programme zeigt ein DEBUGGER immer 16 Bytes pro Zeile. Damit alles für Euch schön adrett aussieht, wurde der Beginn eines jeden DPB auf eine glatte Adresse gelegt: xxx0

Da der DPB 64 Bytes umfaßt, handelt es sich folglich um exakt 4 Zeilen. Es steht also alles schön sauber untereinander. Dabei bedeuten:

1. Zeile: STANDARD-DPB gem. üblicher CPM-Konvention. Dieser normale DPB umfaßt allerdings nur 15 Bytes. Deshalb wurde aus optischen Gründen ein sog. "Führungsbyte" spendiert. Es enthält die DRIVE-Kennung für A,B usw., also HEX 41, 42 usw. Im ASCII-Feld könnt Ihr dann im Abstand von jeweils 4 Zeilen die Buchstaben A...E finden, so daß Ihr immer eine Kontrolle habt, wo Ihr gerade herum-DEBUGGERT. 2. Zeile: Zeile 2 und 3 enthalten die Codierung für die 3. Zeile: Reihenfolge der Sektor-Tabelle. Hiermit wird Seitenumschaltung per Sektorübersetzung und das Setzen von SKEW-Faktoren ermöglicht. Nicht vorhandene Sektoren sollten mit NULLEN aufgefüllt werden. 4. Zeile: Dies ist der eigentl. EXTENDED DPB. Die ersten 8 Bytes (xxx0 -xxx7) definieren das Format. Die letzten 8 Bytes (xxx8 - xxxF) enthalten die oben genannten HARDWARE-PARAMETER, die Ihr ja noch ändern sollt!

Die 5 DPBs (A - E) stehen auf folgenden Adressen:

```
A EA90 1. ZEILE
EAA0 2. ZEILE
EAB0 3. ZEILE
EAC0 4. ZEILE
B EAD0

C EB10

D EB50

E EB90
```

Bei dieser glatten Adressen-Einteilung sollte auch der unerfahrene MOPPEL-Freund keine Angst beim Einsatz eines DEBUGGERS haben! Falls Ihr Probleme hinsichtlich der Werte für die eigentlichen PARAMETER habt, ruft doch einfach per SDPB das gewünschte Format für ein nicht "aktives" Laufwerk auf und schaut Euch das erzeugte PARAMETER-Ergebnis per DDT einfach im RAM an und schreibt es ggf. einfach ab!

Schließlich noch abschließend der HINWEIS, daß Ihr die SYSTEM Spuren ja auch mit den MOPPEL-FDC-UTILITIES ändern könnt. Vielleicht ist diese DEBUGGER-Art für den einen oder anderen doch die vertrautere und damit risikoloseste Form! Wie dem auch sei, Ihr müßtet alle in der Lage sein, irgendwie Eure ureigene maßgeschneiderte SYSTEM-MASTER-DISKETTE herzustellen. Rückfragen sollten daher wirklich vermieden werden. Wenn aber alle "Stricke reißen", dann schätzt Euch glücklich, daß Ihr für die vorhandene SOFTWARE eine HOT-LINE zu den SYSTEM-PROGRAMMIEREN habt. Macht aber bitte vom Telefon und nicht vom Drucker Gebrauch. Mündlich lassen sich wirklich alle Probleme lösen, nicht aber mit zeitraubender und daher schleppend verlaufender Korrespondenz.

Ich betrachte daher auch alle zwischenzeitlich mir zugegangene schriftliche und noch unerledigte Anfragen zur Vorabversion hier mit für erledigt. Das Leben geht nun endlich weiter bzw. fängt erst an. Ich freue mich, vielleicht demnächst einmal den MOPPEL auch für mich privat nutzen zu können!

