

```

##      ##      #####      ##      ##
###     ###     #####     ##     ##
#####  #####  ##         ##     ##
## ##### ##         ##         #####
##  ##  ##         ##         #####
##      ##         ##         ##     ##
##      ##         ##         ##     ##
##      ##         ##         ##     ##
##      ##         ##         ##     ##

```

Info  
7

**MTX User - Club Deutschland**

=====

**Zweck:** Austausch von Tips & Tricks u.s.w. und Hilfestellung bei allen möglichen Problemen.

**Programme (nur Selbstgeschriebenes):** Tausch von kurzen und einfachen Routinen. Besprechung von guten Programmen damit der Autor diese dann an Clubmitglieder verkaufen kann. Programme an uns schicken, und wir liefern Verbesserungshinweise, Besprechung, ...

**Mitglied** kann jeder werden! Keine Aufnahme oder Beitragsgebühr!

**Verpflichtungen:** Einsendung unseres Fragebogens (liegt Info's bei)

**Bitte:** Einsendung von Tips & Tricks, Fragen, Antworten, kurzen Routinen, Programmen, Hinweisen auf preiswerte Hard- und Software, und was noch so zusammenkommt und andere interessieren könnte.

**Club-Info,** unser Blatt, verschicken wir ca. monatlich. Inhalt ist alles was uns über den MTX in die Hände fällt (und kein Copyright hat).

**Kosten:** Wir berechnen ausschließlich Selbstkosten (Porto, Verpackung, Vervielfältigung, Datenträger, ...).  
Wir verschicken **nichts**, wenn's Guthaben nicht reicht! (s.u.)

Da wir unseren Steckbrief nicht nur gegen Freiumschlag verschicken, ziehen wir denen, die ihn geschickt bekommen dafür DM -.70 vom Konto ab. (Einspruch ist selbverständlich jederzeit möglich.)

**Geld/Konto:** Für jedes Mitglied führt Herbert Herberg ein Konto, von dem die entstehenden Kosten jeweils abgehen. Der Kontostand wird bei **jeder** Sendung mitgeteilt (er steht über der Anschrift), und kann selbverständlich jederzeit erfragt werden!

Einzahlungen bitte möglichst auf's Club-Konto: (oder V-Scheck)  
(Absender! incl Name und Anschrift nicht vergessen!)

Postgiroamt Hamburg, BLZ 200 100 20,  
Herbert Herberg Sonderkonto C, Nr. 3480 00-200

**Kontaktadressen:**

Herbert Herberg  
Sonnenau 2  
2000 Hamburg 76  
(040) 200 87 04

Thomas Pflaum  
Leipziger Platz 1  
8500 Nürnberg 20  
(0911) 51 35 21

Moin, moin!

Entschuldigt bitte, daß es mit Info 7 so lange dauerte, aber dafür haben wir jetzt einen Bestellzettel anbei legen können. Laufwerke und Akustikkoppler zu guten Preisen ... Am 20. Juli 1985 habe ich meine Prüfungen hinter mir, und hofflich bestanden! Das versprochene Inhaltsverzeichnis von Thomas Pflaum ist noch nicht ganz fertig. Das kommt daher, daß er erst ein Programm dafür schreiben mußte, und dann alle Infos druchlesen, da im Verzeichnis nicht nur die Überschrift und die Quelle vermerkt ist, sondern außerdem noch bis zu 5 Stichpunkte je Eintrag hinzukommen, damit wir dann auch mal ein Verzeichnis sortiert nach verschiedenen Gesichtspunkten erstellen können. Außerdem übersieht bitte nicht, daß Thomas einen Beruf hat, und auch andere Dinge um die Ohren hat. Jedenfalls ist dieses Info mal wieder recht dick, was die Seitenpreise auch drückt. Da ich eh über die 47-Seiten-Grenze (d.h. Porto 1.40 -> 2.-) bin lege ich auch die Beschreibung von STAT.COM anbei, die ich schon etwas länger habe, und nach der ich immer wieder gefragt werde. Immerhin nehme ich mir die Freiheit eingesandte Artikel zu untersuchen, und falls sie falsch sind, oder ich einen Artikel exakt gleichen Inhalts schon habe diesen auch wegzulassen. Ansonsten verzögere ich höchstens mal etwas, aber ich bringe Eure Artikel alle ins Info!

Ich habe drei Neuerungen im Info! Ja, das gibts, der ändert mal was. Das Deckblatt ist anders aufgebaut, und es gibt eine Verpflichtung für die Clubmitglieder (auch Rückwirkend): Der Fragebogen soll eingeschickt werden. Jede Seite, die hier aus NewWord kommt hat außerdem 'MTX User-Club Deutschland' eine zweite unterstrichene Überschrift, aus der hervorgeht, was auf dem Blatt zu finden ist. Ich hoffe, daß das das Lesen und Blättern erleichtert. Und die Mitgliederliste: Nachname vor Vorname.

Ich werde oft angerufen: 'Ich habe letzte Woche DM .. auf das Clubkonto überwiesen, wann kann ich mit den Unterlagen rechnen?'. Nun ist es so, daß die Post auf diesem Gebiet sehr gründlich (also langsam) ist. Laufzeiten von 10 Tagen sind fast normal! Wer es eilig hat, kann mir einen Verrechnungs-Scheck (muß nicht Euro sein - aber gedeckt) schicken. Ich glaube an Eure Ehrlichkeit, und schicke dann die Unterlagen/Programme umgehend raus. Bitte keine Einschreiben oder Einzahlungsbelege! Wenn ich einen Scheck einreiche fülle ich dafür auch einen Einzahlungsbeleg aus, den ich dann leicht als von mir geschrieben erkenne, und dann wenn die Kontoauszüge kommen weiß ich 'das ist schon gutgeschrieben'. Anders schreibe ich gut und es kommt noch ein Beleg, so daß ich mir immer einen deutlichen Vermerk machen muß!

Wenn Ihr Programme bestellen wollt, wendet Euch bitte immer an denjenigen, unter dessen Namen es steht. Den Flugsimulator von Andreas habe ich nur vorübergehend vertrieben, solange sein MTX 512 in Reparatur war.

**Telefonisch bin ich zu folgenden Zeiten erreichbar:**

**Dienstags 19.30 - 21.30 und Samstags 9.00 - 12.00**

(Ich bin nur Mensch, mit Verpflichtungen, ... Deshalb passiert es ab und zu mal, daß ich zu diesen Zeiten nicht erreichbar bin. Entschuldigt bitte!)

Herbert Herberg

**MTX 500 + FDX + 2 Laufwerke + Drucker + Grün-Monitor + Papier**  
gibt's für DM 3500.- (VB, NP 4300.-) wegen Systemwechsel nach 10 Stunden von Andreas Fuchs, Lindenstr. 5, 8221 Grabenstätt, 08661-1335.

V o n F r ü h e r / H a r d w a r e

**Bisher erschienen** und noch erhältlich: (Herbert Herberg)

ROM-Unterlagen (32 Seiten, 8.-), Info 1 (18 Seiten, DM 4.70), Info 2 (45 Seiten, DM 8.15), Info 3 (46 Seiten, DM 8.60), Info 4 (65 Seiten, DM 11.05), Info 5 (33 Seiten, 6.95), Info 6 (48 Seiten, 9.05). Allerdings dauert die Lieferung ggf. etwas, da ich nur begrenzte Vorräte habe. Wer etwas davon haben möchte muß es mich wissen lassen! Nur das aktuelle Info gibt's automatisch! (Wenn's Geld auf dem Konto reicht!)

**Alt-Info's**

Wie Ihr Euch sicherlich vorstellen könnt sammeln sich allmählich recht imposante Papiermassen in Form alter Info's bei mir, da ich diese ja immer noch habe und weitergebe. Nun muß ich da etwas tun, um nicht in Papier zu ersticken. I.a. wollen nur neue Mitglieder die alten Info's, und dann gleich alle. Deshalb werde ich ab **1. September 1985** nur noch die neuesten 2-4 Info's einzeln haben. Die älteren gibt es dann als gesammelte Werke.

**Hardware**

Zeichensatz-PROM mit eckigen Klammern und echten Oberlängen für die 80-Zeichenkarte (s. Info 6): DM 30.- bei Herbert Herberg.

Tastaturentprellung (d.h. Auseinandernehmen und Zusammenlöten) macht Klaus Korger für DM 30.-

**K o r r e k t u r - N a c h t r a g**

Artikel 'VS4-Grafik unter CP/M' aus Info 6 von Olaf Krumnow:

Wie diejenigen, die die Plot-Routine bereits ausprobiert haben, sicherlich schon merkten, ist die Berechnung von adr falsch. In der vorhergehenden Tabelle muß es richtig heißen:

Spalte	INT(X/8)	X div 8	3* SRL A	(y-Wert)
Position im Zeichen	7-MOD(Y,8)	7-(Y mod 8)	AND 7 NEG ADD 7	(x-Wert)

und in der Plot-Prozedur

```
adr:= 256 * ( 23 - (y DIV 8) ) +
      8 * (      x DIV 8 ) +
      ( 7 - (y MOD 8) ) ;
```

**Nachtrag zu PIP.COM**

Vielleicht hat schon jemand probiert PIP.COM mit einer der möglichen Optionen aufzurufen und sich dann geärgert, das es nicht funktioniert hat. Der Grund ist folgender, PIP.COM erwartet, daß die Optionen in **eckigen** Klammern stehen, und **nicht**, wie von Herbert Gollnik angegeben, in runden Klammern.

**Suchmeldung**

Wo sind die Eckigen Klammern auf der Tastatur abgeblieben. Bei mir PiePt es doch nicht so richtig .... Nur die elementarsten Dinge gehen! Sachdienliche Hinweise nimmt jede Clubdienststelle entgegen.

Programme

Preise sind netto, d.h. ohne Datenträger und P+V (DM 8,-).  
 Die mit einem \* gekennzeichneten Programme gibt es auch als Listing (DM -.30 je Seite); die mit # sind im Info gelistet. Die Programme mit einem ! brauchen den 80-Schirm in der FDX **und** die 40-Zeichen-Gra-  
 Eine Zahl am Ende der Zeile ist die Nummer des Info's mit Besprechung.  
 Ein (C) bedeutet Copyright, d.h. Kopierverbot!

Andreas Viebke (keine Nachnahme - wird zu teuer)

- 16.- Flugsimulator für Linienmaschinen (C) 2
- 12.- # D/E Zeichenvergrößerung und 40 Zeichen auf VS 4 (C) 4
- 20.- Schach-Editor (Schachpartien komfortabel auswerten) (C) 7
- 5.- Kleiner FDXB-Disassembler mit Druckerausgabe (C)
- 42.- XB (Extended FDX-BASIC) (C) 7

Frank Dersewski

- 17.- MDM712 (Modem), FORTH, DU (Disk Utility), RTTY (Teletype), ...

Olaf Krumnow

- 5.- TURBO-Pascal-LISTER (C) 7

Michael Möwe

- 5.- PRT (Druckerinitialisierung mit Zeichensatz) (C) 7

Kurt-Bernd Rohloff

- 0.- SETPRT.COM (Druckereinstellung) 6

Herbert Herberg: (Ich tausche auch!)

- 20.- Quibic (3D-Grafik 4 gewinnt) (C) Herbert Herberg 6,7
- 11.- RAM-Disc V.1 mit Druckerpuffer,.. (C) Bernd Preusing 5,7
- 3.- Fastcopy komfortables COPY (C) Herbert Herberg 6
- 4.- # Charactergenerator incl Zeichensatz
- 1.- \* Character-Designer
- 4.- Labyrinth (durch ein Labyrinth hindurchfinden)
- 3.- Liner (plötzlich auftauchenden Linien ausweichen)
- 3.- Miner (Gold Berg finden)
- 1.- \* Poker
- 2.- Shuttle (Bomben von einer Space-Shuttle fallen lassen)
- 3.- Brio (Geschicklichkeitsspiel: Kugel durch Labyrinth)
- 2.- \* Breakout (Mit Ping-Pong-Verfahren Steine zerstören)
- 2.- \* Mamind (Mastermind = Kombinationsraten)
- 2.- \* Jigsaw (Puzzle)
- 4.- Missile (Städte vor Zerstörung schützen, ASSEMBLER)
- 2.- \* Anschiss (Mind over Electons)
- 1.- \* Was1 (Nimble-Thimble, was immer das sei)
- 2.- Sandburg
- 6.- ! Railroad (Schienen legen mit bis zu 9 Spielern)
- 6.- ! Pferde (Pferdelauf mit mehreren Spielern)
- 4.- War-Plan (Flottenvernichtung)
- 2.- Lunar (Mondlandung, **gut**)
- 2.- # Willie (Ein gefräßiger Wurm)
- 5.- Vier-Gew (Vier gewinnt)
- 0.- Disketten-Konvertierung TA-PC -> MTX
- 2.- Hbigcopy: DMXB0 Hardcopy mit Vergrößerung um den Faktor 2
- 3.- Biorhythmus

NewWord**Encyclopedia**

Nun erst mal zu der berühmt-berüchtigten Encyclopedia: Das ist eine alphabetisch sortierte abgemagerte Version des Do-It-Yourself! Mehr nicht! Darum kann ich sie also auch nicht besorgen. Aber ich habe etwas viel besseres!!! Das User-Area-Listing, und da drin stehen viele schöne Dinge. Dazu seht Euch bitte unten die Patch-Tabelle an!

**So nun noch einige andre Hinweise:**

Verwendung der Trennhilfe (Hyphen-Help). Diese Funktion wird mit ^O^H ein- und ausgeschaltet. Ist sie eingeschaltet fragt NewWord beim Randausgleich mit ^B bei längeren Worten, wo diese getrennt werden dürfen. Dazu wird das Wort gezeigt, und man kann auf den 1. Buchstaben, der in die nächste Zeile soll gehen und ein - (Bindestrich) eingeben. Dann wird dort getrennt. Darf nicht getrennt werden, so gebe ^B ein. NewWord merkt sich übrigens diese erlaubten Trennstellen. Wer sich die Drucker-Kommandos auf dem Bildschirm anzeigen läßt (Ein- und Ausschalten mit ^O^D) sieht diese erlaubten Trennstellen als =, wenn sie nicht am Ende einer Zeile sind. Sie werden nicht gedruckt! Beim Eingeben eines Textes wird natürlich nicht nach Trennstellen gefragt, da das da nur stören würde. Ihr müßt also noch einmal an den Anfang des Abschnittes und ^B eingeben.

Wie rücke ich Teile des Textes mit ^O^G (Zeitweiliges Einrücken = temporary indent) ein, und nutze dies auch beim ^B??

Nun das geht folgendermaßen: In die Spalte, in die eingerückt werden soll muß ein Tabulator: Hingehen und mit ^O^I ESC Setzen. In der Zeile mit L-----R erscheint in der Spalte ein !. Jetzt so oft ^O^G eingeben, bis der auserwählte Tabulator statt mit ! mit V angezeigt wird. Dann wird bis zur Eingabe eines RET (d.h. der ganze Abschnitt) eingerückt. Aber dabei gibt's keine Trennhilfe. Also nachher ggf. zurück zum Anfang des Abschnittes, wieder in die Spalte gehen, so oft ^O^G bis das V da ist wo es hin soll und dann ^B abschicken. Auch einen Text, der jetzt in Spalte 1 beginnt kann ich so beim ^B einrücken. Achtung: gebt den Text nicht ein, indem Ihr mit TAB oder Leerstellen einrückt. NewWord merkt sich diese eingegebenen Leerzeichen, und die werden beim Randausgleich nicht gelöscht: Es tauchen dann einige dumme Leerzeichen auf!

Bei der Eingabe der Laufwerke wird erwähnt, daß das zuerst eingegebene Laufwerk als Defalut-Drive verwendet wird. Das heißt, das auf dem die Dateien NW.OVR, u.s.w. gesucht werden!!! Wer zwei Laufwerke hat möchte sicherlich in B: NewWord, und auf C: die Texte haben. Dann bei der Installation als erstes Laufwerk B angeben, und dann unter CP/M erst C: (Umschalten auf C) und NewWord mit B: NW aufrufen.

Das Problem der 12-Zoll-Seitenlänge ist keins. Am Anfang eines Dokuments muß einmal das 'Dot Command' '.PL72' für 72 Zeilen pro Seite stehen. Zum anderen muß beim Aufruf des P-Kommandos aus dem Hauptmenü bei der Nachfrage ob Formfeeds benutzt werden sollen mit Nein geantwortet werden. Bei 1/8 Zoll Zeilenvorschub muß '.PL96' benutzt werden, da  $8 \times 12 = 96$ .

Use Form feeds?: N. Das N veranlasst Neword keine Seitenvorschübe an den Drucker zu senden, sondern den Rest der Seite mit Leerzeilen vollzuschreiben so, daß die Einstellung der Seitenlänge am Drucker nicht zum Tragen kommt. (Michael Möwe)

'No space' ist immer ein Problem! Zur Not könnt Ihr mit ^K^J die Datei NWPRINT.OVR löschen und später wieder vom Original holen!

New Word

Ersetzen von mehrfachen Leerzeichen durch einfache im ganzen Text: Wer dazu an den Anfang des Textes geht und ^Q^A eingibt, dann als Originalstring zwei Leerzeichen, als neuen eines und die Optionen GN (G = global = alles, N = no ask = ersetzen ohne zu fragen), der wird enttäuscht sein, da dreifache Leerzeichen doppelte werden. Wer das aber von hinten macht (Optionen GNB B = backwards = von hinten), der hat was er will. Damit werden alle Ketten von Leerzeichen durch einfache ersetzt! Das geht übrigens auch mit anderen Zeichen!

Als Help-Level finde ich den Level 2 den besten. Das sogenannte EDIT-MENU kennt man im Laufe der Zeit auswendig, und hätte den Platz gerne für Text. Und folgendes bietet Level 2, was Level 1 nicht hat: 1. Frage, ob ein nicht vorhandenes Dokument überhaupt angelegt werden soll. 2. Beim Löschen von Blocks mit ^K^Y kommt eine Warnung, wenn der Block zu groß ist, um in den UNERASE-BUFFER zu passen, d.h. daß mit ^U nicht der ganze Block zurückkommt.

Viele der obigen Einstellungen (Page length, Hyphen-Help an, Help-Level, TAB-Stops können leicht mit NWINSTAL, Menü's D und E voreingestellt werden.

Wie mische ich zwei Schriftbreiten in einer Zeile? Da gibt's einen Trick: NewWord arbeitet ja mit STANDARD-PITCH und ALTERNATE-PITCH, was mit ^P^N und ^P^A umgeschaltet werden kann. Schalte auf A-PITCH mit ^P^A und gebe das Kommando .CW 25 in Spalte 1 ein (Achtung, das ist Punkt-CW). Damit stelle ich die Zeichenbreite auf 25, also breit. Drucke ich nun die Zeile ^Ntest ^Atest ^Ntest so habe ich folgendes:

test test test.

Mit .CW (Character-Width = Zeichenbreite) und .LH (Line-Height = Zeilenhöhe) kann ich bei einigen Druckern (allerdings nicht DRAFT) den Papierverbrauch steuern. Ob das geht müßt Ihr einfach mal mit Eurem Drucker probieren. Hier ist eine Tabelle, die die Zusammenhänge zwischen den Parametern der o.g. Kommandos und dem Ergebnis:

Zeichen/Zoll	30	24	20	15	12	10	8	6	5	
Kommando	.CW4	.CW5	.CW6	.CW8	.CW10	.CW12	.CW15	.CW20	.CW24	
Zeilen/Zoll	9.6	8.0	6.8	6.0	5.3	4.8	4.0	3.0	2.0	
Kommando	.LH5	.LH6	.LH7	.LH8	.LH9	.LH10	.LH12	.LH16	.LH24	

**Patch - Tabelle** Patch = Flicker! Motto: **Besseres NewWord!**

Ziel der Aktion	Wie mache ich's
AÖÜ als Buchstaben erklären, damit bei ^T das gesamte Wort verschwindet:	In die Speicherstellen 06E0, 06F0: E0 -> FC 06E4, 06F4: E0 -> FE
Trennhilfe (^D^H) soll nicht erst bei 5 Buchstaben nachfragen, sondern früher:	In die Speicherstelle 0742: 05 -> 03
Bei jeder Eingabe im Text soll die Zeile/Spalte Angabe in der Statuszeile sofort erneuert werden:	In die Speicherstelle 0424: ? -> 00 00
Leerzeichen mit unterstreichen:	In die Speicherstelle 070B: 00 -> FF

New Word

**Wie wird's gemacht:**

NWINSTAL aufrufen. Dann mutig das Menü **H** Special Patches aufrufen. Ja, jetzt wird es spannend. Ich habe oben immer die zu ändernde Speicherstelle angegeben (als vierstellige Zahl vor dem Doppelpunkt). Da müssen wir jetzt hinkommen. Dazu müßt Ihr in einem der 'Patch Menus' (Nummern 1-3) mal irgendeine Option aufrufen, und dabei versuchen eine zu finden, bei der unten diese Zeile mit Hex-Zahlen kommt. In den ersten Spalten vor dieser Zeile steht eine vierstellige Zahl: die Adresse. Mit RET kann man dann nach und nach weiterblättern. Irgendwann kommt auch die nächste Zeile u.s.w.. Das halt so lange, bis man an der richtigen Adresse angekommen ist. Nun ist eine Zeile sechzehn Hex-Päärchen (also Bytes) lang. D.h. steht bei der 1. Zeile als Adresse eine 065E, so steht in der nächsten 066E, also 16 mehr! Die Ziffern sind ja 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F. (Keine Angst, unten ist ein Beispiel und das Protokoll der Änderungssitzung von NWINSTAL!). In der Zeile müßt Ihr selbst mitzählen! Deshalb habe ich auch den alten Inhalt mit angegeben! Dabei steht in der Tabelle oben alt -> neu.

