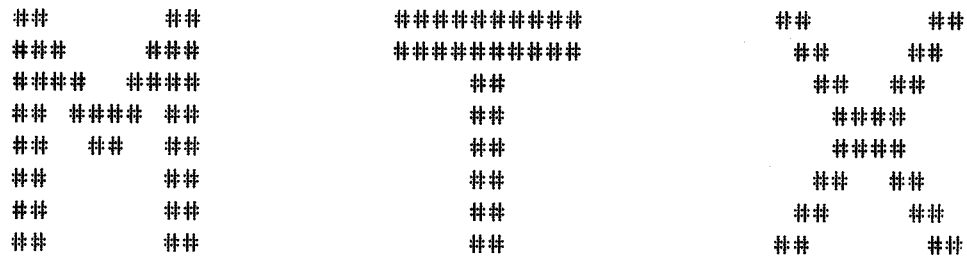


Info 3

M T X U s e r - C l u b D e u t s c h l a n d

1. **Zweck:** Austausch von Tips & Tricks u.s.w.

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

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

4. **Verpflichtungen** keine!

   Bitte: Einsendung von Tips & Tricks, Fragen, Antworten, kurzen Routinen, und was noch so zusammenkommt und andere interessieren könnte.

5. **Club-Info** wollen wir ca. monatlich verschicken. Das hängt von allen ab, da wir ja nicht rund um die Uhr am Computer sitzen können. Da brauchen wir die Hilfe aller Mitglieder!

6. **Kosten:** Wir berechnen ausschließlich Selbstkosten (Porto, Verpackung,..). Verständlicherweise verschicken wir **nichts**, wenn kein Geld da ist (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.)

7. **Geld/Konto:** Für jedes Mitglied führt Herbert Herberg ein Konto, von dem die enstehenden Kosten jeweils abgehen. Der Kontostand wird regelmäßig mitgeteilt, und kann selbverständlich jederzeit erfragt werden!

   Einzahlungen bitte möglichst auf's Club-Konto:
   (**Absender!** incl. Name **und** Anschrift nicht vergessen!)
   Postgiroamt Hamburg, BLZ 200 100 20,
   Herbert Herberg Sonderkonto C, Nr. 3480 00-200

8. **Kontaktadressen:** (beide derselbe Club!!)

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!

Im letzten Info waren zwei Seiten von Carsten Schmidt Software & Elektronik beigefügt. Diese Seiten, sowie jegliche anderen Listen eines kommerziellen Anbieters sind selbverständlich gratis, d.h. für die Kosten kommt der jeweilige Anbeiter auf! Clubmitglieder, die ihre Programme an den Mann bringen wollen (im kleinen Stil) brauchen für die Ankündigung (und ggf. Besprechung) im Info natürlich nichts zu bezahlen!

Ganz toll finde ich die Menge an Informationen, die an uns geschickt wurden!!! Danke! Fast alles habe ich in das Info aufgenommen! Deshalb ist es auch so umfangreich! Ich wurde u.a. gefragt: 'Warum ist das Info einseitig gedruckt ?'. Nun dazu: Beidseitig macht es nur dünner, spart aber nicht, und es muß auf beiden Seiten genug Rand zum Lochen bleiben ... und die Vervielfältigung wird aufwendiger! Außerdem finde ich die leeren Rückseiten für Notizen recht brauchbar.

Bisher wurden 32 Fragebogen an uns geschickt:
9x MTX 500, 23x MTX 512, 25x Drucker (teilweise Typenrad), 20x FDX.
Dies ist kein versuch, denjenigen ohne Floppy gegenüber das letzte floppyintensive Info zu rechtfertigen; immerhin haben wohl deutlich über 50% ein FDX und viele Startprobleme. Aber auch die ohne diesen großen schwarzen Kasten mit ein bis zwei Schlitzen und diesem angenehmen Gebläse (man hört einfach, daß der Computer auch was tut, und nicht nur das eigene Tastengeklappere und diverse Frustausbrüche, weil dieser Sch..-Kasten nicht das tut, was man meint ihm beigepult zu haben) sollen natürlich nicht zu kurz kommen!

Ich bitte die mittelmäßie Qualität der Kopien der ROM-Unterlagen, aber ich habe sie als recht schlecht Kopie bekommen, und versucht das beste daraus zu machen.

Anbei liegt evtl. eine Programm-Preisliste der Firma Windmill (Herzog-Franz-Str. 12, 3170 Gifhorn), auf der jedem Mitglied des MTX User-Club Deutschlands 10% Preisnachlaß gewärht werden. Als Mitglied für Windmill gilt die Mitgliederliste des als letztes erschienenen Infos! Wer noch nicht darauf steht muß etwas warten! (Wenn nicht, so ist die Liste noch nicht eingetroffen!)

Übrigens wird jeder, der dieses Info mal genauer betrachtet feststellen, daß in einer Zeile alle Leereichen gleichlang sind, also nicht mal ein und mal 2 Leerzeichen. Nun das macht NewWord, indem es für Leerstellen statt mit ganzen Blanks zu arbeiten mit Pixel-Blanks arbeitet (d.h. mit Blanks die einen Matrixdruckerpuknt breit sind, und nicht 8 Punkte breit). Das geht allerdings nur bei grafikfähigen Druckern, da diese evtl. 10 Punkte breiten Leerzeichen nur als Grafik an den Drucker gegeben werden können. Das ist auch der Grund, warum der Drucker die meisten Zeilen langsam und nur in eine Richtung druckt. Jedenfalls sieht das viel besser aus!!

Kontostand: Steht unter meinem Absender! Wer dieses Info noch nicht in Händen hat, der hat sein Konto anscheinend nicht aufgefüllt!! Wie gesagt: **'Wenn das Geld nicht reicht, dann verschicke ich nichts!'** Es ist doch wohl zumutbar (wie in Info 2 erbeten), daß jeder mindestens DM 10.- auf dem Konto Guthaben hat! Und ich kann nicht immer hinter dem Geld hinterherlaufen. (Es gibt bei mir Konten, die noch unter Null sind!!)

Telefonisch bin ich zu folgenden Zeiten erreichbar:
Dienstags 19.30 - 21.30 und Samstags 9.00 - 12.00

Herbert Herberg

32 Seiten ROM- bzw. BASIC-Info (Herbert Herberg)
Wir haben leider kein kommentiertes ROM-Listing und auch keine Aussicht bald eines zu bekommen. Für jeden Hinweis sind wir dankbar!
Inhalt der 32 Seiten: Voll disassembliertes & kommentiertes Cassettenhandling, Jump-Tables für die BASIC-Befehle, Joystick & Keyboard-Tips (u.a. eine Routine, wie man für den Joystick R, d.h. Cursortasten, Mehrfachdruck erfragt) sowie VRAM-Aufteilung & Videocontrollerdaten.
Diese Unterlagen sind für das ROM-BASIC sowie das FDXB (Diskette) gültig! Nur: ROM Page 1 ab Hex 2000 liegt bei FDXB ab #4000.
Diese Seiten gibt es bei Herbert Herberg für DM 8.- (incl P+V)

Bisher erschienen und noch erhältlich:
Info 1 (18 Seiten, DM 4.70), Info 2 (45 Seiten, DM 8.15) Allerdings dauert die Lieferung ggf. etwas, da ich nur begrenzte Vorräte habe.

Programme
Andreas Viebke: Flugsimulator DM 20.-

Herbert Herberg:
Die mit einem Stern gekennzeichneten Programme gibt es auch als Listing (DM -.30 je Seite). Ich tausche auch!
Preise netto, d.h. ohne Datenträger (DM 6.-) und P+V.
Die Programme mit einem Ausrufezeichen brauchen den 80-Schirm in der FDX und die 40-Zeichen-Grafik im MTX-Grundgerät.
Viele der u.g. Programme enthalten Maschinensprache (Assembler) und laufen nur auf dem MTX 500 und unter FDX-BASIC. Wer einen MTX 512 ohne FDX benutzen will, bei dem das Programm bei Hex 4000 (und nicht bei 8000) beginnt, muß vor dem laden des Programmes POKE 15*16^3+10*16^2+7*16+10,0 eingeben, um so den 512-er in einen 500-er zu wandeln.

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 suchr Labyrinth)
2.- * Breaktout (Mit Ping-Pong-Verfahren Steine zerstören)
2.- * Mamind (Mastermind = Kombinationsraten)
2.- * Jigsaw (Puzzle)
2.- * Missle (Städte vor Zerstörung schützen)
2.- * Anschiss (Mind over Electons)
1.- * Wasi (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)

DMX 80 - EPSON Druckercodes (Herbert Herberg)
Ich verwende die folgenden Abkürzungen/Bezeichnungen:
n: Binärzahl, **m:** Binärwort in der Reihenfolge Low-Byte, High-Byte,
#: Präfix für Hex-Code, **::** Beginn der Beschreibung,
/: Mehrere Optionen (A B/C heißt A B oder A C),
...: Hinweis, daß noch Bytes folgen müssen (s. Druckerhandbuch),
Bei Mehrteiligen Befehlen: Keine Leerzeichen einfügen!
Beachte: Bei Kommandos die #00, #01 oder #04 verlangen läßt der EPSON statt dessen auch #30, #31 oder #34 (d.h. 0, 1 oder 4) zu. Der DMX ist da anderer Ansicht!

Gemeinsamkeiten der Drucker
Control-Codes: (Ich verwende die standard ASCII-Abkürzungen: FF=#0C)
CR, LF, FF, HT, VT, BS, BEL, DEL (Bedeutung sicherlich klar)
SO/DC4 : Breitschrift für aktuelle Zeile an/aus
SI/DC2 : Schmalschrift an/aus
Escape-Sequenzen (d.h. vorne weg ein ESC = #1B)

#64	: Initialisierung
W #01/#00	: Breitschrift an/aus
E/F	: Fettdruck an/aus
G/H	: Doppeldruck an/aus
4/5	: Kursiv an/aus
- #01/#00	: Unterstreichen an/aus
S #01/#00	: Subscript/Superscript an
T	: " und " aus
R n	: Wähle Zeichensatz (DMX 0-7, EPSON 0-10)
C n	: Zeilen je Seite
C #00 n	: Inch (=Zoll) je Seite
0	: LF liefert Vorschub um 1/8 Zoll
1	: - " - 7/72 "
2	: - " - 1/6 "
3 n	: - " - n/72 "
A n	: - " - n/216 "
J n	: Einmaliger Vorschub um n/216 Zoll
8/9	: Papierende ignorieren/beachten
N n	: Oben und unten je Seite n Leerzeilen
O	: - " - 0 "
U #01/#00	: Unidirektional/Bidirektional drucken
<	: Eine Zeile unidirektional drucken
Q n	: Rechter Rand auf Spalte n
K m ...	: Grafik normal
L m ...	: Grafik doppelte Dichte

Gleiche Funktion, anderer Code (EPSON in Klammern)
Escape-Sequenzen:

P #01 (P)	: Pica Schrift
P #00 (M)	: Elite Schrift

Der EPSON gestattet vor den Controlcodes SO und SI ein ESC, welches ignoriert wird. Der DMX mag das nicht!

Nur DMX 80
Control-Codes
```
   DC1/DC3              : Online/Offline
```
Escape-Sequenzen
```
   D n n ... n #00      : Setzt horizontale Tabulatoren (Control HT)
   B n n ... n #00      : Setzt vertikale Tabulatoren (Control VT)
   Y n n1 ... n9        : Definiert ASCII-Zeichen n (Zeilenweise n1,...)
   Z n                  : Löscht Zeichendefinition von ASCII-Zeichen n
   >/=                  : Bit 7 wird auf 1/0 gesetzt
   #                    : Bit 7 bleibt unverändert
```

Nur EPSON
Escape-Sequenzen
```
   e #00/#01 n          : Setzt hor/vert Tabulator-Abstand
   f #00/#01 n          : Setzt hor/vert Tabulator-Einheit
   Y m ...              : Grafik doppelte Dichte schnell
   Z m ...              : Grafik normal schnell
   * 0/1/2/3            : Entspricht Escape-Sequenzen ESC K/L/Y/Z
   * 4/6                : Sondergrafikzeichen I/II
   m #00/#04            : Controlcodes als solche/Grafikzeichen
   l n                  : Linker Rand (das ist ein kleines L)
   s                    : Halbe Druckgeschwindigkeit (leiser)
```

Allgemeines zu beiden Druckern
Ein Punkt der Matrix hat den Durchmesser 2/216, und die Punkte haben einen vertikalen Abstand von 1/216. Also sind die Mittelpunkte der Matrixpunkte 3/216 = 1/72 Zoll auseinander! Dieser Text ist mit 1/6 Zoll Zeilenvorschub geschrieben. Die Mitgliederliste mit 1/8 Zoll.

Mal was Komisches:

W E R T S A C K
==========

Ein tiefsinniger Beitrag zu der alten Volksweisheit "Ein voller Sack hat's in sich" findet sich im Merkblatt, das die Deutsche Bundespost zum Paragraphen 49 ihrer ADA (Allgemeine Dienst-Anordnung) herausgegeben hat. Darin heisst es ua.: "Der Wertsack ist ein Beutel, der aufgrund seiner besonderen Verwendung im Postbefoerderungsdienst nicht Wertbeutel, sondern Wertsack genannt wird, weil sein Inhalt aus mehreren Wertbeuteln besteht, die in den Wertsack nicht verbeutelt, sondern versackt werden. Das aendert aber nichts an der Tatsache, dass die zur Bezeichnung des Wertsackes verwendete Wertbeutelfahne auch bei einem Wertsack mit Wertbeutelfahne bezeichnet wird und nicht mit Wertsackfahne, Wertsackbeutelfahne oder Wertbeutelsackfahne. Sollte es sich bei der Inhaltsfeststellung eines Wertsackes herausstellen, dass ein in einem Wertsack versackter Versackbeutel statt im Wertsack in einem der im Wertsack versackten Wertbeutel haette versackt werden muessen, so ist die in Frage kommende Versackstelle unverzueglich zu benachrichtigen. Nach seiner Entleerung wird der Wertsack wieder zu einem Beutel, und er ist auch bei der Beutelzaehlung nicht als Sack, sondern als Beutel zu zaehlen."

O welche Lust, Postbeamter zu sein ...

```
#####     ###      #####
#        #   #     #
###      #   #     ###
#        #   #     #
#####     ###      #
```

Der EOF-Befehl im FDXB!!!

