

MTX *User-Club Deutschland*

Info 27
15.06.1988

Zweck: Zusammentragen und Austausch von Tips & Tricks u.s.w., Hilfestellung bei allen möglichen Problemen, Aufbau einer Programmbibliothek und Basteln von Hardware-Erweiterungen.

Programme (nur Selbstgeschriebenes): Tausch von kurzen und einfachen Routinen. Gute Programme (mit Dokumentation) können über den Club an alle Mitglieder verkauft werden. Wer solche Programme an uns schickt erhält ggf. Verbesserungshinweise und eine Besprechung im Info.

Mitglied kann jeder werden! Keine Beitragsgebühr! Anmeldung kostet DM 1.-.

Verpflichtungen: Einsendung unseres Anmeldeformulars.

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

Club-Info, unser Blatt, verschicken wir ca. 8-wöchentlich. Inhalt ist alles was uns über den MTX/FDX (ohne Copyright) in die Hände fällt. Es kostet nicht über DM 12.- (90 Seiten) je Exemplar. Jeder kann dazu Beiträge liefern und hier gratis Kleinanzeigen veröffentlichen.

Kosten: Wir berechnen ausschließlich Selbstkosten und verschicken nichts, wenn Ihr persönliches Guthaben nicht reicht! (s.u.)
Schüler, Studenten, Auszubildende, Grundwehrdienstleistende, Rentner und Arbeitslose erhalten einen Nachlaß von 40% auf die zukünftigen Infos nach Einsendung einer entsprechenden Bescheinigung für deren Gültigkeitszeitraum.

Geld/Konto: Für jedes Mitglied führt Herbert zur Nedden ein Konto, von dem die jeweils entstehenden Kosten abgehen. Der Kontostand wird bei jeder Sendung mitgeteilt (er steht über der Anschrift) und kann selbstverständlich jederzeit erfragt werden! Wir verschicken nur gegen Vorkasse!

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

Kontaktadressen: (nach PLZ geordnet)

Herbert zur Nedden Sonnenau 2 2000 Hamburg 76 (040) 200 87 04	Christian Löhrmann Grevenbleck 24 3005 Hemmingen 1 (0511) 41 78 77	Thomas Wulf Roritzer Str. 8 8500 Nürnberg 90 (0911) 33 52 52	Hans Gras Statenhoek 49 NL 1506 VM Zaandam (0031-75) 17 49 91
------------------------------------------------------------------------	-----------------------------------------------------------------------------	-----------------------------------------------------------------------	------------------------------------------------------------------------

Telefon-Sprechzeiten

Herbert zur Nedden: Do 18 - 22 Uhr, Sa 10 - 14.30 Uhr

Inhaltsverzeichnis**C L U B**

Programmvorstellungen und -besprechungen	Seite 2
Korrektur & Nachtrag	Seite 3
Fragen & Antworten	Seite 3
Kurz Notiert	Seite 3

S o f t w a r e

Neuigkeiten	Seite 4
NODE Tool Box	Seite 5
Kopieren schneller ?	Seite 6
Biorhythmus	Seite 19

H a r d - u n d S o f t w a r e

Edicta-Grafikkarte	Seite 24
--------------------	----------

H a r d w a r e

Booten von 09 und 1A und ...	Seite 26
IM2-Interrupts und DMA für den ECB-Bus	Seite 29
Variable Taktfrequenz	Seite 34
8 MegaHertz	Seite 54
8 MegaHertz-Kochbuch	Seite 55

C P / M

Nsweep-Macke	Seite 35
--------------	----------

L e s e r b r i e f

Kurt-Bernd Rohloff	Seite 35
Hans Gras, NL-1506	Seite 36
Hans Gras, NL-1506 / Michael Kessler, 5600	Seite 38
Dieter Ritter, 8520	Seite 39
Andreas Fischer, CH 4303	Seite 40
Stefan Hößler, 1000	Seite 42
Thomas Mäurer, 4200	Seite 44

d B A S E

Tips	Seite 45
Vergleiche	Seite 51

T u r b o - P a s c a l

Binominalkoeffizienten	Seite 51
------------------------	----------

S u p e r C a l c

Spritverbrauch	Seite 52
----------------	----------

Preis für dieses Info: DM 12.-

Redaktionsschluß für Info 28: 15. Juli 1988

Thema Info-Beiträge:

(Herbert zur Nedden, 2000)

1. Bitte schickt mir Druckvorlagen (d.h. schriftliche Beiträge) für's Info auf einseitig bedrucktem weißen Papier, damit der Kontrast besser ist und nichts durchschimmert. Und bitte mit kräftigem Farbband!
2. Gelegentlich erhalte ich auch Neuigkeiten über's Telefon: Ich notiere mir diese Zwar so gut es geht, kann aber nicht alles inhaltlich nachvollziehen, d.h. verstehe nicht immer alle internen Zusammenhänge. Diese Informationen erscheinen dann halt soweit ich sie verstanden habe im Info - also evtl. etwas unklar.
3. Bitte kennzeichnet auf Disketten, die Ihr mir schickt, die Teile, die für's Info vorgesehen sind in irgend einer Form, damit ich nicht beim Versuch aus dem Brief die für's Info gedachten Teile herauszusuchen Teile übersehe.
4. Bitte nennt die Datei auf der Diskette, die ich als erste lesen sollte BRIEF, ohne jegliche Extension wie .DOC oder .TXT.
5. Am besten: Beiträge auf Diskette als NewWord-Dokument, 80-spaltig oder ohne Silbentrennung, Zeichnungen auf weißem Papier!

Liebe Leserin, lieber Leser,

Wie doch die Zeit vergeht - schon wieder sind die paar Wochen Schonzeit um, und das nächste Info wird erstellt.

Nachdem Thomas Wulf (geb. Pflaum) mit dem Club-Info-Inhaltsverzeichnis angefangen hat, und dann Uwe Beythien, Ulrich Hönisch und ich (geb. Herberg) nach und nach dieses Machwerk in der Mache hatten, hat sich nun Peter Würfel dieser Arbeit angenommen. Aus Erfahrung weiß ich, daß das garnicht so einfach ist, da wir versuchen eine gewisse Struktur einzubringen, und Inkonsistenzen schnell auffallen bzw. den betroffenen Artikel überhaupt nicht finden lassen. Außerdem hat Peter eine Liste der Stichworte, die innerhalb der einzelnen Rubriken vorkommen zusammengestellt, die ich um die Seitenangaben der Rubriken ergänzt habe. Das Ergebnis von 2 Wochen Arbeit liegt anbei. (Danke Peter!)

Die Laufwerke der Sammelbestellung sind längst bestellt. Leider dauert die Lieferung - insbesondere TEAC FD 55 GFR noch etwas! (Stand 30.05.88: noch 10 Tage für GFR, HFN müßten jeden Tag kommen!) Wer sucht eigentlich HD-Disketten für die 3 1/2"-Lw ? Unlängst wurde ja beschrieben, wie den TEAC FD 55 GFV die 1.76 MB beigeputt werden können. Wer will das für sein FD 55 GFR haben ? Das Technical Manual mit Schaltplan kostet DM 150.- (evtl. + MwSt). Evtl. könnten sich die Interessenten zusammen tun - jeder gibt ein Schärfflein dazu - und einer tüftelt die Lösung mittels Manual aus!

Wie Ihr sicherlich schon gesehen habt, verwende ich nun im Info eine schmalere Schrift (.cw10), und erhalte dadurch natürlich auch mehr Zeichen in der Zeile. Seit ich unter NewWord mit 96 Zeichen je Zeile auf dem Bildschirm arbeiten kann, war die Umstellung des Infos auf 80 Zeichen je Zeile an Stelle der bisherigen 70 kein Problem.

In Anbetracht einiger Mißgeschicke des Infos 26 muß ich mal wieder das Thema Beiträge zum Info aufgreifen, was ich unter dem Inhaltsverzeichnis getan habe.

Es wurde mehrfach geäußert, daß die im Club angebotenen Programme etwas mehr Programmbesprechung und eine teilweise ausführlichere Vorstellung bzw. Ankündigung vertragen würden. Dieses Thema findet Ihr auf der nächsten Seite. Übrigens erstellt Olaf für Seine KLICK-PD's eine PD-Beschreibung, die in Info 28 kommt. Dank seiner Stationierung bei der Bundeswehr in Husum geht's nicht schneller!

Das nächste nordische Clubtreffen, vermutlich in Hemmingen, organisiert von Christian Löhrmann, wird im (August)/September 1988 stattfinden. Leider geht das nicht früher! Wer von Euch Lust hat, daran teilzunehmen, möge es mir bitte möglichst früh mitteilen und auf zeitliche Einschränkungen in diesen Monaten hinweisen. Wir wollen wieder versuchen einen Raum incl. Kaffee und Mittagessen für ca. DM 25-30.- pro Person zu bekommen.

Als ich die letzte Sammlung von CLUB-PD's fertig hatte, und die Kopien für die PD-Abonnenten erstellte, mußte ich feststellen, daß meine Briefumschläge zur Neige gingen -und es war Samstag. Also ging ich her, und habe gebrauchte Umschläge verwendet. Aus dieser Not habe ich nun eine Tugend gemacht, und verwende seit dem Tag immer häufiger gebrauchte Umschläge - auch, wenn ich sie evtl. erst etwas mit der Schere zurechtstutzen muß. Immerhin spare ich so Geld, und entlaste die Umwelt etwas.

Leider scheint es beim Assembler-Kurs Probleme zu geben: Kurt-Bernd Rohloff hat keine Artikel für mich. Als Lückenfüller möge mein Assembler-Listing eines Kopierprogrammes dienen, welches recht flott ist ...

Da ich dank des sommerlicheren Wetters Samstags nicht bis 16.00 Uhr zu hause bleiben möchte, habe ich die Sprechstunde am Samstag auf 10.00-14.30 vorverlegt (allerdings bin ich ggf. mal eine Stunde zwischendurch einkaufen!).

Wer sucht noch eine Quelle für Panasonic-Druckerfarbbänder, d.h. für die guten für unseren DMX 80 ? Dank Peter Lang kann ich diese Suche einstellen; die Farbbänder gibt's bei (Preisstaffeln an 10 und 50)

Weber Elektronik, Eisenbahnstr. 53, Würzburg, Tel. 0931 - 704091

Eur

[Handwritten signature]

CLUB: Programmvorstellungen und -besprechungen**Programmbesprechungen**

(Herbert zur Nedden, 2000)

Leider erscheint im Info immer seltener eine 'überparteiliche' Besprechung (Lob und/oder Kritik) für Programme, die angeboten werden.