Nun das Beispiel: Ich such z.B. die Adresse 06E0, und finde:

```
06D4 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
      D4 D5 D6 D7 D8 D9 DA DB DC DD DE DF E0 E1 E2 E3
```

In dunkel die Ausgabe von NINSTAL. In der 2. Zeile habe ich die beiden letzten Ziffern der Adresse für die 1. Zeile druntergeschrieben und die Speicherstelle 06E0 unterstrichen.

**NWINSTAL-Sitzung**

Eure Eingaben in fett und unterstrichen, der Zweck der Änderung ist *karsiv* (der darf weder eingegeben werden noch auf dem Bildschirm erscheinen!).

Bei den verschiedenen Menü's habe ich nur die Überschrift angegeben, da das zur Identifizierung sicherlich ausreicht. Wenn einige Speicherstellen bei Euch andere Werte aufweisen, so liegt das daran, daß ich mit dem nach meinen Vorstellungen installierten NewWord arbeite. Ich habe als Original NW.COM angegeben, und als Datei, in die das geänderte NewWord hinein soll NWTEST.COM genannt. Das hat den Sinn, daß ich dann erst mal testen kann, ob alles geklappt hat, und ggf. einfach NWTEST.COM lösche und von vorne beginne. Nach dem NWINSTAL solltet Ihr alle geänderten Dinge einmal Ausprobieren! Hat alles geklappt, kann mit ERA NW.COM und dann REN NW.COM=NWTEST.COM das verbesserte NewWord in betrieb genommen werden. (NWTEST.COM verwendet die selben .OVR-Dateien wie NW.COM!)

A>nwi

NWINSTAL 1.00 Release of June 11, 1984  
Copyright (C) 1983 Newstar Software Incorporated.

Memotech version

Name of file to install? nw

File to hold installed Newword? nwtest

New World

MAIN MENU

What is your choice? H

PATCH MENU #1

What is your choice? 2

PATCH MENU #2

What is your choice? A

*Statuszeilen - Update schneller*

041A 00 00 D0 07 01 00 64 00 01 00 64 00 F4 01 C3 32 .....d...d....2  
00 00 D0 07 01 00 64 00 01 00 00 00 X

PATCH MENU #2

What is your choice? R

*Leerzeichen unterstreichen*

070A FF 00 0C 0B 0C 0A 03 0B 01 46 02 04 37 00 00 00 .....F..7...  
FF FF X

PATCH MENU #2

What is your choice? 3

PATCH MENU #3

What is your choice? M

*Umlaute als Buchstaben (^T, ^A und ^F)*

06AB 50 41 53 01 FF 50 4C 49 03 FF 43 20 20 03 FF 00 PAS..PLI..C ...  
50 41 53 01 FF 50 4C 49 03 FF 43 20 20 03 FF 00 PAS..PLI..C ...

06BB 00 00 07 00 00 00 00 00 00 00 00 00 00 00 00 .....  
00 00 07 00 00 00 00 00 00 00 00 00 00 00 00 .....  
06CB 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....  
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....  
06DB 00 00 00 FF C0 7F FF FF E0 7F FF FF E0 00 00 00 .....  
00 00 00 FF C0 7F FF FF FC 7F FF FF FE 00 00 00 .....  
06EB 00 00 00 FF C0 7F FF FF E0 7F FF FF E0 00 00 00 .....  
00 00 00 FF C0 7F FF FF FC 7F FF FF FE X

PATCH MENU #3

What is your choice? L

*Trennhilfe ab 3 Buchstaben*

0735 10 06 0A 00 00 20 20 20 20 20 20 FF FF 05 00 FF .....  
10 06 0A 00 00 20 20 20 20 20 20 FF FF 03 X

PATCH MENU #3

What is your choice? X

MAIN MENU

What is your choice? X

The changes you requested have been made.

Do you want to change this? (Y/N) N

A>

NewWord / Hashing**NewWord Version 2.16**

Ich habe sie, die neue NewWord Version (wird nicht von Memotech mitgeliefert!). Kosten waren ca. DM 120.- ... und hat sich finde ich gelohnt! Die Neuerungen sind u.a.:

1. Spaltenblöcke
2. Mehr ^P-Kommandos (zusätzlich ^P^A und ^P^Ü) frei definierbar
3. DMX80 und FX80 -Druckertreiber
4. Besseres Unterstreichen
5. Ich kann das Document-Laufwerk voreinstellen
6. Im Opening-Menu werde ich als Besitzer gemeldet!

Wer dieses NewWord haben möchte, kann es über mich bestellen. Es kostet (ab 5 Bestellungen) je 40 US\$. Siehe Bestellzettel! Ich habe an Newstar geschrieben und versuche den Preis (insbesondere die Versandkosten) zu drücken!

**Hashing Verfahren** (Michael Möwe)

Das Hash-Tabellenzugriffsverfahren dient dazu, aus einem gegebenen Schlüssel den dazugehörigen Tabellenplatz zu errechnen. Im folgenden werde ich mich auf diese Arrays beziehen:

DIM KEY\$(13),PTR(13),TXT\$(13).

Der Sinn der Sache ist, man will mit dem Schlüssel KEY\$ auf den Inhalt des Arrays TXT\$ zugreifen, z.B. sei der Schlüssel ein Name und das Array TXT\$ enthalte Adresse und andere zum Namen gehörende Daten.

Das Verfahren: Eingegeben wird zum Beispiel der Schlüssel 'ABC' der in eine Tabellenadresse in KEY\$ umgerechnet werden soll. Die Umrechnung geschieht auf folgende Art:

$ZEIGER = (ASC('A') - 32) * 64^2 + (ASC('B') - 32) * 64 + ASC('C') - 32 = 137379$

ZEIGER enthaelt jetzt einen eindeutigen Zeiger auf einen Tabellenplatz. Wie jedoch zu Erkennen ist, ist dieser Zeiger zu gross fuer die Tabelle, die uns zur Verfuegung steht; d.h.: der Zeiger muss noch weiter bearbeitet werden:  $ZEIGER = INT(MOD(ZEIGER, 13)) + 1$

$137379 \text{ mod } 13 = 10567 \text{ Rest } 8$  so, dass  $ZEIGER = 8 + 1 = 9$  ist. Der eindeutige Zeiger 137379 wurde also in den nicht eindeutigen Zeiger 9 umgerechnet. Wenn der Platz  $KEY$(ZEIGER)$  noch frei ist, dann koennen hier der Schlüssel 'ABC' und in  $TXT$(ZEIGER)$  die dazugehoerigen Daten eingetragen werden. Ist  $KEY$(ZEIGER)$  jedoch schon belegt, stehen wir ganz schoen belaeummert da. Aber auch dieses Problem laesst sich loesen, mit Hilfe des Arrays PTR. Ein Hilfszeiger HZGR wird gleich ZEIGER gesetzt ( $HZGR = ZEIGER$ ) und solange um 1 erhoehrt, bis ein freier Tabellenplatz gefunden wird. Dann erfolgen diese Aktionen:

$PTR(ZEIGER) = HZGR$ ; PTR(ZEIGER) enthaelt einen Verweis auf den gefundenen freien Platz, der mit dem Schlüssel 'ABC' belegt werden wird.

$KEY$(HZGR) = 'ABC'$ ; Belegung des freien Platzes mit 'ABC'

$TXT$(HZGR) = \text{Text}$ ; den mit dem Schlüssel korrespondierenden Text eintragen.

Bei der Suche in der Tabelle wird aehnlich vorgegangen. Wenn der Schlüssel  $KEY$(ZEIGER) <> 'ABC'$  ist, dann wird  $ZEIGER = PTR(ZEIGER)$  gesetzt; und das solange bis  $KEY$(ZEIGER) = 'ABC'$  oder  $PTR(ZEIGER) = ZEIGER$  ist, d.h.: der Pointer PTR(ZEIGER) auf sich selber verweist, was bedeutet, daß der gesuchte Schlüssel nicht in der Tabelle eingetragen ist. Voraussetzung dafür ist, daß vorher alle Pointer so initialisiert wurden, daß sie auf sich selber verweisen. Das zu dem. Anbei ein kleines Programm, das die angesprochenen Aufgaben erfuehlt.

Drucker (speziell DMX80)**Druckeransteuerung** (Michael Möwe)

Zur Nachfrage von Peter Würfel bezüglich einiger Druckerbefehle.

## 1. Standard bit image designation

Die Anwahl der S.b.i.d bewirkt, dass eine vorher festgelegte Anzahl von empfangenen bytes als Bitmuster fuer den Druckkopf interpretiert wird d.h.: das Bit 0 eines bytes spricht die oberste Nadel des Druckkopfes an, Bit 7 spricht die unterste an so, daß auf diesem Wege die Erstellung einer Graphik ermöglicht wird. Maximal 576 Punkte können auf diese Art in einer Zeile untergebracht werden.

## 2. Double density b.i.d.

wie oben, nur das 1152 Punkte in einer Zeile erzeugt werden können.

## 3. MSB operation command

Wenn ich das richtig verstanden habe, dann ist dieser Modus fuer Apple-Benutzer gedacht, weil diese Scheisskiste wohl nur 7 bit an den Drucker schickt d.h.: der Drucker weiß nicht so recht, welche Hälfte seines Zeichensatzes er benutzen soll. Um dieses dem Drucker zu erzählen kann das MSB per Befehl auf 1 oder 0 gesetzt werden. (sic est)

## 4. Font registration command

Datt is een fien tüch. Das ist eine feine Sache. Dieser Befehl ermöglicht es einen eigenen Zeichensatz im Drucker zu installieren. D.h.: es wird also möglich auch im Italic-font die Umlaute zu benutzen, die allerdings vorher installiert werden müssen. (z.B. mit PRT.COM).

**PRT.COM** (Michael Möwe, siehe Programmliste weiter vorne)

Ich habe mir die Mühe gemacht ein Programm zu schreiben, das die lästige Druckereinstellung (z.B.: 12 Zoll Papier) wesentlich vereinfacht. Zunächst eine Beschreibung dessen, was dieses Programm kann. PRT.COM kann in 2 Modi benutzt werden, es kann mit dem Aufruf PRT.COM<RET> im interaktiven Modus benutzt werden, d.h.: der Benutzer wird per Menue nach den Druckerparametern gefragt (z.B.: Seitenlänge in Zoll ?). Man kann es aber auch wie folgt aufrufen:

```
PRT.COM plen 12 lines 72 skip 6 use xyz
```

Was passiert???

plen steht für Page LENght (Seitenlänge in Zoll); hier 12

lines steht für die Anzahl der Zeilen pro Seite; hier 72

skip beeinflusst die skip perforation; hier werden nach 66 Zeilen 6 übersprungen.

use ruft eine-vorher erstellte-Datei xyz auf, in der ein selbsterstellter Zeichensatz s\_ehn uMd stellt den Drucker auf diesen Zeichensatz ein. Ein Editor für diesen Zeichensatz steht in PRT.COM zur Verfügung.

P.S. wenn PRT.COM so aufgerufen wird PRT <RET>, d.h.: ein Blank nach PRT, dann wird der Drucker auf einige sinnvolle Default-Werte eingestellt.

Die Reihenfolge, in der die Einstellungen angegeben werden ist egal.

Das wär's zunächst zu PRT.COM, was aber nicht bedeutet, daß das schon alles ist, PRT.COM enthält noch einige Möglichkeiten mehr!!!

Turbo-Pascal**Wer hat sich noch nicht über TLIST.COM geärgert?**

Zumindest scheint Olaf Krumnow davon nicht begeistert zu sein! Ich übrigens auch nicht. Olaf hat einen Turbo-LISTER geschrieben, der keine Wünsche offen läßt. Man kann damit:

1. Reservierte Worte oder nur PROGRAM, PROCEDURE und FUNCTION markieren und/oder großschreiben, wobei auf Wunsch Kommentare berücksichtigt werden.
2. Überschrift und Seitennummerierung, Überspringen der Perforation.
3. Ausgabe eines Teillistings.
4. INCLUDE-Dateien mitlisten oder nicht, und markieren.
5. Ausgabe auf den Drucker oder in eine Datei.
6. Seitenlänge einstellen und US-Zeichensatz für Klammern anwählen und linken Rand einstellen.

Ich habe das Programm angeschaut (und einige Vorschläge eingebracht) und kann es nur wärmstens empfehlen! Und es ist billig (DM 5.-)! Siehe Programmliste weiter oben. Natürlich mit ausführlicher Dokumentation und Installationsprogramm!

**Turbo-Pascal 2.0 hat einige Macken:**

1. Der MARK/RELEASE-Mechanismus funktioniert nicht
2. Der Editor arbeitet nicht mit Funktionstasten (die nicht mittels F.COM oder der RAM-Disc geändert wurden) zusammen, da das 7. Bit von Turbo gelöscht wird.
3. Ich kann ja bekanntlich mit Turbo .COM-Dateien erzeugen (beim Compilieren), und beim Aufruf unter CP/M Parameter angeben. Turbo löscht alle Zeichen hinter dem 32-ten. Also kann ich so nur 31 Zeichen erhalten.
4. Beim Aufruf werde ich immer gefragt, ob ich die Fehlermeldungen (Message-Overlay) haben will (d.h. volle Fehlermeldungen), und ich sage immer Ja.
5. Der Divisionsrest (Befehl MOD) liefert bei einigen Zahlen falsche Ergebnisse:  $-11 \text{ MOD } 10 = -1$  aber  $-5 \text{ MOD } 10 = 5$ , und das sollte eigentlich eine minus-fünf liefern.

**Reparatur**

3. Haben Michael Möwe und ich dadurch gelöst, daß wir den Beginn des Programms (Start Adress) verschoben haben, und das .COM-Programm nach jedem Kompilieren so geändert, daß die Kommandozeile in den so erhaltenen Freiraum geschoben wird. Das ist unbequem, aber wir wußten, das es klappt, und hatten es schnell fertig. Olaf Krumnow hat sich da mehr Gedanken gemacht, und eine Möglichkeit gefunden TURBO.COM so zu verändern, daß alle von Turbo erzeugten .COM-Dateien gleich entsprechend geändert sind ohne, daß man deshalb jedes mal darüber nachdenken muß. Seht Euch dazu mal weiter hinten Olaf's drei Seiten an.

- 1., 2. 4. und 5. geht leicht mit dem DDT:

Damit es einfacher für Euch ist, habe ich die DDT-Sitzung hier protokolliert. Wie bei der NWINSTAL-Sitzung oben sind Eure Eingaben **fett und unterstrichen**, und der Zweck der jeweiligen Änderung (der nicht eingegeben wird) *kursiv*:

Das Ergebnis solltet Ihr als TURBON.COM abspeichern und erst mal testen, ehe Ihr TURBO.COM ersetzt!

Turbo-Pascal / Hardware

A>ddt turbo.com

NEXT PC

7900 0100

- S1E65 MARK / RELEASE

1E65 CD ED

1E66 00 .

- S1E70

1E70 CD ED

1E71 00 .

- 2E5B Bit 7 beim Editor

2E5B 28 30

2E5C 03 .

- S2129 Fehlermeldungen automatisch laden

2129 CD 3E

212A 42 0B

212B 2C B7

212C 32 .

- S0713 MOD reparieren

0713 CB F0 <Ist in Turbo 3.0 auch falsch!>

0714 7A 1B

0715 C8 37

0716 18 C7

0717 35 .

- ^C Fertig

A>SAVE 119 TURBON.COM Abspeichern

Das war alles, es sei denn Ihr wollt gleich noch Olaf's Patches mit erledigen.

**Lautsprecher an die 80-Zeichenkarte**

Wenn ich hinter der FDX (diese möglichst offen) sitze, so ist auf der 80-Zeichenkarte links ein Stecker mit Kabel zur Rückwand. Dieser Stecker ist auf nebenstehendem Bild abgebildet. Dahinter (also zur Vorderseite der FDX hin) ist parallel zu dieser Stiftleiste eine Diode, deren Anschlüsse unter der Karte mit den Pins 13 und 14 (der Pin 15 ist der ganz linke) verbunden sind. An diese Pins kann man einen kleinen Lautsprecher direkt anlöten und im Gehäuse z.B. an die Rückwand kleben (Doppelseitiges Klebeband). An den Stecker könnt Ihr den Lautsprecher natürlich auch anschließen, aber der eine der Pins ist schon an die neunpolige Buchse in der Rückwand angeschlossen, aber der andere nicht. Das ist also etwas friemelig.



Hardware / Allgemeines**Umtausch des Speichererweiterungs-PAL (Andreas Viebke)**

Was mein FLA OLL betrifft - da bist Du jetzt bestimmt baff: Zwei Tage hat's gedauert, Montag abgeschickt, Mittwoch zurück und das ohne Eilzustellung. Jetzt hab' ich 32K mehr, die FDXB aber nicht sieht, wohl aber der Rechner in der Betriebsart MTX.

**Speichererweiterung**

Wenn ich dort 64k-IC's in die freien Sockel (Typ 4164, 64k mal 1 Bit DRAM) einsetze, so habe ich nur zusätzliche 32k Byte!

**Speicher für die RAM-Disc von Bernd Preusing**

Ich werde versuchen die 32k-Karte mit größeren Speichermengen ausrüstbar zu machen. Im Prinzip sind schon 16 kByte verwendbar, zumindest als Druckerpuffer.

**Funktstörung**

Bei wem ist der MTX ein Feind von Rundfunk und Fernsehen? Das Problem kann ganz leicht beseitigt werden. Man nehme einen Innensechskant für die Tastatur, einen Schraubenzieher, eine Spitzzange, einen LötKolben und etwa 5-10 cm nicht allzu dünnen isolierten Draht. Dann schraubt die Tastatur auf und zieht den Stecker von der Tastatur von der Grundplatine links ab (evtl. mit dem Schraubenzieher etwas hebeln) und schiebt das Oberteil zur Seite runter. Jetzt müssen die Masseanschlüsse, d.h. die äußeren Kontakte, der Monitor- und der HIFI-Buchse miteinander verbunden werden (natürlich mit dem LötKolben ...), und das eine Ende des Drahtes dort angeschlossen werden. Dann muß das andere Ende des Drahtes in einen Ring geformt werden, durch den die mittlere Schraube vom Kühlblech (d.h. der MTX-Heizung) paßt. Diese Schraube ist die, die unten am MTX herausschaut. Diese Öse muß verzinnt werden. Dann löst Ihr die Mutter der Schraube (mit der Zange festhalten, und die Schraube von unten drehen) und die gelötete Öse unter die Mutter auf die Schraube schieben, und alles fest zusammenschrauben. Also werden die Massekontakte der Monitor- und der HIFI-Buchse mit dem Kühlkörper verbunden. Nun muß der MTX nur noch zusammengebaut werden (Tastaturstecker nicht vergessen).

**SUPERTAPE für MTX???**

Wer hat diese Softwarequelle schon erschlossen?

**CP/M Z80-Assembler**

Wer weiß, ob es eine Z80-Assembler in der sogenannten Public Domain gibt, d.h. einen der weitergegeben werden darf. Der Makro-Assembler MACRO 80 ist ein sehr gutes Programm, kostet aber auch ca. DM 500.-

**FDXB und DISC QUIT**

Ich bin maßlos faul, wenn's gut ist. FDXB verlasse ich grundsätzlich durch Drücken der Reset-Tasten, das geht schneller und besser als das eintippen von DISC QUIT. Außerdem ist das FDXB so merkwürdig, daß nach einem DISC QUIT nach Aussage von Fa. Memotech NewWord nur dann läuft, wenn die Datei NCFM.COM auf der FDXB-Diskette ist. Warum wissen nur die Götter und Programmierer von Memotech.

Dies und Das / Wer tut Was**BASIC - Probleme**

Wer den Inhalt einer Speicherstelle auf dem Bildschirm als HEX-Code oder dezimal haben möchte, der wird weder mit PRINT PEEK(...) oder entsprechenden Assembler-Programmen wenig Erfolg haben. Um den HEX-Code von z.B. PEEK(60000) zu erhalten muß folgendes in BASIC abgeschickt werden:

```
LET I=INT(PEEK(60000)/16): LET J=PEEK(60000)-16*I
IF I<10 THEN LET I=I+48 ELSE LET I=I+55
IF J<10 THEN LET J=J+48 ELSE LET J=J+55
PRINT CHR$(I);CHR$(J)
```

**Mailbox**

Die erste MTX-Mailbox steht jetzt in Frankfurt:

COMBO (Commodore Mb), Tel. 069 - 6638191

Hauptmenü - Infoecke - Computerecke - MTX 512

Hinterlaßt Eure Meldung in Mb111 oder HCUF Mb1600

Wie ? .... :

Akustikkoppler anschließen und CONTACT.COM starten.

Anrufen, wenn frei, auf den Piepton warten, und den Hörer auf den Akustikkoppler legen und RET drücken. Daraufhin meldet sich COMBO.

Für Info 8 bereitet Frank Dersewski eine Beschreibung vor, wie man mit dem Modem-Programm MDM 712, das es bei ihm zusammen mit u.a. einer Disk-Utility, FORTH und einem Teletype-Programm für DM 25.- gibt und dem Dataphon s 21 d Akustikkoppler arbeitet. Dieses MDM 712 ist übrigens auch hervorragend geeignet, um Daten zwischen verschiedenen Rechnern zu übertragen (auch Apple!).