Entdeckt und untersucht von Herbert Herberg!!
Die im FDXB-Handbuch angegebene Syntax ist falsch!!! Korrekt:

DISC EOF kanal,sprungadresse

Dabei ist kanal die Nummer wie im OPEN-Befehl, also z.b. #1, und sprungadresse eine Zeilennummer.
Folgendes Programm liest die Datei TEST und gibt sie auf dem Bildschirm aus:

```
10 OPEN #1,"TEST","I"
20 DISC EOF #1,60
30 DISC LINE INPUT #1,A$
40 PRINT A$
50 GOTO 20
60 PRINT: PRINT "Ende": PRINT: PRINT: PRINT: PRINT
70 DISC CLOSE #1
```

In Info 2 ist eine Liste aller anderen FDX-BASIC Disc-Befehle!

Achtung (Herbert Herberg)
Es gibt mindestens zwei versionen vom FDX-BASIC!

1. Eine alte defekte, bei der der EOF-Befehl wie im Handbuch als Function enthalten ist, d.h. PRINT EOF(1) liefert eine Zahl, und nicht 'undefined'.

2.Eine neue heile. In der neueren (die u.a. von Vobis geliefert wird) funktioniert der EOF-Befehl wie oben beschrieben. Wer in dieser Version PRINT EOF(1) eingibt wird mit 'undefined' belohnt.

Bei wem also PRINT EOF(1) funktioniert, der sollte sich schleunigst nach einer neuen FDXB-Version umsehen, da die Verarbeitung von Dateien nicht zuverlässig klappt und Daten zerstört. Z.B. überschreibt das alte FDXB bei Random-Dateien (= Direktzugriffsdateien) ohne weiteres einen Teil des nächsten Satzes, und DISC INPUT #1,A$ liest nicht einen Satz, sondern auch mal gleich den ganzen Rest der Datei. Naja, damit macht das Abreiten sicherlich keinen Spaß!

TAXAN - Farbmonitor
Das kleine Netzteil, daß mit dem MTX 500/512 ausgeliefert wird ist für englische Netzspannungen ausgelegt: 240 Volt!! Und wir sitzen mit lumpigen 220 Volt dran: Diese fehlenden 9% Netzspannung bewirken eine etwas zu niedrige vom Netzteil gelieferte Spannung. Für die meisten an den MTX angeschlossenen Geräte langt das, aber beim TAXAN kann das zu Farbfehlern und/oder Farbweggfall des Bildes führen. Der Fehler liegt also da im kleinen Netzteil des MTX. Mit dem Netzteil in der Floppystation gibt es keine derartigen Probleme.

Schach (Andreas Viebke)
Das Schachprogramm CHESS von Continental ist sein Geld nicht wert. Das Superchess III für den Spektrum 48k des selben Autors ist um Längen besser. Folgendes fehlt dem CHESS:

- Keine Anzeige der verbrauchten Zeit
- Rechenvorgang läßt sich nicht unterbrechen
- Unheimlich langsam: In Stufe 4 kann man einkaufen gehen, und in Stufe 6 wahrscheinlich seine Tante in Australien besuchen. Superchess braucht max. 5 Minuten und spielt verdammt gut.
- Löst keine Problemstellungen
- Zeigt keine Zugbewertung an
- Keine Widerholung einer Partie
- Häßliche Figuren
- Meiner Meinung nach geringe Spielstärke

Zeitmessungen: (Zug 1 & 2 aus der Bibliothek)

Zug	Lev. 4	Lev. 3	Lev. 2	Lev. 1	Lev. 0
3	8'45''	2'49''	0'24''	0'08''	0'02''
4	23'18''	6'29''	0'27''	0'08''	0'02''
5	15'25''	6'12''	0'34''	0'13''	0'02''

Schach (Herbert Herberg)
Zu o.g. von mir folgendes: Man kann jedes Spiel in CHESS (Continental) mit X unterbrechen, dann in den Analysemodus gehen, Figuren umstellen (also auch Züge revidieren, wenn auch umständlich), dann wieder X eingeben um den Analysemodus zu verlassen. Danach wird man nach Frabe (also ggf. Brett umdrehen), Level und Spieler (B oder W) gefragt; ... und dann kann man weiterspielen. Analyse von Stellungen und lösen von Problemem kann mit dem Analysemodus (Brett entsprechend gewünschter Stellung füllen) zu Leibe rücken!
Ich habe mir mal das MYCHESS Schachspiel angesehen und auch mal die Zeiten gemessen: (Ply=Halbzug)

Zug	3 Ply	2 Ply	1 Ply
3	1'	1'	< 1'
4	43'	9'	4'
5	37'	10'	6'

(so genau wie Andreas konnte ich das nicht messen,... aber ich finde die Zeiten im vergleich zu CHESS recht happig.) Wenn mein 2. MTX wieder repariert ist, werde ich mal CHESS gegen MYCHESS spielen lassen, allerdings nicht auf allzu hohem Level.

40 Tracks ? - 48 Tracks ?
Wer hat dazu zuverlässige Daten? Unser CP/M glaubt 2 System-Tracks + 312 kByte auf ? Tracks mit je 26 Sektoren a 128 Byte, also: 96 Tracks. Folglich 48 je Seite ... und die System-Spuren? die zählen nicht???

Das Programm Fract'1 aus Info 1 (Thomas Pflaum)
Das Programm erstellt ja recht eigenwillige und interessante Grafiken und Ausschnitte von solchen. READ X1,X2,Y1,Y2 liest die Koordinaten eines Rechteckes ein, die einen Ausschnitt definieren, der vom Gesamtbild gemacht werden soll. Dazu ist zur Veranschaulichung ein Zettel mit einem Beispiel anbei. Die eingezeichnenten Rechtecke in den oberen beiden Zeichnungen zeigen den Ausschnitt an, der geliefert werden soll und als Ausdruck direkt darunter ist. Anbei sind selbverständlich die Werte der DATA-Zeile für das READ zu den drei Zeichnungen.
Zur Namensgebung: fract = fraction = engl.€Teil§

Characterhandling auf dem MTX ist immer für eine Überraschung gut
(Herbert Herberg)
DIM T$(10) liefert einen Characterstring (=Zeichenfolge) der **Maximallänge** 10, d.h. reserviert Platz für z.B. **bis zu** 10 Buchstaben. LET T$="12345678901" ist also unzulässig, da das 11 Zeichen sind. Mit T$(3) kann man auf das 3. Zeichen von T$ zugreifen, T$(3,2) liefert eine 2 Zeichen lange Folge, die das 3. und 4. Zeichen beinhaltet. Zur Erläuterung:

```
DIM T$(10) : T$="1234567890"
LET  T$(2)="0"                          liefert T$="1034567890"
PRINT T$(3,3)                           liefert 345
```

Nun ist es oft notwendig gleichzeitig mehrere solche Zeichenfolgen zu haben, z.B. 3 Folgen a 9 Zeichen. Den Speicher reserviert DIM T$(3,9). ACHTUNG: Hier werden **genau** 3 mal 9 Zeichen Platz reserviert und bei jedem Zugriff werden alle neun zusammen angesprochen, und nicht wie bei DIM T$(10) Platz für **bis zu** 10 Zeichen. (DIM S$(1,10) liefert also einen String der genauen Länge 10.) Auf den ersten String von T$ kann man mit T$(1) zugreifen, d.h. damit greift man auf T$(1,1), T$(1,2), ..., T$(1,9) gleichzeitig zu. Will man nur einen Teil der Zeichenfolge, z.B. 3. bis 6. Zeichen des 2. Strings so nimmt man T$(2,3,4) (d.h. String 2, ab Pos. 3, Länge 4). Nur gibt es eine Tücke: Will ich den 3. String von T$ (T$ ist ja eine Ansammlung von 3 Strings der Länge 9) löschen, d.h. mit CHR$(0) auffüllen, so lautet der Befehl dafür nicht LET T$(3)=CHR$(0), sondern FOR I=1 TO 9 : LET T$(3,I)=CHR$(0) : NEXT. Unten ist ein BASIC-Programm, mit dem ich diesen Sachverhalt nocheinmal verdeutlichen will. Dabei stehen hinter den ! keine Teile des Programms, sondern Begleitkommentare. Der Resultierende String wird in ' eingeschlossen; ein Zeichen mit dem ASCII-Code Null (d.h CHR$(0)) wird mit einem . Bezeichnet. Bemerke: Wenn der String die Länge 9 hat, dann ist er leer, falls er mit ASCII-Nullen gefüllt ist! Die Abfrage LEN (T$) liefert stets die mit dem DIM-Befehl festgelegte Größe, falls T$ mehr als eine DIMension hat.

```
DIM T$(3,9)                     ! 3 Strings a 9 Zeichen
LET T$(1)="123456789"           ! 1. String = '123456789'
LET T$(1)="ABC"                 ! 1. String = 'ABC456789'
LET T$(1)=""                    ! Versuch 1. String zu löschen
                                !  liefert '123456789'
LET T$(1)=CHR$(0)               ! Nochein Versuch, auch ohne Erfolg
                                !  liefert '.23456789'
FOR I=1 to 9                    ! Letzter Versuch (s.o.) liefert
T$(1,I)=CHR$(0)                 !  das gewünschte: '.........'
NEXT I                          !  ein  leerer  String  der  Länge  9
```

Also wird mit T$(1) imme eine Zeichenfolge der genauen Länge 9 angesprochen, und LET T$(1)=B$ bewirkt, daß T$(1) linksbündig mit B$ überschrieben, und ist B$ zu kurz, so bleibt ein Teil von T$(1) unverändert; ist B$ zu lang, so gibt's eine Fehlermeldung 'no space'. Mein Vorschlag ist es, einfach einen String der passenden Länge als Löscher zu denfinieren: DIM NU$(9): FOR I = 1 to 9: LET NU$(I)=CHR$(0): NEXT. Dann kann man zum Löschen einfach schreiben: LET T$(1)=NU$.

Also: DIM T$(10) = ein String der Maximallänge 10: T$
DIM T$(1,10) = ein String der genauen Länge 10: T$(1)
DIM T$(3,10) = 3 Strings der genauen Länge 10: T$(1),T$(2),T$(3)

Was ist ein Diskettenfile ? (Herbert Herberg)
Nun erst mal kann man sagen, das ist eine Menge von Speicher auf der Diskette, statt im RAM (also Speicher im MTX), aber etwas genauer werde ich noch: Es gibt zwei Arten von Dateien (Datei = File, aber deutscher!): sequentielle und Direktzugriff.

Sequentielle Dateien:
Das sind Dateien, die man mit einem Stapel Papier mit irgendwelchen Notizen vergleichen kann, die in Zeilen unterteilt werden können. Wenn ich diese Datei zum Beschreiben eröffne, so werden die Daten hinten angehängt; beim Lesen wird am Anfang gestartet. Das klingt irgendwie vernünftig: entweder erweitere ich meine Notizen durch weitere Zettel, oder lese von vorne. Nun fragt sich jeder, warum kann ich nicht in der Mitte etwas einfügen, oder dort anfangen zu lesen. Nu ja Einfügen bedeutet, daß alles was dahinter ist bewegt werden muß (mein Gott kostet das Arbeit und Zeit!!!) und in der Mitte mit dem Lesen beginnen ist etwas schwer, da ja die Zeilen unterschiedliche Länge haben. Also bleibt es dabei Lesen vom Anfang an, Schreiben hängt hinten an. Wer partout in der Mitte lesen will, muß entsprechend oft DISC LINE INPUT auf die Datei loslassen. Natürlich kann ich meinen Stapel Papier auch wegwerfen (DISC ERA). Aber nun etwas BASIC:
Nennen wir die Datei mal STAPEL (warum auch nicht). Nun brauchen wir eine Kanalnummer (das will das BASIC), damit man nicht bei jeder Ein- oder Ausgabe (d.h. Lesen oder Schreiben) den Namen der DATEI angeben muß. Da BASIC Kanal 1 bis 4 erlaubt, können **gleichzeitig** vier Dateien bearbeitet weden. Nehmen wir mal Kanal 2. Wollen wir etwas an die Datei anhängen, also schreiben, so brauchen wir den Befehl:
DISC OPEN #2,"STAPEL","O" ("O" = Output = Ausgabe = Schreiben);
wollen wir hingegen Lesen, so:
DISC OPEN #2,"STAPEL","I" ("I" = Input = Eingabe = Lesen).
Zum Schreiben haben wir DISC PRINT #2,...;
zum Lesen: DISC INPUT #2,... und DISC LINE INPUT #2,...
Der Unterschied zwischen INPUT und LINE INPUT ist ganz einfach (Line = Zeile): INPUT liest was kommt, LINE INPUT liest auch was kommt, aber nur bis zum Zeilenende (und ignoriert ggf. den Rest der Zeile). Beim Lesen ist es wichtig, daß die Variablen im INPUT (mit oder ohne LINE) zu den Daten in der Datei passen! (Als wenn mit INPUT ohne DISC gearbeitet wird, wobei dann die Daten eingetippt werden.) Außerdem muß auch was da sein zu lesen, und dafür gibt es den EOF-Befehl (EOF = End Of File = Ende der Datei) der erst mal schaut, ob noch etwas da ist:
DISC EOF #2,40 macht folgendes: Wenn in der Datei (in unserem Fall STAPEL) nichts mehr steht, dann springt BASIC zu Zeile 40 - ist noch etwas da, so geht es in der nächsten Programm-Zeile weiter. (D.h. wenn Ende der Datei, dann GOTO 40, sonst nächste Programm-Zeile).
ACHTUNG: Die Datei muß nach dem man fertig ist mit dem DISC CLOSE geschlossen werden!!, sonst kann es zu Problemem kommen (DISC EOF funktioniert nicht, u.v.a)!!
Nun ein Anwendungebeispiel: Zeilen 10 bis 70 lesen Paare von X und Y - Werten ein und schreiben sie in die Datei "STAPEL". Dann werden die Werte in Zeilen 100 bis 190 auf dem Bildschirm ausgegeben (Ich gehe davon aus, das STAPEL noch nicht als Datei auf der Diskette existiert). X=0 und Y=0 heißt fertig.:
(Natürlich können X,Y auch irgendwelche Werte sein, die das Programm berechntet, wobei dann Zeile 20 und ggf. 30 anders aussehen.)

```
10 DISC OPEN #2,"STAPEL","O"              ! Eröffnen
20 INPUT "Bitte X,Y eingeben";X,Y         ! Einlesen X und Y
30 IF X=0 AND Y=0 GOTO 60                 ! Ende der Eingabe
40 DISC PRINT #1,X,Y                      ! Schreiben von X,Y in Datei
50 GOTO 20                                ! Weiter
60 DISC CLOSE #2                          ! Fertig, Datei schliessen
70 STOP                                   ! Halt
100 DISC OPEN #2,"STAPEL","I"             ! Eröffnen
110 DISC EOF #2,150                       ! Ende ?
120 DISC INPUT #1,X,Y                     ! Einlesen X,Y
130 PRINT "Eingelesene Werte : ";X,Y      ! Ausgeben X,Y
140 GOTO 110                              ! Weiter
150 DISC CLOSE #2                         ! Datei schliessen
160 STOP                                  ! Halt
```

ACHTUNG: Wenn Character-Strings (d.h. Variablen mit $ wie A$) eingelesen werden mit DISC INPUT , dann wird das Zeilenende ignoriert!!; bei DISC LINE INPUT wird das Zeilenende beachtet.