Klar doch, daß der Autor sein eigenes Programm im Info in den höchsten Tönen lobt - nur übersieht oder verschweigt er leicht mal etwas. Für viele ist es sehr wichtig zu wissen, unter welchen Systemvoraussetzungen das Programm läuft. (Benötigt es RAM 4, oder gar FDXB - oder ist es nur für den Cassette gedacht). Ich kann mich noch gut an den Unmut einiger NLQ-Besteller erinnern, die das Programm garnicht einsetzen, weil es FDXB benötigt! Wie mag das nur denen gehen, die ein Programm auf Diskette erhalten, und dies weder mit dem Cassetten-Recorder noch dem CD-Player einlesen können?

Und dann gibt es noch diese vielen Public-Domain Disketten, auf denen (vereinzelt) richtig gute Programme zu finden sind. Ich habe nicht die Zeit jede PD gründlich durchzuforschen, um all diese Leckerbissen aufzustöbern. Außerdem sind die Interessen & Geschmäcker verschieden. Auch bei den CLUB-PD's teste ich die Programme nicht immer alle aus, sondern halte mich i.a. an die Dokumentation und einen kurzen Test.

Es stellt sich daher die Frage, wie diesem Übel abzuhelpen ist, und mein **Vorschlag** lautet:

- Für PD's erhält ab sofort jeder die CLUB.000, d.h. die Diskette, auf der die Inhaltsverzeichnis-Dateien aller PD's stehen gratis gegen Einsendung einer Diskette mit selbstadressiertem Freiumschlag. Außerdem lege ich diese Scheibe bei Bestellungen ab 10 PD's gratis anbei.
Übrigens kostet eine Diskette als Warensendung (bei richtiger Verpackung) genauso viel Porto wie ein Brief: DM 0.80.
(In den Inhaltsverzeichnis-Dateien stehen die Datei-Namen, und dazu eine mehr oder weniger ausführliche Funktionsbeschreibung.)
- Wer auf einer PD oder sonstwo ein 'habenswertes' Programm entdeckt, möge dies doch bitte im Info kundtun - am besten gleich mit einer kleinen Programmbeschreibung.
- Wenn Du sogar ein Programm im Club gekauft hast, raff Dich doch bitte auf, mir mal Deine Erfahrungen mit diesem (sei es Lob oder Kritik) zu schicken, damit ich sie ins Info aufnehmen kann.

Ich denke da noch an den Slogan, auf den Manfred Flume und ich am Telefon kamen: 'Ein Programm, über das nichts außer der Ankündigung des Autors zu lesen ist, kann so toll nicht sein!'

Da ich dieses Thema nun schon beim Wickel habe, möchte ich auch gleich etwas zu neuen Programmen sagen:

Viele, die unter RAM 4.x arbeiten lassen sich die Uhrzeit 8-stellig in der rechten oberen Ecke des Bildschirms anzeigen. Bitte tragt diesem Umstand beim Design Eurer Bildschirme Rechnung, und schreibt dort nur unwichtiges (wie Leerzeichen) hin. Es ist zwar richtig, daß alle RAM4-Benutzer oft mehr als 24 Zeilen auf dem Bildschirm haben, aber noch mehr Clubmitglieder RAM 4 nicht benutzen (schließlich kostet eine Speichererweiterung auch Geld). Versucht doch daher entweder die Programme installierbar zu machen, oder beschränkt Euch auf 24 Zeilen - es sei denn, daß das Programm eh nur unter RAM 4 Sinn macht. Auch die vielen Bildschirm-Steuercodes, die es erst mit RAM 3 oder RAM 4 gibt, hat nicht jeder! Übrigens muß ein Programm nicht unbedingt installierbar sein, Ihr könnt auch einfach mehrere Versionen erstellen, um evtl. den verschiedenen Systemumgebungen Rechnung zu tragen (mein MTX-Edit liefere ich z.B. gleich in 3 Versionen aus: MTX, RAM3, RAM4).

Und gebt bei Euren Programm-Ankündigungen das Geheimnis preis, in welcher Sprache das Teil geschrieben wurde, und unter welchen Bedingungen es läuft.

C L U B: Korrektur & Nachtrag / Fragen & Antworten / Kurzes**Korrektur & Nachtrag****Edicta-Grafik-Software:**

(leider sind einige Patches erforderlich:)

EDICTA.COM 4F45H 40H (alt 80H), 460EH & 460FH FFH (alt 00H)
 EDI*.CMD 4931H 40H (alt 80H), 3FFAH & 3FFBH FFH (alt 00H)

Fragen & Antworten**F: NewWord (Hartmut Traber, 5270)**

Beim Schreiben dieses und früherer Briefe fiel mir auf, daß unter NW mit Druckkommando's Spalten schlecht dargestellt werden.

Die Druck-Steuerzeichen von NW muß man immer in der Spalte abziehen. Gibt es da eine Lösung ?

F: Schneider Formate (Peter Würfel, 7262)

Wer kann mir möglichst ausführliche Informationen über die verschiedenen Schneider Disketten-Formate geben ?

F: M80 / L80 (Herbert zur Nedden, 2000)

Früher gab es MBASIC samt M80/L80 für unter DM 250.-, jetzt kostet der Kram wieder stolze 1000,-.

Wer weiß, wo es das Teil günstiger gibt, oder will seine Lizenz verkaufen ?

F: M80/L80 und Tubro-Pascal 3.0 (Anton Reiser, 8977)

Suche diese beiden Programme BILLIG!, aber als Lizenz!

A: Info 26, Seite 47, Frage von Holger Göbel: (Jan Bredereke, 2000)

Floating-Point-Arithmetik in Assembler ist schon mal auf CLUB.028.

Zum C-Compiler gehören Floating-Point-Routinen, die alles bieten, was das Herz begehrt, einschließlich transzendenten Funktionen, ...

Die Routinen sind dort einfach mit #asm in den C-Quelltext eingebettet, also sehr leicht herauskopierbar.

A: Info 26, Seite 3, Frage von Uwe Grass: (Jan Bredereke, 2000)

nach den falschen Dateigrößen berechnenden XDIR3:

Laut c't 5/88, S. 52 handelt es sich um eine CP/M-Schwäche. Bei 512 k Dateigröße läuft im Directory-Eintrag das Extent-Byte über, aus dem Programme vorzugsweise die Dateigröße berechnen. Der Überlauf erfolgt anscheinend in das 15. Byte des Eintrags, dieses ist aber undokumentiert. Bei der Suche nach dem letzten Extent mit 'Search Next' erkennt das BDOS weitere Einträge über 512 k nicht. Was das P2DOS von RAM4 dazu sagt, weiß ich nicht und kann ich mangels so großer Scheiben auch nicht ausprobieren.

Kurzes**Turbo-Pascal 3.0**

(Herbert zur Nedden, 2000)

Hier stehen die Integer-Routinen: Mult 6F5 - 70E

Div 70F - 73A

Mod 745 - 74D

Neg 783 - 78A

TEAC FD 55 GFR und 1.76 MB

(Herbert zur Nedden, 2000)

Bislang ist hier die Lösung noch nicht gefunden! Firma NBN bietet ein technisches Handbuch (mit Schaltplan e.t.c.) für DM 150.- (einhundertfünfzig) an, mit dessen Hilfe es sicherlich zu lösen wäre. Wenn genug Interessenten da sind, würde ich das Handbuch bestellen, und jeden an den Kosten beteiligen. Meldet Euch daher bitte, wenn ...

KLA & PLAN

(Manfred Flume, 6500)

Meine Programme

PLAN: VS 4-CAD und

KLA: Klassenarbeitszusammenstellungsprogramm

laufen nur unter RAM 4 wegen RAM-Disc = schnell (PLA) bzw. Windows (KLA)

Software: Neuigkeiten**ALARM**

(Holger Hansen, 3300)

ALARM ist ein Public-Domain-Programm, welches eine Datei mit Terminen (Datum + Anzeigetext) einliest, und diese Anzeigt. Außerdem kann man mit ALARM natürlich auch die Termin-Datei editieren. Was ALARM tut, und wo es die Termini-Datei sucht, wird beim Aufruf in der Kommandozeile angegeben.

ALARM ist nur sinnvoll auf P2DOS/RAM4.xx Systemen mit Hardware-Uhr. Aber auch mit Software-Uhr kann man arbeiten, wenn man DATE4 vor ALARM aufruft.

Ann.d.HzN.: Wenn nun noch ALARM von Holger mit WECKER von Jan (KLICK.003) zusammengeschmissen wird, wäre das eigentlich toll.

Aber Holgers ALARM hat den bedeutenden Vorteil auch auf anderen CPU-getriebenen Systemen, wie PC's zu laufen, da es keine besonderen Anforderungen an das System stellt - wohl aber ohne System-Datum recht dümmlich aus der Wäsche schaut.

DND = Disk Based Named Directory

(Michael Keßler, 5600)

DND ist ein kleines Programm zum Erzielen einer besseren Übersichtlichkeit bei großen LW-Formaten (wie z.B #1B). Unabhängig von NAMES.DIR und den damit verbundenen Hilfsprogrammen wie LD, PWD etc. kann damit unter Ausnutzung der verschiedenen Userebenen ein komfortables Hauptdirectory/Subdirectory - System aufgebaut werden. DND analysiert nur die jeweils eingeloggte Diskette, auch ein Diskettenname kann vergeben werden.

Wie wird DND benutzt ?

Aus User 0 wird eine Datei mit dem Namen der Diskette, und zusätzlich in jedem User-Bereich eine Datei mit dem Namen des User-Bereiches angelegt. Diese Dateien belegen natürlich nur 0 kB, und werden zu System-Dateien erklärt (STAT datei \$SYS), damit DIR sie ignoriert.

Wird nun DND aufgerufen, wird der Name der Diskette und aller User-Bereiche angezeigt, und man kann einen User-Bereich auswählen, und sich dessen Directory anzeigen lassen oder auch diesen einloggen.