:HILFE: INSTALLATION DES DRUCKERS  
Sa,07.01.89;21:53:30h

Im Boot-Sektor (Track 0/Sektor 1) befinden sich 3 Bytes, die beim BOOTEN ins BIOS kopiert werden. Lest den BOOT-Sektor nach 2800 ein und ändert wie folgt und schreibt entsprechend zurück:

```
28D8: 80 I/O-Byte: 10 00 00 00
                   /  /  /  /
                   /  /  /  +----- CON: =TTY
                   /  /  +----- RDR: =TTY
                   /  +----- PUN: =TTY
                   +----- LST: =LPT
```

```
28D9: 00 USER/DRIVE ----> hier also: User 0/Drive A (0A>)
28DA: 20 Printer im MODE "PARALLEL-IF"
      10 Thermo-Drucker
      20 Parallel-IF
      40 Centronics
      80 Seriell
```

:HILFE: INSTALLATION SONSTIGER PARAMETER

Sa,07.01.89;23:32:09h

Es gibt einige voreingestellte Parameter (=DEFAULT), die aber geändert werden können:

```
TVMOD: 00 IK-Mode (DEFAULT BY NS)
      01 TVI-Mode (DEFAULT BY IK)
```

```
ASCII: 00 DIN (DEFAULT)
      FF ASCII
```

TVMOD=00 bedeutet, daß kein echter Televideo-Mode und TVMOD=01 bedeutet, daß echter TVI-Mode vorliegt. Der Unterschied macht sich nur bei "<sup>^</sup>L" bemerkbar. Bei TVI bedeutet dies <CLS>+<HOME>, was den Bildschirm löscht und den Cursor nach oben links positioniert. Mit Rücksicht auf evtl. vorliegende Software kann es erforderlich sein, daß der Cursor besser in der letzten Zeile erscheint. Dann kann durch Setzen dieses alten IK-Modus diese Option erreicht werden. Auf Dauer sollte natürlich die Software geändert werden! Zur Erleichterung kann die Umschaltung auch im RAM-BIOS durch ESC-Sequenz erreicht werden:

```
ESC^VV0 = IK-Mode
ESC^VV1 = TVI-Mode
```

In der vorliegenden BIOS-Version ist also - anders als bei früheren Versionen - DIN voreingestellt. Diese Option läßt sich nach wie vor durch ESC-Sequenz ändern:

```
ESC^VZD = DIN
ESC^VZA = ASCII
```

Darüber hinaus kann DIN durch <FCT+D> und <CR> und ASCII durch <FCT+A> und <CR> erreicht werden.

:HILFE: TERMINAL-SEQUENZEN  
Mo,09.01.89;00:50:59h

Bisher enthielt das IK-BIOS viele zusätzliche Terminal-Sequenzen, die beim HMS-BIOS nicht vorhanden, bzw. falsch waren. Das IK-BIOS war insoweit umfassender. Nun hat HMS einiges nachgebessert. Leider hat er hierfür abweichende Kodierungen gewählt, so daß es nun keine Kompatibilität mehr gibt. Da er meine Codierungen aufgrund einer früheren Muster-Diskette und meiner Abhandlungen in den MOPPEL-NEWS kannte, muß hier insoweit bewußte Inkompabilität angestrebt worden sein. Was soll's!

Die HMS-Terminal-Eigenschaft ist weder normgerecht noch ausreichend, da ja nicht einmal die wichtige OPTION LININS vorhanden ist. Von daher ist praktisch überhaupt keine STANDARD-SOFTWARE halbwegs brauchbar installierbar. Viel schlimmer ist der schräge OFFSET (Zeile=01 - Spalte=00), worauf die Standard-Software nun mal wirklich nur ausnahmsweise (z.B. WORDSTAR) Rücksicht nimmt.

HMS hat ab Version xx.6 neue ESCAPE-Sequenzen eingeführt, die im IK-BIOS bisher fehlten und für V 4.0 versprochen worden sind:

```
ESC ^N 1B 0E INVERS OFF - nichts Neues
ESC ^O 1B 0F INVERS ON - " "
```

```
...
ESC ^Z YY XX 1B 1A YY XX JUMP XX YY
ESC ESC 1B 1B EZU LESEN + SCHREIBEN
ESC ^\ II IO 1B 1C II IO PRINTER INIT + IO-FLAG
ESC ^] XX YY 1B 1D XX YY PRINTER OUT: XX YY
```

Diese neuen Codes sind nun implementiert worden. Darüber hinaus können alle Terminal-Features dieses BIOS komplett abgeschaltet werden, so daß ein reines HMS-Terminal emuliert wird. Natürlich kann auch das HMS-Terminal wieder abgeschaltet werden. Diese Umschaltung erfolgt ausschließlich über die Tastatur per HMS-IO-Flag