Direktzugriffs-Dateien

Das sind Dateien, die mit Zeichen-Feldern verglichen werden können die mittels DIM T$(a,b) festgelegt werden, wobei b die Länge eines Strings ist, und a die Anzahl der Strings (siehe hierzu auch vorvorherige Seite) - nur, daß für eine Datei zwar b auch festgelegt werden muß, aber a nicht, d.h. die Anzahl der Strings ist varaibel (wie schön!). Wer weiß z.B. bei einer Adressenliste vorher wieviele es werden - und was wenn es mehr oder weniger werden?

Der große Unterschied zu sequentiellen Dateien: Man kann schreiben und lesen gleichzeitig (bei sequentiellen Dateien heißt Schreiben ich bin am Ende (der Datei natürlich), Lesen ich in am Anfang), bei einer Direktzugriffsdatei halt wo immer DISC REC mich hinpackt. Bei einer Datei spricht man von Sätzen statt Strings, da das auch etwas anderes ist. Nun fragt sich jeder, woher weiß der MTX, welcher Satz. Nun dazu gibt es den Befehl DISC REC. Damit wird die Nummer des Satzes bestimmt. (REC = Record = Satz, damit ist ein String der Datei gemeint). Beim Schreiben darf natürlich nicht die Länge eines Satzes überschritten werden (wohin sollte auch der Rest?). Ganz schön verwirrend, nicht wahr?

Da die Sätze aus denen die Datei besteht eine fest vorgegebene Länge haben, sieht es BASIC gerne, daß mit **einem** String der gleichen Länge auf die Datei zugegriffen (gelesen oder geschreiben) wird. Zahlen kann man mit STR$ in Strings umwandeln.

Eröffnen: OPEN #2,"STAPEL","R",L (R = Random Access = Direktzugriff, L ist die Länge der Sätze der Datei)

Postionieren: DISC REC #2,I (I ist die Nummer des Strings, die von Eins an durchnummeriert sind)

**Lesen: DISC INPUT #2,A$** (Dabei sollte A$ die Länge L haben)

Schreiben: DISC PRINT #2,A$ (- " -)

Wenn in einen Satz mehrere Werte bzw. Strings gespeichert werden sollen, ist es ratsam alle diese Daten in einen String der Länge eines Satzes umzuspeichern.

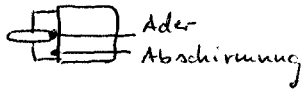
Ansonsten ist zu bemerken: DISC PRINT und DISC INPUT positionieren automatisch auf den nächsten Satz. Sonst funktioniert alles wie für sequentielle Dateien.

An die Aufschrauber
Der freie Sockel neben der Z80 CPU im MTX-Grundgerät ist der Parallelport, der mit LET I=INP(7) und OUT 7,I (BASIC) bzw. IN A,(7) und OUT (7),A (Assembler) angesprochen wird. Siehe deutsches Handbuch Seiten 240 & 237 bzw. englisches Handbuch Seiten 252, 249. Dort ein IC einzusetzen dürfte höchstens dem MTX entsetzen!

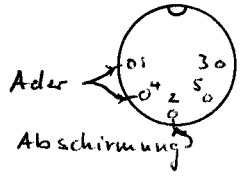
Cassettenhandling (Herbert Herberg)
Wer hat schon einmal erfolgreich die Baudrate (d.h. Übertragungsgeschwindigkeit) für SAVE und LOAD auf Kassette verändert?
VERIFY scheint ja vielen Probleme zu bereiten. Nun ich mache es folgendermaßen: CLEAR : SAVE "Programm" : CLEAR : VERIFY "". Damit hat es bislang immer noch geklappt. Da ich faul bin, schreibe ich diese vier Befehle in eine Zeile im Programm, die nie angelaufen wird, und um zu Speichern (SAVEn) gebe ich EDIT zeilennummer ein, lösche die Zeilennummer, und schicke die vier Befehle ab. Allerdings darf man die vier Befehle nur dann auf ein mal abschicken, wenn man hört, wann SAVE fertig ist. Sonst ggf. erst CLEAR : SAVE "Programm", Kassette zurückspulen und dann CLEAR : VERIFY "" abschicken.

Anschluß MTX an HIFI-Anlage (Herbert Herberg & Jürgen Adamczak)
Wer am Verstärkereingang CINCH-Buchsen (das sind solche, wie der MTX als HI-FI-Buchse hat) hat, der braucht nur ein einfaches Überspielkabel in beide Geräte zu stöpseln, und los geht's. Wenn das Kabel zu kurz ist, einfach zwei Stecker und ein paar Meter einpoliges abgeschirmtes Kabel (kann ruhig dünn sein!) kaufen, und zusammenlöten: Die innere Ader des Kabels mit den inneren Kontakten der Stecker verbinden, und die Abschirmung mit dem äußeren Steckerkontakten. Wer aber nur diese 5-Poligen (sog. DIN-Stecker) hat braucht einen Chinch-Stecker, einen solchen 5-poligen (sog. Diodenstecker) und ebenfalls ein paar Meter Kabel (wie oben). Der Chinch-Stecker wird genauso wie oben beschrieben angeschlossen. Beim 5-poligen muß man erst mal die kleinen Zahlen neben den Kontaktstiftchen finden, die in Halbkreis in der Reihenfolge 1-4-2-5-3 nummeriert sind, und die Ader mit 1 und 4, die Abschirmung mit 2 verbinden.

Chinch-Stecker



5-Pol-Stecker



FDXB Warmstart (Herbert Herberg)
Wenn man unter CP/M eine Diskette wechselt, so muß man dies dem System mit CTRL-C oder BRK mitteilen, damit der Rechner weiß 'Aha, neue Diskette'. Beim FDX-BASIC geht das so nicht! Die Gewaltlösung, die ich im Info 2 vorschlug klappt auch nicht immer, besonders dann nicht, wenn schon eine Datei mit dem gleichen Namen wie im DISC SAVE auf der Diskette existiert.
CTRL-C ruft die CP/M Warmstart-Routine auf. Wer weiß, wo diese (und ggf. die anderen) BIOS-Routinen im FDX-BASIC stehen?

Noddy

Noddy-Zeilen stehen hinter dem BASIC-Programm.
Das Obere Ende von BASIC ist in den Speicherstellen BASTPO (#FAAC) abgespeichert: Je zwei Bytes je RAM-Seite (FDXB arbeitet nur mit einer Seite, hingegen das ROM-BASIC kann auch mehrere Seiten verwalten). In #FAAC,#FAAD steht die Obergrenze von BASIC-Seite 0, in #FAAE,#FAAF die von Seite 1, u.s.w. In #FAA7,#FAA8 steht die Obergrenze der aktuellen Seite, und in #FAA9 die höchste verwendete Seite. Für FDX-BASIC ist also zwangsläufig in #FAA9 eine #00.
Die Obergrenze von Noddy ist in #FAA4,#FAA5 und die zugehörige Seite in #FAA6.
Die einzelnen Noddy-Seiten werden wie folgt kodiert:
1. & 2. Byte: Länge der Seite (d.h. Länge = 256* 1.Byte + 2.Byte), dahinter steht der Name der Seite gefolgt von #C1. Dahinter steht der Text der Noddy-Seite und am Ende ein #FF. Um Platz zu sparen werden Leerzeichen (Blanks) nicht als solche gespeichert, sondern eine Liste von n Blanks wird durch ein Byte mit dem Code #80+n gekennzeichnet. Dabei darf #80+n max. #FE sein (d.h. max 126 Blanks je Byte), und ggf. werden mehrere Bytes verwendet. Beispiel: 77 Blanks werden als #CD kodiert, hingegen 200 Blanks als #FE,#CA. Unten ein BASIC-Programm, das einen Text in A$ in Noddy-Speicher-Format konvertiert: (Nur den Text, ohne Länge, #C1 und #FF) (Die LET's habe ich weggelassen!)

```
100 REM Ich gehe davon aus, daß A$ den Inhalt schon hat.
110 DIM B(1000)
120 REM Das Ergebnis kommt nach B$
130 IB=1: BLANK=0
140 FOR IA=1 TO LEN (A$)
150 IF MID$(A$,IA,1)=32 THEN BLANK=BLANK+1: GOTO 230
160 IF BLANK=0 THEN GOTO 220: REM Keine Blanks da gewesen
170 K=INT(BLANKS/126) : REM Anzahl der Bytes mit 126 Blanks (#FE)
180 L=MOD(BLANKS,126) : REM Anzahl der Blanks im Letzten Byte
190 IF K>0 THEN FOR I=1 TO K: B(IB)=254: IB=IB+1: NEXT
200 IF L>0 THEN B(IB)=128+L: IB=IB+1
210 BLANKS=0
220 B$(IB)=ASC(MID$(A$,IA,1)): IB=IB+1
230 NEXT IA
240 REM Die Restblanks werden nicht gespeichert!
```

Wer nun mal in Hex-Code sehen will, wie die Zeile dann abgespeichert wird mit Länge u.s.w., muß die Länge berechnen, und die Zahlen im Feld B als Hex-Zahlen ausgeben. Zur Konvertierung benutze ich die DEZ-HEX-Routine auf der nächsten Seite in diesem Info. Dabei mut Zeile 1060 so aussehen:
1060 LET X$="#"+RIGHT$("00"+X$,2) Wir wollen ja nur zwei Hex-Ziffern.
Das Programm dazu sieht folgendermaßen aus:
(In NAME$ steht der Name der Seite)

```
200 LAENGE=2 + LEN (NAME$) + 1 + IB-1 + 1
210 REM (Länge: 2 Bytes Länge, Namenlänge, #C1, Text, #FF)
220 X=INT(LAENGE/256): GOSUB 1000: PRINT X$;" ";
230 X=MOD(LAENGE,256): GOSUB 1000: PRINT X$;" ";
240 PRINT "#C1 ";
250 FOR I=1 TO IB-1: X=B(I): GOSUB 1000: PRINT X$;" ";: NEXT
280 PRINT "#FF": PRINT: PRINT: PRINT
```

VOBIS - Verkaufstrategie
Bei Vobis weiß man anscheinend nicht was man will! Jetzt gibt es von Vobis für knapp 900.- dem MTX 512 und angeblich zwei verschiedenen FDX-Systeme mit einem Laufwerk: eines für MTX 500 und eins für MTX 512 (ich habe eins für den 500'er am MTX 512 und es läuft!). Dabei wird dann mit dem für den 500-er die 32k-Erweiterung mitgeliefert, und beim anderen FDX-System nicht. Und wer einen 512-er hat kann mit der 32-Erweiterung nichts anfangen, da nun auch die PAL's nicht mehr getauscht werden sollen. Verwirrend, nicht wahr! Mal sehen was das noch gibt. Die fehlenden IC's für die serielle Schnittstelle (und was noch so da fehlt) gibt es bei Vobis nun für DM 75.-

Umwandlung HEX-DEZ-HEX in BASIC (Herbert Herberg)
X ist die Dezimalzahl, X$ die Hex-Zahl.
Folgende Variablen werden verwendet und müssen ggf. umbenannt werden:
H$ = Char-String mit den Hex-Ziffern; I,J = Hilfsvariablen

```
1000 REM Umwandlung DEZ->HEX  Parameter: Rein X, Raus X$
1010 LET H$="0123456789ABCDEF"
1020 IF X<0 THEN LET X$="ERROR": RETURN
1030 LET I=X: LET X$=""
1040 LET X$=H$(MOD(I,16)+1)+X$: LET I=INT(I/16)
1050 IF I>0 GOTO 1040
1060 LET X$="#"+RIGHT$("0000"+X$,4)
1070 RETURN

2000 REM Umwandlung HEX->DEZ  Parameter: Rein X$, Raus X
2010 LET X=0
2020 IF LEN (X$)=0 THEN RETURN
2030 IF LEFT$(X$,1)="#" THEN J=2 ELSE J=1
2040 FOR I=J TO LEN (X$)
2050 LET X=X*16+ASC(MID$(X$,I,1))-48+7*(MID$(X$,I,1)>"9")
2060 NEXT I
2070 RETURN
```

Bemerkung: (MID$(X$,I,1)>"9") hat den Wert 0 (Falsch) oder -1 (Wahr)! Negative Dezimalzahlen werden nicht verarbeitet! Die Zeile 1060 hat den Nachteil, daß große positive Zahlen nicht verarbeitet werden können. Sie kann weggelassen werden, da sie die Zahlen nur auf ein vierstelliges Hex-Format mit vorangestelltem # bringt. Zeilen 2020, 2040: hinter LEN ein Leerzeichen lassen!!! (Sonst: undefined)

Billige, einfache Disketten
Ich verwende zwar die billigen single side, single density Multilife-Disketten von Vobis (10 Stück DM 39.-), und das **ohne** irgendwelche Probleme wie Datenverlust, ... - **aber** Frank Dersewski hat mit diesen Disketten nur Probleme wie bad sector! **Also:** ausprobieren, und für wichtige Sicherheitskopienen sollte man sowieso gute (d.h. double sided, double density) Disketten verwenden!!