Anwendung mit RAM4:

Bei mir ist das ZCPR2-System wie folgt organisiert:

Nach dem Booten lade ich mit LD NAMES **einmalig** ein Named-Directory, welches max. 10 Einträge besitzt (nur MEMORY-BASED). Da ich stets auf A0: arbeite und meine Systemutilities stets auf A1: sind, sieht mein NAMES.DIR wie folgt aus:

A0: = WORK B0: = B C0: = C D0: = D E0: = E
A1: = SYS

Auf der User-Ebene 0 einer JEDEN Diskette befindet sich bei mir nur ST.COM (Inhalt: A1:DND). Wenn ich auf Drive B zugreifen will, gebe ich CD B ein, dadurch wird (über ST.COM) DND in A1 aufgerufen, das Haupt-directory der Diskette in B: erscheint und ich habe nun die Möglichkeit, sofort in das gewünschte SubDir einzuloggen oder mir erst die Inhalte eines oder mehrerer SubDirs anzusehen.

Software: NODE Tool Box**MEMOTECH NODE TOOLBOX**

=====

This software has been developed by Eugen Kaschubinsky, Holland for those who own a standard Memotech and want to make more out their machine.

This package offers about **20 extra commands**. The NODE TOOLBOX is available in ROM-version with the handy ROM-case to connect at the left cartridge port.

The NODE TOOLBOX is booted with the command ROM 7 and you can use these extra commands:

NODE COPY : copies a part of the RAM to VRAM.
NODE DUMP : dumps the screen (text or graphics screen) to the printer (DMX80 or EPSON).
NODE FLSH : flashes the text screen.
NODE FREE : determines the amount of free memory left for BASIC.
NODE HELP : displays the HELP page.
NODE INVT : inverts all lowercase characters.
NODE NRML : restores the normal character sets.
NODE NUMB : renumbers the BASIC program (include GOTO, GOSUB and RESTORE!).
NODE PAND : performs logical AND on two given HEX numbers.
NODE PDEC : displays the DEC value of the given HEX number.
NODE PHEX : displays the HEX value of the given DEC number.
NODE POR_ : performs logical OR on two given HEX numbers.
NODE PROG : send and receives a BASIC program from a remote computer via the RS232.
NODE PXOR : performs logical XOR on two given HEX numbers.
NODE QUIT : hooks off all NODE commands
NODE TAPE : allows you to save and load a portion of memory to a cassette.
NODE TIME : displays the time in the right bottom of VS 5.
NODE TOFF : disables the clock.
NODE VRAM : copies a piece of VRAM to RAM.
NODE ZERO : installs a slashed zero and a copyright sign.

The software is designed to run in high memory and this means that you can use NEW and CLEAR without destroying the code.

The software does not run on disc based system. The FDX-users must type A>MTX to use the extra commands.

This ROM-pack with manual costs only **DM 50,-** (carriage included):

D'Hondt Erik
 Wilgstraat 25A
 B-9440 EREMBODEGEM
 Belgium

tel.: 053/77.44.97

Software: Kopieren schneller ?

1440 MB (Format 1B) in unter 3 1/2 Minuten oder
390 kB (Format 09) in 80 Sekunden kopiert

(Herbert zur Nedden, 2000)

'Warum habe ich dieses Programm nicht schon früher geschrieben ???' habe ich mich gefragt, als ich sah, wie schnell ganze Scheiben gleichen Formats von einem Laufwerk auf ein anderes kopiert werden können - und zwar ALLES, d.h. mit Systemspuren, allen User-Bereichen, und unter RAM4 auch mit Zeiteinträgen!

Es ist sicherlich kein Problem, COPYD auch für den Betrieb mit nur einem Laufwerk anzupassen. Es muß lediglich bei den Labels 'WriteBfr:' und 'GoBack:' die Aufforderung die Diskette zu wechseln ausgegeben, und ein Tastendruck abgewartet werden, falls Quell- und Ziellaufwerk gleich sind. Natürlich muß bei 'Parse:' dann die Abfrage, ob die Laufwerke gleich sind entfallen. Allerdings dauert dann der Kopiervorgang deutlich länger, da die Diskettenwechselzeiten hinzukommen.

Geschrieben habe ich COPYD, weil Olaf Krumnow bei der KCLICK.003-Public-Domain darauf bestand, daß die Dateien auf 3 User verteilt werden sollen, um die verschiedenen Dinge auseinander halten zu können. MDC kann das nicht, NSWEEP und MCOPY sind zu langsam!

Da erinnerte ich etwas aus Info 2 - lange lange ist's her. Da ist so ein Listing eines Programmes, welches spurweise kopiert. Also kurz mal angesehen - nein doch genauer hingesehen, und dann unnötigen Krempel weg, dafür jedoch optimiert in Sachen Geschwindigkeit.

Das Resultat ist COPYD, welches schnell ganze Disketten gleichen Formats vollständig kopiert, d.h. mit Systemspuren, und allen User-Bereichen.

Übrigens benötigt das Kopieren einer 03-er Diskette mit seinen 320 kB länger als eine 09-er mit 390 kB, da das 09-er Format größere physikalische Sektoren hat, und die Verteilung der Sektoren (Sektorverschränkung) einer Spur auf der Diskette besser ist.

COPYD gibt nach jeder Spur die Anzahl der gelesenen Sektoren aus. Würde diese Meldung nach jedem gelesenen Sektor ausgegeben, wird COPYD wesentlich langsamer, da dann zu viel Zeit zwischen dem Lesen aufeinanderfolgender Sektoren vergeht, um innerhalb einer Diskettenumdrehung mehrere Sektoren zu lesen - schließlich wurden die Sektoren für einen schnellen, und nicht einen langsamen Zugriff innerhalb der Spur verteilt.

Nun zum technischen Ablauf:

Eigentlich funktioniert das Spurweise kopieren einer Diskette ganz einfach:

Spur = 0; Sektor = 1;

Loop: Lies Sektor von Quelldiskette

Wenn Puffer voll:

Schreib Puffer auf Zieldiskette

Wenn Ende der Quelldiskette:

Schreib Puffer auf Zieldiskette (falls nicht schon leer)

Goto Ende

Positioniere auf nächsten Sektor

Goto Loop.

Ende: Tschüß

Software: Kopieren schneller ?

Und genau so kann man es tun, was leider bei 'Einfach-drauf-los-Programmierung' noch lange nicht der schnellste Weg ist! Bei dieser Lösung dauert das Schreiben etwa 2 1/2x so lange wie das zum Lesen der Sektoren. Die Ursache für dieses Phänomen ist, daß CP/M 2.2 nur an Sektoren von 128 Bytes Länge glaubt:

Ein Sektor auf der Diskette hat nicht nur Platz für 128 Bytes, sondern oft für 256, 512 oder gar 1024 Bytes, da sonst deutlich weniger Daten auf der Diskette Platz hätten, weil zum einen zwischen zwei physikalischen Sektoren Platz bleiben muß, und zum anderen zu jedem Sektor auf der Scheibe Verwaltungsinformationen stehen (u.a. eine Prüfsumme).

Um die Sektoren auf der Diskette von denen, an die CP/M glaubt zu unterscheiden, nenne ich die auf der Diskette physikalische, und die 128-Byte-Sektoren von CP/M logische Sektoren.

Da CP/M 2.2 wie gesagt nur 128 Bytes lange Sektoren mag, hat das BIOS die ehrenvolle Aufgabe, das sogenannte Blocking/Deblocking, d.h. das Umsetzen von logischen (128-Byte-Sektoren) in physikalische Sektoren (128, 256, 512 oder 1024 Bytes-Sektoren) und zurück vorzunehmen.

Zu diesem Zweck befindet sich im BIOS ein Sektor-Puffer, in den ein physikalischer Sektor hineinpaßt, der für das Blocking/Deblocking dient.

Bei Lesen eines CP/M-Sektors (also 128 Bytes) geht das BIOS wie folgt vor:

1. Lesen des entsprechenden physikalischen Sektors, falls dieser nicht im Puffer steht.
2. Extrahieren der geforderten 128 Bytes aus dem Puffer.

Beim Schreiben eines Sektors muß das BIOS natürlich sicherstellen, daß nur die an das BIOS übergebenen 128-Bytes (CP/M-Sektor) auf die Diskette kommen, jedoch der Rest des physikalischen Sektors unverändert bleiben.

Dazu geht das BIOS wie folgt vor:

1. Lesen des entsprechenden physikalischen Sektors, falls dieser nicht im Puffer steht.
2. Einfügen der geforderten 128 Bytes in den Puffer.
3. Zurückschreiben des Puffers, falls ausdrücklich verlangt, jedoch spätestens beim nächsten Warm-Boot, oder beim Lesen eines anderen Sektors von Diskette (da dann ja der Puffer dafür benötigt wird).

Daher erfordert ein Schreibzugriff auf eine Diskette i.a. zuvor einen Lesezugriff, was die Sache verlangsamt. Wenn man nun noch den Verwaltungsaufwand für das Blocking/Deblocking mitrechnet, erntet man (wie oben angedeutet) einen deutlich langsameren Schreibzugriff als Lesezugriff.

Da ich jedoch bei meinem Kopierverfahren, d.h. beim Schreiben ganzer Spuren in einem Rutsch eigentlich auch die Daten eines physikalischen Sektors in einem Zug bereitstelle, wäre es doch eigentlich toll, wenn das BIOS diesem Umstand Rechnung tragen könnte, und beim Schreiben glauben würde, daß es alle Daten für den physikalischen Sektor vom Programm auf einen Schlag erhält, sich also um den alten Inhalt der Diskette nicht kümmern müßte (weil eh alles neu geliefert wird). Es müßte ja nur warten, bis alle Daten des physikalischen Sektors in Häppchen von 128 Bytes eingetroffen sind, um dann den Sektor wegzuschreiben.

Das gleiche in kurz: Es soll einfach nicht auf der Zieldiskette lesen!

Die Macher von CP/M haben sogar schon an diesen (und einen anderen) Umstand gedacht: Man dem BIOS beim Aufruf der BIOS-Funktion SektorSchreiben im C-Register eine Verhaltensregel mitgeben:

Software: Kopieren schneller ?