**Wer tut Was**

Hier die neue Rubrik, die ich im Info 6 mal versuchshalber gestartet habe, und wo ich immer noch auf Hinweise hoffe:

Allround:	H. Herberg, C. Löhrmann, M. Schlüter
FDX-BASIC	A. Viebke
CAD, Ditracer	D. Path
CP/M System incl. CP/M 3.0	B. Preusing
CP/M	K. Rohloff
Modem, Funk	F. Dersewski
Steuerungsprobleme	H. Riebesehl
Turbo-Pascal	O. Krumnow
Grafik	M. Köster, A. Schmitt
Einkauf und Industriekontakte	F. Bueschler

Wer sich auf dieser Liste fehl am Platz fühlt oder vermißt ... schreibe mir. (Bitte nur ernstgemeinte Zuschriften, d.h. Ihr solltet im genannten Bereich firm sein, und es soller Euer Hauptinteresse bzgl. MTX sein).

**Hier noch eine Anmerkung für MTX500-User mit FDX.** (Frank Bueschler)

Mit der FDX habt ihr eine 32kB-Karte für das Grundgerät bekommen. Auf dieser Karte sind acht Steckplätze nicht belegt. Einfach acht 64 kB-IC's einstecken und schon habt ihr eine 96kB-Version und könnt mit der RAM-DISC von Bernd Preusing eine 47kB-Ram-Disc oder Druckerpuffer installieren. (IC-Typ: 4164. Anm. d. Red.)

Was sonst noch zusammen kam**Schacheditor**

Andreas Viebke hat zu viel Freizeit und nutzt sie indem er diverse gute Programme mit FDX-BASIC schreibt. So auch einen guten Schacheditor, d.h. ein Programm mit dem man Schachpartien anschauen, eingeben und editieren, kommentieren, ausdrucken, sich vorführen lassen, .... kann. Die Grafik ist hervorragend und das Programm recht bedienerfreundlich - vorausgesetzt man macht sich die Mühe die Beschreibung zu lesen statt wie ich einfach zu sehen, daß das Programm menügesteuert ist und herumprobiert. Da blieb ich doch glatt im Korrekturmenü hängen und wußte nicht, daß man da nur durch Eingabe eines korrekten Zuges rauskam.

**80-Zeichen-Interrupt-Uhr**

ACHTUNG, das Programm hat seine Tücken. Bernd Preusing hatte auch schon soetwas überlegt, mußte aber feststellen, daß die an die 80-Z-K übergebenen Werte in Zwischenspeicher wandern, und wenn diese Uhr eine Ausgabe auf die 80-Z-K unterbricht, so kann es passieren, das eine ganze Menge Mist zusammenkommt. Die Uhr landet wo sie soll, aber die unterbrochene Ausgabe weiß von nichts, glaubt an die Richtigkeit der bis dahin übergebenen Daten und macht einfach weiter. Und das kann doch etwas merkwürdig aussehen! Bei der 40-Z-K geht diese Ausgabe ja stets durch POKen ins VRAM, dort gibt es keine Zwischenspeicher!

**RAM-Disc mit Druckerpuffer für CP/M von Bernd Preusing**

Nun hier eine Beschreibung, was RAM alles kann:

1. Verwaltung von überschüssigem Speicher (d.h. Speicher über 64k und ggf. die 16k VRAM aus der Tastatur) als RAM-Floppy, d.h. funktioniert wie eine Diskette, ist nur schneller und nach dem Abschalten futsch.

2. Ein Teil des überschüssigen Speicher als Druckerpuffer.

3. Ein Funktionstastenmodul! D.h. Belegen der F-Tasten aus Datei,

Und wie wird's gemacht:

1. Initialisierung von nur RAM-Disc:

RAM a 0, wobei a die Anzahl der freien 16k-Seiten ist. Wer einen MTX 500 mit 32k-Karte hat: a=1. Ist die 32k-Karte mit 4164-er IC's aufgerüstet, dann ist a=3. Dann CONFIG F:50 oder CONFIG F:60 je nachdem ob die RAM-Disc mit oder ohne Systemspuren sein soll. Bei 50 muß ein SYSCOPY F: folgen. Dann kann mit PIF F:=B:... alles mögliche auf die RAM-Disc gepackt werden.

2. Initialisierung von nur Spooler (= Drucker-Puffer)

RAM 0 b, wobei hier b die Zahl der 16k-Seiten ist. Fertig!!!

Die Kommandos RAM S S (Puffer löschen und neu Initialisieren) sowie RAM S P (Pause, d.h. Puffer bleibt erhalten mit Inhalt, aber wird nicht beachtet) und RAM S R (Weiterdrucken, d.h. jetzt gehen Ausgaben wieder über den Puffer und das was im Puffer noch ist geht auf den Drucker) braucht man i.a. garnicht!

3. Initialisierung für beides:

RAM a b, wobei a+b die Zahl der 16k-Seiten ist. Alle weiteren Kommandos gehen genauso wie oben! Die RAM-Disc hat mit dem Puffer nichts gemeinsam außer, daß sie sich Speicher teilen und beide mit dem Programm RAM.COM bearbeitet werden!!

Wer das VRAM aus dem Grundgerät für Grafik braucht (z.B. unter Turbo-Pascal) darf für a,b bzw. a+b nur eins weniger angeben, da dann die 16k VRAM nicht mitzählen. Das Programm weiß dann schon, das das VRAM freizulassen ist. Die RAM-Disc arbeitet nicht mit FDXB.COM!

H a r w a r e**Die Mühe hat sich gelohnt** (Frank Bueschler)

Ich glaube, ich bin nicht der einzige, dem einige Zubehörteile für den MTX zu teuer sind. Also habe ich mich zusammen mit Herbert auf die Suche nach neuen Möglichkeiten gemacht. Neben der von Herbert geplanten bis zu 8MB ausbaubaren Speichererweiterung ist es mir gelungen eine Firma ausfindig zu machen, die sämtliche Schalteile direkt aus Japan bezieht und diese dann sehr günstig an uns weiterverkaufen kann. Die Preise für Speicher-IC's sind fantastisch und mit der weiteren Möglichkeit, die PAL's zu kopieren können wir uns ohne große Kosten die RS232 Aufrüstung, als auch eine Speichererweiterung auf 128kB leisten. Weiterhin wäre es möglich, aus dem MTX500 einen MTX512 mit 64kB auf auf der Grundplatine zu machen, denn dann könnte die gesamte Speichererweiterung genutzt werden.

Gibt es Interessenten für einen Farbmonitor mit guter Auflösung für DM 650,-, ( baugleich zu den Monitoren in Spielhallenvideogeräten ), die von Philips, bzw. Tochterfirma Valvo hergestellt werden.

**Zum Thema : Laufwerke** (Frank Bueschler)

Das auf dem Club-Treffen angesprochene Radial-Laufwerk mit 12MB Speicherkapazität war angeblich ein Aprilscherz der Zeitschrift c't, auf den viele Leser reingefallen sind ( der Bericht war viel zu ernst ). Ansonsten sind wir am Überlegen, ob wir statt der 77 Spur Laufwerke nicht lieber umschaltbare von 77 auf 80 (1,6 / 1 MB ) nehmen sollten, da man dann eben zwei Fliegen mit einer Klappe schlägt. Da Epson diese zur Zeit nur in begrenztem Umfang liefern kann und somit die Preise recht hoch sind, sind wir dabei, uns für TEAC zu entscheiden, da diese auch um einiges leiser sind. Leider müssen wir nun wieder in neue Preisverhandlungen treten, also bitte etwas Geduld, bis wir näheres wissen.

Anmerkung zum reinen 77 Spur Laufwerk: Das Laufwerk SD560 läßt sich mit unserem Formatprogramm nicht formatieren. Auf unsere Bitte hat sich Memotech auch ein Epson SD560 besorgt und hatte die gleichen Probleme. Memotech verspricht aber hier in näherer Zukunft Abhilfe zu schaffen.

**Frage :** Hat schon jemand versucht 3,5 Zoll Laufwerke anzuschließen, da immer mehr Computerhersteller auf dieses Format gehen ?

**Dazu etwas neueres** (Herbert Herberg)

Auf dem Bestellzettel findet Ihr eine recht große Auswahl an Laufwerken, in der Ihr hoffentlich das richtige findet. Es gibt u.a. auch ein Laufwerk der Firma Philips, Zweiseitig, 80 Spuren, Slim-Line (d.h. passt in die FDX) für ca. DM 400,-. Dieses Laufwerk haben wir nicht auf der Liste, da es einen immensen Stromverbrauch an den Tag legt: 12.9 Watt. Zum Vergleich: EPSON 8.7 und TEAC 4.9 Watt, und wir wissen nicht, was das Netzteil abkann! Es müssen ja auch mal beide gleichzeitig arbeiten können (PIP, NewWord, ...). Wie Frank erwähnte wissen wir auch noch nicht, wie wir die 77-Spur fähigen Laufwerke korrekt ansprechen (insbesondere damit auch die umschaltbaren), aber das werden wir. Ich habe mittlerweile ein TEAC FD 55-GF. Wichtig ist, daß die 77 Spuren nur mit High-Density Disketten (HD-Disketten heißen auch Quad-Density (QD), Fujji und Maxell stellen solche her) funktionieren. Die anderen Disketten sind zu schlecht. HD-Disketten kosten i.a. über DM 15,-, aber es paßt mehr drauf, und die 77-er sind schneller! Kommt Zeit (wenig natürlich) - kommt Funktionieren!

Ein Hamburger in München**Ein Hamburger in München**

Frank Bueschler

Wie wird sie noch genannt -- ach ja -- München, die schöne Landeshauptstadt Bayerns. Ja sicherlich, schön ist die Stadt schon, aber Kontakte zu "Einheimischen" zu knüpfen ist hier ganz schön schwer. Die Münchner sind eben ein Volk für sich.

Fünf Wochen München im Rahmen meiner Ausbildung sind aber dennoch etwas sehr erholsames und in dieser Zeit habe ich auch Kontakte zu unseren süddeutschen Mitgliedern geknüpft. Interessante Programme konnte mir besonders Andreas Schmitt zeigen, der sich auf dem Grafiksektor recht stark betätigt. Von 3-D Grafiken, die sich mit von ihm erstellten Programmen in kurzer Zeit abspeichern und wahnsinnig kurzer Zeit wieder aufrufen lassen, bis zu einer komfortablen Dateiverwaltung in FDXB, die er wohl demnächst auch der Allgemeinheit zur Verfügung stellen wird.

Gespräche mit unserem Mitglied Martin Reuther zeigten mal wieder eine Unfähigkeit von VOBIS auf. Martin, hauptsächlich an NewWord interessiert, kam einige Male umsonst in die VOBIS-Niederlassung, um sich NewWord demonstrieren zu lassen. Entweder waren die Versionen nicht installiert, das Gerät nicht mehr komplett, oder aber kein VOBIS-Freak anwesend, der mit dem Gerät umgehen konnte. Das alles hielt Martin bisher davon ab, sich einen MTX zuzulegen, obwohl er von dem Gerät überzeugt ist. Na ja, ich glaube wir hatten oder haben alle unsere kleinen Probleme mit VOBIS.

Ansonsten muß ich sagen, obwohl man sagt München sei eine Computerhochburg, dieses zwar durch die Anzahl der Hard- und Softwarehäuser bestätigt werden könnte, aber der Schein trügt. Beratung und Information lassen meiner Meinung nach zu wünschen übrig und verkaufen kann man hier unten nur mehr, weil der ganze südeuropäische Markt über München beliefert wird, obwohl das nach Aussagen von einigen Händlern auch langsam abflaut, bestätigt dadurch, daß selbst die großen Kaufhäuser ihre Computerabteilungen radikal verkleinern bis auflösen.

Mit den Ohren schlackern kann man hier unten, wenn man die Preise in den Elektronikläden sieht. Drei marktbeherrschende Unternehmen in der Innenstadt, gegen die keine Konkurrenz ankommt.

Jedenfalls habe ich Kontakt zu einem "Computerversandmensch" aufgenommen, und der ist interessiert daran, daß ich ihm elektronische Bauteile und IC's aus Hamburg schicke. Dieser kann uns allerdings auch das langersehnte Modem für etwa 200,00 DM beschaffen.

Na ja, ich freue mich jedenfalls wieder auf Hamburg, wo ich allerdings schon wieder sein werde, wenn dieses Info erscheint und sage nur noch TSCHUSS bis zum nächsten Mal.

**Gefunden**

Die eckige Klammer auf ist auf dem großen Ä und die e.K. zu auf dem großen Ü. Bei den Ermittlungen wurde außerdem der schräge Strich (nein nicht die Reeperbahn) oben-links-nach-unten-rechts auf dem großen ö gesehen!

**Kosten für Info 7: DM 11.96**

Dieses erweiterungsfähige Melodieprogramm ermöglicht das Spielen einer einstimmigen Melodie mit Hilfe des Soundbuffer. Dazu wird die Melodie in einem BASIC-String gespeichert und durch ein Unterprogramm abgespielt.

Das Programmlisting ist zwar etwas kommentiert, dennoch möchte ich einige Hinweise geben, damit Ihr das Programm nach Euren Wünschen ergänzen könnt. (In Klammern gesetzte Zahlen sind Zeilennummern)

Grundlage des Programms sind die Tonwerttabelle (110) und der Melodiestring (510ff). Die Tonwerttabelle enthält die für den Continuous-Sound-Befehl erforderliche Frequenzangabe. Diese Werte werden einmalig aus der mit den notwendigen Vorzeichen versehenen Tonleiter (130) erzeugt (140-180).

Die hier verwendete Tonleiter (130) ist D-Dur; daher sind C und F einen halben Ton (=eine Position) im String nach oben verschoben. Die C-Dur-Tonleiter, die ja keine Vorzeichen enthält, lautet:

"C D E F G A Hc d e f g a h"

Das Array TW enthält also an der Stelle ASC("C"), also TW(67), den Wert für das tiefe C und so weiter.

Es müssen nicht nur Töne abgespielt werden, sondern auch Sonderfunktionen vorgesehen sein. Dafür dienen die Zeichen - ' # und b (180). Beim Minuszeichen "ertönt" eine Pause, das Doppelkreuz und das kleine b entspricht der gewohnten Musiknotation (Halbton rauf bzw runter). Auch das Apostroph findet man in der Musik wieder: C' ist eine Oktave tiefer als C. Dagegen ist c' eine Oktave höher als c. Bei großen Buchstaben ist der Ton eine Oktave tiefer, bei kleinen Buchstaben der Ton eine Oktave höher gemeint. Nun muß noch das Tempo der Melodie bestimmt werden. Dazu dient die Grundzeit Z0 (210), die als Vielfaches von 16 ausgelegt ist, um Rundungsfehler zu vermeiden. Die Ziffern 1,2,4,8,9,0 bestimmen dann Ganze, Halbe, Viertel, Achtel, Sechzehntel und Zweiunddreißigstel (220-255). Bei den beiden letzten mußte ich die Ziffern 9 und 0 verwenden, da 16 und 32 zwei Ziffern sind.

Wird ein laufendes Programm abgebrochen, so läßt es meist einen Ton stehen. Um dies beim folgenden Programmlauf abzu-

brechen, wird vor Dimensionierung des Soundbuffer (320) jeder Soundkanal abgeschaltet (310). In (330) wird der er-

zeugte Klang eingestellt. L ist die Anfangslautstärke eines Tones; DL bestimmt die Lautstärkenabnahme. DL muß negativ sein; der Ton muß ausklingen! In (340) kann zusätzlich eine Tonhöhenänderung vorgenommen werden; die Melodie bekommt dann einen leiernden Klang.

Als letztes muß dem Programm die Melodie vorgegeben werden. Die hier abgedruckte Melodie ist ein recht bekannter Klassiker, der auch in dieses Jahr paßt.

Zur Syntax der Melodie:

Ein Ton wird angegeben als Tonhöhe und Tondauer. Zuerst wird die Tonhöhe angegeben. Sie besteht aus dem Buchstaben des Tones (groß und klein beachten!), gefolgt von Oktavangaben und Halbtonangaben. Beispiele: C D' e' f' g' # hb h#

Als Angabe für die Tondauer dienen die Ziffern. 4 ist eine Viertelnote. 8 ist eine Achtelnote. Man kann auch kombinieren: 48 ist eine Punktierete Viertelnote, nämlich Viertel + Achtel.

Bei Pausen werden statt dem Buchstaben das Minuszeichen angegeben; es folgt die Länge der Pause.

Wenn Ihr Euch den vorgegebenen String M\$ (510-516) ansieht und ein wenig experimentiert, dann habt Ihr es schnell drauf.

Zum Programm, das die Melodie in M\$ dann abspielt (GOSUB 1000), ist noch eine Besonderheit zu erwähnen.

Betrachten wir uns einen Soundbefehl:

SOUND CH, T, L, DT, DL, Z, 1

CH gibt den Tonkanal an. T und L sind Tonhöhe und Lautstärke beim Tonanfang. Z ist die Zeit, während der der Ton in Höhe und Lautstärke gemäß DT und DL verändert wird. Betrachten wir nur die Lautstärkenänderung, da das Melodieprogramm davon lebt. Wenn die Zeit so lang ist, daß die Lautstärke unter Null gerät, so schaltet die Lautstärke von 0 auf 1023 wieder voll hoch. Der Effekt ist aber nicht erwünscht. Es muß darauf geachtet werden, daß DL nicht bleibt. Sonst kann dieser unerwünschte Effekt auftreten.

Das Programm verkürzt den Ton auf die Zeit, bei der dieser Effekt nicht auftritt (1060-1065) und hängt gegebenenfalls einen Korrekturton ohne Lautstärke hinten an (1072), damit der Rhythmus erhalten bleibt. Damit ist es möglich, sehr starke Lautstärkeänderungen vorzugeben (330).

Natürlich ist dies nur ein Ansatz zu einem Musik-Editor, da der Editor fehlt und man hier im Programm editieren muß. Aber ich denke, daß sich die Arbeit des Eintippens lohnt (oder Programmservice!)

Ich wünsche Euch viel Spaß beim Musikmachen.

Als Beilage noch ein bekanntes Synthesizer-Stück aus den 70er Jahren, einer Passage aus den Brandenburgischen Konzerten nachempfunden.

Es mag Euch vielleicht verwundern, daß die beiden hier angegebenen Musikstücke aus dem klassischen Bereich kommen. Aber für Popmusik fehlt mir bisher der Sound für ein Schlagzeug. Wer schafft Abhilfe und ergänzt dies Programm oder nennt seine Schlagzeugtechnik ?!

Vielleicht hat auch jemand eine Idee, einen einfachen Editor zu ergänzen, der das Komponieren ohne Noten vereinfacht.