(Bit ---- --X-). Die Umschaltung erfolgt also per ^1!

```
IO-Flag: xxxx xx0x IK-Mode
IO-Flag: xxxx xx1x HMS-Mode
```

Wesentlich ist aber, daß das TELEVIDEO-Terminal (=TVI) emuliert wird. Damit auch möglichst viele andere Terminals, die TVI nachahmen, eingebunden sind, wurden noch einige andere Codes aus der ADM-Serie eingeschlossen. Ihr müßt aber einsehen, daß Farb- oder Grafik-Befehle nicht emuliert werden können! Ähnliches gilt für Sonderfunktionen wie BREITSCHRIFT, INDEXSCHRIFT usw. Es wurde aber versucht, diese Codes abzufangen und damit zu neutralisieren. Bei der Installation von Marken-Software könnt Ihr zwar eine Installation mit TELEVIDEO 950 versuchen und bei Mißerfolg ggf. auf TELEVIDEO 910 heruntergehen. Als heikel hat sich lediglich MULTIPLAN erwiesen, das generell voll von den grafischen Möglichkeiten Gebrauch macht. Dort muß ggf. per Hand und Einzelanweisung eine Installation vorgenommen werden. Mir ist es z.B. gelungen, sogar die Grafik-Zeichen für WINDOWS in MULTIPLAN einzubinden, so daß "RAHMEN" und "AUSSCHNITT" richtig professionell aussehen. Hier ein paar Möglichkeiten ausgetesteter Fertig-Installationen per Aufruf über Installations-Menü:

MULTIPLAN:

```
-----
- ADM 3A Eingabefeld-Begrenzung als Ä.....]
- ADM 21 "
- SOROC IQ120 "
```

TURBO-PASCAL:

```
-----
- SOROC
- APPLE CPM
- TELEVIDEO
- ADM 3A/31
- KAYPRO
- OSBORNE
```

WORDSTAR:

```
-----
- TELEVIDEO 910/950
- SOROC 120/140
```

dBASE II:

```
-----
- TELEVIDEO
```

So ganz nebenbei sind auch noch einige ITT3030-Besonderheiten berücksichtigt. Ihr werdet Euch wundern, wieviele Software ohne irgendwelche Änderungen sofort auf dem MOPPEL lauffähig ist.

So schön wie diese MEHRFACH-TERMINAL-EMULATION auch ist, besteht doch kein zwingender Grund, die HMS-Codes und alten IK-Codes weiter mitzuschleppen, wenn die TVI/ADM-Familie ähnliche oder gleichlautende Funktionen bietet. Stellt also bitte Eure Software alsbald auf TELEVIDEO um, da die überflüssigen Codierungen

aus Platzgründen irgenvann einmal entfallen sollen. Es folgt nun eine Beschreibung der TVI/ADM-Sequenzen und anschließend eine Synopse aller Codes. Es werden folgende Kurzformen gewählt:

HOME1: Cursor nach links oben  
HOME2: Cursor nach links unten  
CPOS: Cursor-Positionierung  
CLS: Clear SCREEN  
LINDEL: Cursor-Zeile löschen  
LININS: Leerzeile an Cursor-Position einfügen  
ERAEOL: ab Cursor bis Zeilenende löschen  
CPCLS: Clear SCREEN ab Cursor  
INVON: Invers EIN  
INVOFF: Invers AUS

#### TVI/ADM-SEQUENZEN:

-----

ASCII HEX DEZ Bedeutung

-----

#### ONE BYTE:

^G 07 7 BELL  
^H 08 8 BACKSPACE  
^I 09 9 TAB  
^J 0A 10 LF  
^L 0C 12 FF = <CLS>+<HOME1>  
^M 0D 13 CR  
^Z 1A 26 wie ^L  
^^ 1E 30 HOME

#### TWO BYTES:

ESC \$ 1B 24 27 36 MP-Strich-Grafik EIN <in Vorber.>  
ESC % 1B 25 27 37 MP-Strich-Grafik AUS <in Vorber.>  
ESC ( 1B 28 27 40 INVOFF  
ESC ) 1B 29 27 41 INVON  
ESC + 1B 2B 27 43 wie ^L  
ESC : 1B 3A 27 58 wie ^L  
ESC E 1B 45 27 69 LININS  
ESC R 1B 52 27 82 LINDEL  
ESC T 1B 54 27 84 ERAEOL  
ESC Y 1B 59 27 89 CPCLS  
ESC j 1B 6A 27 106 INVON - ONLY ADM!  
ESC k 1B 6B 27 107 INVOFF - ONLY ADM!