- C=0 bedeutet, daß der Sektor sofort auf die Diskette zu schreiben ist, was z.B. bei Directory-Sektoren auf jeden Fall erforderlich ist, wobei ggf. zuvor von der Diskette der entsprechende physikalische Sektor gelesen wird, um die Richtigkeit der Daten im Puffer sicherzustellen,
- C=1 bedeutet, daß der Sektor in den Puffer zu stellen ist, wobei ggf. zuvor von der Diskette der entsprechende physikalische Sektor in den Puffer gelesen wird, um die Richtigkeit der Daten im Puffer sicherzustellen, es jedoch ausreicht, wenn der Sektor erst dann auf die Diskette geschrieben wird, wenn der Puffer für einen anderen physikalischen Sektor benötigt wird.
- C=2 bedeutet, daß der Sektor in den Puffer zu stellen ist, jedoch weder zuvor sichergestellt wird, da der Puffer richtig initialisiert wird (d.h. kein Lesezugriff) und der Puffer auch nicht sofort auf die Diskette wandert.

Klarerweise ist dies der schnellste Weg Daten auf Diskette zu schreiben, allerdings muß durch den Anwender sichergestellt werden, daß die richtigen Daten im Puffer stehen, bevor dieser auf die Diskette kommt.

Das ganze in Kurz:

- C=0 sofort auf Diskette schreiben, mit Blocking/Deblocking
 C=1 gepuffert auf Diskette schreiben, mit Blocking/Deblocking
 C=2 gepuffert auf Diskette schreiben, ohne Blocking/Deblocking

Bei beiden zweiten Schreibzugriffen (C=1, C=2) bleibt der physikalische Sektor im Puffer stehen, bis entweder der Sektor-Puffer anderweitig benötigt wird, oder explizit das Wegschreiben des Sektor-Puffers auf Diskette verlangt wird.

COPY-DISC verwendet diese Möglichkeit des 'ungeblockten Schreibens' (C=2), also den schnellsten Weg. Dies ist daher zulässig, daß sichergestellt wird, daß beim Kopiervorgang immer ganze physikalische Sektoren auf ein Mal transferiert werden, d.h. bei Schreiben stets der Sektor-Puffer 100%-ig gefüllt wird, bevor evtl. zwischen Lesen der Quelldiskette und Schreiben der Zieldiskette gewechselt wird - schließlich ist der Kopier-Puffer von COPYD nicht unendlich groß. Um dies sicherzustellen wird die Größe des Kopier-Puffers auf ein Vielfaches von 1kB (1024 Bytes) gebracht. ($1024 = 2*512 = 4*256 = 8*128$)

Klar, ich hätte diese ganze Problematik dadurch beseitigen können, daß ich immer abwechselnd eine Spur lese und dann gleich schreibe u.s.w., aber schließlich sprach ich von SCHNELL, und außerdem sollen diejenigen von Euch, die sich COPYD für das Kopieren mit nur einem Laufwerk anpassen nicht nach jeder Spur disc-jockeyen müssen.

Da nach dem Schreiben des allerletzten Sektors auf die Zieldiskette dieser noch im Sektor-Puffer des BIOS steht (schließlich beinhaltet ungeblocktes Schreiben auch gepuffertes Schreiben), muß noch ein anderer Sektor von der Zieldiskette gelesen werden, damit das BIOS den noch nicht weggeschriebenen Puffer auf die Diskette packt (da schließlich der Sektor-Puffer bei dieser Aktion für den zu lesenden Sektor benötigt wird).

Damit die Chose richtig interessant wird, habe ich die Teile von COPYD, die nur vor dem eigentlichen Kopiervorgang benötigt werden nach hinten gestellt, wo sie bei Beginn des Kopierens überschrieben werden, d.h. sie stehen im Kopier-Puffer. So ist es immerhin gelungen, einen 440-Sektor-Puffer (55 kB) bereitzustellen, da auch das BDOS überschrieben wird, da ich mit dem BIOS alleine bestens auskomme, außer für den Disc-Reset, den ich aber vor dem Beginn des Kopierens aufrufe.

Software: Kopieren schneller ?

```

;*****
;*
;* COPYD - Spurweises Diskettenkopieren *
;*
;* (C) 1988 Herbert zur Nedden, Hamburg *
;*
;*****

        .Z80
        ASEG
        ORG     100h

Stack    equ    100h        ; Stack

Beginn:  jp     Start      ; da geht's los

DefSour: db     'D'-'A'    ; Default-Quell-Laufwerk
DefDest: db     'B'-'A'    ; Default-Ziel-Laufwerk
ScreenReset: db   Esc,'XR',0 ; Bildschirm-Reset für RAM4-System
                          ; ohne RAM4: db Esc,'XX',0

;*****
;* Konstanten und MACROs *
;*****

ConOut   equ    0ffd3h     ; Bildschirmausgabe

Reset    equ    0dh        ; BDOS-Funktion Disc-Reset

SelDrv   equ    09h        ; BIOS-Funktion Select Drive
SetTrk   equ    0ah        ;      "      Set Track (Spur)
SetSec   equ    0bh        ;      "      Set Sector
SetDma   equ    0ch        ;      "      Set DMA-Adresse
RdSec    equ    0dh        ;      "      Read Sector (Lesen)
WrSec    equ    0eh        ;      "      Write Sector (Schreiben)
SecTran  equ    10h       ;      "      Sector-Translate
                          ; (= Sektorverschränkungs-Umrechnung)

CmdLin   equ    80h        ; Da steht die CP/M-Kommandozeile

Bell     equ    7          ; Klingelingeling
cr       equ    13        ; Carriage-Return
lf       equ    10        ; Line-Feed
tab      equ    9         ; Tabulator
esc      equ    27        ; Esc

; MACRO um Ctrl-Codes zu erzeugen
;   Liefert die Konstanten Cntl_A, Cntl_B, ...
;   in der Form
; Cntl_A   equ    01h ; ^A
; u.s.w

        IRPC   X,ABCDEFGHIJKLMNPOQRSTUVWXYZ
Cntl_&X    equ    '&X' and 1fh

        ENDM

```

Software: Kopieren schneller ?

; BDOS-Aufruf für faule

```

Bdos      MACRO   Funk
          ld      c,Funk
          call    5
          ENDM

```

; dto. für BIOS

```

Bios      MACRO   Funk
          ld      a,Funk
          call    CallBIOS
          ENDM

```

; klar, oder ?

```

GetNext   MACRO
          ld      a,(hl)
          inc    hl
          ENDM

```

; wie GetNext, jedoch gleich mit Prüfung, ob gültige Laufwerksangabe

```

GetDrv    MACRO
          ld      a,(hl)
          inc    hl
          sub    'A'
          jp     c,ErrCmd
          cp    0fh
          jp     nc,ErrCmd
          ENDM

```

; LD HL,(IX+Nr) die gelegentlich fehlende Adressierungsart

```

LDIX      MACRO   Nr
          ld      l,(ix+Nr)
          ld      h,(ix+Nr+1)
          ENDM

```

; LD HL,(IY+Nr)

```

LDIY      MACRO   Nr
          ld      l,(iy+Nr)
          ld      h,(iy+Nr+1)
          ENDM

```

; INC (Par) brauche ich öfters

```

INCHL     MACRO   Par
          ld      hl,(Par)
          inc    hl
          ld      (Par),hl
          ENDM

```

; CP HL,(Par)

```

CMPDE     MACRO   Par
          ld      de,(Par)
          or     a
          sbc    hl,de
          ENDM

```

Software: Kopieren schneller ?

```
;*****
;* Dieser Teil wird nicht überschrieben *
;*****
```

```
; Einlesen einer Puffer-Ladung vom Quell-Laufwerk
```

```
ReadBfr:   ld      a,(SourDrv)
           ld      c,a
           BIOS   SelDrv      ; Laufwerk anwählen
           ld      bc,Buffer  ; Da ist der Puffer
           ld      (DMA),bc   ; Also Pufferanfang setzen

ReadLoop:  ld      bc,(DMA)    ; Da soll der Sektor hin
           ld      hl,BOh     ; so lang ist er
           add     hl,bc      ; und da hin später der nächste
           ld      (DMA),hl   ; merken
           CMPDE  BfrEnd     ; ist der Puffer voll ?
           jp     nc,WriteBfr ; ja, dann wegschreiben
           BIOS   SetDma     ; nein, dann DMA-Adresse setzen
           ld      bc,(SourTrk)
           BIOS   SetTrk    ; und Spur setzen
           ld      bc,(SourSec)
           ld      de,(SourTran)
           BIOS   SecTran   ; Sektorverschränkung ...
           ld      b,h
           ld      c,l
           BIOS   SetSec    ; und Sektor setzen
           BIOS   RdSec     ; Sektor Lesen
           or     a
           jp     nz,ErrRd   ; Fehler! -> Ja, dann AbOrt
           INCHL  ReadCount  ; Zähler 'gelesene Sektoren' +1
           CMPDE  NoOfSec   ; Ende ?
           jp     z,WriteBfr ; ja, auch den Rest wegschreiben
           INCHL  SourSec    ; nächster Sektor
           CMPDE  SecPerTrk ; schon Ende der Spur
           jp     nz,ReadLoop ; nein
           ld     (SourSec),hl ; ja, dann Sektor = 0
           call   OutMsgRead ; Meldung ausgeben
           INCHL  SourTrk   ; Spur +1
           jp     ReadLoop  ; weiter
```

```
; Meldung ausgeben, wieviele Sektoren gelesen wurden
```

```
OutMsgRead: ld      hl,MsgReadPos
            call   Print
            ld      hl,(ReadCount)
            call   NumOut    ; Umsetzen in lesbare Zahl
            ld      hl,MsgReadSec
            jp     Print
```

Software: Kopieren schneller ?

```

; Puffer auf Ziel-Laufwerk schreiben
;
; da beim Schreiben fast die selben Dinge getan werden
; (DMA setzen, Sektor setzen, ...), wie beim Schreiben,
; ist dieser Teil von COPYD mit weniger Kommentaren gesegnet

WriteBfr:    call    OutMsgRead ; Sagen, was geschafft wurde
            ld      a,(DestDrv)
            ld      c,a
            BIOS    SelDrv
            ld      bc,Buffer
            ld      (DMA),bc

WriteLoop:   ld      bc,(DMA)
            ld      hl,80h
            add     hl,bc
            ld      (DMA),hl
            CMPDE   BfrEnd      ; Puffer abgearbeitet ?
            jp      nc,GoBack   ; ja, dann weiterlesen
            BIOS    SetDma
            ld      bc,(DestTrk)
            BIOS    SetTrk
            ld      bc,(DestSec)
            ld      de,(DestTran)
            BIOS    SecTran
            ld      b,h
            ld      c,l
            BIOS    SetSec
            ld      c,2          ; ungepuffert schreiben !!!!
            BIOS    WrSec
            or      a
            jp      nz,ErrWr
            INCHL   WriteCount
            CMPDE   NoOfSec
            jp      z,Done      ; Jippiihhhh .. FERTIG!
            INCHL   DestSec
            CMPDE   SecPerTrk
            jp      nz,WriteLoop
            ld      (DestSec),hl
            call    OutMsgWrite
            INCHL   DestTrk
            jp      WriteLoop

GoBack:     call    OutMsgWrite ; Meldung ausgeben
            jp      ReadBfr     ; weiterlesen

```

Software: Kopieren schneller ?