Joysticks (Herbert Herberg)
Das Problem: Diagonalen und Richtungen zusammen mit Feuer abfragen! Nun für den Joystick R = Cursortasten steht in unseren 32 Seiten ROM-Unterlagen ein Assembler- und ein BASIC-Programm, für den Joystick L = Tasten YCBM und Leertaste hatten wir im Info 1 ein BASIC-Programm. Aber eben dieses Programm hatte so seine Tücken: Der Druckfehler ist sicherlich jedem rasch aufgefallen: hoch = 251 und nicht 281 (was ja auch unmöglich ist, da ein Byte <256 ist). Aber es gab da noch eine Tücke: fragt man Port 6 für Feuer in BASIC ab so erhält man 123 bzw. 122, hingegen wird IN A,(6) mit einer 11 bzw. 10 im Register A quittiert. Desterhalb und diesterwegen ist ein Programm mit BASIC und Assembler-Routine dafür anbei! Dabei wird der Wert von Port 5 noch umgerechnet: I=15-MOD(INP(5),16). Dann haben die vier Himmelsrichtungen als Wert gerade Zweierpotenzen, und die Diagonalen sind die Summe der zugehörigen Himmelsrichtungen (Nord = 4, West = 1, also Nordwest = 5). Zweierpotenzen entsprechen obendrein genau den Bits und können also mit BIT 1,A u.s.w. in Assembler abgefragt werden.

FEHLER in INFO 2 (Jens Ebert & Herbert Herberg)
Ja, auch das gibt's: Auf Seite 10 der mit MTX User-Club Deutschland überschriebenen Seiten hat das Uhr-Programm zwei Fehler: Die 5. und 7. Assemblerzeile von unten heißt LD (#FD5B),A bzw. LD (#FD5C),A. Da muß natürlich LD A,(#FD5B) bzw. LD A,(#FD5C) heißen (also LD A,... statt LD ...,A).

FDX - Probleme
Ärger mit Disketten,...: Aufschrauben, Kontaktspray zur Hand! Dann:
1. alle Steckkontakte trennen - sprayyyy - und verbinden
2. alle DIP-Schalter - sprayyyy - und 50x hin und her bewegen

NewWord auf FDX
Wenn der Speicher voll ist (und das meldet NewWord ziehmlich früh) einfach ignorieren!!! NewWord legt automatisch Teile des Textes auf Diskette ab, so daß man sich um nichts (außer Platz auf der Diskette) kümmern muß.

Disketten-Format
Ja, wir haben Triumpf Adler Alphatronic PC - Format!!! bis auf zwei Unterschiede: TA hat 4k Directory, wir nur 2k und beim TA beginnen die Daten (Directory und Dateien) 76 Sektoren weiter hinten!! Das ist alles! Und das stimmt!!
Konvertierung: TA -> MTX: Alles 76 Sektoren nach vorne holen. Es dürfen nur max. 32 Directory-Einträge auf der Diskette sein.
Konvertierung: MTX -> TA: Alles 76 Sektoren nach hinten schieben. Dabei muß nur als erstes eine 2k Datei in der nur Hex E5 steht als erstes auf die Diskette, und direkt vor der Konvertierung gelöscht werden.

Directory auf den Drucker:
In CP/M schaltet CTRL-P den Drucker ein, so daß alles was auf den Bildschirm kommt auf den Drucker wandert. Abschalten mit CTRL-C.
Also eingeben: CTRL-P DIR RET (dabei DIR = Tasten D,I,R)

Kosten für dieses Info: DM 8.60

Ansteuerung der 80-Zeichen-Karte (Bernd Preusing)

Die 80-Zeichen-Karte benutzt folgende Ports:

```
#38 (out)      Ausgabe der 6845-Registernummer
#39 (in/out)   Lesen/Schreiben eines 6845-Registers
#30 (out)      Ausgabe von A0..A7 der Schirm-Adresse
#30 (in)       Strobe für Tonausgabe (CTRL-G)
#31 (out)      D0..D2: A8..A10 der Schirm-Adresse
               D3..D4: unbenutzt
               D5:     Schreib-Freigabe für Attribut-RAM
               D6:     Schreib-Freigabe für Zeichen-RAM
               D7:     1=schreiben, 0=lesen
#32 (in/out)   Zeichen (0..255)
#33 (in/out)   Attribut-Byte (0..255)
```

Die Karte enthält je 2K RAM für die Zeichen und die Attribute. Dieses RAM ist 'zirkulär' organisiert, d.h. nach Adresse #7FF kommt wieder Adresse 0. Das macht es möglich, ein Hardware-scrolling durchzuführen, indem einfach die RAM-Start-Adresse in den Registern 12 und 13 des 6845 um 80 erhöht wird. Andererseits wird es dadurch schwierig, eine bestimmte Bildschirmposition x,y anzusteuern, da man immer erst den RAM-Start als Offset auf die entsprechende Adresse berücksichtigen muss. Diesen Offset kann man durch Auslesen der 8645-Register (AND #7FF) erhalten, oder, falls der Bildschirm-Treiber im CP/M benutzt wird, aus den Adressen #F411,#F412 holen.
Ganz oben links ist Adresse 0, oben rechts ist 79, die zweite Zeile beginnt bei 80, usw. bis unten rechts=24*80-1. Die Adresse errechnet sich also aus (Offset+x+80*y) MOD #7FF. Im FDXB kann man von Offset=0 ausgehen, da hier nicht gescrollt wird.

Will man nun z.B. ein 'A' an Adresse HL schreiben, so kann man dies mit folgender Assembler-Routine erreichen:

```
LD   A,#0C          ;REGISTER 12 DES 6845
OUT  (#38),A        ;..ANWÄHLEN
IN   A,(#39)        ;HIGH-BYTE RAM-START
LD   D,A
LD   A,#0D          ;REG 13
OUT  (#38),A
IN   A,(#39)        ;LOW-BYTE RAM-START
LD   E,A
ADD  HL,DE          ;+OFFSET
                    ;bis hier nur ,falls nötig!
LD   A,'A'
OUT  (#32),A        ;ZEICHEN ANLEGEN
LD   A,0            ;ATTRIBUT-BYTE
OUT  (#33),A        ;..ANLEGEN (FALLS ERWÜNSCHT)
LD   A,H            ;HIGH-BYTE DER ADRESSE
AND  #07            ;NUR 3 BIT GÜLTIG
                    ;ODER AND #1F, DA D3..D4 UNBENUTZT
OR   #E0            ;SCHREIBEN IN BEIDE RAMS
                    ;ODER OR #C0, FALLS NUR ZEICHEN
OUT  (#31),A        ;ADR UND WRITE-ENABLE ANLEGEN
LD   A,L            ;LOW-BYTE DER ADRESSE
OUT  (#30),A        ;ADR ANLEGEN UND STROBE
```

Wichtig ist, daß OUT #30 immer als letztes erfolgt, da hierdurch erst die anderen Ports freigegeben werden! Die Ports #31..#33 sind 'latched', d.h. sie behalten ihre Inhalte solange, bis etwas anderes eingeschrieben oder der Video-Speicher ausgelesen wird. Dadurch ist es auch möglich, einen ganzen Bereich zu löschen, indem nur einmal die Werte in #32 und #33 gesetzt werden, und nur jeweils die Adresse erhöht wird.

Wer nun meint, er könne eine interrupt-gesteuerte Uhr auf den Bildschirm zaubern, wie es im letzten Info für den 40-Zeichen-Schirm gezeigt wurde, der irrt. Die Bildschirmausgabe wie oben beschrieben darf nämlich nicht von einer anderen Ausgabe-Routine unterbrochen werden, da sonst die bereits angelegten Adressen und/oder Daten völlig undefiniert werden, was z.B. beim Drücken von CLS im FDXB dazu führen kann, daß plötzlich der halbe Schirm voller Ziffern ist! Dies Problem ließe sich nur umgehen, wenn die Interrupt-Routine feststellt, ob sie gerade eine Bildschirm-Ausgabe unterbrochen hat und in diesem Falle nichts tut.

Eine Beschreibung der 6845-Register habe ich beigelegt; die Register 0..15 werden beim RESET folgendermaßen initialisiert:

```
Reg:   0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
Inh#: 77 50 5C 09 1E 03 18 1B 00 09 60 09 00 00 00 00
```

Dies bedeutet im Einzelnen:

R0:	Eine Zeile dauert #77+1=120 usec
R1:	Horizont. Aufl. = 8*#50 = 640 Punkte
R2:	Zeilenlage #5C = 92 usec
R3:	Sync-Breite = 9 usec
R4,5,9:	Zeilenzahl= (#1E+1)*(9+1)+3 = 313 Zeilen (R9)+1 = 10 Bytes untereinander = 1 Block
R6:	Vertikale Auflösung R6*(R9+1) = 24 Zeilen a 10 Punkte
R7:	Bildlage
R8:	Zeilensprung 00 = aus
R10,11:	Cursor-Block von 0..9 (also 10 Punkte hoch) + Blinken
R12,13:	RAM-Start = #0000
R14,15:	Cursor-Position = #0000 (oben links)

Durch Ändern einiger Register-Inhalte kann man sicher noch ein paar andere Bildschirm-Formate herstellen, aber das ist ein Fall für Hacker, die den RESET-Schalter schon als Fußpedal unter dem Computer-Tisch haben!

Noch eine Ergänzung zur Beschreibung der ESCAPE-Sequenzen für die 80-Zeichen-Karte:

ESC	R	legt das Zeichen und das Attribut-Byte der aktuellen Cursor-Position in #F40B und #F40C ab. (Screen-Peek-Funktion)

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

Fragen und Bemerkungen (Bernd Freusing)

Die neuen Diskettenlaufwerke mit dem 'Druckknopf' sind vom Typ EPSON SD 521, (ds,dd,40Trk) und sind erst seit einigen Monaten auf dem Markt. Ich persönlich finde sie sehr schnell, leise und zuverlässig. Wer nur ein Laufwerk hat (wie ich), kann so ein Ding für DM 596,- incl. Mwst erstehen und braucht dann nur noch ein Stromversorgungskabel, ein paar Schrauben und etwas Geschick. Warum VOBIS dafür fast DM 900 haben will, ist mir schleierhaft!
Ich überlege allerdings noch, ob ich mir nun ein zweites Laufwerk, Silicon-Discs (falls es die endlich mal zu kaufen gibt) oder lieber eine 128- oder 256K-Karte in den MTX einbaue (die man ja bei entsprechender Software auch als SIDISC nutzen kann). Das letztere hätte dann noch den Vorteil, daß man dann auch CP/M 3 fahren kann!
Frage: Wer weiß, wo es SIDISCs oder 256K-Karten gibt?

Das Disketten-Format der FDX ist leider wirklich einmalig; aber nicht verzweifeln: ich bastele gerade ein CONFIG fürs TRS80-Format, dann sind wir die Sorgen der CP/M-Software-Bestellung los, denn das ist tatsächlich ein sehr gängiges Format! Eine genaue Beschreibung unseres Formates ist fürs nächste Info schon in Arbeit.
Frage: Wer kennt die genauen Formate von einigen anderen Rechnern ? (physikalisch, logisch, CP/M-logisch)

Daß die RS232-Karte nicht voll bestückt ist, kann man VOBIS ja noch verzeihen, aber daß da extra ein FPLA drauf ist, das die Nachbestückung verhindern soll, betrachte ich gelinde gasagt als Schikane und Profitgier! Das was da fehlte, habe ich für ca. DM50 nachgekauft, nur leider spielt dieses eine IC nicht mit. Wie ich mit einem 74151 die Adresse dekodiere habe ich schon herausgefunden, nur ist mir absolut unklar, warum auf einem Eingang des FPLA das DTIEO-Signal liegt?
Frage: Wer weiß warum? Wer hat ein Datenblatt des FPLA?

Suche weiterhin Datenblätter des Z80-DART und des CTC.

VDEB.COM (Bernd Preusing)

Ich halte VDEB wirklich für ein exzellentes Werkzeug zum 'debuggen' von Maschinenprogrammen, nur leider stieg das Programm jedesmal aus, sobald ich die Funktionen 'rEad' oder 'Write' aufrief.

Das liegt daran, daß VDEB zum Lesen und Schreiben von Sektoren direkt ins BIOS springt und nicht etwa vorher die Einsprungpunkte berechnet, was ja leicht aus dem Sprung bei #0000 geschehen könnte. Meine Version von VDEB springt jeweils nach #D1xx, während sich mein BIOS bei #E5xx befindet.

Hier nun eine 'Reparatur-Anleitung': (Eingaben unterstrichen)

```
A>DDT VDEB.COM
-S2                 (Speicherstelle 2 ansehen)
0002 E5 .           (hier ist der Beginn des BIOS: merken!)
-S770
0770 D1 E5          (#770 von D1 auf E5 ändern)
0771 xx .           ('.'= ändern beenden)
```

Auf diese Weise folgende Adressen von #D1 auf den Wert von Adresse #0002 ändern:

0770 0771 0794 07C0 07E6 07FC 082D 0838 0841

Wer möchte, kann sich jetzt noch eine deutsche Tastatur herstellen, indem er die Original-Tabelle des Monitors ins VDEB kopiert:

```
-MF198 F239 12BC    (BOOT-Version 3 vorausgesetzt!)

-ctrl-c oder BRK    (DDT beenden und Rückkehr nach CP/M)

A>SAVE 19 VDEBN.COM ('N'eues VDEB abspeichern)
A>VDEBN             (und testen)
```

Noch einige Anmerkungen zu VDEB selbst:

Die gelesenen oder geschriebenen Sektoren sind keine physikalischen, sondern logische Sektoren, da die Operationen ganz normal über das BIOS laufen, d.h. (außer bei CONFIG 08) 26 Sektoren pro Spur a 128 Bytes. Der einzige Unterschied ist, daß kein 'sector skew' stattfindet, den es sowieso nur bei CONFIG 10 (also 8"-sd) gibt.

Nach dem Schreiben von Sektoren sollte immer ein Lesen eines ganz anderen Sektor-Bereiches erfolgen, da sonst evtl. der letzte Sektor nicht auf die Diskette geschrieben wird! Das liegt am Sector-blocking-deblocking des BIOS bei double-density-Formaten, was im normalen CP/M-Betrieb nichts ausmacht, da die letzte Operation nach dem Schliessen eines Files immer ein Directory-Lesen ist.