Reinhard Weigel

Brandenburgisches Konzert:

c8H8c4G8F8G4 c8H8c4E8D8E4c8H8c4C8D8E4F#4G8A8G8F#8G8A8G8f8G8A8G8F#8G8A8G8f8G8A8G8F#8G8A8G8g8e4d8c8d4c8H8c4H8A8G8c8G8c8A4G8F8E8c8E8c8 F4E8D8C4D4E4F#4G4A4H4c4d4d8d8G4H4c4H8A8G8F8E8D8E4D8C8G4G'4C2

```

100 REM *** MUSIC-EDITOR ***
110 DIM TW(127): REM TONWERTTABELLE
120 LET T0=262: REM TIEFSTER TON IN Hz
122 LET T0=1E6/T0
125 LET HALB=.5^(1/12): REM HALBTONSCHRITT
130 LET T$=" CD E FG A H cd e fg a h ": REM TONLEITER MIT VORZEICHEN
140 FOR I=1 TO LEN(T$): REM TONLEITER BILDEN
150 LET TT$=MID$(T$,I,1)
160 IF TT$(">") THEN LET TW(ASC(TT$))=INT(T0+.5)
170 LET T0=T0*HALB
180 NEXT I
185 REM *** SONDERFUNKTIONEN ***
190 LET TW(ASC("-"))=-1: REM PAUSE
191 LET TW(ASC("'"))=-2: REM OKTAVE
192 LET TW(ASC("#"))=-3: REM KREUZ
193 LET TW(ASC("b"))=-4: REM b
200 REM *** ZEITWERTE EINTRAGEN ***
210 LET Z0=7*16: REM GRUNDZEITEINHEIT
220 LET TW(ASC("1"))=Z0
225 LET TW(ASC("2"))=1/2*Z0
235 LET TW(ASC("4"))=1/4*Z0
245 LET TW(ASC("8"))=1/8*Z0
250 LET TW(ASC("9"))=1/16*Z0
255 LET TW(ASC("0"))=1/32*Z0
300 REM *** SOUNDBUFFER AKTIVIEREN ***
310 FOR I=0 TO 3: SOUND I,0,0: NEXT
320 SBUF 100
330 LET L=1023: LET DL=-50: REM LAUTSTARKE UND LAUTSTARKENÄNDERUNG
340 LET DT=0: REM TONHÖHENÄNDERUNG
500 REM *** MELODIESTÜCK M$ ***
505 DIM M$(1000)
510 LET M$="h8d'9h9 f8h9f9d8f9d9 H4F9H9d9H9 c9H9c9H9A#9c9e9c9 d8H8h8d'9h9 f8h9f9d8f9d9 H4d8dB "
511 LET M$=M$+"d8d8h8d8 d8c8f8f8 f8f8d'8f8 f8e88c9f9a9f9 g#9f9g#9f9e#9g#9h9g#9 a9g#9a9g#9f9a9f9e#9 f9h9f9e#9f9c'9f9e#9 "
512 LET M$=M$+"f9d'9f9e#9f9d'9c'9h9 c'9a9g#9f9a8g#8 f4"
513 LET M$=M$+"f8a9f9 c8f9c9A8c9A9 F4cb8H8 e8d#9f9a8g9f9 g8e8g8h9g9 "
514 LET M$=M$+"e8g9e9c8e9c9 A49d9f9d9 e9d9e9d9c9e9g9e9 f9e9f9e9d9f9d9c9 d9g9d9c9d9a9d9c9 d9h9d9c9d9h9a9g9 "
515 LET M$=M$+"a9f9e9d9f8e8 d4f8f8 f8f8d'8f8 f8e8e8e8 e8e8c'8e8 e8d8h8d'9h9 g48h0a0g0f0 e48g0f0e0d0 "
516 LET M$=M$+"cb9e9g9e9cb9H9cb9H9 A#8F8G8F8 H8A#9c9e8d9c9 d8H0c0d0e0f8d9f9 h8f8e9d9c9d9 H4"
530 GOSUB 1000: REM SPIELEN
540 STOP : REM PROGRAMMENDE
000 REM *** MUSIK IN M$ SPIELEN ***
1010 LET P=1
1015 LET LENM=LEN(M$)
1020 REM *** SCHLEIFE ***
1030 LET T=TW(ASC(M$(P))): IF T<0 THEN GOSUB 1300
1032 LET NT$=M$(P): REM FÜR OKTAVE
1035 LET P=P+1: IF TW(ASC(M$(P)))<0 THEN GOSUB 1300: GOTO 1035
1040 LET Z=0: REM TONZEIT ERMITTELN
1045 LET NZ=TW(ASC(M$(P))): IF NZ<0 OR NZ>200 THEN GOTO 1060
1050 LET Z=Z+NZ: LET P=P+1
1055 IF P<LENM THEN GOTO 1045
1060 LET ZM=-L/DL: REM MAX SINNVOLL
1065 IF ZM>Z THEN LET ZM=Z
1070 IF T>0 THEN SOUND 0,T,L,DT,DL,ZM,1 ELSE SOUND 0,0,0,0,0,ZM,1
1072 IF Z-ZM=1 THEN SOUND 0,T,0,0,0,Z-ZM,1
1075 PRINT "*";: REM PROTOKOLL
1080 IF P<LENM THEN GOTO 1020
1100 REM *** MUSIKSTÜCK ZU ENDE ***
1110 SOUND 0,0,0,0,0,1,1
1120 RETURN
1200 REM *** MELODIEFEHLER ***
1210 SOUND 0,1000,0,0,0,1,1: PRINT "ERROR: M$(;P;)"=;CHR$(34);M$(P);CHR$(34): PAUSE 1100: LET ERR=P-1: GOTO 1100
1300 REM *** SONDERFUNKTIONEN ***
1310 ON -1-TW(ASC(M$(P))) GOSUB 1400,1500,1600,1700
1330 RETURN
1400 REM *** PAUSE ***
1410 LET T=0
1420 RETURN
1500 REM *** OKTAVE ***
1510 IF "A"<=MT$ AND MT$<="Z" THEN LET T=T*2 ELSE LET T=T/2
1520 LET T=INT(T+.5)
1530 RETURN
1600 REM *** KREUZ ***
1610 LET T=T*HALB
1620 LET T=INT(T+.5)
1630 RETURN
1700 REM *** b ***
1710 LET T=T/HALB
1720 LET T=INT(T+.5)
1730 RETURN
65000 CLEAR : SBUF 1: SAVE "MUSIC-EDITOR": RUN

```

## ROM-Routinen zum Einbinden in eigene Assemblerprogramme

Die folgenden Einsprungsadressen sind Adressen des ROM's, also nur für einen MTX ohne Floppy gültig. Wo diese Adressen für das FDXB liegen ist bislang noch nicht bekannt. Mit ihnen lassen sich u.U. eigene Assemblerprogramme verkürzen. Vielleicht hat auch irgend jemand eine Idee, wozu man die eine oder andere Routine gut einsetzen kann. Diese Liste stammt von C. Romanazzi, der sich die Mühe gemacht hat, das ROM auf brauchbare Teile hin zu untersuchen. Also los geht's:

Einsprungsadresse: #1850                    Übergabeparameter: in BC

Funktion:

Gibt den Inhalt von BC nibbleweise als ASCII-Code aus.

z.B.    BC = #1234            Ergebnis: 1234

        BC = #FB34            Ergebnis: FB34    auf dem Bildschirm

Damit lassen sich HEX-Zahlen gut auf den Bildschirm bringen !

---- \* ----

Einsprungsadresse: #1343                    Übergabeparameter: HL=Zeilenanfang

Funktion:

Stellt die Länge einer Basic-Zeile fest, das Ergebnis wird in BC festgehalten.

---- \* ----

Einsprungsadresse: #3B87 oder    Übergabeparameter: Adresse in HL  
(RST 28, #BD(=CP L))

Funktion:

Stellt fest, ob das Zeichen in (HL) ein Buchstabe (außer äöüß) ist. Ist dies der Fall, dann ist das Carry-Flag gesetzt.

---- \* ----

Einsprungsadresse: #3B96 oder    Übergabeparameter: Adresse in HL  
(RST 28, #B7(=OR A))

Funktion:

Stellt fest, ob das Zeichen in (HL) eine Zahl ist. Wenn ja, ist das Carry-Flag gesetzt.

---- \* ----

Einsprungsadresse: #1225                    Übergabeparameter:  
DE = RAM-Adresse                    HL = 0

Funktion:

Hinter einem GOTO wird die Zeilennummer dezimal im RAM abgelegt. Diese Routine rechnet nun Dezimal in Hex um. Das Ergebnis steht in HL. Das Carry-Flag ist gesetzt wenn die Zahl >65535 war --> Out of range. War alles in Ordnung, ist das Zero-Flag gesetzt.

---- \* ----

Einsprungsadresse: #124F                    Übergabeparameter: Adresse in DE

Funktion:

Prüft, ob das Zeichen in (DE) ein "+" oder "-" ist. Das Ergebnis wird über Carry- und Zero Flag mitgeteilt:

Bei "+":                    Zero Flag ist gesetzt,

Bei "-":                    Zero Flag und Carry Flag sind gesetzt

Keins von beiden: Nur das Carry-Flag ist gesetzt.

---- \* ----

Einsprungsadresse: #06A3                    Übergabeparameter: Wert in HL

Funktion:

Subtrahiert #80 von HL.

---- \* ----

## SYSTEMVARIABLEN (1. Teil)

ADRESSE (Hex.)	NAME	BYTELAENGE	ERKLAERUNG
FA52	CTRBADR	40	Kontrolbuffer für Sound.
FA7A	LSTPG	1	Beinhaltet Anzahl von bereiten 32K RAM Seiten +1.
FA7B	VARNAM	2	Beinhaltet die Adresse vom Beginn der Variablen- namen
FA7D	VALBOT	2	Beinhaltet FFh. VALPOT plus 1 ist die Adresse des Beginns der Variablenwerte.
FA7F	CALCBOT	2	Beinhaltet die Adresse des Anfangs des Kalkula- torstacks.
FA81	CALCST	2	Beinhaltet die Adresse des oberen Endes des Kal- kulatorstacks +1, das heisst das nächste verfügbare freie Byte.
FA83	KBDBUF	2	Beinhaltet die Adresse des Keyboardbuffer.
FA85	USYNT	4	

USYNT beinhaltet die vier Syntax-Bytes, die gebraucht werden, um dem Computer mit-  
zuteilen, was er ausführen soll, wenn das BASIC-Kommando USER gebraucht wird. Diese  
Bytes können vom Programmierer definiert werden (s.S.1/6). Die vier Bytes werden  
vom obersten zum untersten überprüft, wobei das letzte Byte eine RET-Instruktion  
enthalten muss.

FA89      USER          3

USER beinhaltet die Adresse der Routine, in welche gesprungen werden soll, wenn  
das BASIC-Kommando USER gebraucht wird. Normalerweise beinhaltet es RET, aber es  
kann vom Programmierer umdefiniert werden. Wenn Sie eine neue Adresse in USER  
eingeben wollen, so müssen Sie darauf achten, dass die Adresse in umgekehrter  
Reihenfolge eingegeben wird, d.h. zuerst FA8Bh, dann FA8Ah, dann FA89, ander-  
falls springt der Computer nach 0000, was einem RESET gleichkommt.

### Ein Beispiel:

Ursprünglicher Inhalt von USER:

```
FA89h  C9  RET
FA8Ah  00  nicht belegt
FA8Bh  00  " "
```

Sprung nach DOFFh:

```
FA89h  C3  JUMP-Anweisung nach...
FA8Ah  FF  niedriges Byte zuerst, dann
FA8Bh  D0  höheres Byte
```

## Die Wirkung von verschiedenen Syntax Bytes in USYNT

Syntax Byte (dezimal)	Syntax-Kontrolle für:
0	Numerischer Ausdruck
1	String-Ausdruck
2	Arithmetischer Ausdruck
3	Liste von Ausdrücken getrennt durch "," oder ";"
4	Liste von Zahlen getrennt durch "," zwischen 0 und 64K
5	Liste von arithmetischen Ausdrücken
6	Einzelne Zahl zwischen 0 und 64K
7	Erlaubt alles, d.h. keine Kontrolle
8	INPUT statement
9	IF statement
10	STEP in FOR statement
11	GOTO oder GOSUB
12	I=(arithmetischer Ausdruck) in FOR statement
13	Numerische Variabel oder nichts

ADRESSE (Hex.)	NAME	BYTELAENGE	ERKLAERUNG
FA8D	----	3	Diese Bytes werden vom Computer nicht gebraucht. Die Sprungadresse, die Sie hier eingeben können, wird abgeSAVED.
FA8F	IOPL	1	Liste der Peripherie.
FA90	REALBY	1	Panel Breakpoint. Speichert das Byte vom Breakpoint in GO.
FA91	KBFLAG	1	s. weiter hinten.
FA92	STKLIM	2	Oberes Ende des freien Speicherplatzes.
FA94	SYSTOP	2	Oberes Ende der zu SAVEnden Variablen.
FA96	SSTACK	2	Adresse des Maschinenstacks. Dieser Wert ist im SP (Stack Pointer) Register gespeichert, wenn immer die Maschine ins BASIC zurückkehrt, oder sich ein Error ereignet.
FA98	USERINT	3	Beinhaltet die abhängige Sprungadresse, die gebraucht wird, um das Bit in INTFFF (FD5Eh) zu setzen.
FA9B	NODLOC	3	Beinhaltet die Sprungadresse, die im MTX RING SYSTEM gebraucht wird.

## WICHTIGE MATHEMATISCHE FUNKTIONEN

$$\text{ASN}=\text{ATN}(X/\text{SQR}(1-X*X))$$

$$\text{ACS}=-\text{ATN}(X/\text{SQR}(1-X*X))+\text{PI}/2$$

$$\text{HSN}=(\text{EXP}(X)-\text{EXP}(-X))/2$$

$$\text{HCS}=(\text{EXP}(X)+\text{EXP}(-X))/2$$

$$\text{HTN}=-2*\text{EXP}(-X)/(\text{EXP}(X)+\text{EXP}(-X))+1$$

$$\text{AHS}=\text{LN}(X+\text{SQR}(X*X+1))$$

$$\text{AHC}=\text{LN}(X+\text{SQR}(X*X-1))$$

$$\text{AHT}=\text{LN}((1+X)/(1-X))/2$$

Michael Künzli

## NOTEN: SOUNDWERTE UND FREQUENZEN

NOTE	SOUNDWERTE: Frequenz	Frequenz'	Frequenz''	Frequenz'''
A	1136 (110)	568 (220)	284 (440)	142 (880)
A#,Bb	1068 (117)	536 (233)	268 (466)	134 (932)
B	1016 (123)	506 (247)	253 (494)	127 (988)
C	994 (131)	477 (262)	239 (523)	119 (1047)
C#,Db	899 (139)	451 (277)	226 (554)	113 (1109)
D	850 (147)	425 (294)	213 (587)	106 (1175)
D#,Eb	801 (156)	402 (311)	201 (622)	100 (1245)
E	757 (165)	379 (330)	190 (659)	95 (1319)
F	714 (175)	358 (349)	179 (698)	89 (1397)
F#,Gb	675 (185)	338 (370)	169 (740)	84 (1480)
G	637 (196)	319 (392)	159 (784)	80 (1568)
G#,Ab	601 (208)	301 (415)	150 (831)	75 (1661)

Die eingeklammerten Zahlen sind die Frequenzen der links stehenden Noten.  
Die nicht eingeklammerten Zahlen sind die Werte, die man in SOUND K,N,L bei N eingeben muss.

N berechnet sich mit folgender Formel: 
$$N = \frac{4'000'000}{(32 * \text{Frequenz})}$$

Michael Künzli

## FDXB-ERWEITERUNG XB (EXTENDED BASIC)

Hallo Leute, vielleicht hat Herbert schon in dieser Info etwas über meine Disc-Basic-Erweiterung gesagt, ich will mich an dieser Stelle etwas genauer dazu äußern.

Also, meine Erweiterung umfaßt bis jetzt 24 (vierundzwanzig) neue Befehle (ein paar folgen noch), die insgesamt maximal vier KByte beanspruchen. XB liegt am oberen Ende des Speichers, und vereinnahmt auch die Stelle, die von USERERROR angesprungen wird. Das heißt, daß USERERROR an eine Stelle von XB gelenkt wird, die das Paging übernimmt und dann nach 18AF springt. Alle Routinen, die normalerweise sonst noch über EE8E liegen, werden vom FDXB nämlich nicht benötigt, wie ich herausgefunden habe.

Die Befehle von XB werden alle über den unbenutzten und von mir in PROC umbenannten NODE-Befehl angesprochen, der nun diese Syntax hat: PROC BEFEHL DO (PARAMETER). Die Befehle sind unheimlich vielseitig, sehr sehr schnell, und vereinfachen das Programmieren in Basic ungemein. Hier eine Liste mit Kurzbeschreibung. Ich hätte weder soviel Platz noch Zeit, alle Möglichkeiten ausführlich zu beschreiben.

SWAP DO (A,B,C,...)	Vertauschen numerischer Variablen aller Art. Ist eigentlich ein Rotieren, denn im Beispiel wird A zu B, B zu C und C zu A. Natürlich werden auch Variablen mit (berechneten) Subscripts verarbeitet (zum Sortieren gut geeignet)
SWAP\$ DO (A\$,B\$,C\$...)	Wie SWAP, nur mit Strings.
KILL\$ DO (S\$,A\$,X,I)	Löschen bestimmter Zeichen aus einem String. KILL\$ DO ("A",A\$,X,I) würde das Zeichen "A" maximal X-mal aus A\$ entfernen. I (optional) enthält hinterher die Anzahl der tatsächlich gelöschten Zeichen.
POS DO (A\$,B\$,I)	Sucht in B\$ nach in A\$ definierter Zeichenfolge und legt ihre Position in I ab. Wird I nicht angegeben, geht die Position (0 für nicht da) auf den Schirm. Die Wild Card ist erlaubt. POS DO ("*ACH",B\$,F(100)) mit B\$="BIENENWACHS" legt das Ergebnis (7) in F(100) ab.
WILD\$ DO (A\$)	Umdefinieren der Wild Card. WILD\$ DO ("?" ) definiert ? als Wild Card.
PUT\$ DO (P\$,A\$,B\$(A),...)	Füllen von Strings mit einer in P\$ definierten Zeichenkette. Anders als bei den meisten anderen Stringfunktionen wird die Länge eines Strings nicht verändert. PUT\$ DO ("*-",A\$) legt in A\$ (mit seiner momentanen Länge) eine Folge von Sternen und Strichen ab.
UCASE\$ DO (A\$,...)	Umwandlung von Kleinbuchstaben in Großbuchstaben. Mit SET ist Anpassung an deutschen Zeichensatz möglich.
LCASE\$ DO (A\$,...)	Umkehrung von UCASE\$.

- CONV DO (A\$(5,4),I) Umwandlung von Hexadezimal in Dezimal. Das Ergebnis wird in I abgelegt oder, falls I nicht angegeben wird, auf den Schirm gebracht. Die Länge von A\$ darf nicht null sein und vier Zeichen nicht überschreiten. CONV DO ("10A",I) legt in I den Wert 266 ab.
- CONV\$ DO (I,A\$) Umwandlung von Dezimal in Hexadezimal. Das Ergebnis (immer vier Stellen) geht in A\$ oder auf den Schirm. CONV\$ DO (4\*K) hat als Ergebnis 007C, wenn K=31 ist.
- DOKE DO (ADR,WERT) Das berühmte Double-Poke. ADR und Wert können beliebige Ausdrücke sein, also numerisch oder Strings mit Hex-Werten drin. DOKE DO ("B000", "FAA") legt in B000 den Wert AA und in B001 den Wert F (=15) ab. Wie bei allen Hex-Angaben können für die Werte A bis F (10 bis 15) auch Kleinbuchstaben verwendet werden.
- DEEK DO (ADR,I) Natürlich auch ein Double-Peek. ADR kann numerisch oder ein String sein. I ist numerische Variable, die weggelassen werden kann. Dann erscheint das Ergebnis auf dem Schirm.
- DEEK\$ DO (ADR,I\$) Wie DEEK, nur ist das Ergebnis eine Hex-Zahl.
- FIND\$ DO (A\$,ADR,P\$) Durchsuchen des Speichers ab Adresse ADR nach einer in A\$ definierten Zeichenfolge. ADR wie bei DEEK und DOKE. Die Hexadresse, bei der die gesuchte Zeichenkette steht, wird in P\$ abgelegt oder auf dem Schirm ausgegeben.
- ASCII\$ DO (A\$,B\$) Umwandlung eines Strings mit Hex-Werten in einen String mit ASCII-Werten. Beide Parameter müssen angegeben werden. ASCII\$ DO ("8081FF", B\$) legt in B\$ CHR\$(128)+CHR\$(129)+CHR\$(255) ab. Ideal im Zusammenhang mit FIND\$.
- GET DO (A\$,I) Komfortable Tastaturabfrage. GET DO ("YnN",I) wartet so lange, bis eine der angegebenen Tasten gedrückt wird und schreibt den ASCII-Code der gedrückten Taste in I. I ist optional, wird es weggelassen, dann trotzdem keine Ausgabe auf dem Schirm. GET DO ("") ist viel praktischer als 100 IF INKEY\$="" THEN GOTO 100.
- JUMP DO (AUSDRUCK) Kann GOTO auf spektakuläre Weise ersetzen. AUSDRUCK kann alles sein. Eine berechnete Zeilennummer (ohne lästiges "No Line"), oder ein Label (auch in einer Stringvariablen), das in einer REM-Zeile definiert ist. Beispiel (springt zu Zeile 500):
- ```

100 LET A=1
110 IF A=1 THEN PROC JUMP DO ("TASTATUR")
120 REM      * weitere Berechnungen *
500 REM      :TASTATUR: * weitere REMarks *
510 PROC GET DO ("1234567890",A)

```

|                                                                                                                             |                                                                                                                                                                                                                                                                                                                             |
|-----------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CALL DO (AUSDRUCK)                                                                                                          | Wie JUMP, nur als Unterprogrammaufruf, der ganz normal mit RETURN abgeschlossen wird.                                                                                                                                                                                                                                       |
| EXIT DO (CALL)                                                                                                              | Nimmt eine RETURN-Rücksprungadresse vom Stapel.                                                                                                                                                                                                                                                                             |
| EXIT DO (LOOP)                                                                                                              | Beendet die zuletzt aufgestellte FOR-NEXT-Schleife, ohne den momentanen Inhalt der Laufvariablen zu verändern.                                                                                                                                                                                                              |
| SCREEN\$ DO (A\$)                                                                                                           | Betrifft die Schirmausgabe. Was bei PROC-Befehlen auf den Schirm kommt, wird normalerweise mit anschließendem line feed und carriage return ausgegeben. Das läßt sich ändern. SCREEN\$ DO (CHR\$(9)) beispielsweise gibt stattdessen ein Tab aus (und eigentlich auch ein CHR\$(0)). Man kann maximal zwei Zeichen angeben. |
| COM\$ DO (A\$)                                                                                                              | Wem PROC als Befehlsname nicht gefällt, der kann hier was anderes vereinbaren. Aber nur genau vier Zeichen lang.                                                                                                                                                                                                            |
| SET DO (A\$)                                                                                                                | Ersetzt bestimmte POKEs. SET DO ("B") aktiviert die BRK-Taste. Die Kürzel sind: B=Break, I=Interrupt, N=Numeric keypad, C=Caps lock, T=Tabulator. Letzteres ersetzt ein an den Drucker geschicktes CHR\$(9) durch eine entsprechend berechnete Anzahl von Spaces.                                                           |
| RES DO (A\$)                                                                                                                | Die Umkehrung zu SET.                                                                                                                                                                                                                                                                                                       |
| So, das sind die bis jetzt fertig programmierten XB-Befehle. Jetzt die, die mit größter Wahrscheinlichkeit noch dazukommen. |                                                                                                                                                                                                                                                                                                                             |
| PRINTER\$ DO (N,A\$,B\$)                                                                                                    | Für die Druckerausgabe, besonders bei Typenrad-druckern. Sechs einzelne auszugebene Zeichen können durch jeweils bis zu acht andere ersetzt werden. N ist eine Zahl zwischen 1 und 6. PRINTER\$ DO (1,"?"," * ").                                                                                                           |
| DUMP\$ DO (ADR,ANZAHL,A\$)                                                                                                  | ADR und ANZAHL wie bei DOKE. Ein Speicherbereich der Länge ANZAHL, ab Adresse ADR wird Als Folge zweistelliger Hex-Zahlen nach A\$ geschrieben. Wird A\$ weggelassen, dann Ausgabe auf dem Schirm mit Spaces zwischen den Hex-Zahlen.                                                                                       |

Vielleicht kommen noch mehr Befehle dazu. Ich finde aber, daß es jetzt schon ganz schön viele sind. Was die Fehlermeldungen betrifft, so ist das ein Kapitel für sich. Ihre Aus-gabe ist die gleiche wie bei FDXB, sie erscheinen aber nur dann, wenn es unbedingt nötig ist. "No line", wie gesagt, bekommt man nur, wenn man mit Labels arbeitet. PUT\$ erzeugt keine Fehlermeldung, wenn in einen String nichts hineingeschrieben werden kann. Nicht vorhandene Variablen oder falsche Subscripts werden natürlich mit "Undefined" bzw. "Subscript" belohnt, CONV DO ("G") gibt natürlich "Out of range", usw.