#### THREE BYTES:

ESC . 0 1B 2E 30 27 46 48 Cursor: unsichtbar  
ESC . 1 1B 2E 31 27 46 49 Cursor: Blinkender Block  
ESC . 2 1B 2E 32 27 46 50 Cursor: Stehender Block  
ESC . 3 1B 2E 33 27 46 51 Cursor: Blinkende Linie (=DEFAULT)  
ESC . 4 1B 2E 34 27 46 52 Cursor: Stehende Linie  
ESC . 5 1B 2E 35 27 46 53 Cursor: FAST Block  
ESC . 6 1B 2E 36 27 46 54 Cursor: FAST Line  
ESC . 7 1B 2E 37 27 46 55 NOTHING for "USER TWINKLING"  
ESC G V 1B 47 -- 27 71 -- Mit Ausnahme von V=0 und V=4 ist  
alles auf "NOTHING" eingestellt,  
da V Video-Attribute enthält  
ESC G 0 1B 47 30 27 71 48 INVOFF  
ESC G 4 1B 47 34 27 71 52 INVON für UNTERSTREICHEN

#### FOUR BYTES:

ESC = ZS 1B 3D ZS 27 61 ZS Cursor-Positionierung

Z= Zeile + Offset 20H/32D  
S= Spalte + " "

Zeile: 0 ... 23  
Spalte: 0 ... 79

SYNOPSIS DER ESC-SEQUENZEN:  
-----

ASCII Bedeutung TVI IK ITT HMS  
-----

ONE BYTE:

^G BELL x x x x  
^H BACKSPACE x x x x  
^I TAB x x x x  
^J LF x x x x  
^L FF = <CLS>+<HOME1> x  
FF = <CLS>+<HOME2> x x  
CLS x  
^M 0D 13 CR x x x x  
^Z 1A 26 wie ^L x x  
^^ 1E 30 HOME x

TWO BYTES:

ESC ^N INVOFF x x  
ESC ^O INVON x x  
ESC ^P CLS x  
CLS und verlorener Cursor x  
ESC ^Q CPCLS x  
HOME1 x x  
ESC ^R INVON x  
Cursor unsichtbar x x  
ESC ^S INVOFF x  
Cursor: Blinking Line x x  
ESC ^T SCROLL UP ab 5. Zeile x x  
ESC ^U SCROLL DOWN ab 5. Zeile x x  
ESC ^W LINSINS x  
CPCLS x x  
ESC ^X LINDEL x x  
ERAEOL x  
ESC ^Y ERAEOL x x x  
ESC ESC EZU lesen + Ausgabe per C0 x x  
ESC \$ MP-Strich-Grafik EIN <in Vorber.> x  
ESC % MP-Strich-Grafik AUS <in Vorber.> x  
ESC ( INVOFF x  
ESC ) INVON x  
ESC \* wie ^L - nur ADM 42 x  
ESC + wie ^L x  
ESC : wie ^L x  
ESC ; wie ^L - nur ADM 31 x  
ESC E LININS x  
ESC R LINDEL x  
ESC T ERAEOL x  
ESC Y CPCLS x  
ESC j INVON - ONLY ADM! x  
ESC k INVOFF - ONLY ADM! x

THREE BYTES:

ESC . 0 Cursor: unsichtbar x  
ESC . 1 Cursor: Blinkender Block x  
ESC . 2 Cursor: Stehender Block x

ESC . 3 Cursor: Blinkende Linie (=DEFAULT) x  
ESC . 4 Cursor: Stehende Linie x  
ESC . 5 Cursor: FAST Block x  
ESC . 6 Cursor: FAST Line x  
ESC . 7 NOTHING for "USER TWINKLING" x  
ESC G V Mit Ausnahme von V=0 und V=4 ist x  
alles auf "NOTHING"  
eingestellt, da V  
Video-Attribute  
enthält  
ESC G 0 INVOFF x  
ESC G 4 INVON für UNTERSTREICHEN x