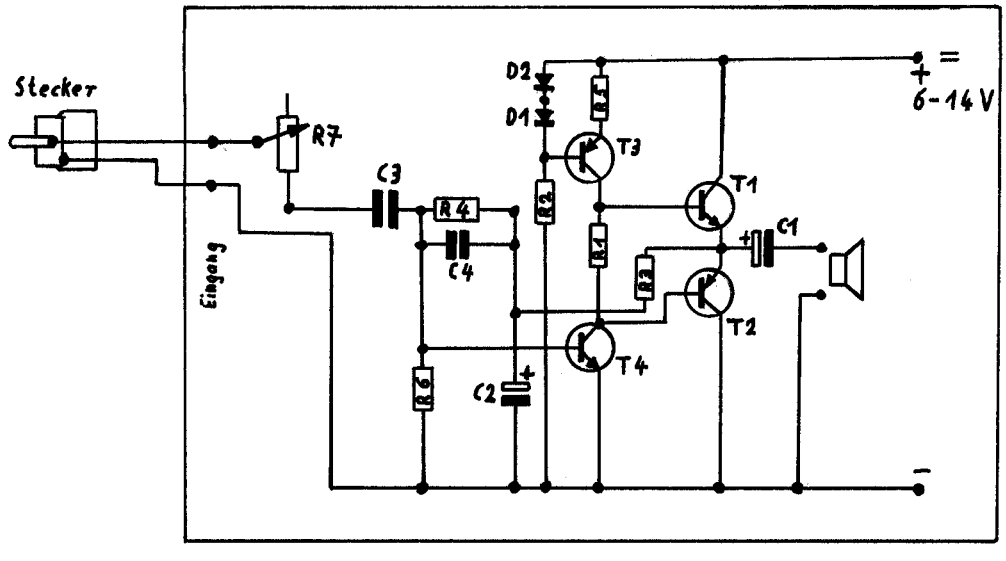
Jede Eingabe in VDEB kann jederzeit durch ESCAPE abgebrochen werden, was manchmal sehr nützlich ist, falls man sich vertippt hat. Die Eingabe 'Number' erwartet eine Hex-Zahl, RETURN und nimmt dann nur noch 'S' oder 'T' an, was zu großem Chaos führen kann, falls der PC nicht sinnvoll gesetzt ist.

LAUTSPRECHERANSCHLUß AN HI-FI TONAUSGANG

Ich betreibe meinen MTX an einem Monitor ohne Tonanschluß und mußte deshalb bei Computerspielen auf den Ton verzichten. Da der MTX jedoch über einen separaten Tonausgang verfügt, habe ich mich nach einer einfachen und billigen Verstärkerschaltung umgesehen. Die unten aufgeführte Schaltung liefert bei weitem keine HI-FI Qualität, dafür kostet sie aber nur ca. 12-15.-DM und erfüllt voll meine Ansprüche.

Stückliste:

Halbleiter:	D1,D2	1 N 4148, (1 N 914 o.ä.)
	T1	BC 140, (BC 141 o.ä.)
	T2	BC 160, (BC 161 o.ä.)
	T3	BC 307, (BC 308 o.ä.)
	T4	BC 237, (BC 208 o.ä.)
Widerstände:	R2,R4,R6	100 K-OHM (br-schw-ge)
	R1	39 OHM (oran-weiß-schw)
	R3	390 OHM (oran-weiß-br)
	R5	22 OHM (rot-rot-schw)
Elektrolytkondensatoren:	C1	470 MF 16V
	C2	2,2 MF 16V
Kondensatoren:	C3	0,15 - 0,18 MF
	C4	680 pF
Drehpoti:	R7	1 M-OHM
Lautsprecher:	0,5-4 Watt,	4-8 OHM



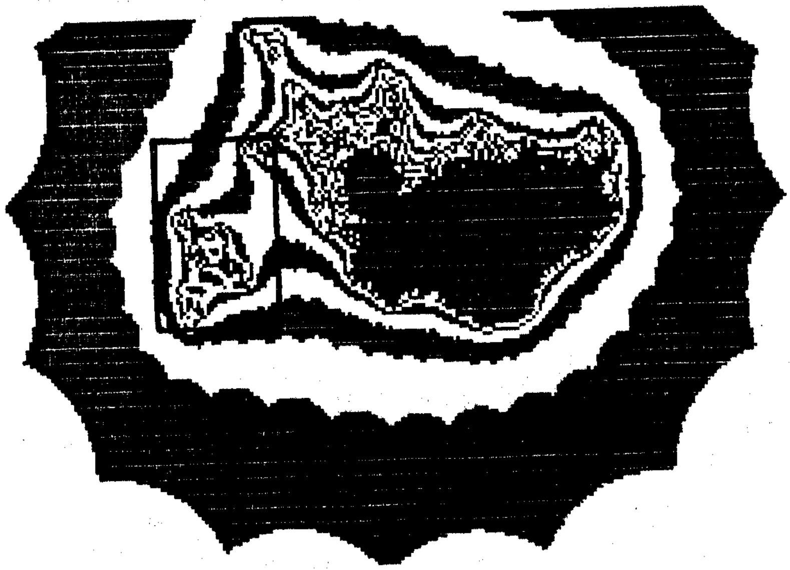
Jürgen Adamczak

P.S.: Viele Elektronik-Fachgeschäfte haben kleine Bausätze und fertige Verstärker, die etwa 2 Watt liefern und die sind ebenfalls geeignet!! Und kosten auch ca. 10-15.- DM.

(Herbig)

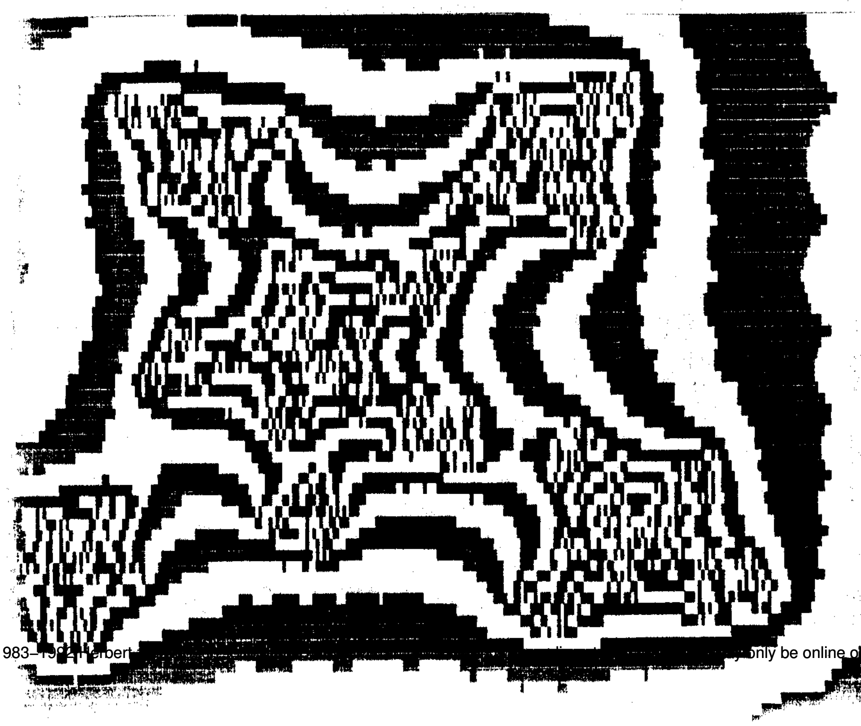
zu fract'1

(T. Pflaum)



READ X1,X2,Y1,Y2 : -4,-1,-0.25,2

: -3.3,-2.8,0.5,1.1



B A S I C — T o k e n s

Uwe Schüler & Herbert Herberg

Das Programm listet die BASIC-Befehle, -Tokens & -Adressen auf

Ausgabe auf den Drucker: PRINT durch LPRINT ersetzen

ACHTUNG: Zeile 1040 nur für FDX-BASIC!! im ROM-BASIC weglassen!

```
0 GOTO 1000
10 REM Umwandlung Dez -> Hex
11 REM ----------------------
20 LET H$="0123456789ABCDEF"
30 IF X<0 THEN LET X$="ERROR": RETURN
40 LET I=X: LET X$=""
50 LET X$=H$(MOD(I,16)+1)+X$: LET I=INT(I/16)
60 IF I>0 THEN GOTO 50
70 RETURN

1000 REM MTX-Befehle
1010 PRINT "Befehl                    Token                              Adresse"
1011 PRINT "-------------------------------------------------------------------"
1012 PRINT
1020 LET ADR=9531: LET JUMP=9975
1030 REM Die Zeile 1040 nur für FDX-BASIC! Im ROM-BASIC weglassen!!
1040 LET ADR=ADR+2*16^3: LET JUMP=JUMP+2*16^3
1100 FOR TOKEN=128 TO 194
1110 LET C=PEEK(ADR): LET ADR=ADR+1
1120 IF C>127 THEN PRINT CHR$(C-128); ELSE PRINT CHR$(C);
1130 IF C<128 THEN GOTO 1110
1140 LET X=TOKEN: GOSUB 10: PRINT ,,"    #";RIGHT$("00"+X$,2);"    ";X;
1150 LET X=PEEK(JUMP)+256*PEEK(JUMP+1): LET JUMP=JUMP+2
1160 GOSUB 10: PRINT "              #";RIGHT$("0000"+X$,4);
1170 PRINT "    ";RIGHT$("     "+STR$(X),5)
1180 NEXT TOKEN
1190 PRINT : PRINT : PRINT : PRINT : PRINT
```

N O D D Y — auf den D r u c k e r

Herbert Herberg & Uwe Schüler

```
10 REM NODDY Ausdrucken
20 REM Uwe Schüler & Herbert Herberg
30 DIM A$(2000): LET K=1
40 INPUT "Ist der Schirm 80 Zeichen breit ? (J/N)";B$
50 IF B$="J" OR B$="j" THEN LET B=80 ELSE LET B=39
90 REM Einlesen auf A$
100 PLOD "test": REM Noddy-Seite ausgeben
110 CSR 0,0: REM Oben links positionieren
120 FOR I=0 TO 23: REM 24 Zeilen
130 FOR J=1 TO B: REM B Spalten
140 LET A$(K)=SPK$: LET K=K+1
150 NEXT : NEXT
190 REM Drucken
200 FOR I=0 TO 23
210 LPRINT A$(I*B+1,B)
220 NEXT
```

In Zeile 100 muß test durch den Namen der entspr. Noddy-Seite ersetzt werden!

Joystickabfrage L (H. Herberg)

```
1 REM Joystickabfrage in Assembler und BASIC: Links
2 REM
3 REM Herbert Herberg
4 REM
5 REM Joystick R (bzw. Cursortasten) siehe 32 Seiten ROM-Unterlagen
6 REM
7 REM Der Joystick L ist auch auf der Tastatur:
8 REM Y,C,B,M = links,rechts,hoch,runter; Leertaste = Feuer
9 REM
10 REM Die gedrückte Richtung ist im C-Register:
11 REM Bit 1-4 = links, rechts, hoch, runter
12 REM ggf. sind mehrere Bits gesetzt
13 REM
14 REM Will man wissen, ob hoch gedrückt ist: BIT 3,B   ;Prüfe Bit 3
15 REM                                       JR NZ,UP  ;Verzweige
16 REM Nur hoch gedrückt:                    LD A,B    ;Hole B
17 REM                                       CP #04    ;Prüfe 'nur Bit 3'
18 REM                                       JR Z,UP   ;Verzweige
19 REM
20 REM Wenn auch Feuer gedrückt ist, ist das B-Register =1, sonst 0
21 REM
22 REM Die Zeile 220 holt die Register B und C mit dem USR-Befehl,
23 REM trennt sie, und gibt sie aus
24 REM
25 REM Zeile 230 macht dasselbe in BASIC
26 REM
27 REM
28 REM Im Info 1 ist ein Druckfehler: Dort steht für hoch 281 statt 251
29 REM
30 REM Achtung: BASIC: Feuer wenn INP(6)=122
31 REM          ASSEM: Feuer wenn INP(6)=10
32 REM
33 REM Die Diskrepanzen kommen daher, daß Ports 5 & 6 Tastaturports
34 REM sind, die regelmäßig wegen der BREAK-Taste angesteuert werden!
35 REM
36 REM

99 GOTO 200
100 CODE

8526          IN A,(5)    ;Lese Port 5 für Richtung
8528          CPL         ;Einsen und Nullen vertauschen
8529          AND #0F     ;Unteren 4 Bits isolieren
852B          LD B,0      ;Lösche Register B
852D          LD C,A      ;Nach Register C
852E          IN A,(6)    ;Lese Port 6 für Feuer
8530          CP 10       ;Feuer ? (in Assembler nicht 122!!)
8532          JR NZ,NOF   ;Feuer nicht gedrückt
8534          INC B       ;Register B = 1 für Feuer
8535 NOF:     RET
8536          RET

Symbols:
NOF     8535

200 CLS : LET K=8*16^3+5*16^2+2*16+6: REM K = #8526
201 CSR 17,10: PRINT "C"," B": CSR 33,10: PRINT "C"," B"
210 FOR J=1 TO 200
220 LET I=USR(K): CSR 16,12: PRINT INT(I/256),MOD(I,256);"   "
230 CSR 32,12: PRINT 123-INP(6),15-MOD(INP(5),16);"   "
240 NEXT
```

```
**********************          ABBA                    Seite 1
* Hallo MTX-Freunde! *
**********************
```

Die Laufgeschwindigkeit der BASIC-Programme zu verbessern ist das Ziel aller Hobby-Programmierer.
Wenn Sie in einem Unterprogramm eine sogenannte Suchschleife eingebaut haben,wird das nachfolgende Beispiel Ihnen sicherlich helfen,die Rechenzeit der Schleife zu verkuerzen.

```
DIM A(1000):    REM Array die durchsucht werden soll
LET POS=-1 :    REM Position des gesuchten Wertes innerhalb der Array
```

In etlichen BASIC-Handbuechern steht unter der Rubrik 'FOR-NEXT' unter anderem:
1. DAS HINEINSPRINGEN IN EINE SCHLEIFE IST NICH ERLAUBT!
2. DAS HERAUSSPRINGEN AUS EINER SCHLEIFE IST ERLAUBT!
Punkt 2.ist aber nicht richtig.
Schauen Sie sich bitte einmal das folgende Programm an.

```
10 LET I=0
20 FOR B=1 TO 1000
30 IF B=3 THEN GOTO 50
40 NEXT B
50 LET I=I+1
60 PRINT I
70 GOTO 20
```

Wenn Sie sich die Arbeit machen und tippen dieses kleine Programm ein, koennen Sie nach dem 255.Schleifendurchlauf feststellen,dass es zu der Fehlermeldung > NO FOR < kommt.
In anderen BASIC-Versionen kann es auch zu der Fehlermeldung > OUT OF MEMORY < kommen.Diese Meldung erfolgt allerdings erheblich spaeter.

Deshalb koennen Sie mit dem MTX bzw.FDX-BASIC maximal 255 ineinander verschachtelten Schleifen aufbauen.Da in diesen kleinen Programm nie eine Schleife beendet wurde,wird jeder Neubeginn wie eine neue Schleife behandelt.Belegt somit wieder neuen Speicherplatz.