So, nun zu Erscheinungsdatum und Preis. XB wird ungefähr zwei Wochen nach dem Erscheinen dieser Info fertig sein, wenn mich nicht gerade irgendwelche Leute anrufen und mir sagen "also ich brauche unbedingt noch den und den Befehl zu meiner absoluten Zufriedenheit". Eine umfangreiche Bedienungsanleitung wird dann auch fertig sein. Der Preis ist, gemessen an der Arbeit, die ich hatte und noch habe, extrem niedrig, nämlich 42,- DM netto, Datenträger 6,- DM, P+V 2,- DM (per Nachnahme wirklich nur in Ausnahmefällen, ich hab' schon so genug zu tun). Und bitte fragt nicht, ob es das Ganze nicht auch für MTX ohne Floppy gibt; leider noch nicht, obwohl die Möglichkeit generell besteht, wenn auch so mancher Befehl dann wegfallen müßte.

#### QUIBIC 3D von Herbert Herberg

Als interessierter, wenn auch nicht leidenschaftlicher Schach- und Damenspieler reizen mich natürlich auch andere Strategiespiele, und was fordert einen da mehr heraus als ein dreidimensionales Vier-Gewinnt. Und gleich beim Antesten fiel mir auf: Das Programm erkennt gefährliche Situationen sehr schnell. Die Spielstärke scheint also wirklich hoch zu sein. Alle Partien, die ich bis jetzt gegen den MTX gespielt habe, hab' ich verloren, was aber - das muß man zugeben - zum Teil daran lag, daß ich die Spielsteine mangels Kontrast auf meinem monochromen Monitor nicht erkennen konnte. Auch wären vielleicht die Ebenen und damit die Räumlichkeit besser zu erkennen gewesen, wenn man sie als dunkle Flächen mit Trennlinien in der Hintergrundfarbe gestaltet hätte. Als sehr gut gelöst empfand ich die Art der Eingabe. Mit dem Joystick oder den Cursor-Tasten bewegt man seinen Spielstein, der links unten darauf wartet, auf das gewünschte Feld und drückt dann die Feuertaste bzw. HOME. Der Stein wird dann zurechtgerückt, und gleich legt der MTX seinen Stein hin. Spielstark und schnell und jedem zu empfehlen, der ein Farb-Sichtgerät und keine Schwierigkeiten mit nur einfacher räumlicher Darstellung hat.

## H A S H I N G

Ein Beispielprogramm zu Michaels Artikel auf Seite 8  
(Selbverständlich von Michael Möwe!

```

10 REM Dieses Programm von mir entstand leider nicht synchron mit
   der
20 REM HASH-Beschreibung (weiter vorne in diesem Info)!
30 REM Deshalb sind einige Bezeichner anders!
40 REM
50 REM Der Text      TXT# der Beschreibung ist hier in HATAB#
60 REM Der Zeiger    PTR  der Beschreibung ist hier in PTR
70 REM Der Schlüssel KEY der Beschreibung ist hier nicht gesonde
   rt
80 REM
90 REM
990 DIM HATAB$(113,5),PTR(113)
993 REM HATAB & PTR LOESCHEN
995 FOR I=1 TO 113: LET HATAB$(I)="      ": LET PTR(I)=I: NEXT I
1000 REM HASHING
1100 REM NUR GROSSBUCHSTABEN UND ZIFFERN
1200 INPUT "GIB EINEN STRING EIN :";EIN$
1300 GOSUB 2000
1400 GOSUB 3000
1500 GOSUB 4000
1700 GOTO 1200
2000 REM SCHLUESSEL-BERECHNUNG
2050 LET KODE=0
2060 LET SAV$=EIN$
2070 IF LEN(EIN$)>5 THEN LET DIS=5 ELSE LET DIS=LEN(EIN$)
2100 LET EIN$=MID$("      "+EIN$,1+DIS,5)
2200 FOR I=1 TO 5
2300 LET KODE=64*KODE
2400 IF I<=LEN(EIN$) THEN LET KODE=KODE+ASC(MID$(EIN$,I,1))-32
2500 NEXT I
2600 RETURN
3000 REM TABELLEN-EINTRAG
3050 LET ZLR=0
3100 LET ZGR=MOD(KODE,19)+1
3125 LET ZGR=INT(ZGR)
3150 LET HZGR=ZGR
3170 PRINT ZGR,HATAB$(ZGR)
3200 IF HATAB$(ZGR)="      " THEN LET HATAB$(ZGR)=EIN$: LET PTR(HZ
   GR)=ZGR: RETURN
3300 REM TABELLENPLATZ IST BELEGT
3400 LET HZGR=ZGR
3500 IF PTR(ZGR)=ZGR THEN GOSUB 3702 ELSE LET ZGR=PTR(ZGR)
3600 LET ZLR=ZLR+1
3700 IF ZLR<113 THEN GOTO 3200
3702 REM FREIEN PLATZ SUCHEN
3710 LET ZGR=ZGR+1
3715 IF ZGR=20 THEN LET ZGR=1
3720 IF HATAB$(ZGR)="      " THEN RETURN
3730 LET ZLR=ZLR+1
3750 IF ZLR<19 THEN GOTO 3702
3800 PRINT "TABELLE VOLL"
3900 STOP
4000 REM TABELLE ANZEIGEN
4010 PRINT CHR$(12);
4020 PRINT "INDEX", "INHALT", "ZEIGER": PRINT : PRINT
4100 FOR I=1 TO 19: PRINT I,HATAB$(I),PTR(I): NEXT I
4190 INPUT QQQ$
4200 RETURN

```

## Echtzeituhr für den 80 - Zeichen - Bildschirm

20 CODE

```

8007 START:  LD HL,UHR;           Siehe Info 2,
800A         LD DE,#E001;       Uhr für 40 - Zeichen-Schirm
800D         LD BC,#FO
8010         LDIR
8012         LD A,#C3
8014         LD (#FA98),A
8017         LD HL,#E001
801A         LD (#FA99),HL
801D         XOR A
801E         LD (#E000),A
8021 INI:    LD HL,#76F;         Bildschirmposition
8024         LD C,#39;         Initialisierung
8026         LD A,12;         der 6845-Register
8028         OUT (#38),A
802A         IN D,(C)
802C         INC A
802D         OUT (#38),A
802F         IN E,(C)
8031         ADD HL,DE
8032         LD (#E100),HL;     hier steht die aktuelle Bildschirmposition
8035         RET
8036 UHR:    LD A,(#E000)
8039         INC A
803A         LD (#E000),A
803D         CP #40
803F         JR NZ,END
8041         DI
8042         LD DE,(#E100);     Bildschirmposition holen
8046         LD HL,#FD57
8049         LD C,02
804B         LD B,06
804D LOOP:   LD A,(HL)
804E AUSG:   OUT (#32),A
8050         LD A,#04;         bright-up
8052         OUT (#33),A
8054         LD A,D
8055         AND #07
8057         OR #E0
8059         OUT (#31),A
805B         LD A,E
805C         OUT (#30),A
805E         DEC C
805F         JR NZ,WEIT
8061         LD A,#3A
8063         INC DE
8064         LD C,03
8066         JR AUSG
8068 WEIT:   INC HL
8069         INC DE
806A         DJNZ LOOP
806C         XOR A
806D         LD (#E000),A
8070         EI
8071 END:    RET
8072         RET

```

Symbols:

```

UHR8036START8007
INI8021END8071
LOOP804DAUSG804E
WEIT8068

```

```

30 PRINT : PRINT : PRINT
40 INPUT "           Bitte Uhrzeit eingeben! 'hhmmss': ";C$
50 CLOCK C$
60 POKE 64862,31
70 PRINT : PRINT
80 PRINT "           Uhr ausschalten mit: POKE 64862,15 !"
90 STOP

```

Viel Spaß beim Ausprobieren!

Gruß Rüdiger J. Hollmann

CP/M Assembler Programme mit FDXB erstellen

## 1. Programm CODEB200.BAS

Statements 0 bis 50 eintippen, danach mit ASSEM 100 das Assemblerprogramm eingeben. Es muß ab #B200 beginnen. Nach erfolgreicher Assemblierung mit RUN das Basic-Programm durchführen. Es speichert den Z80-Maschinencode für die Laufadresse #B200 und dessen Länge in der Datei CODEB200.HEX.

## 2. Programm CODEB100.BAS

In CODEB200.BAS sind folgende Änderungen durchzuführen:

Statement 0: "CODEB200." -> "CODEB100."

Statement 1: 512 -> 256

Löschen Statements 20-50

ASSEM 100 eingeben, damit das Assemblerprogramm mit der neuen Startadresse #B100 assembliert wird. (Dies ist notwendig)

Assembler wieder verlassen, ohne Änderungen durchzuführen

Nun wird auch dieses Basic-Programm mit RUN gestartet. Es speichert nun den Maschinencode für die Startadresse #B100 in die Datei CODEB100.HEX.

## 3. Erzeugen des .COM Files (Startadresse #100)

Starten des Programmes RELOCATE.BAS. Es liest die beiden .HEX Files und vergleicht sie Byte für Byte. Wenn Ungleichheit vorliegt, wird vom Wert des entsprechenden Bytes aus CODEB100.HEX #B0 = 128 subtrahiert und die so geänderte Datei CODEB100.HEX als Command-File CODE.COM abgespeichert.

## 4. Durchführung des Assemblerprogrammes unter CP/M

Na klar, nach dem Verlassen des Basic mit dem Aufruf: CODE<ret>.

## 5. Das war es schon.

## Programm RELOCATE.BAS

```

10 LET N=33000
20 DISC READ "CODEB100.HEX",N
30 LET L=PEEK(N+1)*256+PEEK(N)
40 DISC READ "CODEB200.HEX",N+L+2
50 FOR I=N+2 TO N+L+1
60 IF PEEK(I)<>PEEK(I+L+2) THEN POKE I,PEEK(I)-128
70 NEXT
80 DISC WRITE "CODE.COM",N+2,L
90 STOP
100 DISC SAVE "RELOCATE.BAS"

```

## Programm CODE8100.BAS

```

0 DIM N$(1,9): LET N$(1)="CODE8100."
1 LET N=8*4096+256
2 LET A=PEEK(N-1)*256+PEEK(N-2)
3 REM <A>=LAENGE DES MASCH.CODES
4 DISC ERA N$(1)+"HEX"
5 DISC WRITE N$(1)+"HEX",N-2,A-2
6 DISC SAVE N$(1)+"BAS"
7 STOP
10 REM HIER SIND 60 BELIEBIGE ZEICHEN UNTERZUBRINGEN ( => #8100 )
100 CODE

8100      JP START      ; ABSICHTLICH EIN ABSOLUTER SPRUNG
8103 MELDUNG:DB #0D,#0A,"ZUM BEENDEN BITTE EINE TASTE BETAETIGEN","$"
812D START: LD DE,MELDUNG
8130      LD C,9        ;STRING-AUSGABE
8132      CALL 5
8135      CALL WAIT     ;WARTEN AUF KONSOL-EINGABE
8138      JP 0          ;WARMBOOT
813B WAIT: LD C,1
813D      CALL 5
8140      CP #0D
8142      JR NZ,WAIT
8144      RET

```

## Symbols:

```

START 812D      MELDUNG 8103
WAIT 813B

```

## Programm CODE8200.BAS

```

0 DIM N$(1,9): LET N$(1)="CODE8200."
1 LET N=8*4096+512
2 LET A=PEEK(N-1)*256+PEEK(N-2)
3 REM <A>=LAENGE DES MASCH.CODES
4 DISC ERA N$(1)+"HEX"
5 DISC WRITE N$(1)+"HEX",N-2,A-2
6 DISC SAVE N$(1)+"BAS"
7 STOP
10 REM HIER SIND 60 BELIEBIGE ZEICHEN UNTERZUBRINGEN ( => #8100 )
20 REM 12345678901234567890123456789012345678901234567890123456789012345678901234567890123
30 REM 12345678901234567890123456789012345678901234567890123456789012345678901234567890123
40 REM 12345678901234567890123456789012345678901234567890123456789012345678901234567890123
50 REM 123456789
100 CODE

8200      JP START      ; ABSICHTLICH EIN ABSOLUTER SPRUNG
8203 MELDUNG:DB #0D,#0A,"ZUM BEENDEN BITTE EINE TASTE BETAETIGEN","$"
822D START: LD DE,MELDUNG
8230      LD C,9        ;STRING-AUSGABE
8232      CALL 5
8235      CALL WAIT     ;WARTEN AUF KONSOL-EINGABE
8238      JP 0          ;WARMBOOT
823B WAIT: LD C,1
823D      CALL 5
8240      CP #0D
8242      JR NZ,WAIT
8244      RET

```

## Symbols:

```

START 822D      MELDUNG 8203
WAIT 823B

```

**TURBO-PASCAL Version 3.0**

Was bietet die dritte Version des 'Volkscompilers' ? Zunächst einmal muß in aller Deutlichkeit gesagt werden: Eine Enttäuschung. Die so hoch gepriesene Geschwindigkeitssteigerung ist gleich null. Es muß sogar gesagt werden, daß die Bildschirmausgabe in der neuen Version deutlich langsamer ist als in Vers.2.0 (mir ist schleierhaft, warum). Die Neuerungen:

Zwei Funktionen, um an ein Programm übergebene Parameter auszuwerten. Die Neuerung ist hierbei allerdings nur insofern zu gebrauchen, daß sie die selbstgeschriebenen Prozeduren und Funktionen ersetzen und damit das Programm etwas verkürzen kann. Aber auch mit diesen Funktionen ist die max. Anzahl an Zeichen, die aus der Kommandozeile übernommen werden können, dreißig, obwohl CP/M ja durchaus 127 zuläßt. Was sich Borland dabei gedacht hat, mag der Teufel wissen. Ein Patching für Turbo-Pascal, das die Kommandozeile rettet (zumindest 124 Zeichen), ist auf Seite 2 beschrieben.

Zwei zusätzliche File-Funktionen, ähnlich EOLN und EOF, melden aber auch dann schon true, wenn noch Leerzeichen oder CRLF folgen, aber keine anderen Zeichen mehr.

Eine Prozedur namens EXIT, nichts anderes als ein GOTO Blockende, also ein vorzeitiges Beenden des aktuellen Blockes. Diese Ergänzung ist mehr kosmetischer als praktischer Natur.

Ferner eine neue Standard-Variable namens ErrorPtr, die, wenn gesetzt, auf eine Fehlerbehandlungsroutine zeigt. Bei einem auftretenden Fehler wird dann zu dieser Routine verzweigt. Damit lassen sich noch abschließende Maßnahmen im Falle eines Runtime- oder I/O-Fehlers unternehmen, z.B. Files schließen, detaillierte Fehlermeldung ausgeben, und so weiter. Es ist jedoch kein Error-Handling im Sinne der Basic-Anweisung 'ON ERROR GOTO', da in Pascal das Programm mit 'halt' abgebrochen werden muß.

Und letztlich eine Prozedur namens OVRDRIVE, mit der erreicht werden kann, daß Overlay-Files auch auf anderen als dem Bezugslaufwerk untergebracht werden können.

Fazit: Es ist fraglich, ob sich ein Umtausch lohnt, weil die Neuerungen wirklich spärlich ausgefallen sind und eine Geschwindigkeitssteigerung (zumindest unter CP/M 80) nicht merklich ist.

Noch ein Tip, der nicht im Handbuch steht, aber gelegentlich ganz nützlich ist:

Turbo zählt Zeilen/Spalten bei 1 beginnend. Gibt man nun bei gotoXY eines der Argumente als 0 an, so wird die momentane Cursor-Zeile/Spalte beibehalten. Ein Beispiel:

Cursor steht auf 27,10:

- gotoxy(1,0) bringt den Cursor an den Anfang von Zeile 10, also Position 1,10.
- gotoxy(0,6) bringt den Cursor auf Position 27,6.

## Patching für Turbo-Pascal zum Retten der CP/M-Kommandozeile:

Diese kurze Routine wird an den Anfang von Turbo-Pascal geschrieben (über das Copyright - Pfui! Aber woanders ist kein Platz) und kommt so auf jedes COM-File, das von Turbo-Pascal geschrieben wird, arbeitet deshalb in jedem Programm. Sie arbeitet folgendermaßen:

Am Anfang eines jeden COM-Files findet eine Initialisierung statt. Der erste Befehl des ersten dabei aufgerufenen Upros wird durch einen Sprung ersetzt, der zum Patch führt. Dort wird zuerst geprüft, ob es sich um den ersten Aufruf handelt. Wenn nein, erfolgt ein Rücksprung, jedoch nicht, ohne vorher den ersetzten Befehl auszuführen. Ist es der erste Aufruf, so wird der zweite BIOS-Sprung (Warmstart) verbogen, so daß daran erkannt werden kann, ob die Routine schon gelaufen ist. Dann wird die Kommandozeile in den Speicher ab FREE, also den Free Space Pointer geschrieben, falls dort genug Platz ist. Dort kann sie durch eine einfache Routine ausgewertet werden. Am Ende eines jeden Turbo-Pascal-Programms (eigentlich am Ende eines jeden CP/M-Programms) steht der Sprung nach 0, also der Warmstart. Der verbogene Sprung in der BIOS-Leiste führt zu einer kurzen Routine, die denselben wieder herstellt, damit das System danach wieder ordentlich läuft.

Zum Nutzen dieser Routine läßt sich sagen, daß sie nur dann sinnvoll anzuwenden ist, wenn im Aufruf eines Programms mehr als zwei Worte übergeben werden. Die ersten zwei Worte werden von CP/M nochmals in Speicher ab \$005C bzw. \$006C abgelegt und werden von Turbo-Pascal nicht zerstört. Im Allgemeinen reichen diese zwei Worte völlig aus.

Diese Änderungen werden am Besten mit VDEB unter der Assemble-Option vorgenommen, wo die Befehle genau wie im Listing geschrieben eingetippt werden können. Deshalb fehlt auch bei allen Zahlen eine Angabe über das Zahlensystem; es sind alles Hex-Zahlen. Wenn die Änderungen eingegeben wurden, VDEB mit <BRK> beenden und das neue Turbo-Pascal mit **SAVE 121 TURBON.COM** abspeichern.