; Damit auch der Puffer des letzten Sektors weggeschrieben wird,
 ; einfach mal von Spur 0 etwas einlesen.

```
Done:      ld      bc,Buffer    ; Verblüffend, was man so alles
          BIOS   SetDma      ; tun muß, um nur so einen popeligen
          ld      bc,0        ; Sektor zu lesen, niwwa ?
          BIOS   SetTrk
          ld      bc,1
          ld      de,(SourTran)
          BIOS   SecTran
          ld      b,h
          ld      c,l
          BIOS   SetSec
          BIOS   RdSec

          call   OutMsgWrite
          ld     hl,MsgFertig
          jp     Exit
```

; Meldung über geschriebene Sektoren ausgeben

```
OutMsgWrite: ld     hl,MsgWritePos
            call  Print
            ld     hl,(WriteCount)
            call  NumOut
            ld     hl,MsgWriteSec
            jp     Print
```

; Diverse Fehlermeldungen ausgeben

```
ErrCmd:     ld     hl,MsgErrCmd
            jp     ErrExit
ErrDrv:     ld     hl,MsgErrDrv
            jp     ErrExit
ErrDPB:     ld     hl,MsgErrDPB
            jp     ErrExit
ErrRd:      ld     hl,MsgErrRd
            jp     ErrExit
ErrWr:      ld     hl,MsgErrWr

ErrExit:    call  Print
            ld     hl,MsgErr
Exit:       call  Print
            ld     de,0        ; Warm-Boot
            push  de          ; Statt Call Print und JP 0
            ld     hl,ScreenReset
```

Print: ; Zeichenfolge, die mit Oh endet auf die Glotze ausgeben

```
GetNext
or         a
ret        z
ld         c,a
call      ConOut
jp        Print
```

Software: Kopieren schneller ?

; HL dezimal ausgeben

```

NumOut:      ld      b,' '      ; führende Leerstellen
             ld      de,10000
             call   ZifferOut  ; erst die 10000-er
             ld      de,1000
             call   ZifferOut  ; dann die 1000-er
             ld      de,100
             call   ZifferOut  ; die 100-er
             ld      de,10
             call   ZifferOut  ; die 10-er
             ld      a,l
             jp     DigitOut   ; und ihn auch noch

```

```

; Von der Zahl (in HL) wird die gewünschte Zehnerpotenz (in DE), also
; 10000, 1000, 100 bzw. 10 so lange abgezogen, bis das Ergebnis unter
; Null ist. Wie oft abgezogen wird, wird natürlich gezählt, denn diese
; Anzahl ist ja die Anzahl, wie oft die Zehnerpotenz in der Zahl drin
; steckt, also die gesuchte Ziffer.
; Da wir einmal zu oft abziehen (d.h. bis HL <0), beginnen wir die Zählung
; bei -1.

```

ZifferOut:

```

ld      a,Offh      ; -1

```

ZiffOutL:

```

or      a           ; Carry löschen
sbc    hl,de       ; Zehnerpotenz abziehen
inc    a           ; Zähler erhöhen
jp     nc,ZiffOutL ; <0 ?, wenn nein: nocheinmal
add    hl,de       ; Zehnerpotenz dazuaddieren (da 1x zu viel)
or     a           ; =0 (prüfung wegen führender 0)
jp     z,NibbleOut ; ja, dann ausgeben (als '0' oder ' ')

```

DigitOut:

```

ld      b,'0'      ; nun kommen die Ziffern

```

NibbleOut:

```

add    a,b
ld     c,a
jp     ConOut

```

; BIOS-Aufruf

; Im Akku (d.h. im A-Register) steht die Funktionsnummer

```

CallBIOS:  push   de          ; retten
           dec    a           ; -1
           ld     d,a
           sla   a           ; *2
           add   a,d         ; +1 = *3
           ld     d,0
           ld     e,a        ; DE = 3* Funktionsnummer
           ld     hl,(1)     ; HL = Adresse WarmBoot
           add   hl,de       ; Nun (HL) = gewünschte Funktion
           pop   de         ; da war doch noch was
           jp    (hl)       ; Los

```


Software: Kopieren schneller ?

; Ein paar erforderliche Speicherstellen

```

SourDrv:      db      0      ; Laufwerke
DestDrv:      db      0

SourTran:     dw      0      ; Translate-Tables
DestTran:     dw      0

SourSec:      dw      0      ; Zu lesener/schriebender Sektor
DestSec:      dw      0

SourTrk:      dw      0      ; Zu lesene/schriebende Spur
DestTrk:      dw      0

ReadCount:    dw      0      ; Gelesene/schriebene Sektoren
WriteCount:   dw      0

NoOfSec:      dw      0      ; Anzahl zu Kopierender Sektoren

Sektor:       dw      0
Track:        dw      0
DMA:          dw      0

DPB:          ; Disk Parameter Block
SecPerTrk:    dw      0      ; Sektoren / Spur
BlkShif:      db      0      ; Block Shift
RecPerBlk:    db      0      ; Block-Größe
              db      0
Blocks:       dw      0      ; Anz. Blöcke
              dw      0
              dw      0
              dw      0
SysTrk:       dw      0      ; System-Spuren

MsgReadPos:   db      Cntl_C,32,5+32,0
MsgReadSec:   db      ' Sektoren gelesen',0
MsgWritePos:  db      Cntl_C,32,7+32,0
MsgWriteSec:  db      ' Sektoren geschrieben',0

MsgErrRd:     db      cr,lf,lf,'Lesefehler aufgetreten - Abbruch '
              db      cr,lf,0
MsgErrWr:     db      cr,lf,lf,'Schreibfehler aufgetreten - Abbruch '
              db      cr,lf,0

MsgErr:       db      cr,lf,Bell,0
MsgFertig:    db      cr,lf,lf,'Fertig!'
              db      cr,lf,0

BfrEnd:      dw      0

```

Software: Kopieren schneller ?

```
;*****
;* Ab hier wird schon überschrieben *
;*****
```

Buffer:

```
Start:      Id      hl,MsgHallo ; Begrüßung
            call    Print

            Id      a,(CmdLin) ; Kommandozeile Leer ?
            or      a
            jp      nz,Parse   ; nein

            Id      a,(DefSour) ; sonst Default-Werte nehmen
            Id      (SourDrv),a
            Id      a,(DefDest)
            Id      (DestDrv),a
            jp      CmdOK      ; Default-Werte glauben!
```

Parse:

```
Id      hl,CmdLin+2 ; Dorten geht's los
GetDrv   ; Ziel-Laufwerk
Id      (DestDrv),a
Id      b,a         ; Merken
GetNext  ; Doppelpunkt
cp      ':'
jp      nz,ErrCmd

GetNext  ; Gleichheitszeichen
cp      '='
jp      nz,ErrCmd

GetDrv   ; Quell-Laufwerk
Id      (SourDrv),a
cp      b           ; Laufwerke gleich ?
jp      z,ErrDrv   ; also nein
GetNext  ; Doppelpunkt
cp      ':'
jp      nz,ErrCmd
```

CmdOK:

```
Id      sp,Stack   ; das wichtigste, der Stack

Id      a,(SourDrv) ; Quell-Laufwerk
Id      c,a
add     a,'A'      ; Aufbereiten für Meldung
Id      (MsgSour),a
BIOS    SelDrv     ; Quell-Laufwerk selektieren
push    hl         ; DPH-Adress
pop     ix
ldix   00h         ; Translate-Table
Id      (SourTran),hl
ldix   0ah         ; DPB
push    hl

Id      de,DPB     ; Quell-DPB retten
Id      bc,15
ldir
```

Software: Kopieren schneller ?

```

        ld      a,(DestDrv) ; Ziel-Laufwerk
        ld      c,a
        add     a,'A'       ; aufbereiten für Meldung
        ld      (MsgDest),a
        BIOS   SelDrv      ; Ziel-Laufwerk selektieren
        push   hl          ; DPH-Adress
        pop    iy
        ldiy   00h        ; Translate-Table
        ld      (DestTran),hl
        ldiy   0ah        ; DPB 2

        pop    ix          ; DPB 1
        ld      b,15      ; Länge DPB

CpDPB:
        ld      a,(ix)     ; DPB's vergleichen
        cp     (hl)
        jp     nz,ErrDPB  ; nicht gleich: schade
        inc    ix
        inc    hl
        djnz   CpDPB      ; weitermachen - oder auch nicht

; Berechnen, wieviele Sektoren auf der Scheibe sind:
;
; Sektoren = System-Spuren * Sektoren je Spur
;           + Blöcke * Blockgröße / Sektor-Größe
;           = System-Spuren * Sektoren je Spur
;           + Blöcke * 2 hoch BlockShift

        ld      bc,(SysTrk) ; System-Spuren
        ld      de,(SecPerTrk) ; Sektoren / Spur
        call   multbcde     ; multipliziert:
        push   hl          ; Sektoren in Systemspuren
        ld      a,(BlkShif) ; BlockShift
        ld      b,a
        ld      hl,(Blocks) ; Blöcke
        inc    hl
Shift:  add     hl,hl        ; mal zweierpotenz
        djnz   Shift
        pop    de
        add    hl,de
        ld      (NoOfSec),hl; Zahl Sektoren

        ld      hl,MsgLos
        call   Print
        ld      hl,(NoOfSec)
        call   NumOut
        ld      hl,MsgSekt
        call   Print

; Erst mal die verschiedenen Werte initialisieren

Init:   ld      hl,0
        ld      (ReadCount),hl ; noch nix gelesen
        ld      (WriteCount),hl ; und nix geschrieben
        ld      (SourTrk),hl   ; bei Spur 0 geht's los
        ld      (DestTrk),hl   ; dto.
        ld      (SourSec),hl   ; und Sektor 0
        ld      (DestSec),hl   ; dto.