Nun zurueck zu unserem Problem,der sogenannten Suchschleife.
Angenommen es wird die Position des Wertes X in der Array A gesucht.

```
1000 REM Unterprogramm SUCHSCHLEIFE  UP1
1010 FOR B=1 TO 1000
1020 IF X=A(B) THEN LET POS=B
1030 NEXT
1040 IF POS=-1 THEN PRINT 'Wert nicht vorhanden.'
1050 RETURN
```

Dieses Unterprogramm arbeitet fehlerfrei.Aber jeder kann nun erkennen: Ist der Wert X gefunden,ist jeder weitere Schleifendurchlauf nur Zeitverlust.

Ein GOTO oder RETURN-Befehl verhindert diesen Zeitverlust.

```
1000 REM Unterprogramm SUCHSCHLEIFE  UP2
1010 FOR B=1 TO 1000
1020 IF X=A(B) THEN LET POS=B:RETURN
1030 NEXT
1040 PRINT 'Wert nicht vorhanden.'
1050 RETURN
```

Dieses Unterprogamm arbeitet jetzt wesentlich schneller,da die unnoetigen Schleifendurchgaenge nicht mehr ausgefuehrt werden.
Aber jetzt besteht die Gefahr dass es wieder zur Fehlermeldung > NO FOR < kommmt.

Das naechste Unterprogramm arbeitet mit einem kleinen Kunstgriff fehlerfrei wie UP1,hat aber die Vorzuege von UP2.

```
1000 REM Unterprogramm SUCHSCHLEIFE  UP3
1005 LET Y=1000
1010 FOR B=1 TO Y
1020 IF X=A(B) THEN LET POS=B:LET B=Y
1030 NEXT
1040 IF POS=-1 THEN PRINT 'Wert nicht vorhanden.'
1050 RETURN
```

Sie sehen nun:Ist der Wert X in der Arry A gefunden,wird zuerst seine Position in der Variable POS gespeichert und anschliesend wird die Schleifenzaehlvariable auf ihren Maximalwert gebracht.

Die Vorteile liegen nun auf der Hand:

1. Keine unnoetigen Schleifendurchlaeufe.
2. Die Schleife wird beendet.

Der Zeitgewinn kann sehr gewaltig sein.Er ist abhaenig von:
a. Wie oft wird das Unterprogramm aufgerufen.
b. Position von X in der Array A.

****
****

KRITIK: Profisoft/Taxan

Ich bedauere es sehr das die Verbindung MEMOTECH/PROFISOFT nicht mehr besteht.Die Kundenbetreuung der FA.PROFISOFT ist vorbildlich und perfekt.Einen spezielen Dank an den Herrn Oelmann.
Ich besitze einen TAXAN-RGB-vision-II Monitor und habe noch niemals irgenteine Fehlfunktion beobachten koennen.Also auch ein Lob an TAXAN.

****
****

Bdos Err On C:Bad Sector

Im Falle eines Bedienungsfehlers an der Disketten-Station kann es vorkommen,dass Sie keinen Diskettenzugriff mehr haben.
Bevor Sie die beiden RESET-Tasten druecken und den Fehler beseitigen, koennen Sie Ihr neues Programm retten ,indem Sie Ihren Cassetten-Recorder anschliesen und mit SAVE 'NAME',Ihr Programm speichern.

****
****

M-BASIC *

In letzter Zeit wurde in FDX-Kreisen immer oeffter ueber M-BASIC gesprochen.

Wann immer ein BASIC-Programm geschrieben wird,ohne HRG oder ASSEMBLER, ist M-BASIC dem MTX bezw.FDX-BASIC in jedem Falle vorzuziehen.

Fragen zu M-BASIC:
* Wie wird die Echtzeituhr gesteuert?
* Wie wird mit der Funktion >EDIT< die Zeilennummer geaendert?

Das nun folgende Listing ermoeglicht jedem der einen Farbmonitor besitzt und demjenigen dem eine M-BASIC Discette zur Verfuegung steht, etliche selbstdefinierte Funktionen.

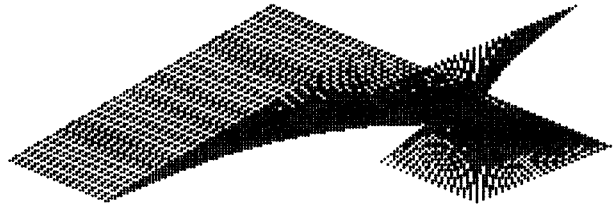
* M-BASIC = Microsoft BASIC (Anm. d. Red.)

```
10  REM ******************** PAPER-Funktionen **********************

11  DEF FNPLI$=CHR$(4):            REM *** Lila
12  DEF FNPRO$=CHR$(6):            REM *** Rot
13  DEF FNPWE$=CHR$(4)+CHR$(7):  REM *** Weiss
14  DEF FNPGR$=CHR$(4)+CHR$(10):REM *** Grün
15  DEF FNPGE$=CHR$(4)+CHR$(11):REM *** Gelb
16  DEF FNPBL$=CHR$(4)+CHR$(12):REM *** Blau
17  DEF FNPHB$=CHR$(4)+CHR$(6):  REM *** Hellblau
18  DEF FNPSC$=CHR$(4)+CHR$(9):  REM *** Schwarz

100 REM ******************** INK-Funktionen ************************

101 DEF FNISC$=CHR$(16):REM *** Schwarz
102 DEF FNIRO$=CHR$(17):REM *** Rot
103 DEF FNIGR$=CHR$(18):REM *** Grün
104 DEF FNIGE$=CHR$(19):REM *** Gelb
105 DEF FNIBL$=CHR$(20):REM *** Blau
106 DEF FNILI$=CHR$(21):REM *** Lila
107 DEF FNIHB$=CHR$(22):REM *** Hellblau
108 DEF FNIWE$=CHR$(23):REM *** Weiss

200 REM ***************** Einfache Sonderfunktionen ******************

201 DEF FNBAN$=CHR$(14):              REM *** Blinkmodus AN
202 DEF FNBAU$=CHR$(15):              REM *** Blinkmodus AUS
203 DEF FNCLS$=CHR$(12):              REM *** Bildschirm löschen
204 DEF FNCAU$=CHR$(31):              REM *** Cursor AUS
205 DEF FNGRA$=CHR$(27)+CHR$(199):REM *** Schachgrafik
206 DEF FNSAS$=CHR$(27)+CHR$(33):  REM *** Spezial ASCI-Code

300 REM ************** Globale Sonderfunktionen ********************

301 DEF FNCRP$(S,Z)=CHR$(3)+CHR$(31+Z)+CHR$(31+S)
    REM *** Cursor-Position
302 DEF FNSZI$(Z,S,X,A$)=FNCRP$(Z,S)+STRING$(X,A$)
    REM *** Erzeugt X-Zeichen(A$) ab Position Z,S
303 DEF FNLZI$(Z,S,X)=FNCRP$(Z,S)+SPACE$(X)
    REM *** Erzeugt X-Leerzeichen ab Position Z,S
304 DEF FNCAN$(F)=CHR$(24)+CHR$(F)+FNCRP$(1,1)
    REM *** Cursor AN (Position 2,1):INK-Farbe F
305 DEF FNDOT$(Z,S)=CHR$(1)+CHR$(31+S)+CHR$(31+Z)
    REM *** Erzeugt einen Punkt auf Position Z,S (Monitor)
306 DEF FNLIN$(Z,S,X,Y)=CHR$(2)+CHR$(31+S)+CHR$(31+Z)+CHR$(31+Y)+
    CHR$(31+X)
    REM *** Erzeugt eine Linie von Z,S nach X,Y

Z.B. PRINT FNSZI$(3,1,20,"*") Erzeugt ab Position 3,1 eine Zeichen-
     kette von zwanzigmal "*"

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+   Binder Alois jun.   (6728)Germersheim I    Tel.07274/3993      +
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

L I N E S . B A S

incl Hardcopy-Routine für DMX 80

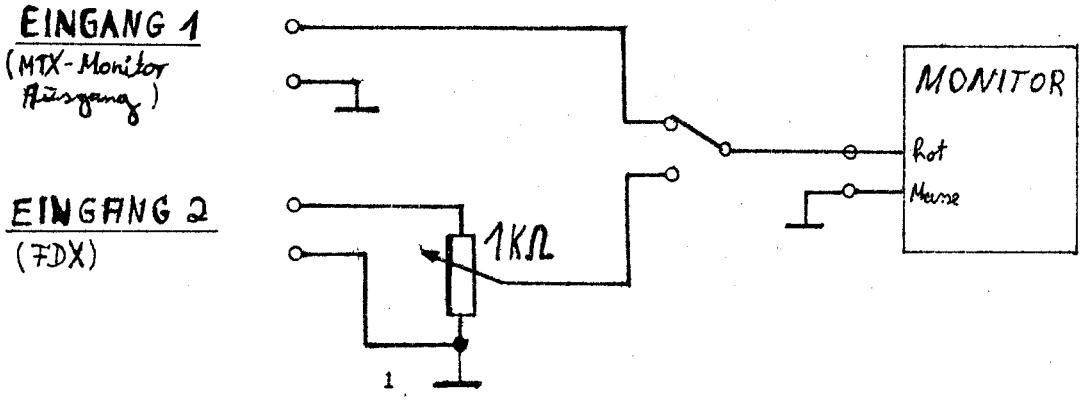
```
10 REM     LINES
20 REM
30 REM  Frank Dersewski  2.5.1984
40 REM
50 RAND -5
100 VS 4: ATTR 3,0: CLS
150 LET Z=0
200 LET X1=50: LET Y1=50: LET X2=30: LET Y2=40
250 GOTO 1000
300 LINE X1,Y1,X2,Y2
350 LET Z=Z+1
360 IF Z=90 THEN GOTO 900
400 LET X1=X1+RX1: LET Y1=Y1+RY1
500 LET X2=X2+RX2: LET Y2=Y2+RY2
510 IF X1>240 OR X1<10 THEN LET RX1=-RX1
520 IF Y1>170 OR Y1<20 THEN LET RY1=-RY1
530 IF X2>240 OR X2<10 THEN LET RX2=-RX2
540 IF Y2>170 OR Y2<20 THEN LET RY2=-RY2
600 GOTO 300
800 REM Nach Erstellung des Bildes warten,
801 REM ob Leertaste gedrückt wird, d.h. Hardcopy
802 REM Sonst und danach: Neue Grafik
900 FOR D=1 TO 100
910 IF INKEY$=" " THEN GOSUB 20000
920 NEXT
1000 LET X1=INT(RND*161)+40: LET Y1=INT(RND*101)+30
1010 LET X2=INT(RND*161)+40: LET Y2=INT(RND*101)+30
1040 LET RX1=INT(-9*RND)+4: LET RY1=INT(-17*RND)+8
1050 LET RX2=INT(-9*RND)+4: LET RY2=INT(-17*RND)+8
1075 IF RX1=0 OR RX2=0 OR RY1=0 OR RY2=0 THEN GOTO 1040
1100 LET Z=0: CLS : GOTO 300
19000 REM ------------------------------------------
20000 REM HARDCOPY NORMAL-RESOLUTION
20100 REM Angepasst für DMX80 Drucker von H. Herberg
20200 LPRINT CHR$(27);"A";CHR$(8);
20300 FOR Y=191 TO 0 STEP -8
20400 LPRINT : LPRINT CHR$(27);"K";CHR$(0);CHR$(1);
20500 FOR X=0 TO 255
20600 LET A=ASC(GR$(X,Y,8))
20700 LPRINT CHR$(A);
20800 NEXT
20900 NEXT
21000 LPRINT : LPRINT : LPRINT : RETURN
22000 DISC SAVE "LINES.BAS"
```



MTX-User Club Allgemeines Frank D. 1

Nun habe ich es also doch noch einmal geschafft,einen kleinen Beitrag fürs INFO 3 fertigzustellen.Bisher habe ich immer nur mit Herbert (HH) zusammengearbeitet, in der Hoffnung, daß meine Ideen so ihren Weg nehmen.(Was wohl auch geschehen ist).Doch genug des Geplauders:

FDX-System: Da ich wohl einer der ersten glücklichen(?) Besitzer des FDX-Systems bin, kann ich inzwischen eine Menge Erfahrung nachweisen(Datum der Erstinbetriebnahme vom MTX-512 mit FDX-Doppellaufwerk,Monitor und Drucker ist der 3.7.1984- genau einen Tag vor meiner kleinen Urlaubsreise). Bis heute kann ich eigentlich nicht groß meckern, bis auf das seltsame Diskettenformat,doch dazu später. Ein einziger Fehler trat nach dem Versuch auf, ein Modemprogramm auf CP/M laufen zu lassen.Auf RS232-1 kam pro Versuch nur ein einziges Byte heraus, danach kam immer die Meldung Buffer voll.Des Rätsels Lösung kam nach mehreren Stunden Suchens mit Ossi und Ohmmeter.Ein kleiner Kurzschluß zwischen SER2 und einem Bein eines anderen IC's. Der CTC lieferte also das Clock-Signal für den DART, nur kam es nicht an.Ich hoffe, das der Rest der MTX von solchen versteckten Fehlern verschont geblieben ist. Noch etwas: Das FDX-System verweigert seine Arbeit bei Temperaturen unterhalb ca. 12 Celsius.Bei winterlichen Transporten und sofortigem Aufbau also nicht wundern. Die Möglichkeit, mit 2 Bildschirmen zu arbeiten, habe ich durch einen einfachen Umschalter in meinem Monitor gelöst. (Zenith-ZVM123e grün).Eine zweite Cinch-Buchse,2 geschirmte Kabel zur Frontplatte und ein normaler Umschalter machen das Umschalten zum Spaß.Ja , wenn nicht der Helligkeitsunterschied zwischen den Bildschirmen wäre.Mein FDX-Monitorsignal hat ca. 5.5 V (0-Spitze), der MTX dagegen nur ca. 2.25 V (alle Spannungen im Leerlauf gemessen). Dagegen hilft ein einfacher Spannungsteiler,siehe Schaltplan.Mit dem Poti läßt sich nun der Pegel des FDX auf die gleiche Helligkeit bringen.Doch Vorsicht!!, auch Monitore führen Hochspannung (ca.14 kV).Vor dem Basteln also tunlichst Gerät ausschalten und Stecker aus der Steckdose;wer seine Garantie nicht verlieren möchte,kann die kleine Schaltung natürlich auch außen aufbauen.



Folgende CP/M -Programme sind bei mir kostenlos zu erhalten, was mich dabei natürlich besonders freut, ist die Tatsache, das keine Copyright-Rechte verletzt werden.

1.MDM 717 Ein Modem-Programm von Irv Hoff (W6FFC) Für Datenübertragung per Telefon und Kabel. Mit Docu, von DK3UZ und mir (DG3LO) auf den MTX angepaßt. (1985 ist eine MTX-Mail-box in Kiel geplant.)

2.Z80FORTH fig-FORTH für den Z80 unter CP/M Überarbeitet von DK3UZ für die FIG (Forth Interest Group Deutschland)

3.SQ-USQ File-Sqeezer und File-Unsqeezer von Dick
XTYPE-14 Greenslaw , Ohio USA Komprimiert und expandiert Text-Files, spart Platz auf Diskette und Geld bei Telefon-übertragung von Text. (z.B. 40K normal ergaben bei mir ca. 20k gesqueezt)

4.RTTY10 Ein RTTY-Programm von DK3NY, modifiziert für den MZ-80K von DK3UZ Ist leider noch nicht auf dem MTX lauffähig aber wer Lust hat, immer ran an die Arbeit.

Kommen wir nun zu den Basic-Programmen, von mir selbst geschrieben, nie 100%-ig , aber ganz nett.

5.POLYN11 Ein Kurvenannäherungs-Programm mit Graphic auf dem VS 4. Man gibt Messwerte ein und erhält die Funktion, die die geringste Abweichung hat; mit Plot auf dem Bildschirm.

6.LIFE Die lieben Bakterien vermehren sich immer noch. Eine Assembler-Version, die in einer Sekunde ca. 11.5 Generationen berechnet. (P.S. Aufforderung zum Wettbewerb: Wer berechnet schneller? mit Generationszähler)

7.KARTEI Datenbank-Programm (noch in Arbeit)

8.HANOI Türme von Hanoi Kleines Spiel zum Nachdenken.

Für DM 25.- gibt't eine Diskette mit o.g. Programmen.
Bitte keine Disketten und keine Briefmarken schicken!

Bei meinem Versuch, ein Datenbank-Programm zu erstellen,mußte ich leider einen erheblichen Fehler im Basic feststellen. Eine sequentielle Datei ließ sich noch einwandfrei abspeichern, was man gut mit TYPE nachprüfen kann, als ich sie jedoch wieder einlesen wollte,bekam ich immer nur den ersten String zu fassen. Doch das Schlimmste war, das der Rest nicht nur unerreichbar, sondern sogar auf der Diskette gelöscht war. Profisoft sagte mir ,ich möge mein Probeprogramm vorbeischicken, was ich aber noch nicht geschafft habe.Die FDXBasic von mir und Herbert Herberg sind auch unterschiedlich lang, meine Version hat 266 Records, Herberts 272. Ein weiterer Fehler ist, daß ich keine längeren Utilities mit DISC RUN laden kann, dafür funktioniert aber mein EOF. Wer Spaß daran hat, kann mein kleines Probeprogramm mal eintippen und schauen, ob es einwandfrei läuft.