Um anschließend den String auswerten zu können, liest ihn folgende Funktion ein:

```
FUNCTION Read_CMD-Line : AnyStr;    (* AnyStr ist type String(.128.)
                                     oder länger *)
var S      : AnyStr;
    i      : byte;
    Free: integer absolute $40 ; (* Speicherstelle für FREE-Pointer *)

begin
  S:='';      (* Aufnahmestring löschen *)
  for i:=1 to mem(.Free.) do
    S:=S+chr(mem(.Free+i.));  (* Zeichen für Zeichen auslesen *)

  Read_CMD_Line := S      (* Funktionszuweisung *)
end;
```

Die Beschreibung bezieht sich auf Turbo 2.0, die Änderungen für Turbo 3.0 sind in Klammern angegeben.

| Adresse                                                       | Befehl       |                                                                      |
|---------------------------------------------------------------|--------------|----------------------------------------------------------------------|
| 0103                                                          | LD HL, (110) | ; DIESES IST BEREITS DAS WIEDERHERSTELLEN DES ZEIGERS                |
| 0106                                                          | LD DE, (10E) |                                                                      |
| 010A                                                          | LD (HL), E   |                                                                      |
| 010B                                                          | INC HL       |                                                                      |
| 010C                                                          | LD (HL), D   |                                                                      |
| 010D                                                          | JP 0         |                                                                      |
| 0110                                                          | NOP          | ; HIER UND IM VORIGEN SPRUNG WERDEN ADRESSEN ABGELEGT                |
| 0111                                                          | NOP          |                                                                      |
| ; JETZT KOMMT DIE EIGENTLICHE ÜBERTRAGUNGSROUTINE             |              |                                                                      |
| 0112                                                          | EXX          | ; ALLE REGISTER RETTEN                                               |
| 0113                                                          | LD HL, (1)   | ; ZEIGER AUF BIOS-SPRUNG                                             |
| 0116                                                          | INC HL       | ; ZEIGER AUF ZIELADRESSE                                             |
| 0117                                                          | LD (110), HL | ; RETTEN                                                             |
| 011A                                                          | LD A, (HL)   | ; LOW-BYTE DER ZIELADRESSE                                           |
| 011B                                                          | INC HL       |                                                                      |
| 011C                                                          | LD H, (HL)   | ; HIGH-BYTE                                                          |
| 011D                                                          | LD L, A      | ; ZIELADRESSE JETZT IN HL                                            |
| 011E                                                          | DEC H        | ; FALLS H=1, NICHT ERSTER AUFRUF                                     |
| 011F                                                          | JR Z, 142    | ; WENN JA, ZUM ENDE SPRINGEN                                         |
| 0121                                                          | INC H        | ; H WIEDER INCR.                                                     |
| 0122                                                          | LD (10E), HL | ; UND RETTEN                                                         |
| 0125                                                          | LD HL, (110) | ; ADRESSE DES BIOS-SPRUNGES NACH HL                                  |
| 0128                                                          | LD DE, 103   | ; NEUER WERT NACH DE                                                 |
| 012B                                                          | LD (HL), E   | ; UND EINTRAGEN                                                      |
| 012C                                                          | INC HL       |                                                                      |
| 012D                                                          | LD (HL), D   |                                                                      |
| ; DAMIT SIND DIE ZEIGER VERBOGEN, JETZT FOLGT DIE ÜBERTRAGUNG |              |                                                                      |
| 012E                                                          | LD HL, (46)  | ; (TOAM) NACH HL LADEN                                               |
| 0131                                                          | LD DE, (40)  | ; (FREE) NACH DE                                                     |
| 0135                                                          | LD BC, 80    | ; LANGE NACH BC                                                      |
| 0138                                                          | SBC HL, DE   | ; FREIEN SPEICHER BERECHNEN                                          |
| 013A                                                          | SBC HL, BC   | ; GROSS GENUG ?                                                      |
| 013C                                                          | JP M, 142    | ; FALLS NEIN, SPRUNG ZUM ENDE                                        |
| 013F                                                          | LD H, B      |                                                                      |
| 0140                                                          | LD L, C      | ; HL = 80 WIRD QUELLADRESSE                                          |
| 0141                                                          | LDIR         | ; ÜBERTRAGEN: DE ALS ZIEL UND BC ALS LANGE SIND NOCH KORREKT GESETZT |
| 0143                                                          | EXX          | ; REGISTER WIEDER HERSTELLEN                                         |
| 0144                                                          | LD (0DB), HL | ; DEN ERSETZTEN BEFEHL AUSFÜHREN                                     |
| (0144                                                         | LD (0D2), HL | ; FÜR TURBO 3.0)                                                     |
| 0147                                                          | JP 390       | ; UND ZURÜCK INS UPRO                                                |
| (0147                                                         | JP 367       | ; FÜR TURBO 3.0)                                                     |
| ; JETZT MUSS NOCH DER SPRUNG IM UPRO GESETZT WERDEN           |              |                                                                      |
| 038D                                                          | JP 112       |                                                                      |
| (0364                                                         | JP 112       | ; IN TURBO 3.0 STEHT ES AN ANDERER STELLE )                          |

## Steuerung der Bildschirmattribute mit TURBO-Pascal

TURBO-Pascal-Programme stellen Text normalerweise in einfacher heller Schrift dar. Um auch die anderen Bildschirmattribute zu nutzen, habe ich folgende Prozeduren geschrieben, die als Include-File in ein eigenes Pascalprogramm aufgenommen werden können.

Das Include-File SCHIRM.ATT bietet folgende Möglichkeiten:

Underline(an/aus) : Unterstreichen  
Hell(an/aus) : Hell- bzw. Normaldarstellung  
Reverse(an/aus) : Inverse Darstellung  
Background(an/aus): Hintergrund  
Flash(an/aus) : Blinken

Der Aufruf einer dieser Prozeduren läßt alle anderen unverändert. Es ist also möglich, mehrere Attribute zu mischen, um andere Effekte zu erzielen. Einige Beispiele zeigt das Programm auf Seite 2. Damit es läuft, muß jedoch zunächst das File SCHIRM.ATT auf der Diskette sein. Dieses Programm ist auf Seite 3.

ClrAttribute : Alles auf aus

Hiermit können alle Attribute auf aus gesetzt werden. Danach ist also wieder normale Darstellung gewählt.

Die Aufrufe der Prozeduren aus SCHIRM.ATT sind unterstrichen. Ansonsten erklärt sich das Programm von selbst; am besten einfach mal laufen lassen.

Olaf Krumnow

```
program Teste_Bildschirm_Attribute; (* Di af Krumnow *)
(*$I SCHIRM.ATT Include-File muß schon auf Diskette sein !!*)
begin
  ClrScr;
  Hell(an);
  Underline(an);
  writeln('Dieses ist ein Test');
  Hell(aus);Underline(aus);
  writeln;writeln;
  writeln('Normaltext');
  Reverse(an);
  writeln('Einfach Invers');
  Background(an);
  writeln('HellInverse an');
  Background(aus);
  writeln('Wieder aus');
  Reverse(aus);
  background(an);
  writeln('Einfacher Hintergrund');
  Hell(an);
  writeln('Hell auf Hintergrund');
  Underline(an);
  writeln('Auch hier unterstrichen möglich');
  ClrAttribute;
  writeln('Alles gelöscht');
  background(an);
  writeln('Wieder einfacher Hintergrund');
  background(aus);
  flash(an);
  write('Blinken ');
  Hell(an);write('ist in ');
  Underline(an);write('ALLEN');Underline(aus);
  background(an);write(' Kombinationen ');
  reverse(an);writeln('möglich !');
  ClrAttribute;
  writeln;writeln('Ende des Tests!')
end.
```

```

(*)          Include-Datei SCHIRM.ATT          *)
(*)  Prozeduren zur Bildschirm-Attribut-Steuerung *)
(*)          geschrieben von Olaf Krumnow      *)

```

```

type tSwitch = (an,aus);
var attribute : byte;

```

```

procedure setze (var nr, wort : byte);

```

```

begin
  wort:=wort or (1 shl nr)
end;

```

```

procedure resetze (var nr, wort : byte);

```

```

begin
  wort:=wort and ((1 shl nr) xor $ff)
end;

```

```

procedure set_byte (nr : byte; switch : tSwitch);

```

```

begin
  case switch of
    an  : setze(nr, attribute);
    aus : resetze(nr, attribute);
  end;
  write(^F, chr(attribute));
end;

```

```

procedure ClrAttribute;

```

```

begin
  attribute:=0;
  write(^F, chr(attribute))
end;

```

```

procedure Hell(switch : tSwitch);

```

```

begin
  set_byte(2, switch)
end;

```

```

procedure Underline(switch : tSwitch);

```

```

begin
  set_byte(0, switch)
end;

```

```

procedure reverse(switch : tSwitch);

```

```

begin
  set_byte(4, switch)
end;

```

```

procedure background(switch : tSwitch);

```

```

begin
  set_byte(5, switch)
end;

```

```

procedure flash(switch : tSwitch);

```

```

begin
  set_byte(6, switch)
end;

```

```

PROGRAM TURM(INPUT,OUTPUT);
TYPE POINTER = ^SCHEIBEN;
   SCHEIBEN = RECORD
       GROESSE : INTEGER;
       ZEIGER  : POINTER;
   END;
   BEREICH = 1..10;

VAR STAB_A, STAB_B, STAB_C, STAB : POINTER;
    ANZAHL : BEREICH; (* ANZAHL DER SCHEIBEN *)
    I, ANZAHL_BEWEGUNG : INTEGER;

    (* STAB_A  : ANFANGSSTAB
       STAB_B  : HILFSSTAB
       SRAB_C  : ENDSTAB      *)

PROCEDURE DRUCK;
VAR J : INTEGER;

PROCEDURE WELCHE_RINGE(STAB_X : POINTER);
    (* DIESE PROZEDUR STELLT FEST, WELCHE RINGE IN WELCHER REIHENFOLGE
       AUF DEM STAB LIEGEN, WELCHER MIT DEM AUFRUF UEBERGEHEN WURDE. *)

BEGIN
    IF NOT (STAB_X = NIL)
    THEN BEGIN
        WHILE NOT (STAB_X^.ZEIGER = NIL) DO
            BEGIN
                WRITE(' ', STAB_X^.GROESSE:1, ' " ', ' ');
                STAB_X := STAB_X^.ZEIGER;
                END;
                WRITE(' ', STAB_X^.GROESSE:1, ' " ', ' ');
                END;
            END; (* OF WELCHE_RINGE *)

BEGIN
WRITE(' ', ANZAHL_BEWEGUNG:3, ' . BEWEGUNG');
WRITELN('          SCHEIBEN MIT DER GROESSE : ');
WRITELN('          -----');
ANZAHL_BEWEGUNG := ANZAHL_BEWEGUNG + 1;
WRITELN;
WRITE('ANFANGSSTAB ENTHAELT : ');
WELCHE_RINGE(STAB_A); WRITELN;
WRITE('HILFSSTAB ENTHAELT  : ');
WELCHE_RINGE(STAB_B); WRITELN;
WRITE('ENDSTAB ENTHAELT   : ');
WELCHE_RINGE(STAB_C); WRITELN;
FOR J:=1 TO 50 DO WRITE(' - ');
FOR J:=1 TO 4 DO WRITELN;
END; (* OF DRUCK *)

PROCEDURE SETZE(VAR STAB_1, STAB_2, STAB_3 : POINTER; ZAHL : BEREICH);

PROCEDURE BEWEGE(VAR STAB_ANF, STAB_ENDE : POINTER);
    (* DIE OBERSTE SCHEIBE VOM ANFANGSSTAB WIRD AUF DEN ENDSTAB GESETZT. *)

    VAR HILFSZEIGER : POINTER;
    BEGIN
        HILFSZEIGER := STAB_ANF;
        STAB_ANF := STAB_ANF^.ZEIGER;
        HILFSZEIGER^.ZEIGER := STAB_ENDE;
        STAB_ENDE := HILFSZEIGER;
    
```

```

DRUCK;
END;

BEGIN
IF ZAHL >= 1
  THEN BEGIN
    SETZE(STAB_1, STAB_3, STAB_2, (ZAHL-1));
    BEWEGE(STAB_1, STAB_3);
    SETZE(STAB_2, STAB_1, STAB_3, (ZAHL-1));
    END;
END; (* OF SETZE *)

BEGIN (* OF MAIN *)
FOR I:=1 TO 5 DO WRITELN;
WRITE('GEBEN SIE DIE ANZAHL DER SCHEIBEN AN (1..10): ');
READLN; READ(ANZAHL);
FOR I:=1 TO 4 DO WRITELN;
(* AUFBAU EINER KETTENSTRUKTUR FUER DEN ANFANGSSTAB. *)
STAB_A := NIL; STAB_B := NIL; STAB_C := NIL;
FOR I:=ANZAHL DOWNT0 1 DO
  BEGIN
    NEW(STAB);
    STAB^.GROESSE := I;
    STAB^.ZEIGER := STAB_A;
    STAB_A := STAB;
  END;
(* DRUCK DER AUSGANGSSITUATION. *)
ANZAHL_BEWEGUNG := 1;
DRUCK;
SETZE(STAB_A, STAB_B, STAB_C, ANZAHL);
END.

```

JRT-Pascal: Das von mir im Info 3 erwähnte JRT-Pascal ist jetzt in der März-Ausgabe der Zeitschrift "MC" recht ausführlich vorgestellt worden (S.103-105).

In dem Buch "CP/M USER GUIDE" von Osborne/McGraw-Hill ist JRT-Pascal 3.0 sogar unter den "most popular versions of Pascal for CP/M" verzeichnet. Ein echter Außenseiter ist es also nicht mehr. In der Anlage ist ein Hanoi-Programm mit einigen Laufzeiten für verschieden viele Scheiben. Ich möchte die Clubmitglieder mit anderen Pascal-Dialekten bitten, das Programm **unverändert** einzugeben und mir dann ihre Laufzeiten zu schicken (Jens Ebert). Wenn einiges zusammen kommt, können wir den Vergleich ja mal veröffentlichen.

| Anz. Scheiben | Anz. Bewegungen | Benötigte Zeit (min:sec:..) |
|---------------|-----------------|-----------------------------|
| 2             | 4               | 1:73                        |
| 4             | 16              | 6:29                        |
| 6             | 64              | 24:55                       |
| 8             | 256             | 1:41:90                     |

Sorry, aber diese Seite musste ich entfernen, weil ich für eine Veröffentlichung außer in der ehemaligen, gedruckten Form keine Freigabe habe.

Sorry, aber diese Seite musste ich entfernen, weil ich für eine Veröffentlichung außer in der ehemaligen, gedruckten Form keine Freigabe habe.

Sorry, aber diese Seite musste ich entfernen, weil ich für eine Veröffentlichung außer in der ehemaligen, gedruckten Form keine Freigabe habe.

Sorry, aber diese Seite musste ich entfernen, weil ich für eine Veröffentlichung außer in der ehemaligen, gedruckten Form keine Freigabe habe.

Wer hat sich nicht schon geärgert, daß der DMX80-Drucker zwar acht internationale Zeichensätze hat, aber in der Kursivschrift (italic font) immer nur eckige Klammern usw. druckt. Da man aber das Aussehen der Buchstaben einzeln undefinieren kann, habe ich ÄÖÜäöüß entworfen und auch gleich in normaler Schrift ein schöneres ß mit Unterlänge dazugesetzt. Um den Drucker nicht immer von Hand umschalten zu müssen, lasse ich nach jedem Einschalten automatisch durch STARTUP einen SUBMIT-File ausführen, in dem durch PIP ganz schnell die nötigen Steuerzeichen an den Drucker geschickt werden. Auch die Umstellung auf die deutsche Papierlänge von 72 Zeilen bzw. 12 Zoll und das Überspringen der Papierperforation werden gleich mit erledigt.

### Die Installation

Zuerst tippt man das BASIC-Programm (fehlerfrei) ein und läßt es einmal laufen. Dadurch wird ein File namens LPT.INI erzeugt, der die DMX80-Steuerzeichen enthält. Wenn man schon ein File hat, das bei jedem Einschalten ausgeführt wird, setzt man einfach die Zeile

```
PIP LST:=LPT.INI<RET>
```

hintendran. Sonst erzeugt man sich mit Newword ein Non-Dokument-File START.SUB folgenden Inhalts:

```
CONFIG B:03,C:03<RET>
```

```
INITIATE<RET>
```

```
PIP LST:=LPT.INI<RET>
```

Anschließend tippt man direkt unter CP/M:

```
A>STARTUP SUB START<RET>
```

```
A>STARTUP<RET>
```

Und schon wird beim nächsten Einschalten der Drucker automatisch richtig initialisiert.

Natürlich muß das File SUB.COM auf der Diskette vorhanden sein.

### Ohne FDX, nur mit DMX80

Auch ohne FDX geht es, nur nicht automatisch. Dann muß man die Befehle DISC ERA, OPEN und CLOSE entfernen und aus DISC PRINT #1 stattdessen LPRINT machen. Allerdings muß man selbst das BASIC-Programm laden und es laufen lassen.

### Benutzung der kursiven Umlaute

Von Basic, Pascal usw. aus muß man nur zum Zahlenwert jedes Buchstabens  $128_D=80_H$  addieren, z.B. in BASIC:

```
FOR I=1 TO LEN(A$):LET A$(I,1)=CHR$(ASC(A$(I,1))+128):NEXT
```

Es ist leider **unmöglich**, die Kursiv-Umschaltung (italic font designation) zu benutzen, da nicht die undefinierten Zeichen in kursiv, sondern die in normal angesprochen werden. Wenn der normale Code nicht undefiniert wurde, sieht man leider wieder die eckigen Klammern. Wenn das viele Addieren zu viel Aufwand ist, dem bleibt allerdings noch eine Möglichkeit. Statt jedesmal das achte Bit (MSB) selbst durch die Addition zu setzen, kann dies auch der DMX80 tun durch ESC >. Abgeschaltet wird die dadurch erzeugte Kursivschrift durch die erneute Freigabe des MSB mit ESC #. Diese Methode hat allerdings ihre Tücken, denn auch bei der Übermittlung der anderen SteuerCodes wird das achte Bit gesetzt. Bei einigen stört das nicht, aber die Codes NULL, 01 und bei Grafik die Anzahl der folgenden Bytes werden gemeinerweise nicht mehr verstanden. Daher muß man vor jedem Pica-, Breitschrift-, Unterstrich-, Hoch- und Tiefstell-, Zoll-pro-Zeile-, Grafik- usw. Kommando das MSB erst wieder freigeben. Daher ist auch diese

Methode von Newword aus ohne weitere Hilfen mit Vorsicht zu benutzen im Hinblick auf andere Erweiterungen wie z.B. den echten Unterstrich. Auch ist es dann unmöglich, den Drucker als EPSON zu definieren, da dabei der Randausgleich feiner durch weiße Grafik gemacht wird. Wenn Herbert allerdings endlich die Newword-Enzyklopädie in Händen hält und damit die Möglichkeit besteht, für Newword einen speziellen DMX80-Druckertreiber zu schreiben, kann man dies ja berücksichtigen.

#### **Eigene Experimente mit neuen Zeichensätzen**

Mit Hilfe der font registration (DMX80-Handbuch Seite 39) kann man bis zu 62 Buchstaben umdefinieren, also sogar einen kompletten neuen Zeichensatz erstellen. Das Format ist:

ESC + Y(59<sub>H</sub>, D9<sub>H</sub>=89<sub>D</sub>, 217<sub>D</sub>) + <Code des neuen Zeichens> + D1 + D2 + ... + D9

D1 bis D9 sind die Daten für das neue Zeichen. Es besteht aus einer 9 Spalten mal 8 Zeilen Matrix. Die neunte Nadel, die für die zwei Punkte weiten Unterlängen von g, j, y usw. und auch für die Unterstreichung sorgt, läßt sich so leider nicht ansteuern. Die Bits von D1 erzeugen die erste Spalte links. Das niederwertigste Bit ist, wie bei Grafik, unten und erzeugt einen Punkt weite Unterlängen. Ab dem zweiten Bit beginnen die normalen Buchstaben. Da die Punkte horizontal in Halbnadelabstand gedruckt werden, dürfen nicht in zwei aufeinanderfolgenden Spalten dieselben Bits gesetzt sein. Falls dies doch der Fall sein sollte, werden die zweiten Bits zwangsweise rückgesetzt.

Wenn bei bestimmten Zeichen der DMX80 völligen Unsinn drucken sollte, muß man es mit einem etwas anderen Aussehen des Buchstabens versuchen. Vielleicht ist dies ein Fehler in der Software des DMX80.

Man kann auch eine Neudefinition wieder abschalten mit:

ESC + Z(5A<sub>H</sub>, DA<sub>H</sub>=90<sub>D</sub>, 218<sub>D</sub>) + <Code des neuen Zeichens>

Wer mindestens eine Woche nichts anderes zu tun hat, kann ja mal einen kompletten neuen Zeichensatz erzeugen, der noch viel besser als alles Dagewesene ist, und es uns wissen lassen.

Jan Bredereke

```

100 REM Für den Drucker DMX80:
101 REM Erzeugung eines Files für kursive Umlaute
102 REM Jan Brederke
103 REM
105 DISC ERA "LPT.INI"
110 DISC OPEN #1,"LPT.INI","0"
113 REM Reset des Druckers und Seitenlänge von 72 Zeilen
115 DISC PRINT #1,CHR$(27);"5";CHR$(27);"C";CHR$(72);
116 PRINT "Wieviele Zeilen sollen an der ";
117 INPUT "Perforation übersprungen werden? ";LS
118 DISC PRINT #1,CHR$(27);"N";CHR$(LS);
119 REM in kursiv: äöüß
120 FOR I=251 TO 254: DISC PRINT #1,CHR$(27);"Y";CHR$(I);
130 FOR J=1 TO 9: READ A: DISC PRINT #1,CHR$(A);
140 NEXT : NEXT
145 REM in kursiv: ÄÖÜ
150 FOR I=219 TO 221: DISC PRINT #1,CHR$(27);"Y";CHR$(I);
160 FOR J=1 TO 9: READ A: DISC PRINT #1,CHR$(A);
170 NEXT : NEXT
175 REM in kursiv: §
180 DISC PRINT #1,CHR$(27);"Y";CHR$(192);
190 FOR J=1 TO 9: READ A: DISC PRINT #1,CHR$(A);
200 NEXT
205 REM in normal: schöneres ß
210 DISC PRINT #1,CHR$(27);"Y";CHR$(126);
220 FOR J=1 TO 9: READ A: DISC PRINT #1,CHR$(A);
230 NEXT
900 DISC CLOSE #1
990 REM ä kursiv
1000 DATA 4,10,0,42,128,42,4,42,144
1005 REM ö kursiv
1010 DATA 0,12,18,0,162,0,36,24,128
1015 REM ü kursiv
1020 DATA 12,18,32,130,0,4,10,176,0
1025 REM ß kursiv
1030 DATA 3,12,48,68,2,144,2,144,108
1035 REM Ä kursiv
1040 DATA 6,8,144,40,0,72,6,56,128
1045 REM Ö kursiv
1050 DATA 12,18,128,34,0,34,0,36,152
1055 REM Ü kursiv
1060 DATA 12,50,0,130,0,2,0,12,176
1065 REM § kursiv
1070 DATA 2,24,98,8,162,8,164,16,128
1075 REM ß normal, schöner
1080 DATA 63,64,132,18,128,18,96,12,0

```

Für Interessierte: Das Hexdump von LPT.INI  
DUMP LPT.INI

```

0000 1B 40 1B 43 48 1B 4E 00 1B 59 FB 04 0A 00 2A 80
0010 2A 04 2A 90 1B 59 FC 00 0C 12 00 A2 00 24 18 80
0020 1B 59 FD 0C 12 20 82 00 04 0A B0 00 1B 59 FE 03
0030 0C 30 44 02 90 02 90 6C 1B 59 DB 06 08 90 28 00
0040 48 06 38 80 1B 59 DC 0C 12 80 22 00 22 00 24 98
0050 1B 59 DD 0C 32 00 82 00 02 00 0C B0 1B 59 C0 02
0060 1B 62 08 A2 08 A4 10 80 1B 59 7E 3F 40 84 12 80
0070 12 60 0C 00 1A 1A

```

## Die Zweitfloppy-Einbaustory oder 'wie lernt man fluchen?'

Diese Geschichte ist vor allem den unermüdlichen Single - Drive-Akrobaten zugeordnet, die vorhaben sich ein zweites Laufwerk in Ihre schwarzen Kästen zu bauen.