```

Software: Kopieren schneller ?

```

ld      hl,(1)
dec     hl
dec     hl
dec     hl
dec     hl          ; Ende der TPA (BDOS Ade!)
ld      de,Buffer
or      a
sbc     hl,de       ; Puffergröße

```

; Puffergröße muß (ein Vielfaches von 1kB) +1 sein, damit beim unge-
; blocktem Schreiben nicht halbe phys. Sektoren auf der Strecke bleiben!
; Die +1 ist von wegen der Art der Ende-Abfrage des Puffers nötig

```

ld      a,0fch      ; 1k-Blöcke Phys.
and     h
ld      h,a
ld      l,1         ; Wegen nc-Abfrage
add     hl,de
ld      (BfrEnd),hl
jp      ReadBfr    ; Los geht's

```

; HL = BC * DE

```

MultBCDE:  ld      a,c
           ld      c,b
           ld      b,15
           ld      hl,0
MultLoop:  srl     c
           rra
           jr      nc,MultNo
           add     hl,de
MultNo:    ex      de,hl
           add     hl,hl
           ex     de,hl
           djnz   MultLoop
           ret

```

```

MsgHallo:  db      Esc,'AR',Cntl_L,1fh
           db      Esc,'S',Cntl_U,'CopyDisc',Cntl_S
MsgHilf:   db      ' der Diskettenduplizierer '
           db      Esc,'GC',Esc,'S 1988 Herbert zur Nedden, Hamburg'
           ds      76-#+MsgHilf,' '
           db      Cntl_R,cr,lf,lf,0
MsgErrCmd: db      'Fehler in der Kommandozeile',cr,lf,lf
           db      'Aufruf: ',Cntl_T,'COPYD z:=q:',Cntl_R,cr,lf,lf
           db      'Dabei müssen für ',Cntl_T,'z',Cntl_R,
           db      ' das Ziel- und für ',Cntl_T,'q',Cntl_R,
           db      ' das Quell-Laufwerk angegeben werden',cr,lf,lf
           db      'Die beiden Laufwerke müssen das selbe Format haben,'
           db      cr,lf
           db      'jedoch verschiedene Laufwerke sein',0
MsgErrDrv: db      'Die Laufwerke müssen verschieden sein',0
MsgErrDPB: db      'Die Laufwerke haben nicht das selbe Format',0

MsgLos:    db      'Ich kopiere von '
MsgSour:   db      'D: nach '
MsgDest:   db      'B: ',0
MsgSekt:   db      ' Sektoren',0
END

```

Software: Biorhythmus**Biorhythmus mit Partnervergleich**

(Herbert zur Nedden, 2000)

Meine Frau wollte vor einiger Zeit mal wissen, wie unsere Biorhythmen im Vergleich aussehen. Also suchte ich mal die Infos und PD's durch, und fand sogar ein FDX-BASIC-Programm, welches einen Biorhythmus mit simpelster Grafik ausgab. Ich wollte jedoch beide Rhythmen am liebsten gleichzeitig haben, und auch noch das sog. Partner-Dreieck anzeigen, da Biorhythmus-Kenner auch daraus einiges entnehmen können. Um all dieses gleichzeitig anzeigen zu können, kam mir meine Edicta-Grafik-Karte mit einer Auflösung von 700 x 560 Pixeln sehr zu statten.

Da im Club noch kein derartiges Biorhythmus-Programm existiert, außer meinem, finde ich es an der Zeit, diesem Umstand abzuhelpfen.

Zum Biorhythmus eines Menschen gehören drei sinusförmige Zyklen, die am Tage der Geburt bei Null beginnen, und dann lustig (am Anfang steigend) losschwingen. Die drei Zyklen haben drei verschiedenen Phasenlängen. Die Phasenlänge einer Sinuskurve ist die Zeit um eine Schwingung von Null über Eins, Null, minus Eins zurück zu Null zu durchlaufen, d.h. nach einer Phasenlänge beginnt die Chose von vorne.

Die drei Zyklen sind:

1. Körperlicher Zyklus, Phasenlänge 23
2. Intellektueller (=geistiger) Zyklus, Phasenlänge 28
3. Seelischer (=emotioneller) Zyklus, Phasenlänge 33

Die Punkte auf z.B. dem körperlichen Zyklus können also wie folgt berechnet werden: $\text{Sinus}(2\pi/23 * (\text{Tag seit der Geburt}))$

oder auch: $\text{Sinus}(2\pi/23 * ((\text{Tag seit der Geburt}) \bmod 23))$

2π ist die normale Phasenlänge des Sinus; mit $2\pi/23$ wird die Phasenlänge auf 23 gebracht. Da sich der Sinus nach einer Phasenlänge wiederholt, kann ich das Argument (hier 'Tag seit der Geburt') modulo Phasenlänge (hier 23) nehmen.

Mein Programm berechnet für beide Partner die Biorhythmen ab dem aktuellen Tagesdatum für einen Monat, und gibt diese grafisch aus.

Zum Partnervergleich muß bewertet werden, wie 'gleich' die drei Zyklen paarweise sind, d.h. die beiden körperlichen Zyklen werden miteinander verglichen, u.s.w. 100%-ige Übereinstimmung bedeutet, daß sie übereinander liegen, und 0%-ige Übereinstimmung, daß sie entgegengesetzt laufen, d.h. mit genau umgekehrtem Vorzeichen versehen sind. Anders formuliert, bedeutet 100%-ige Übereinstimmung, daß von beide Zyklen die Startpunkte der Phasen gleich sind, und 0%-ige Übereinstimmung, daß diese Startpunkte um die halbe Phasenlänge auseinanderliegen.

Um die Lage der Startpunkte zu vergleichen, muß ich also den Abstand zwischen den Tagen der Geburt der beiden Partner heranziehen, und modulo der halben Phasenlänge nehmen. Ist dieser Wert 0, bedeutet es 100%-ige Übereinstimmung, ist der Wert $(\text{Phasenlänge}/2) - 1$, so ist die Übereinstimmung 0%-ig.

So erhalte ich drei Werte, aber noch kein Dreieck. Auch hier muß der gute alte Sinus herhalten. Zu jedem Wert liefert $-\text{Sinus}(2\pi/100 * \text{Wert})$ einen Punkt auf einer vollen Sinus-Phase, wobei für Wert $< 50\%$ das Ergebnis negativ, anderenfalls positiv ist. Jetzt habe ich also auf einer Sinuskurve zu jedem der drei Übereinstimmungs-Werte, aus denen sich leicht ein Dreieck machen läßt: Das Partnervergleichs-Dreieck.

Eckpunkte des Dreiecks, die im negativen (d.h. d.h. unterhalb der waagerechten Koordinatenachse) Bereich liegen zeigen geringe Übereinstimmung, die anderen größere Übereinstimmung. Desto mehr von dem Dreieck im positiven Bereich (also oberhalb der waagerechten Koordinatenachse) liegen, desto mehr Übereinstimmung ist zwischen den Partnern gegeben.

Software: Biorhythmus

Hier nun das Programm: für eine Grafik-Auflösung von 700 x 560:

Verwendete Grafik-Funktionen:

SetzeLinienMuster setzt das Zeichenmuster für Linien

Linie bzw. Rechteck zeichnet das genannte

PrintStr(t,x,y) gibt String t an (x,y) aus

Program BioRhythmus;

```

Const  TageImMonat: array[1..12] of Integer
        = (31,29,31,30,31,30,31,31,30,31,30,31);
Phys:   string[7] = 'Kirper';
Emot:   string[7] = 'Seele ';
Inte:   string[7] = 'Geist ';
DX = 18; DY = 50;

Type    DatumText = Record (TT.MM.JJ)
        Case Boolean of
        True: (Len:   Byte;
               TT:   Array[1..2] of Char;
               Fill1: Char;
               MM:   Array[1..2] of Char;
               Fill2: Char;
               JJ:   Array[1..2] of Char);
        False: (TTMMJJ: String[8]);
        End;
Koord   = Array[1..30] of Integer;
Str8    = String[8];
AnyStr  = String[99];

(* Prüfen, ob Datum gueltig ist, und ermittle die Tage seit 1900 *)
Function DatumOK(Var Datum: DatumText; Var TagSeit1900: Integer): Boolean;
  Var Tag,Monat,Jahr: Integer;
      ReturnCode,I: Integer;

  Begin
    With Datum Do Begin
      TagSeit1900:=-1;
      Val(TT,Tag,ReturnCode);
      If ReturnCode = 0 Then
        Begin
          Val(MM,Monat,ReturnCode);
          If ReturnCode = 0 Then
            Begin
              Val(JJ,Jahr,ReturnCode);
              If ReturnCode = 0 Then
                Begin
                  If (Monat>0) and (Monat<13) Then
                    Begin
                      If (Tag>0) and (Tag<=TageImMonat[Monat]) Then
                        Begin
                          TagSeit1900:=Tag-1 + Jahr*365 + Jahr div 4;
                          If Monat > 1 Then
                            For I:=2 to Monat Do
                              TagSeit1900:=TagSeit1900+TageImMonat[I-1];
                        End; End; End; End; End; End;
                          DatumOK:=(TagSeit1900>=0);
                        End;
                    End;
                End;
              End;
            End;
          End;
        End;
      End;
    End;
  End;

```

Software: Biorhythmus

```

Procedure Kurve(Periode, Start: Integer; Var Y: Koord);
  Var I,X: Integer;
      R: Real;

```

```

  Begin
    X:=1;
    For I:=Start To Start+30 Do
      Begin;
        R:=I/Int(Periode) - I div Periode;
        Y[X]:=-Trunc(Sin(2.*Pi*R)*DY);
        X:=X+1;
      End; End;

```

```

Procedure Rhythmus(Start, X0,Y0: Integer; Person: Str8);
  Var Y: Koord;
      I: Integer;

```

```

  Begin
    PrintStr(Person,X0,Y0-DY-20);
    X1:=X0-10; X2:=X0+30*DX+10; Y1:=Y0-DY-30; Y2:=Y0+DY+20;
    Rechteck(X1,Y1,X2,Y2);
    X1:=X0; Y1:=Y0; X2:=X0+30*DX; Y2:=Y0; Line(X1,Y1,X2,Y2);
    X1:=X0; Y1:=Y0-DY-10; X2:=X0; Y2:=Y0+DY+10; Line(X1,Y1,X2,Y2);
    For I:=1 to 29 Do
      Begin
        X1:=X0+I*DX; X2:=X1; Y1:=Y0-3; Y2:=Y0+3;
        Line(X1,Y1,X2,Y2);
      End;
    For I:=-10 to 10 Do
      Begin
        Y1:=Y0+I*DY div 10; Y2:=Y1; X1:=X0-2; X2:=X0+2;
        Line(X1,Y1,X2,Y2);
      End;