```
99 REM    DATEN EINGEBEN UND ABSPEICHERN
100 DIM A$(10,10,20)
200 FOR T=1 TO 6
300 INPUT A$(1,T)
400 NEXT
500 DISC OPEN #1,"TEST.DAT","O"
600 FOR T=1 TO 6
700 DISC PRINT #1,A$(1,T)
800 NEXT
900 DISC CLOSE #1
950 STOP

999 REM   DATEN WIEDER EINLESEN
1000 DIM A$(10,10,20)
1100 DISC OPEN #1,"TEST.DAT","I"
1200 FOR T=1 TO 6
1300 DISC INPUT #1,A$(1,T)
1400 NEXT
1500 DISC CLOSE #1
1600 FOR T=1 TO 6
1700 PRINT A$(1,T)
1800 NEXT
```

Ich habe immer 6 Namen eingegeben und leider nur einen wiederbekommen.(Den Teil ab 1000 mit GOTO 1000 starten) Auf der nächsten Seite noch ein kleines Demo für die Graphik, ganz nett.

Z80 Assembler: Mit den vorgeschlagenen Büchern über die Programmmierung des Z80 bin ich nicht ganz einverstanden. Ich arbeite mit dem Buch von RODNEY ZAKS "Programmierung des Z80", was ich nur empfehlen kann.

Dieses RENUMBER
ändert nicht
GOTO, GOSUB, ...

FL = Nr. d. 1. Zeile
ST = Inkrement, d.h.
immer plus ST.

CSR X,Y : PRINT CHR$(A)

MC-Code: Ausgabe eines Zeichens an gegebener Bildschirmpos.

REGISTER	INHALT
A	Zeichen
BC	Zeile
DE	Spalte

```
1 REM Ausgabe eines ASCII-Zeichens
2 REM an best. Bildschirmposition
3 REM (c) Wolfgang Stelzig
10 CODE

406E            LD BC,10;  y-Koordinate
4071            LD DE,10;  x-Koordinate
4074            LD A,50;   ASCII-Code
4076            PUSH AF
4077            PUSH BC
4078            PUSH DE
4079            LD B,C
407A            LD C,0
407C            INC B
407D            LD DE,40
4080            LD HL,#5C00
4083 ADD1:      ADD HL,DE
4084            DJNZ ADD1
4086            XOR A
4087            SBC HL,DE
4089            POP DE
408A            POP BC
408B            POP AF
408C            ADD HL,DE
408D AUSGABE:LD C,2
408F            DI
4090            OUT (C),L
4092            OUT (C),H
4094            DEC C
4095            OUT (C),A
4097            EI
4098            RET

Symbols:
ADD14083AUSGABE408D
```

```
10 REM **** BYTE PACK ****
20 REM **** NOV 1984  ****
100 CODE

803B CHECK:  LD HL,(#FACC)      ; Check for progra
m in memory.
803E         LD A,H
803F         OR L
8040         RET Z       ; Return if no program to
renumber.
8041         LD BC,(#BF1D)
8045         LD B,0      ;  BC=Step between lines 0
 to 255.
8047         LD HL,(#BF1E)      ; HL=First line nu
mber.
804A RENUMR: LD IX,(#FAAA)      ; IX=Start of basi
c.
804E REPEAT: LD E,(IX+0)
8051         LD D,(IX+1)        ; DE=Line length.
8054         LD (IX+2),L
8057         LD (IX+3),H        ; Poke new line nu
mber into place.
805A         ADD HL,BC   ; Add step size to HL.
805B         PUSH HL
805C         ADD IX,DE   ; IX=Address of next line.
805E         PUSH IX
8060         POP DE      ; DE=Address of next line.
8061         LD HL,(#FAAC)      ; HL=Top of basic.
8064         AND A
8065         SBC HL,DE   ; Subtract line address fr
om top of basic.
8067         JR C,END    ; END if address of line >
 top of basic.
8069         LD A,H
806A         OR L
806B         JR Z,END    ; END if address of line =
 top of basic.
806D         POP HL
806E         JR REPEAT   ; Not finished, renumber n
ext line.
8070 END:    POP HL
8071         RET

Symbols:
CHECK   803B    RENUMR  804A
REPEAT  804E    END     8070

110 REM ********************************
120 REM **** RENUMBER BASIC LINES ****
130 REM ****   MTX 500,512 MICROs   ****
140 REM ****   (c) E.Roy  June.84   ****
150 REM ********************************
```

ROM - Adressen MTX 512

Befehl	Anfangsadresse		Bemerk.	Befehl	Anfangsadresse		Bemerk.
ADJSPR	#15BB	05563		ANGLE	#15FC	05628	
ARC	#16CE	05838		ASSEM	#2784	10116	direkt
ATTR	#15E2	05602		AUTO	#0A07	02567	
BAUD	#0D38	03384		CIRCLE	#180F	06159	
CLEAR	#27FB	10235	direkt	CLOCK	#2AD6	10966	
CLS	#1595	05525	direkt	COLOUR	#15B3	05555	
CONT	#2787	10119	direkt	CRVS	#15EA	05610	
CSR	#1583	05507		CTLSPR	#159B	05531	
DATA	#27AA	10154		DIM	#2837	10295	
DRAW	#1687	05767		DSI	#157C	05500	direkt
EDIT	#2846	10310		EDITOR	#28C5	10437	
ELSE	#06C7	01735		FOR	#2867	10343	
GENPAT	#15A3	05539		GOSUB	#289C	10396	
GOTO	#288F	10383		IF	#2961	10593	
INK	#1590	05520		INPUT	#28DC	10460	
LET	#29DA	10714		LINE	#1574	05492	
LIST	#297D	10621		LLIST	#2941	10561	
LOAD	#2AEE	10990		LPRINT	#2A37	10807	
MVSPR	#15D1	05585		NEW	#0205	00517	direkt
NEXT	#2A64	10852		NODDY	#2B44	11076	direkt
NODE	#277F	10111		ON	#2B4F	11087	
OUT	#09FD	02557		PANEL	#2BF8	11256	direkt
PAPER	#158A	05514		PAUSE	#2BE4	11236	
PHI	#160A	05642		PLOD	#2B32	11058	
PLOT	#1670	05744		POKE	#2C02	11266	
PRINT	#2B88	11144		RAND	#2C09	11273	
READ	#2C74	11380		REM	#27AA	10154	direkt
RESTORE	#2C8C	11404		RETURN	#2C11	11281	direkt
ROM	#009E	00158		RUN	#2CAF	11439	direkt
SAVE	#2B03	11011	**	SBUF	#27B3	10163	
SOUND	#2D0C	11532		SPRITE	#15C4	05572	
STOP	#2D55	11605		VERIFY	#2B1A	11034	
VIEW	#15DA	05594		VS	#15F6	05622	

direkt = direkt vom BASIC aus aufrufbar ohne Parameterübergabe

** aufrufbar mit "PRINT USR(11011),A$"

Bei Eingabe eines Befehlswortes im Direktmodus nimmt die Systemvariable #FD82/83 den Wert der Einsprungadresse ins ROM an.

mit Interruptroutine auslesen, da PRINT PEEK nur die Adresse von PRINT liefert

Siehe auch 32 Seiten ROM-Unterlagen!!
und Programm "BASIC-TOKENS".

TELEVIDEO-FORMAT auf MTX mit 2 Laufwerken

Disketten, die im Televideoformat geschrieben sind, lassen sich lesen, wenn das zweite Laufwerk mit **OVERLAY8** und **CFIG8 C:08** darauf eingestellt worden ist. Man kann dann mit **PIP A:=C:*.*** die gesamte Diskette kopieren.
Dieses klappt (bei meiner Version)
jedoch nur mit Files bis maximal 16kB Größe. Von längeren Files werden nur die ersten 16k kopiert. Das liegt an einem Fehler in der Parameterliste für das Televideoformat.
Er kann folgendermaßen behoben werden:

DDT OVERLAY8.COM