Bisher hatte ich in meiner FDX nur ein EPSON SD 521, jetzt sind es zwei. Aber fragt nicht was dazwischen lag .....

Zuallererst düste ich nach München um mir bei Fa. MILDE die Floppy zu holen (DM 396,-- incl. Manual !!!--Vobis verlangt DM 898,--??).

Service bei Fa. MILDE: Hervorragend.

Als handwerklich begabter Mensch holte ich mir gleich nachdem ich mein Zuhause erreicht hatte die Werkzeugkiste und machte mich ans Werk.

Alle (!) Stecker aus der FDX und dieses Ding mit Front zum Körper hingestellt. Rechte Seitenplatte ab, mittleren Deckel raus und linke Seitenplatte ab. Rückfront nach lösen der Verbindung zur 80-Zeichen-Karte und zum Netzschalter wegklappen.

So, alte Floppy rechts und links daneben Platz mit Plastikkappe davor. Darf ja nicht schwer sein ...

Hier gleich ein den Hausfrieden sichernden Tip: Macht die ganze Montiererei auf einer alten Tischdecke o.ä. denn bei Memotech muß das Entgraten der Gehäuseteile unbekannt sein ! Jedenfalls sieht meine furnierte Tischplatte aus wie vom Sperrmüll !

Um dieses Info nicht zur Stärke eines mittleren Telefonbuches anwachsen zu lassen beschreibe ich den Rest ohne meine Fehlversuche !

Drückt zuerst das Entlüftungsgitter mit Netzschalter heraus (Von innen mit Schrauber o.ä. die drei oberen Noppen nach unten und dann das Ding raus), und mit Tesa o.ä. nach oben fixieren.

Das große Aluprofil (Laufwerkbock) wird Frontseitig von zwei Schrauben gehalten. Die rechte ist schnell weg aber die linke ... Mit gewinkeltem 5,5er Schlüssel durch die Netzschalterplattenöffnung 'aufbetteln' (mit den Fingern gehts nicht weil selbstsichernde Schrauben). Dann die zwei hinteren Befestigungen des Laufwerkbocks. Eine neben/unter dem Trafo, die Andere gut zugänglich links neben Netzteilplatte mit 5,5er Schlüssel und Kreuzschrauber entfernen.

Doch Halt ! Vorher die Verbindungen zur alten Floppy lösen !!!

Jetzt läßt sich mit viel Geduld und VORSICHT der Laufwerkbock mit der Floppy leicht nach oben/hinten/rechts herausheben. Aber wie gesagt: Vorsichtig damit dem darunterliegenden Netzteil nichts passiert (Keine Gewalt, es geht auch sanft !!!).

Als ich in diesem Stadium (schon völlig entnervt) die vorgebohrten Löcher des Laufwerkbocks mit denen der EPSON-Floppy verglichen habe, hat sich ein Fluchschwall über Memotech ergossen.

Der Grund:

Die Lage der 4 EPSON-Gewindelöcher ist nicht kompatibel mit den Bohrungen im Laufwerkbock. Nachdem ich die Schrauben des alten Laufwerks entfernt hatte, wußte ich warum ...

Mit windigen Schräubleinchen wurden Adapterbleche (mit Gewinde) an die Floppy montiert (Sieht aus als wenn da mal andere Floppys vorgesehen gewesen wären ...). Was jetzt ???

Nachdem die Floppy trotz nicht festgezogener Schrauben bisher unerschütterlich ihre Dienste tat habe ich kurzerhand die vier Bleche auf beide Laufwerke aufgeteilt. Je zwei Bleche halten diagonal montiert (mit neuen Schrauben) je eines der beiden Floppys. Die jeweils freien Floppygewinde habe ich mit einer Schraube versehen um eine gute Auflage für die Laufwerke auf dem Bock zu erzielen.

Bei den umseitig beschriebenen 'Verschraubungen' solltet Ihr unbedingt neue Schrauben mit passendem Gewinde (M3) verwenden und diese nicht zu fest anziehen (Die Originalen waren zu schwach , eine davon hatte das Gewinde ausgerissen !!).

Oh MEMOTECH , Oh MEMOTECH ...

So,nun sind also beide Floppys auf dem Bock montiert.Da bei mir das B-Laufwerk rechts war,habe ich es (zwecks der Gewohnheit) wieder rechts montiert , die C-Floppy links. Damit die FDX später weiß wo es langgeht muß auf der B-Floppy das Widerstandsarray entfernt werden (Das ist das IC-ähnliche Gebilde auf Sockel,weiß, mit Beschriftung 'BECKMAN..R150'). Bei der C-Floppy muß das Array drinbleiben !! Dann muß noch initialisiert werden welches das Erste und welches das Zweite Laufwerk ist. Dazu dient auf jeder Floppyplatine das 'Wahlsteckerchen' direkt neben dem Vielpoligen Anschluß. Hier sieht man die Stiftreihe mit der Beschriftung : DS 0 1 2 3 . Vom Werk werden die Floppys mit Steckerchen auf '0' verschickt. Dies kann man bei Floppy 1 lassen . Die C-Floppy muß das Steckerchen auf '1' verpasst kriegen. (Weitere wären demnach: 3.Floppy - '2' und 4.Floppy - '3').

Wer jetzt den schwarzen Kasten nicht mehr sehen kann,der sollte jetzt den LötKolben anheizen und sich zur Entspannung das Stromversorgungskabel für die neue Floppy montieren (falls kein fertiges vorhanden). Fa.MILDE hatte übrigens auch den passenden Stecker auf Lager (was leider bei Anderen nicht üblich ist!). Zur Kabelmontage nehme man einfach das alte Kabel als Muster.

Nun muß der Bock mit den beiden Floppys ins Gehäuse. Nicht ungeduldig mit Gewalt sondern rechts ansetzen (der hintere Teil leicht angehoben) und dann vorsichtig übers Netzteil nach vorn einschieben. Jetzt von unten die beiden hinteren Schrauben durch die Bodenplatte stecken (dabei wackelt der ganze Kasten beträchtlich - deswegen: 'Gefühl') und die Muttern aufgeschraubt. Dann die vorderen Schrauben (die sitzen hoffentlich immer noch in der Frontwand ..) in die jeweiligen Löcher ein-fühlen und Muttern aufsetzen (rechts gehts, aber links .....,da mußte meine bessere Hälfte mit den schlanken Fingerchen helfen !) und anziehen.

Der am Flachbandkabel sitzende breite Stecker wird nun auf die linke Floppy und der Ursprüngliche aufs rechte aufgesteckt.Stromversorgungskabel am Netzteil anschließen und das Gehäuse wieder zusammenmontieren. Netzschalterplatte wieder einsetzen (darauf achten,daß Anschlüsse noch isoliert sind !).

(Vorher kann mit allen üblichen Vorsichtsmaßnahmen evtl. ein Test gefahren werden ) .

Hört sich einfach an,gell ? Wenn ich all das was ich jetzt dies-bezüglich weiß,gestern gewußt hätte ... ja dann ...

Für Ungeduldige und Choleriker ist das nichts !!!

Vielleicht gibt es eine bessere Lösung um das teure Stück in den schwarzen Kasten zu praktizieren,aber dann soll sich der mal bitte bei mir melden !

Und nochwas: Bezüglich der Richtigkeit des obigen Senfes kann ich natürlich keine Garantie geben (falls da etwas anders ist oder etwas zu Bruch gehen sollte !!!!!).

Ich hoffe dem Einen oder Anderen geholfen zu haben (evtl.die Finger davon zu lassen) .

Zum Thema Frage und Antwort : März 85 Frage 4 : Störimpulse

Grundsätzliches Problem ist folgendes:

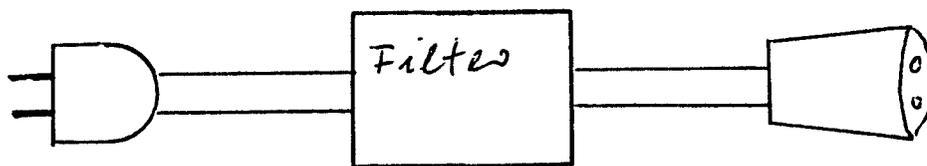
- a) Der Computer stört durch Hf-Ausstrahlung
- b) Der Computer wird durch Hf-Impulse gestört
- c) Der Computer wird durch Induktionsimpulsspitzen gestört , die über 220 V liegen.

Alle drei Arten kommen bei vielen Computertypen immer wieder vor und können User, die an einem sehr unruhigen Netz hängen , durch Programmabstürze manchmal zur Verzweiflung bringen. So ging es mir kurz nach dem Kauf meines MTX 500 +Floppy+Epson.

Zu a+b) Es gibt Breitbandentstörfilter als Netzzwischenstecker, die im Bereich von 100-300 MHz bei störenden wie gestörten Geräten schützen , allerdings nur gegen Hf-Einstrahlungen. Bezug: z.B. Völkner Elektronik-Postfach 5320-3300 Braunschweig : Breitbandnetzfilter Typ EM 504 Preis: 36,-DM.

Zu c) Empfehlenswert ist es, zusätzlich zwischen Phase und Nulleiter zwei entgegenges. Netzdioden vom Typ 1N4003 zu schalten, um Überspannungsimpulse über 300 V Spitze, wie sie beim Abschalten von Induktivitäten (Motoren, Spulen, Relais bei Waschmaschinen, Spülmaschinen, Kühlschränken Heizungen, die auf gleicher Phase wie der Computer liegen) entstehen können, vom Computer fernzuhalten.

Da der o.g. Filter recht einfach aufgebaut, dafür aber doch recht teuer ist, kann man sich ihn auch preiswert selber herstellen:  
VDE BEACHTEN. Zum Einbau empfiehlt sich ein Kunststoffgehäuse ( es geht auch mit einer Aufputzverteilerdose .

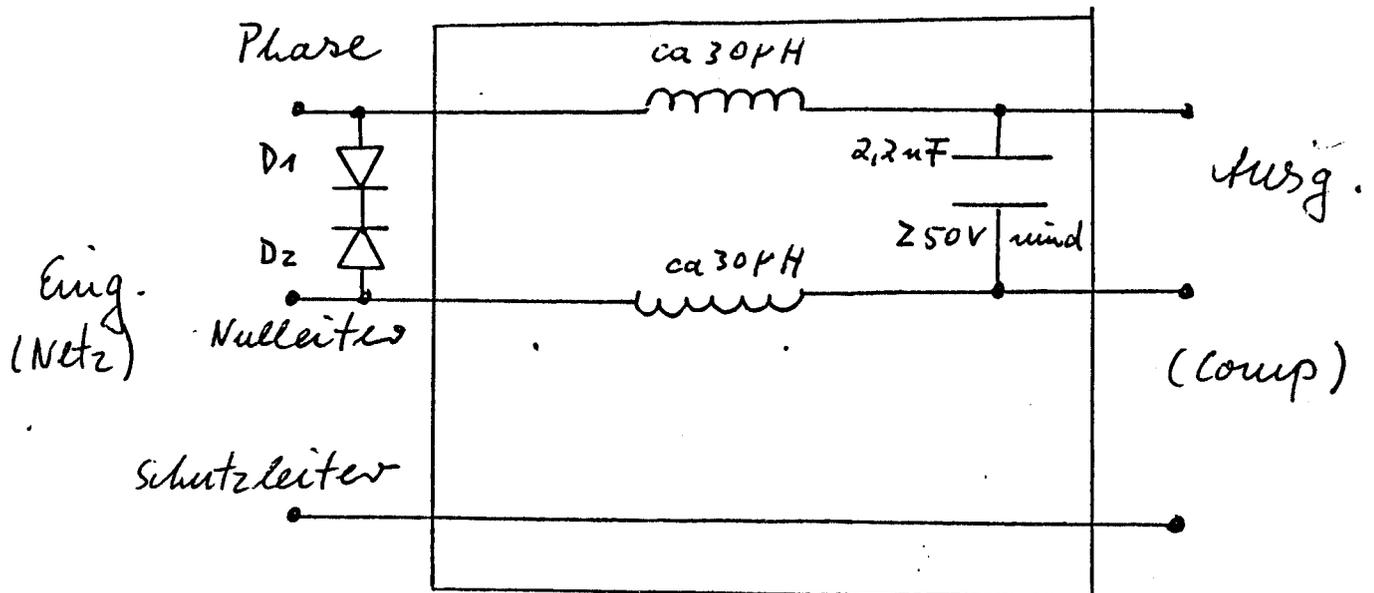


Netzstecker

Netzkupplung

P.S.. Um einen eventuellen Kurzschluß bei (angeblich) möglichem Durchschlagen einer Diode zu vermeiden, habe ich ~~w~~ zwischen beide Dioden eine Glössicherung von 0.5 A flink geschaltet. Statt der beiden Dioden kann auch ein Varistor eingeschaltet werden (mind. 300 V Spannungsfest)

# Selbstgebaute Hardware



Hier das Schaltbild des kompletten Filters:

Eingerahmt der gekaufte Breitbandfilter (auch selbst herzustellen)  
Die beiden Spulen blocken Hf-Störungen ab, der Kondensator schließt übriggebliebene Hf quasi kurz.

Die beiden Dioden lassen Spannungsspitzen über 300 V (daher unbedingt 1 N 4003 benutzen) durch und schließen sie daher kurz.

Der Kondensator sollte keramisch sein und mindestens 250 V Wechselspannung vertragen. Als Spulen eignen sich handelsübliche Thyristor Entstördrosseln mit 2 Ampere Belastbarkeit.

Nach dem Zwischenschalten dieses Filters habe ich seit 2 Monaten keine Programmabstürze mehr gehabt, selbst wenn ich (zum Test) auf derselben Steckdose meine Schlagbohrmaschine ein und ausschalte.

Mit frdl. Grüßen

*Jost Beckendahl*

Sorry, aber diese Seite musste ich entfernen, weil ich für eine Veröffentlichung außer in der ehemaligen, gedruckten Form keine Freigabe habe.

Sorry, aber diese Seite musste ich entfernen, weil ich für eine Veröffentlichung außer in der ehemaligen, gedruckten Form keine Freigabe habe.

Sorry, aber diese Seite musste ich entfernen, weil ich für eine Veröffentlichung außer in der ehemaligen, gedruckten Form keine Freigabe habe.

Sorry, aber diese Seite musste ich entfernen, weil ich für eine Veröffentlichung außer in der ehemaligen, gedruckten Form keine Freigabe habe.

Sorry, aber diese Seite musste ich entfernen, weil ich für eine Veröffentlichung außer in der ehemaligen, gedruckten Form keine Freigabe habe.

Sorry, aber diese Seite musste ich entfernen, weil ich für eine Veröffentlichung außer in der ehemaligen, gedruckten Form keine Freigabe habe.

Eine Übersicht über die STAT-Funktionen  
Übertragen von Jörg-Uwe Oppermann

Der Befehl STAT kommt wie so vieles in der EDV aus dem Englischen Sprachgebrauch und heißt nichts anderes als "Statistics" zu gut deutsch Statistic oder noch besser Status der im moment benutzten Disc und der auf ihr gespeicherten Dateien.

Übersicht der vorhandenen STAT-Befehle

- STAT USR:**  
zeigt die auf der Disc vorhandenen Benutzerbereiche und die aktiven Benutzernummern an.
- STAT DEV:**  
liefert eine Tabelle über die vorhandenen Ein-Ausgabekanäle und deren Zuordnung zu den Geräten.
- STAT <Kanal>:=<Gerät>:**  
weist dem <Kanal> CON:, RDR:, PUN:, oder LST: eines der vordefinierten <Geräte> zu.
- STAT <Laufwerk>:=R/O**  
schützt das angegebene <Laufwerk> gegen Schreibzugriffe.
- STAT**  
liefert Angaben über den Schreibschutz der Laufwerke und über den noch verfügbaren Speicherplatz.
- STAT <Laufwerk>:**  
liefert eine Angabe über den freien Speicherplatz auf der im <Laufwerk> angegebenen Diskette.
- STAT <Dateiname>**  
liefert einen Überblick über die angegebene Datei, über der von ihr belegten Speicherplatz und der Dateimerkmale sowie der auf der Disk noch vorhandenen freien Speicherplatz.
- STAT <Dateiname> \$<Merkmal>**  
setzt bzw. löscht die vordefinierten Dateimerkmale.
- STAT DSK:**
- STAT <Laufwerk>:DSK:**  
bringt einen tabellarischen Überblick über die Eigenschaften der angegebenen Diskette.
- STAT VAL:**  
liefert eine Tabelle der STAT-Befehle und Befehlsargumente.

Benutzerbereich ermitteln

**STAT USR:**  
Meldet die gegenwärtig aktive Benutzerbereichsnummer und die Benutzernummern der im Bezugslaufwerk verzeichneten Dateien:

**Active User :** <Benutzernummer>

**Active Files:** <Liste der Benutzerbereiche mit Dateien>

**Beispiel:**

Der Benutzer mit der Nummer 7 arbeitet mit einer Disk, auf der Dateien in den Benutzerbereichen 1, 7, 13 und 25 angelegt sind.

**A>STAT USR:**

**Active User :** 7

**Active Files:** 1 71325

**Besonderheiten:**

- Bei der Ermittlung der vorhandenen Benutzerbereiche werden nur die Dateien auf der Disk im Bezugslaufwerk untersucht.
- Es werden alle vorhandenen Dateien untersucht.
- Es wird auf das BDOS-Programm zugegriffen, deshalb werden alle 32 Benutzerbereiche erfaßt.

**Fehler:**

--> Eine Laufwerksangabe steht vor USB:  
(z.B. "STAT C:USB:")

**Meldung:**

**File Not Found**

**Abhilfe:**

Geben Sie vor USB: nie ein Laufwerk an! STAT sieht in dem Falle USB: als Filenamen an. Wähle lieber das Bezugslaufwerk neu und rufe dann wieder STAT auf.

**Gerätezuordnung feststellen****STAT DEV:**

Untersucht die im IOBYTE verschlüsselte Gerätezuordnung für die Ein-Ausgabekanäle und gibt die Information folgendermaßen an:

CON: ist <Konsolengerät>  
RDR: ist <Lesergerät>  
PUN: ist <Stanzengerät>  
LST: ist <Druckergerät>

**Beispiele:****A>STAT DEV:**

|              |                            |
|--------------|----------------------------|
| CON: is CRT: | (Konsole: Bildschirmgerät) |
| RDR: is TTY: | (Leser: Fernschreiber)     |
| PUN: is TTY: | (Stanzer: Fernschreiber)   |
| LST: is LPT: | (Drucker: Zeilendrucker)   |

**Besonderheiten:**

- STAT DEV: prüft lediglich die im IOBYTE gespeicherten Werte. Die Auswertung erfolgt in der BIOS-Routine.

**Gerätezuordnung setzen**

STAT <Kanal>=<Gerät>

STAT <Kanal 1>=<Gerät 1>,<Kanal 2>=<Gerät 2>,...

**Wirkung:**

Weist den Kanälen die angegebenen Geräte zu, indem im IOBYTE der entsprechende Schlüssel eingetragen wird.

**Besonderheiten:**

- Der einzige STAT-Befehl in dem Mehrfachargumente übergeben werden können. Das heißt man kann alle vier Zuordnungen bestimmen z.B.:

STAT CON:=CRT:,RDR:=UR1,PUN:=UP1,LPT:=TTY:

**Fehler:**

--> <Kanal> oder <Gerät> sind falsch angegeben.

**Meldung:**

**Invalid Assignment**

**Abhilfe:**

Befehl in richtiger Form wiederholen.

--> Fehlerhafte Zuweisung an die "CON:"-Einheit

**Meldung:**

**Der Rechner reagiert nicht mehr.**

Dies tritt meist dann auf, wenn man der Konsole einen Drucker zuweist (manchmal ist es bequem, längere Bedienungssequenzen so über den Drucker zu protokollieren). Hier wurde entweder die Ausgabe blockiert (der Drucker ist nicht eingeschaltet) oder das System erhält keine Eingabe mehr (weil jetzt z.B. der Fernschreiber statt der Konsoleinheit abgefragt wird).