FOUR BYTES:

ESC^V ZS 1B 16 ZS 27 22 ZS Cursor-Position.  
Z= Zeile + 0  
S= Spalte + 0  
Zeile: 0 ... 23  
Spalte: 0 ... 79 x

Zeile: 1 ... 24  
Spalte: 0 ... 79 x x

ESC^VC0 Cursor unsichtbar x  
ESC^VC1 Cursor als Blinking Line x  
ESC^VT1 TIME 1: Zeit als C0 x  
ESC^VT2 TIME 2: Zeitstring im RAM ablegen (#F900) x  
Der String wird mit "\$" für BDOS beendet!  
ESC^VV0 IK-Mode für ^L x  
ESC^VV1 TVI-Mode für ^L x  
ESC^VZD DIN-Ausgabe x  
ESC^VZA ASCII-Ausgabe x

ESC^ZYX JUMP XY x x  
ESC^IF Init Printer (I= NOT 00)+ IO-Byte (=F) x x  
ESC^]XY Printer OUT: XY x x

ESC^\_ ZS 1B 1F ZS 27 31 ZS Cursor-Position. x  
Z= Zeile + 20H/32D  
S= Spalte + "  
Zeile: 1 ... 24  
Spalte: 0 ... 79  
ESC = ZS 1B 3D ZS 27 61 ZS Cursor-Position. x  
Z= Zeile + 20H/32D  
S= Spalte + "  
Zeile: 0 ... 23  
Spalte: 0 ... 79

Ja, das war sicherlich eine Menge Stoff, aber seid beruhigt, daß die Programmierung etwas mühseliger als Eure Leselektüre war. Ich hoffe, daß Ihr nun überzeugt seid, daß der MOPPEL endlich ein vollwertiger CP/M-Rechner ist.

Die erwähnte MP-Strichgrafik wird als ZG III auf den Systemspuren abgelegt und für die 89er-Video-Karte eingespielt. Dann fehlen die Klein-Buchstaben. Anstelle dieser Codes sind die Groß-Buchstaben im ASCII-Format vorgesehen. Die bisherigen Codes der Groß-Buchstaben werden dann TVI-gerecht einen besonderen Strichkode enthalten!

Diese Option darf also nur aufgerufen werden, wenn ASCII eingestellt ist, da andernfalls sich an der DIN-Einstellung überhaupt nichts ändern wird. ZG III wird dann nur mit ZG II vertauscht!

:HILFE: FCT-CODES

Di,03.01.89;22:54:25h  
Zur Zeit vorhandene FCT-Codes:

als "Ext" ist jeweils die 87er-Tastatur-Erweiterung gemeint  
--- gilt bereits ab V 3.31 !!! ---

FCT zuzüglich  
-----

D Din  
A ASCII

C CPM-Kaltstart zu CPM 60 KB (EA00)  
Rückkehr von Bank 0 (= 89er MEM)  
0 Ext Bank 0 einschalten (z.B. für FDC)

RET RESET to Bank 0 (softwaremäßig!!!), wichtig für RAM-Disk

1 Ext Bank 1 mit Bank 0 parallel für Rückkehr nach CPM mit  
MEM 87  
F FDC-Warmstart

-----  
Codes for WORD STAR

HOME Ext ^QR TOP OF FILE  
Z ^QC END OF FILE  
UP EXT ^QE TOP OF SCREEN  
DOWN EXT ^QX BOTTOM OF SCREEN  
LEFT EXT ^QS LEFT SIDE  
RIGHT EXT ^QD RIGHT SIDE

-----  
. Ext Batch 0 (=D0): Schnellstart

Zur Realisierung dieser Codes ist auf # 2C40-2CFF einiges an Maschinen-Code  
abgelegt!

Die Codierung der FCT-Codes ist ja auf den Systemspuren untergebracht, so daß  
Ihr beliebige Änderungen vornehmen könnt. Da auch hier vom "SYSTEM" von den FCT-  
Codes Gebrauch gemacht wird, wurde für den USER der Bereich der oberen Tastatur-  
Reihe ausgespart:

also FCT+1, FCT+2 ... FCT+9, FCT+0, FCT+ß und FCT++!

\*\*\*\*\*