```
-s0f1d <CR>
  0f1d 01 00 <CR>
  0f1e AA ^C
```

Danach mit **SAVE 16 OVERLAY8.COM** die korrigierte Version speichern.

(Olaf Krumnow)

TURBO-PASCAL auf MTX

Turbo-Pascal gibt es leider nicht im MTX-Format. Wer das Pascal haben möchte, muß es im Televideoformat bestellen und dann wie oben angegeben (2 Files sind länger als 16k) kopieren.

(Olaf Krumnow)

TURBO-PASCAL im MTX-Format

Für Clubmitglieder mit nur einem Laufwerk, die auf TURBO-Pascal nicht verzichten möchten, bin ich bereit von einer **Originaldisk** mit Copyrightvermerk BORLAND Int. und Seriennummer eine Kopie in MTX-Format zu ziehen.
Interessenten schicken bitte die Originaldiskette, eine formatierte Leerdiskette und 20.- DM im Schein für P&V und Aufwand.

↑
Keine Billigdisketten!!

(Olaf Krumnow)

1. **MTX/FDX - Monitorausgänge** Wie mir, scheint es auch anderen zu gehen: Der FDX-Monitorausgang ist o.k., aber wenn man mit Disc-Basic arbeitet und Grafiken erzeugt, die man sich wiederum auf dem Monitor ansehen will, dann stellt man fest: hängt man den Monitor nun an den MTX, so ist dessen Ausgangssignal sehr schwach. Die Monitorintensität muß derart hoch gedreht werden, daß Störungen an den Randbereichen auftreten (Streifen, Unschärfen). Nun liegt die Idee ja nahe, mit Hilfe eines Umschalters den Monitor mal am FDX, mal am MTX zu haben. Das Problem liegt jetzt aber an den unterschiedlichen Pegeln. Beim Umschalten müßte man die Intensität ständig nachregeln. Lösungsmöglichkeiten:

1. Den FDX-Pegel abschwächen. Das ist schlecht, da dann die Störungen immer auftreten.
2. Den MTX-Pegel anheben.

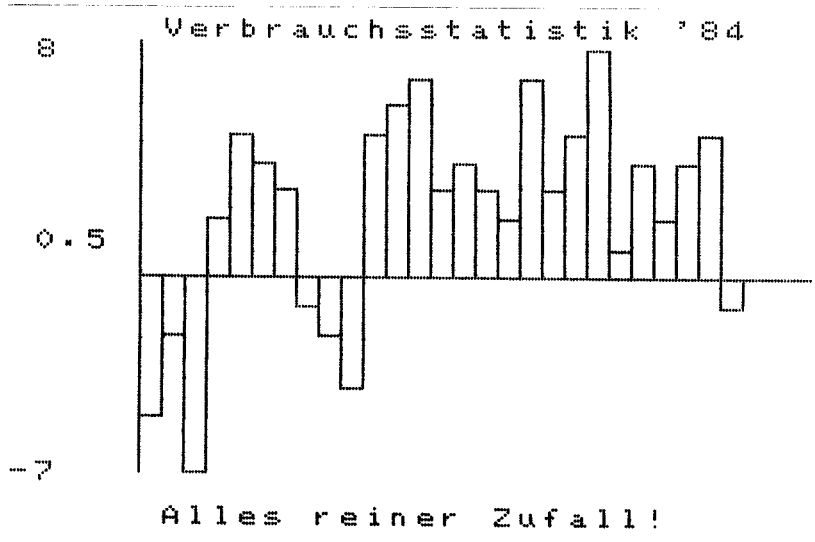
2.1. Im MTX den Pegel anheben. Müßte gehen (sieht man sich den Schaltplan an, ist der betreffende Widerstand auszumachen), aber wer greift schon gerne in das Gerät ein?

2.2. Außerhalb des MTX den Pegel anheben. Dies habe ich gemacht. Bastler können sich den dazu erforderlichen Videoverstärker leicht selber bauen (er sollte regelbar sein, um die Pegel gut anpassen zu können), oder sie können sich in Elektronikshops einen Bausatz kaufen (ca. 20 DM). Ein fertiger Videoverstärker dürfte wohl zu teuer sein (100 DM ?). Man führt das MTX- sowie das FDX-Videosignal in ein kleines Metallgehäuse, verstärkt das MTX-Signal und geht dann an einen Umschalter. Dieser sollte 2xUM sein, da m.E. die Masse unbedingt mit umgeschaltet werden sollte (sonst Gefahr von Masseschleifen). Im Gehäuse ist ein kleines Netzteil unterzubringen (mein Verstärker braucht etwa 50mA; Erdung ans Gehäuse). Wählt man jetzt noch einen griffigen Umschalter, dann kann man zwischen den Ausgängen hin- und herschalten. Mich hat das Ganze etwa 40 DM gekostet.
In das Gehäuse kann man zusätzlich den für die 80-Zeichen-Karte vorgesehenen Lautsprecher einbauen (obwohl der auch gut im FDX-Gehäuse liegen kann).
Wer Näheres wissen möchte, kann sich an mich wenden (z.B. Schaltplan).

2. **Screendump auf DMX** Das entsprechende Basicprogramm mit Assemblereinlage aus "COMPUTING TODAY, 11/84" muß für den DMX etwas verändert werden (andere Steuersequenzen):

1. Für einen X/Y-Ratio von 1:1 muß die Elite-Schrift gewählt werden:
   ```
   32 LPRINT CHR$(27);"P";CHR$(0);
   ```
2. Der Inhalt der Adresse 64161 soll gerettet werden:
   ```
   33 LET NODDYSAV=PEEK(64161)
   ```
3. Um den DMX auf Grafikbetrieb zu schalten muß Zeile 40 wie folgt ersetzt werden:
   ```
   40 LPRINT CHR$(27);"K";CHR$(0);CHR$(2);
   ```
   (standard bit image designation command; column number=512 bytes).
4. Aus dem selben Grund wie bei 3. wird Zeile 63 ersetzt:
   ```
   63 LPRINT:LPRINT CHR$(27);"K";CHR$(0);CHR$(2);
   ```
5. Inhalt der Adresse 64161 zurückschreiben:
   ```
   85 POKE 64161,NODDYSAV
   ```
6. Da das VOBIS-Papier länger ist als die Voreinstellung des DMX, benutze ich folgenden Befehl zur Anpassung:
   ```
   87 LPRINT CHR$(27);"C";CHR$(0);CHR$(12):LPRINT CHR$(27);"N";CHR$(6);
   ```
   Es wird zusätzlich die Perforation mit 6 Zeilen übersprungen.

Eine brauchbare Anwendung der Screendump-Routine ist die Ausgabe von Blockgrafik. Ein derartiges Programm kann jeder von mir haben, der ein nettes anderes Programm zu tauschen hat (max. 70 Balken, Druck normal wie in Abb. oder 4-fach vergrößert).



3. **Pascal für FDX** Das von Memotech angebotene Pascal ist wegen der fehlenden Diskettenunterstützung wertlos. Ich habe JRT-Pascal auf meinem MTX/FDX und dieses bietet alles was begehrt ist, z.B. Diskettenzugriffe (volle Unterstützung der Pascal I/O; Portunterstützung), externe Prozeduren, ein Assembler (Assembler Prozeduren, damit Grafik), Maskengenerator, voller Zugriff auf CP/M 2.2, Zugriff auf sequentielle- bzw. random-Dateien, Debugging für Pascal- und Assemblerprozeduren.
Der absolute Knüller dabei: JRT-Pascal kostete vor einem Jahr ganze $29!!
Anfragen bei Bedarf an mich (Adresse, Bestellmodalitäten).

4. **R232C-Schnittstelle** Vobis bietet diese Schnittstelle für 75 DM zur Nachrüstung von FDX-Systemen an (Austausch der FDX-Karte im MTX). Ich bin damit sehr zufrieden: Die Kopplung des MTX mit einem 68000-System sowie mit einem Heath-Zenith Z80-System klappte recht schnell (trotz der Normung von RS232C-Schnittstellen dauert es erfahrungsgemäß längere Zeit bis die Kopplung auch klappt). Die von uns verwendeten Protokollprogramme kamen allerdings über 9600 Baud nicht hinaus (obwohl sie bei anderen Rechnern noch die 19200 Baud schaffen). Könnte am MTX liegen (?).

---------- Soweit aus Bergedorf ---------- Jens Ebert ----------

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.

SPRITE EDITOR

Dieses, hier als Listing wiedergegebene Programm, unterstützt den Aufbau und die Veränderung von Sprites im Format 16x16 bei Einzelpunktauflösung. Es kann vor jedem beliebigem Programm stehen. Um das Programm durchschaubar zu halten, ist es zum größten Teil in BASIC geschrieben, lediglich die Reservierung des Speicherplatzes für die Sprite-Pattern und die Löschung dieses Bereiches geschehen über ASSEMBLER-Routinen. Das Löschen geht mit dem Befehl RAND USR (17416) schnell und einfach.

Nach dem Laden des Programms werden zunächst alle 32 Pattern ins VRAM kopiert (GENPAT-Befehle). Dies wird auf dem Bildschirm mit "PATTERN XX" angezeigt. Danach startet der eigentliche Editor. Er stellt einige komfortable Funktionen bereit:

- Gezieltes Verändern jedes einzelnen Sprite-Patterns
- Invertieren eines Sprite-Patterns
- Kopieren eines Patterns in ein Anderes (Duplizieren)
- Ansehen aller 32 Sprites auf einen Blick
- Keine zusätzlichen GENPAT-Befehle im Hauptprogramm

Während des Programms wird ständig die Pattern-Nr. angezeigt, das dazugehörige Sprite ebenfalls. Am unteren Rand steht das Editor-Menü. Im linken oberen Teil des Bildschirms erscheint das Sprite-Pattern in 8-facher Vergrößerung. Gesetzte Bildpunkte lassen sich mit F8 (helles Quadrat) oder mit jedem anderen Zeichen erzeugen. Transparente Bildpunkte werden mit der Space-Taste oder durch Überspringen mit Hilfe der Cursor-Steuertasten erzeugt. Besondere Möglichkeiten kann man noch beim DSI-Befehl im Handbuch nachlesen. Nach der Veränderung eines Sprite-Patterns wird durch RETURN die Abtastung ausgelöst. Danach ist das Sprite sofort in der neuen Form verfügbar. Wird bei der Menüsteuerung irgend ein anderes Zeichen als die angegebenen eingetippt, so wird der EDITOR beendet und alle 32 Sprites werden angezeigt. Das Programm wartet dann in einer Endlos-Schleife und muß mit BREAK abgebrochen werden.

WICHTIG: Der EDITOR benutzt die Grafik-Screen VS 4. Wird der Editor aufgerufen, so wird die Grafikseite gelöscht. Also Vorsicht mit Programmen die komplexe Grafiken auf der VS 4 halten! Eigene Änderungen können nach Lust und Laune vorgenommen werden.

```
S P R I T E   E D I T O R
1 CODE

4007            DS 250
4101            DS 250
41FB            DS 250
42F5            DS 250
43EF            DS 24
4407            RET
4408 CLEAR:     LD HL,#4007   ← Startadresse für Löschen
440B            LD C,4
440D LOOP1:     LD B,255
440F LOOP2:     LD (HL),0
4411            INC HL
4412            DJNZ LOOP2
4414            DEC C
4415            JR NZ,LOOP1
4417            RET

Symbols:
CLEAR   4408    LOOP1   440D
LOOP2   440F

2 CLEAR : SAVE "EDITOR"
3 CLS : DIM X(32): CTLSPR 2,32: CTLSPR 6,2: GENPAT 1,135,255,255,255,255,255,
55,255,255
4 FOR S=0 TO 31: CSR 10,10: PRINT "Pattern ";S
5 GOSUB 190: GOSUB 150: NEXT S: GOTO 10
6 FOR A=1 TO 32: LET X(A)=255-X(A): NEXT A: GOSUB 150
7 VS 2: CSR 0,0: FOR A=1 TO 32: LET Q=X(A): FOR B=7 TO 0 STEP -1: IF Q>=2^B T
EN PRINT " USER ";: LET Q=Q-2^B: GOTO 8 ELSE PRINT " ";   'USER' = T8!
8 NEXT B: NEXT A: GOTO 110
10 CRVS 2,1,0,2,16,16,32: COLOUR 4,1: VS 4: COLOUR 0,1: COLOUR 1,10: COLOUR 4
1: PAPER 1: CLS
20 CSR 0,0: INPUT "Welches Pattern soll verändert werden? (alle=-1):";V: IF
=-1 THEN FOR S=0 TO 31 ELSE IF V<32 AND V>=0 THEN FOR S=V TO 31 ELSE GOTO
20
30 CLS : CSR 0,0: PRINT "Sprite Pattern";S: VS 2: COLOUR 0,4: COLOUR 1,15: PR
NT CHR$(29);: CLS : GOSUB 190
40 CSR 0,0: FOR A=1 TO 32: LET Q=X(A): FOR B=7 TO 0 STEP -1: IF Q>=2^B THEN
RINT " USER ";: LET Q=Q-2^B: GOTO 50 ELSE PRINT " ":
50 NEXT B: NEXT A: GOTO 120
60 VS 2: PRINT CHR$(30);: ADJSPR 0,1,S: DSI
70 CSR 0,0: FOR A=1 TO 32: LET X(A)=0
80 FOR B=7 TO 0 STEP -1
90 IF ASC(SPK$)<>32 THEN LET X(A)=X(A)+2^B
100 NEXT B: NEXT A: GOSUB 150.
110 FOR A=1 TO 32: POKE 16390+S*32+A,X(A): NEXT A
120 VS 4: SPRITE 1,S,200,120,0,0,15
130 CSR 0,19: PRINT "(N)ächstes Pattern": PRINT "(V)erändern dieses Patterns"
 PRINT "(I)nvertiere Pattern": PRINT "(S)tart Editor"
135 PRINT "(P)attern-Nr. ändern";: CSR 18,17: PRINT "              ": CSR 18,1
: INPUT A$
140 IF A$="N" THEN NEXT S ELSE IF A$="V" THEN GOTO 60
145 IF A$="I" THEN GOTO 6 ELSE IF A$="P" THEN GOTO 195 ELSE IF A$="S" THE
  ADJSPR 1,1,0: GOTO 10 ELSE GOTO 200
150 GENPAT 4,S,X(1),X(3),X(5),X(7),X(9),X(11),X(13),X(15)
160 GENPAT 5,S,X(17),X(19),X(21),X(23),X(25),X(27),X(29),X(31)
170 GENPAT 6,S,X(2),X(4),X(6),X(8),X(10),X(12),X(14),X(16)
180 GENPAT 7,S,X(18),X(20),X(22),X(24),X(26),X(28),X(30),X(32): RETURN
190 FOR A=1 TO 32: LET X(A)=PEEK(16390+S*32+A): NEXT A: RETURN
195 CSR 18,17: INPUT "Neue Nr. ?";NR: IF NR>31 OR NR<0 THEN GOTO 195 ELSE L
T S=NR: GOSUB 150: CSR 14,0: PRINT S: GOTO 110
200 REM ALLE SPRITEPATTERN ANZEIGEN
210 CLS : FOR A=0 TO 7: FOR B=1 TO 4: SPRITE A*4+B,A*4+B-1,B*30,180-A*20,0,0,
5: NEXT B: NEXT A
220 IF INKEY$="" THEN GOTO 220
230 FOR A=1 TO 32: ADJSPR 1,A,0: NEXT A
240 REM ******START NACH BREAK MIT GOTO 10******
```

AUTOSTART von Programmen

Wie den meisten von Euch von anderen BASIC-Dialekten her bekannt sein wird, startet sich ein Programm nach dem Laden selbst, wenn der SAVE-Befehl im Programm selbst steht. Das Programm startet dann mit dem Befehl nach dem SAVE-Befehl, bzw. mit der nächsten Programmzeile. Allerdings gibt es hier Schwierigkeiten mit dem MTX 512. Nämlich dann, wenn das Programm inkl. Daten länger als 32 kBytes ist. Also dann, wenn der MTX die Speicherbank 0 abschließt und die Speicherbank 1 aktiv wird. Der Autostart klappt dann leider nicht mehr! Egal, ob der SAVE-Befehl am Anfang oder am Ende des Programms steht. Das Spektrum der möglichen Reaktionen beim Laden reicht vom schlichten Nicht-verlassen der LOAD-Routine bis hin zum Absturz des Systems. Eine echte Lösung für dieses Problem konnte bisher leider noch nicht gefunden werden, selbst die Leute bei Profisoft in Osnabrück waren ziemlich ratlos. Allerdings gibt es Möglichkeiten das Problem zu umgehen:

1.) Ist das Programm länger als 32 KB, den SAVE-Befehl nicht ins Programm schreiben, sondern im Dialog ausführen lassen. Der Computer meldet sich nach dem Laden dann wenigstens mit dem BASIC-Ready, beendet also die LOAD-Routine selbsttätig.

2.) Steht der SAVE-Befehl bereits im Programm und läuft der MTX im LOAD weiter, obwohl die Daten von der Cassette zu Ende sind, einfach mit BREAK abbrechen. Das Programm steht dann in den meisten Fällen richtig im Speicher und läßt sich auch normal starten.

Bei beiden Möglichkeiten verliert man aber den Vorteil des AUTORUN, nämlich die Vermeidung eines versehentlich eingegebenen RUN-Kommandos beim Hand-Start. Solche Programme also prinzipiell mit GOTO XXX starten!

Wer eine Lösung für dieses Problem findet, oder schon gefunden hat, möge uns umgehend benachrichtigen, damit wir im nächsten FORUM des Rätsels Lösung veröffentlichen können!

Das der VERIFY-Befehl nur sehr bedingt einsatzfähig ist, dürfte inzwischen bekannt sein. Ein Hinweis darauf steht ja auch im Handbuch. Wer hat eine zuverlässige Lösung gefunden??