**Abhilfe:**

Vergewissern Sie sich, daß die Zuweisung stimmt und daß das Gerät auch wirklich als Konsoleneinheit arbeiten kann. Ist das nicht der Fall hilft nur noch der Kaltstart des Systems.

**Schreibschutz für ein Laufwerk einstellen**

STAT <LAUFWERK>:=R/O

**Wirkung:**

Setzt einen BIOS-internen Vermerk, nach dem die Disk nicht mehr verändert werden kann.

**Fehler:**

- > Laufwerksangabe falsch
- > Laufwerk noch nicht aktiviert und betriebsbereit

**Meldung:**

**Bdos Error On <Laufwerk>: Select**

**Abhilfe:**

Drücken Sie irgend eine Taste, Laufwerk betriebsbereit machen und wiederholen Sie dann den Schreibschutzbefehl.

**Fehler:**

- > Das Gleichheitszeichen fehlt oder die R/O-Anweisung ist falsch.

**Meldung:**

**"Invalid Disk Assignment"**  
("Zuweisung an die Diskette fehlerhaft")

**Abhilfe:**

Befehl in der richtigen Form wiederholen.

**Ermitteln des gegenwärtigen Diskettenzustands**

STAT

**Wirkung:**

Gibt den Zustand aller im System derzeit aktivierten Laufwerke in einer Liste wie folgt an:

**<Laufwerk>: <Schreibschutz>, Space: <n>k**

Laufwerk =A - P  
Schreibschutz= R/O oder R/W      Read Only bzw. Read Writ  
n= eine Zahl    in k Kilobytes.

**Beispiel:**

```
A>STAT
A: R/O, Space: 36k           (schreibgeschützt, 36 KBytes frei)
C: R/W, Space: 147k        (nicht geschützt, 147 KBytes frei)
```

**Ermitteln des freien Speicherplatzes auf einer Disk**

**STAT <Laufwerk>**

**Wirkung:**

Gibt den auf der Disk im angegebenen Laufwerk derzeit verfügbaren Speicherplatz in der Form:

**Bytes Remaining On <Laufwerk>: <n>k**

an, wobei <n> die freie Speichergröße in Kilobytes ist.

**Ermitteln des Zustands von Dateien**

```
STAT <Einfachname>
STAT <Mehrfachname>
```

**Wirkung:**

Gibt die Speicherbelegung den Schreibschutz und den Systemzustand der angegebenen Dateien in einer Liste wie folgt wieder:

```
Recs  Bytes  Ext  Acc
  r1   b1k   e1  z1 <Laufwerk>:<Dateiname>
  r2   b2k   e2  z2 <Laufwerk>:(<Dateiname>)
Bytes Remaining On <Laufwerk>: <n>k
```

- r1, r2 = Anzahl der tatsächlich belegten Aufzeichnungseinheiten zu je 128 Bytes
- b1, b2 = Größe des auf der Disk durch Aufzeichnungsblöcke reservierten Bereichs in Kilobytes
- e1, e2 = Anzahl der (logischen) Verzeichniseinträge für je 16 KBytes Dateiumfang
- z1, z2 = Stand des Schreibschutzes für die Dateien. R/O oder R/W
- (...) = der <Dateiname> bezeichnet eine Systemdatei
- <n> = Anzahl des noch verfügbaren Speicherplatzes in KBytes

**Beispiel:**

**A>STAT B:\*.\***

```
Recs  Bytes  Ext  Acc
  213   27k   2  R/W B:WERTE.DAT
   52    7k   1  R/O B:ED.COM
   58    8k   1  R/O B:(PIP.COM)
Bytes Remaining On B: 74k
```

Die Disk enthält (im gerade aktiven Benutzerbereich) drei Dateien, von denen zwei (WERTE.DAT und ED.COM) im DIR-Verzeichnis aufgelistet werden, während die dritte (PIP.COM) das Systemmerkmal trägt und so im DIR-Verzeichnis nicht auftaucht. Die beiden COM-Befehlsdateien sind schreibgeschützt (R/O), WERTE.DAT kann verändert werden..

Die Datei in der ersten zeile belegt tatsächlich 213x128=27264 Bytes, für die 27x1024=27648 Bytes reserviert sind. Sie muß in zwei logischen Verzeichniseinträgen erfaßt werden, für die beiden anderen reicht je ein Eintrag aus.

Insgesamt sind für diesen Benutzerbereich derzeit 27+7+8=42 KBytes auf der Disk reserviert.

## Besonderheiten

59

Man kann noch weitere Angaben anfordern, wenn man den Befehl durch "\$S" (von "size", Größe) genauer bestimmt. Man erhält dann eine weitere Spalte aufgelistet z.B. so:

A>STAT B:\*. \* \$S

| Size | Recs | Bytes | Ext | Acc             |
|------|------|-------|-----|-----------------|
| 2048 | 213  | 27k   | 2   | R/W B:WERTE.DAT |
| 52   | 52   | 7k    | 1   | R/O B:ED.COM    |
| 58   | 58   | 8K    | 1   | R/O B:(PIP.COM) |

Bytes Remaining On B: 74k

Wurde die Datei sequentiell angelegt, dann fällt "Size" mit der "Recs"-Angabe zusammen. Andernfalls wird hier die "virtuelle" Größe einer unmittelbar angelegten Datei angegeben.

### Dateimerkmale bestimmen

STAT <Einfachname> \$<Merkmal>

STAT <Mehrfachname> \$<Merkmal>

Dabei ist <Merkmal> eine der Angaben:

- R/O - (read only) Schreibschutz setzen
- R/W - (read/write) Schreibschutz aufheben
- SYS - (system) Systemeigenschaft setzen
- DIR - (directory) Systemeigenschaft aufheben

### Wirkung

Setzt bzw. löscht in den betreffenden Dateinamen im Verzeichnis die Merkmalsflaggen wie folgt:

- R/O - setzt Bit 7 im ersten Zeichen der Klassenbezeichnung
- R/W - löscht Bit 7 im ersten Zeichen der Klassenbezeichnung
- SYS - setzt Bit 7 im zweiten Zeichen der Klassenbezeichnung
- DIR - löscht Bit 7 im zweiten Zeichen der Klassenbezeichnung

(Bit 7 ist das höchstwertige, achte Bit im Zeichen; Setzen eines Bits bedeutet, ihm den Wert 1 zu geben; löschen gibt ihm den Wert 0.)

## Besonderheiten

- Nicht alle Programme nutzen diese Merkmale aus. Um mit Programmen arbeiten zu können die vor der Version 2.0 erstellt worden sind, ist es manchmal notwendig, alle Merkmale für die betreffende Datei aufzuheben.
- STAT gestattet nur den Zugriff auf die angegebenen Merkmalsflaggen. Um andere Flaggen zu beeinflussen, insbesondere die für Anwendungsprogramme freigegebenen achten Bits in den ersten vier Zeichen des Dateinamens, muß man unmittelbar die im BDOS-System gegebenen Möglichkeiten benutzen.

### Fehler:

--> Das Leerzeichen zwischen Dateinamen und Dollarzeichen fehlt

### Meldung:

Auslisten der betreffenden Dateieigenschaften

**Abhilfe:**

Befehl in richtiger Form wiederholen

- > Ein Lehrzeichen zwischen Dollarzeichen und Merkmalsangabe
- > Ungültige Merkmalsangabe

**Meldung:**

**Invalid File Indicator**

**Abhilfe:**

Befehl in richtiger Form wiederholen

**Diskettenmerkmale ermitteln**

**STAT DSK:**

**STAT <Laufwerk>:DSK:**

**Wirkung:**

Untersucht die in einer Tabelle im BIOS-Teil angegebenen Disk-eigenschaften und listet sie in folgender Form aus:

```

X: Drive Characteristics
nnnnn: 128 Byte Record Capacity
nnnn: Kilobyte Drive Capacity
nnn: 32 Byte Directory Entries
nnn: Checked Directory Entries
nnnn: Records/ Extent
nn: Sectors/ Block
n: Reserved Tracks
  
```

Die erste Form des Befehls gibt eine Liste für alle derzeit aktiven Laufwerke, die zweite erfaßt nur das angegebene <Laufwerk>. Die Zeilen bedeuten von oben nach unten:

- Eigenschaft für Laufwerk X:
- kann nnnnn Aufzeichnungseinheiten zu je 128 Bytes aufnehmen,
- kann nnnn Kilobytes speichern
- verfügt über nnn Verzeichniseinträge zu je 32 Bit
- davon werden nnn Einträge (für den R/O-Test) untersucht
- ein Verzeichniseintrag beschreibt nnnn Aufzeichnungseinheiten
- ein Aufzeichnungsblock faßt nnn Aufzeichnungseinheiten
- eine Diskettenspur faßt nn Sektoren
- n Spuren sind für das Betriebssystem reserviert

**Beispiel:**

Ein System mit einer Standarddiskette in Laufwerk A, einer Diskette mit doppelter Genauigkeit in Laufwerk B und einem Festplattenlaufwerk als Nummer C kann z.b.folgende Werte liefern:

**A>STAT DSK:**

```

A: Drive Characteristics
1944: 128 Byte Record Capacity
243: Kilobyte Drive Capacity
64: 32 Byte Drive Capacity
64: Checked Directory Entries
128: Records/ Extent
8: Records/ Block
26: Sectors/ Track
2: Reserved Tracks
  
```

```

      B: Drive Characteristics
4800: 128 Byte Record Capacity
      600: Kilobyte Drive Capacity
      128: 32 Byte Drive Capacity
      128: Checked Directory Entries
      128: Records/ Extent
      16: Records/ Block
      64: Sectors/ Track
      2: Reserved Tracks

```

```

      C: Drive Characteristics
65536: 128 Byte Record Capacity
      8192: Kilobyte Drive Capacity
      128: 32 Byte Drive Capacity
      0: Checked Directory Entries
      1024: Records/ Extent
      128: Records/ Block
      58: Sectors/ Track
      0: Reserved Tracks

```

### Besonderheiten:

- Diese Möglichkeit gibt es erst ab Version 2.0 ab der die Disk-eigenschaften nicht mehr vom BDOS-Teil sondern vom BIOS-Teil bestimmt werden.
- Da die Disketteninformationen im BIOS bereitgestellt werden, hängt es von der Auslegung ab, ob die Angaben immer gleich bleiben oder ob sie von Disk zu Disk wechseln. Ausgebaute Systeme erkennen automatisch, um was für eine Diskette es sich handelt und erneuern die Diskettenbeschreibung entsprechend. (Die richtige Beschreibung ist wichtig, da sie zentrale Informationen für die Arbeit des BDOS-Teils enthält!)

### Übersicht über die STAT-Befehle

#### STAT VAL:

#### Wirkung:

Gibt eine Liste der im STAT möglichen Befehle in folgender Form aus:

```

Temp R/O Disk: d:=R/O
Set Indicator: d:filename:typ $R/O $R/W $SYS $DIR
Disk Status  : DSK: d:DSK:
User Status   : USR:
Iobyte Assign:
CON: = TTY: CRT: BAT: UC1:
RDR: = TTY: PTR: UR1: UR2:
PUN: = TTY: PTP: UP1: UP2:
LST: = TTY: CRT: LPT: UL1:

```

Erläuterungen zu den System-Disk-Utilities

- ASM.COM = Assembler für 8080-Maschinencode. Ergibt einen druckreifen .PRN File und einen .HEX File zur Weiterverarbeitung mit LOAD.COM.  
Format: ASM Filename (Das nichtzugebende Extend muß .ASM sein)
- BATCH.COM = Ähnlich SUBMIT.COM, man gibt aber die Befehle direkt ein, die nach einem SEND sofort ausgeführt werden.  
Format: BATCH
- BAUD.COM = Stellt den vorgegebenen Kanal der RS232 auf eine BAUDrate (9600,4800,2400,120,600,300 oder 110) ein.  
Format: BAUD Kanal,Rate (Kanal ist A oder B)
- BIOS.ASM = Der Quellcode von Digital Research des Basic Input/Output System. Dies kann nun angeschaut, verändert, auf die Systemspuren gebracht werden, oder sonst was.
- CBIOS.ASM = Das vollständige BIOS mit den Modifikationen für den MTX.
- CBIOS.HEX = Der Hexddump des CBIOS.ASM. Man kann ihn selber mit dem Linker LOAD.COM herstellen.
- CONFIG.COM = Das CONFIG für OVERLAYB-Disketten. Wird wie CONFIG benutzt.
- COLDBOOT.COM = Führt einen Kaltstart auf dem angegebenen Laufwerk aus und erhebt dieses zum Laufwerk A:, für weitere Warm- und Kaltstarts.  
Format: COLDBOOT d: (Wenn z.Z.angesprochenes Laufwerk gemeint ist : d:->\$)
- CONFIG.COM = CONFIGuriert angewählte Laufwerke nochmal bzw.neu. Die CONFIG-Codes sind im Technical-Manual aufgeführt.  
Format: CONFIG b:03,c:07
- CONTACT.COM = Programm zum Austausch bzw. zur Versendung von Daten über die beiden RS232 Kanäle.  
Format: CONTACT d:Filename (Wenn kein Filename angegeben, wird LAST.LNK gesucht. Das File muß das Extend .LNK haben)
- CONTVI.COM = Ich sehe keinen Unterschied zu CONTACT.COM, außer außer einer geringeren Records Angabe.
- COPY.COM = Kopierprogramm für Einzellaufwerke, mit der Möglichkeit von einem nur CONFIGuriertem Laufwerk (kein echtes - nur ein gedachtes !!!) auf das richtige zu kopieren. Sehr langsam, da es nur immer 16K auf einmal kopiert, dann muß die Diskette gewechselt werden. Auch keine Möglichkeit die ganze Diskette zu kopieren, nur Fileweise.  
Format: COPY d:neuename.ext d:altername.ext

- CPMGEN.SUB = SUBMIT-File zur automatischen Veränderung der CP/M Systemgröße. Die größte Größe ist 59K (nach Angaben von MEMOTECH), die auch problemlos benutzt werden kann. Ein Warmstart sollte dann allerdings von dieser Diskette vermieden werden, obwohl es bei mir meistens klappt.  
Format: SUB CPMGEN nn d: (nn ist die neue Größe folgende Files werden noch benötigt:  
CBIOS.ASM  
WRTBIOS.COM  
WRTCPM.COM  
MVCPM.COM  
ENTER.COM  
ASM.COM  
SUB.COM  
ED.COM)
- DDT = Dynamic Debugging Tool. Ein Werkzeug zur Hilfe von Fehlersuche und deren Beseitigung (Fehler=Buggs=Wanzen). Es arbeitet mit dem 8080-Code, wie ASM.COM.  
Format: DDT Filename.ext (man beendet es mit BRK und kann sein Programm dann mit dem eingebauten Befehl SAVE abspeichern auch bei VDEB.COM)
- DUMP.ASM = Quellcode des DUMP.COM zum besseren Verständnis von CP/M.
- DUMP.COM = Erzeugt einen Auszug eines Files in Hexadezimal.  
Format: DUMP Filename.ext
- ED.COM = ZeilenEDitor. Arbeitet mit einem CP (Charakter Pointer) der immer den Wert des derzeitigen Cursor hat. Umständlich zu benutzen, da man nicht mit dem richtigen Cursor arbeiten kann.  
Format: ED filename.ext (wieder raus mit Q(uit))
- ENTER.COM = Wird in BATCH oder SUBMIT Files benutzt. Simuliert eine Eingabe von der Tastatur.  
Format: ENTER msg1ömsg2ö..msgn (das ö ist der dt.Backslash)
- ERAQ.COM = Löscht Files, fragt aber vorher immer erst mal nach.  
Format: ERAQ Filename.ext (auch "\*" oder "?")
- EXAMPLE.KBD = Ein Beispiel für die Benutzung von F.COM.
- F.COM = Der Funktiostastendecoder. Mit ihm kann man die Funktionstasten bequem einstellen. Man muß ein .KBD File anlegen, wie EXAMPLE.KBD, indem die Funktionstasten wie gewünscht angegeben sind.  
Format: F Filename (ohne Extend!)
- FDXB.COM = Der BASIC-Interpreter als Diskettenversion. Wenn ein Filename mitangegeben, wird der gleichmitgeladen, wie mit DISK LOAD"Filename".  
Format: FDXB Filename

- FORMAT.COM** = FORMATierprogramm für Doppellaufwerke.  
 Format: FORMAT d:
- FRM.COM** = FoRMatierprogramm für Einzellaufwerke.  
 Format: FRM
- H.COM** = Auch ein Hexauszug des angegebenen Programms, allerdings werden alle Hexzahlen ohne Leerzeichen dazwischen ausgegeben, damit ohne gute optische Aufbereitung, dafür ist es aber schneller als DUMP.COM.  
 Format H Filename.ext (Control S=Pause/  
 Control Q=Weiter)
- I.COM** = ? Mit "." wird der Ablauf angehalten.  
 Format: I Filename.ext
- INITIATE.COM** = Stellt die BAUD-Rate auf 9600 Baud ein, erhöht die Tastatur Repeat Rate und testet das Laufwerk.  
 Format: INITIATE
- LAST.LNK** = File in dem die eingestellten Daten für CONTACT.COM abgespeichert sind. Jedes andere File für denselben Zweck muß auch das Extend .LNK haben.
- LOAD.COM** = LOADER für ein mit Intel-Hexsumme-Test erstelltem Hexfile, der den Hexfile in einen .COM File verwandelt. Ein solcher Hexfile wird durch ASM.COM erstellt. Jedes zu wandelnde File muß das Extend .HEX haben, wird aber nicht angegeben.  
 Format: LOAD Filename
- MOVCPM.COM** = Erstellt ein nnK Byte großes CP/M und legt das dann bei 0980H im Speicher ab. Benutzt in CPMGEN.SUB.  
 Format: MOVCPM nn \* (Vorsicht beim probieren!!)
- MTX.COM** = Schaltet wieder auf normalen MTX Betrieb um, als wenn keine Floppy angeschlossen wäre. Durch Reset kommt man wieder zurück.  
 Format: MTX
- NCPM.COM** = Wird für das Node-Ring System benötigt, aber auch für den DISK-QUIT im FDXB.COM.  
 Format: NCPM (im FDXB muß es nur da sein).
- OVERLAY8.COM** = Das OVERLAY für das Televideo Diskettenformat. Es legt sich über den normalen Diskettenhandler im CP/M. Es muß vor CFIG8.COM gestartet werden.  
 Format: OVERLAY8
- PIP.COM** = Vielseitiges Kopierprogramm für 2 Laufwerke, kann aber teilweise auf nur einem Laufwerk verwendet werden, z.B. um etwas auszudrucken.  
 Format: s.Handbuch (PIP <ret> lädt PIP.COM in den Speicher.)
- R.COM** = Supermini Editor. Was man auf tippt wird in das angegebene File geschrieben.  
 Format: R Filename.ext (Control Z = Zurück)
- RCHECK.COM** = List jeden Sektor auf Diskette und überprüft den. Falls ein Fehler auftritt, kommt eine Zahl, die den Fehler spezifiziert. Eine Erklärung für die Zahlen selber habe ich aber leider nicht.

- SIDISC.COM = Richtet die angeschlossenen Silicon Discs ein, diese müssen danach CONFIGuriert und FORMATiert werden.  
Format: SIDISC oder SIDISC CONFIG F:40
- SISPOOL.COM = Drucker Puffer, der die Silicon Discs als Puffer benutzt.  
Format: SISPOOL S(top) Stopt und löscht Puffer  
P(ause) Pausiert  
R(esume) Weiter nach P(ause)
- STARTUP.COM = Bewirkt, daß bei jedem Kalt- oder Warmstart die angegebenen Programme ausgeführt werden.  
Format: STARTUP CONFIGöINITIATE
- STAT.COM = Gibt STATUS jedes Files, Laufwerkes, Device an oder ändert ihn, z.B.File auf Read/Only setzen.  
Format: s.Handbuch
- SUB.COM = Erweitertes SUBMIT.COM um Annahme von Control Zeichen und Aktionen von einem anderen Laufwerk.  
Format: SUB filename (s.SUBMIT.COM)
- SUBMIT.COM = Automatisiert Abläufe unter CP/M.  
Der Ablauf muß in einem .SUB File untergebracht sein.  
Format: SUBMIT Filename (Extend.SUB nicht angeben)
- SYS.COM = Kopiert die Systemspuren der Diskette auf einem Laufwerk.  
Format: SYS
- SYSCOPY.COM = SYS.COM für zwei Laufwerke.  
Format: SYSCOPY d:
- T.COM = Ein sehr schnelles TYPE.  
Format: T Filename.ext (Steuerung wie H.COM)
- VDEB.COM = Die Floppy Version von FANEL, aber mit mehr Möglichkeiten (s.VDEB.DOC).  
Format: VDEB Filename.ext (Filename muß nicht angegeben werden)
- WRTBIOS.COM = Schreibt den BIOS und COLD-START LOADER Teil von CBIOS.HEX auf die Systemspuren.  
Format: WRTBIOS d:
- WRTCPM.COM = Schreibt das BDOS und CCP auf die Systemspuren, was vorher mit MOVCPM in den Speicher gebracht worden sein muß.
- XSUB.COM = Erweitertes SUBMIT.COM um Annahme von Befehlen aus anderen Programmen, in die man mit XSUB.COM gesprungen ist.  
Format: XSUB Filename (Extend .SUB nicht angeben!)

Sorry, aber diese Seite musste ich entfernen, weil ich für eine Veröffentlichung außer in der ehemaligen, gedruckten Form keine Freigabe habe.