```

```

    Kurve(33,Start,Y); SetzeLinienMuster($f0f0);
    X1:=X0; Y1:=Y0+Y[I];
    For I:=2 To 30 Do      { seelisch }
      Begin
        X2:=X1+DX; Y2:=Y0+Y[I]; Line(X1,Y1,X2,Y2);
        X1:=X2; Y1:=Y2;
      End;

```

```

    Kurve(23,Start,Y); SetzeLinienMuster($ff00);
    X1:=X0; Y1:=Y0+Y[I];
    For I:=2 To 30 Do      { kirperlich }
      Begin
        X2:=X1+DX; Y2:=Y0+Y[I]; Line(X1,Y1,X2,Y2);
        X1:=X2; Y1:=Y2;
      End;

```

```

    Kurve(28,Start,Y); SetzeLinienMuster($fff);
    X1:=X0; Y1:=Y0+Y[I];
    For I:=2 To 30 Do      { geistiger }
      Begin
        X2:=X1+DX; Y2:=Y0+Y[I]; Line(X1,Y1,X2,Y2);
        X1:=X2; Y1:=Y2;
      End;

```

```

  End;

```

Software: Biorhythmus

```

Procedure Partner(DP,DE,DI: Real; X0,Y0: Integer);
  Var Y: Koord;
      I: Integer;
      R: Real;
      XP,XE,XI,YP,YE,YI: Integer;

  Begin
    PrintStr('Partner-Vergleich',X0,Y0-DY-20);
    X1:=X0-10; X2:=X0+30*DX+10; Y1:=Y0-DY-30; Y2:=Y0+DY+20;
    Rechteck(X1,Y1,X2,Y2);
    X1:=X0; Y1:=Y0; X2:=X0+30*DX; Y2:=Y0; Line(X1,Y1,X2,Y2);
    X1:=X0; Y1:=Y0-DY-10; X2:=X0; Y2:=Y0+DY+10; Line(X1,Y1,X2,Y2);
    For I:=1 to 29 Do
      Begin
        X1:=X0+I*DX; X2:=X1; Y1:=Y0-3; Y2:=Y0+3;
        Line(X1,Y1,X2,Y2);
      End;
    For I:=-10 to 10 Do
      Begin
        Y1:=Y0+I*DY div 10; Y2:=Y1; X1:=X0-2; X2:=X0+2;
        Line(X1,Y1,X2,Y2);
      End;

    Kurve(29,0,Y);
    SetzeLinienMuster($f0f0);
    X1:=X0; Y1:=Y0+Y[I];
    For I:=2 To 30 Do
      Begin
        X2:=X1+DX; Y2:=Y0-Y[I]; Line(X1,Y1,X2,Y2); X1:=X2; Y1:=Y2;
      End;

    XP:=Trunc(29*DP*0.01); R:=XP/Int(29) - XP div 29;
    YP:=Trunc(Sin(2.*Pi*R)*DY);
    XE:=Trunc(29*DE*0.01); R:=XE/Int(29) - XE div 29;
    YE:=Trunc(Sin(2.*Pi*R)*DY);
    XI:=Trunc(29*DI*0.01); R:=XI/Int(29) - XI div 29;
    YI:=Trunc(Sin(2.*Pi*R)*DY);

    SetzeLinienMuster($ffff);
    X1:=XP*DX+X0; Y1:=YP+Y0; X2:=XE*DX+X0; Y2:=YE+Y0;
    Line(X1,Y1,X2,Y2);
    X1:=XP*DX+X0; Y1:=YP+Y0; X2:=XI*DX+X0; Y2:=YI+Y0;
    Line(X1,Y1,X2,Y2);
    X=XI*DX+X0; Y1:=YI+Y0; X2:=XE*DX+X0; Y2:=YE+Y0;
    Line(X1,Y1,X2,Y2);

    X1:=XP*DX+X0;
    If YP<0 then Y1:=YP+Y0-10
      else Y1:=YP+Y0+18;
    PrintStr('K',X1,Y1);
    X1:=XE*DX+X0;
    If YE<0 then Y1:=YE+Y0-10
      else Y1:=YE+Y0+18;
    PrintStr('S',X1,Y1);
    X1:=XI*DX+X0;
    If YI<0 then Y1:=YI+Y0-10
      else Y1:=YI+Y0+18;
    PrintStr('G',X1,Y1);
  End;

```


Software: Biorhythmus

```

Var Heute,Er,Sie: DatumText;
    TH,TE,TS,TD: Integer;
    DP,DE,DI: Real;
    SP,SE,SI: String[5];
    I,Adr.,Lin: Integer;
    I1,I2,I3: String[2];
    B: Boolean;

```

```

BEGIN

```

```

  ClrScr;
  GotoXY(10,08); Write('Datums-Format: TT.MM.JJ');
  GotoXY(10,10); Write('Heutiges Datum: ',#5);

```

```

  (* Hole RAM4-Datum *)

```

```

  I:=Mem[$FFC6]+256*Mem[$FFC7]; Adr:=Mem[I+$2C]+256*Mem[I+$2D];
  I:=Mem[Adr+0]; Str(I:2,I1);
  I:=Mem[Adr+1]; Str(I:2,I2);
  I:=Mem[Adr+2]; Str(I:2,I3);
  Heute.TTMMJJ:=I1+'.'+I2+'.'+I3;
  If Heute.TT[1]=' ' Then Heute.TT[1]='0';
  If Heute.MM[1]=' ' Then Heute.MM[1]='0';
  If Heute.JJ[1]=' ' Then Heute.JJ[1]='0';
  Write(Heute.TTMMJJ);
  B:=DatumOK(Heute,TH);

```

```

  repeat

```

```

    GotoXY(10,12); Write('Sein Geburtsdatum: 15.06.60',#5);
    read(Er.TTMMJJ);
  until (DatumOK(Er,TE));
  Lin:=80+dy; Rhythmus(TH-TE,100,Lin,'Er');

```

```

  repeat

```

```

    GotoXY(10,14); Write('Ihr Geburtsdatum: 22.06.58',#5);
    read(Sie.TTMMJJ);
  until (DatumOK(Sie,TS));
  Lin:=Lin+60+dy+dy; Rhythmus(TH-TS,100,Lin,'Sie');

```

```

  If TS<TE Then TD:=TE-TS
    Else TD:=TS-TE;

```

```

  { Abweichung = Abstand(Startdifferenz modulo Phase) von Phasenmittel }
  { Pozentsatz = Verhaeltnis Abstand zu halber Phasenlaenge }
  DP:=abs((TD mod 23)-11.5)*200.0/23.0;
  DI:=abs((TD mod 28)-14.0)*200.0/28.0;
  DE:=abs((TD mod 33)-16.5)*200.0/33.0;

```

```

  Lin:=Lin+60+dy+dy; Partner(DP,DE,DI,100,Lin);

```

```

  PrintStr(Phys,100,550); PrintStr(Emot,300,550); PrintStr(Inte,500,550);

```

```

  SetzeLinienMuster($ff00);
  X1:=165; Y1:=550; X2:=X1+70; Y2:=Y1; Line(X1,Y1,X2,Y2);
  SetzeLinienMuster($f0f0);
  X1:=360; Y1:=550; X2:=X1+70; Y2:=Y1; Line(X1,Y1,X2,Y2);
  SetzeLinienMuster($ffff);
  X1:=560; Y1:=550; X2:=X1+70; Y2:=Y1; Line(X1,Y1,X2,Y2);

```

```

END.

```

Hard- und Software: Edicta-Grafikkarte

Was kann die Edicta-Grafikkarte mit ihrer Software ? (Herbert zur Nedden, 2000)

Ich werde immer wieder (wenn auch vereinzelt) gefragt:

'Was kann eigentlich die Edicta-Grafik-Software bzw. die -Karte ?'

Dabei ist der Zweck der Frage, entweder sich zu informieren, ob sich die Beschaffung der Karte lohnt, oder, was alles zu programmieren ist, um die eigene Nicht-Edicta-Karte (TERM1, VS 4, ...) mit einer kompatiblen Software zu versehen, um so einen Programmaustausch zu ermöglichen.

Immerhin hat die Edicta-Grafik-Software so eine Art 'Standardcharakter'.

Daher liste ich hier einfach einmal die eingebauten Funktionen auf:

Globale Setup-Routinen

Init	Initialisierung der Grafik-Software und Grafik-Karte
Interlace	Interlace-Modus an/aus
Display	Display an/aus
Zoom	Vergrößerungsfaktor für die Anzeige auf dem Monitor
Cls	Löschen der Grafik-Ebene
AreaCls	Löschen eines rechteckigen Teils der Grafik-Ebene
Status	Abfragen der Größe der Grafik-Ebene und die Größe des Ausschnitts, der auf dem Monitor angezeigt wird.

Cursor- und Ausschnitt-Routinen

SetCrs	Cursorposition setzen
CrsRead	Cursorposition lesen
CrsLeft	Cursor nach links
CrsRight	Cursor nach rechts
CrsUp	Cursor hoch
CrsDown	Cursor runter
ScrTop	oberen Ausschnitt setzen
ScrBottom	unteren Ausschnitt setzen
ScrLeft	ersten Ausschnitt nach links
ScrRight	ersten Ausschnitt nach rechts
ScrUp	ersten Ausschnitt hoch
ScrDown	ersten Ausschnitt runter
Tab	Schrittweite für Crs*- und Scr*-Bewegungen setzen
Fadenkreuze	zwei Fadenkreuze (Cursor, Marker) zeichnen
Frame	bewegliches Rechteck zum Einlesen von dessen Koordinaten

Linien und Grundfiguren

Line	Linie
Draw	Linie
Rectangle	Rechteck
Box	gefülltes Rechteck
Invert	Rechteck invertieren
Kreis	Kreis
Circle	Kreis
Arc	Kreisbogen
Plot	Punkt
CrsPlot	Zeichenstift an/aus
WdatMod	Zeichenattribut setzen
BitMuster	Linienmuster setzen
CircleMod	Randprüfung für Kreise an/aus

Hard- und Software: Edicta-GrafikkarteZeichen und Flächenfüllen

Print	Zeichen ausgeben
PrintStr	Zeichenkette ausgeben
PrintMod	Normal/Kursiv, Schreib-Richtung, Zeichenbreite und -größe für Print und PrintStr festlegen
SetGchar	Muster für Grafik-Zeichen definieren
Fill	Fläche mit Muster füllen
FillMuster	Muster für Fill und Box definieren

Grafik-RAM-Zugriffe

ReadBlock	16x64 Pixel aus dem Grafik-RAM lesen
WriteBlock	16x64 Pixel in das Grafik-RAM schreiben
SaveBlock	16x64 Pixel-Vielfaches vom Grafik-RAM auf Diskette schreiben
LoadBlock	16x64 Pixel-Vielfaches von Diskette ins Grafik-RAM laden
WordRead	16x 1 Pixel aus dem Grafik-RAM lesen
DotRead	1x 1 Pixel aus dem Grafik-RAM lesen

Hardcopy	Fläche auf den Drucker ausgeben
CopyMod	Druckdichte, lage auf dem Papier für HardCopy festlegen
PrinterTyp	Drucker DMX80, Panasonic 1093, Epson, NEC P9XL selektieren

3D-Umrechnungen

SetZero	2D-Koordinaten des 3D-Ursprunges setzen
SetAngle	3D-Drehungswinkel setzen
DreiD	3D-Koordinaten in 2D-Koordinaten umrechnen

Sonstiges

Version	Version der Grafik-Software einlesen
---------	--------------------------------------

Die Edicta-Grafik-Software ist größtenteils in Assembler geschrieben, und hat, wie das BDOS, nur einen Einsprung. Die gewünschte Funktion wird im Akkumulator, und die zu übergebenen Daten in einem 8-Byte-Puffer, dem sog. GCB (Graphics Control Block) übergeben. Sollen größere Datenmengen zwischen dem Grafik-RAM und dem Programm ausgetauscht werden, steht im GCB natürlich die Adresse eines Daten-Puffers!

Dank der (darum so entwickelten) Aufruf- und Datenübergabemimik ist es möglich (und läuft auch), daß die Grafik-Software entweder hinter das Programm in die CP/M-TPA, oder, wie ein KLICK-Overlay, auf eine andere Speicherbank geladen wird. Letztere Variante bedeutet natürlich, daß die Software nur einmal geladen und im Bereich des BIOS ein Einsprung abgelegt wird, d.h. ist eigentlich eine Systemerweiterung ist.

Um auf die Grafik-Software zugreifen zu können, gibt es für Turbo-Pascal zwei Include-Dateien (1x ohne und 1x mit KLICK), in denen ein leichter Zugriff auf die verschiedenen Funktionen ermöglicht wird, ohne für die Funktionen die internen Funktionsnummern und die Lage der Parameter im GCB kennen zu müssen. Zu diesem Zweck sind dort die Aufrufe als Procedures, und die GCB-Parameter als Variablen definiert worden.

Für Assembler gibt es ebenfalls eine Include-Datei als Schnittstelle zwischen dem Programm und der Grafik-Software, die allerdings nicht den vollen Umfang, wie die Turbo-Pascal-Versionen bietet.

Wer nun vor hat (wie es Manfred Flume für seine TERM 1 schon versucht), eine Grafik-Software für eine andere Karte (z.B. den VS 4 !!!!) zu erstellen, weiß nun schon einmal, was für Funktionen schon vorhanden sind - es werden auf jeden Fall mehr.

Das Handbuch zur Edicta-Software sowie beratende Unterstützung gebe ich gerne weiter.

Hardware: Booten von 09 und 1A und ...

NEUES BOOT-EPROM FÜR FDX

(Michael Kessler, 3300)

JETZT KANN JEDER (AUCH OHNE RAM4) MIT 09, 0A, 1A ODER 1D BOOTEN !
CA. 45 FORMATE OHNE RAM4 MÖGLICH

Das neue BOOT-Eprom für die FDX ist endlich fertig. Es ermöglicht das Booten mit den Formaten 03, 07, 09, 0A, 19, 1A, 1D und 6F. Somit stehen die MaxBP und weitere Formate jetzt auch denjenigen zur Verfügung, die RAM4 bisher noch nicht besitzen, alle Besitzer von RAM4 ersparen sich spezielle "Bootdisks" mit Format 03 oder 07.

Die beiden letzten Formate sind nicht im ursprünglichen FORMATE4.DAT, sondern neue Versionen. Format 1D ist ein 1C (MaxHD 80T) mit Systemspuren, Kapazität 1730 KB. Format 6F ist nur was für Leute mit einem einseitigen 8"er, es ist ein Single-Sided 19, MaxBP 77T mit Systemspuren, Kapazität 682 KB. Da ich auf diesem Format boote, muß es halt zum Leidwesen der übrigen Clubmitglieder im Eprom drin sein.

Das neue Eprom bietet die folgenden Neuerungen:

- Booten mit den Formaten 3,7,9,A,19,1A,1D,6F auf jedem physikalischen Laufwerk B, C, D, E in jeder gewünschten Kombination möglich
- Ein nach individuellen Wünschen eingebauter Suchpfad ermöglicht das Booten auf drei verschiedenen Laufwerken und/oder Formaten
- ohne RAM4 kann aus einem Bestand von ca. 45 Fremdformaten jeweils eines mittels Overlay-Verfahren ins Betriebssystem eingelagert werden

ACHTUNG: RAM3 läuft mit dem neuen Eprom NICHT !!!

Die erste Neuerung besteht aus der Möglichkeit des Bootens mit MaxBP/MaxHD-Formaten. Für alle diejenigen, die RAM4 nicht besitzen, gebe ich eine kurze Übersicht über die implementierten Formate:

03 - normales Memotech 03,	5", 40T, DS, DD,	310k
07 - normales Memotech 07,	5", 80T, DS, DD,	630k
09 - MaxBP	5", 40T, DS, DD,	390k
0A - MaxBP	5", 80T, DS, DD,	790k
19 - MaxBP	8", 77T, DS, DD,	1368k
1A - MaxBP (für 77/80 Lw)	8", 80T, DS, DD,	1420k
1D - MaxHD (s. Info 26)	8", 80T, DS, DD,	1738k
6F - MaxBP	8", 77T, SS, DD,	682k

Die zweite Neuerung ist der Einbau eines Boot-Suchpfades mit drei Bootformaten/-laufwerken. Jede Kombination aus den o.g. Bootformaten und den Laufwerken B: bis E: ist möglich. Jeder Besteller bekommt seinen Wünschen entsprechend ein individuelles Booteprom mit den drei von ihm bevorzugten Bootformaten und Bootlaufwerken.

Hardware: Booten von 09 und 1A und ...

Zur Verdeutlichung:

Ich selbst habe z.B. den Pfad B:1A, B:0A und E:6F gewählt. Beim Einschalten des Rechners erscheint zuerst die Boot-Meldung und der Ramtest wird durchgeführt. Dann meldet mein Rechner

Boot-Versuch auf B:1A

Ist eine Diskette eingelegt, so wird gebootet, falls sie das richtige Format hat

Boot-Versuch auf B:1A ok,booten

... ist das Format der Disk in LW B: aber nicht 1A, so wird sofort eine Fehlermeldung gegeben und zum nächsten Bootversuch mit B:0A weitergesprungen:

**Boot-Versuch auf B:1A kein Bootsektor
Boot-Versuch auf B:0A**

Ist das Laufwerk leer, wartet das System ca. 15 sec. und schaltet erst dann zum nächsten Bootformat weiter.

**Boot-Versuch auf B:1A kein Bootsektor
Boot-Versuch auf B:0A Lesefehler
Boot-Versuch auf E:6F ok, booten**

Um einen Absturz des Rechners während der Bootphase zu verhindern muß ein neues SYSCOPY benutzt werden (egal ob mit RAM4 gearbeitet wird oder nicht), das zusammen mit dem Eprom auf Diskette ausgeliefert wird.

Sollten alle drei Bootversuche nicht erfolgreich gewesen sein, wird ein kleiner "Bootmonitor" aktiviert und das System wartet auf die Eingabe der gewünschten Bootparameter über Tastatur. Dies kann auch direkt durch das Drücken der Return-Taste während der Ramtest-Phase erreicht werden.

Weiterhin ist es möglich, sehr viele Fremdformate (ca. 45) **ohne RAM4** unter dem neuen Boot-System als Overlay (ähnlich dem alten OVERLAY8) einzuladen. Dies kann dann von Vorteil sein, wenn ein Ausfall der Speichererweiterungskarte den Betrieb von RAM4 unmöglich macht, man aber dennoch auf alle Software unter den verschiedensten Formaten zugreifen will. Darüber hinaus haben jetzt auch diejenigen, die RAM4 immer noch nicht besitzen, Zugang zu fast allen Formaten, die auch mit RAM4 möglich sind.

Dem Eprom liegt eine Diskette bei, die folgende Programme enthält

- XSYS : ersetzt SYSCOPY4.COM egal, ob RAM4 geladen oder nicht
- XCFG: ersetzt CONFIG.COM bei Betrieb ohne RAM4
- XOV: ermöglicht Nachladen von derzeit etwa 45 Fremdformaten bei Betrieb ohne RAM4
- BOOT3: generiert Kaltstart a la Memotech mit den normalen Formaten 0-7, 10-13 als Boot B:03 FDxB läuft nicht mit dem neuen Booteprom läuft, sondern nur über den Umweg via BOOT3.COM benutzbar ist. ROMBAS läuft allerdings mit dem neuen Booteprom fehlerfrei (sagt jedenfalls Manfred).

Alles in allem erhöht das neue Boot-Eprom den Komfort des Rechners erheblich, Leute ohne Ram4 kommen endlich von den beknackten Memotech-Formaten weg und alle die Ram4 besitzen brauchen jetzt keine Angst mehr zu haben, daß im Falle des Ausfalls der Speichererweiterung die schönen 09/0A/1A etc. Disketten unlesbar werden. Ab jetzt gibts auch keine Backup-Sammlung auf 03 oder 07 mehr und das Starten von RAM4 direkt von einem MaxBP-Format hat sicherlich auch seine Vorteile durch den Wegfall spezieller Bootdisketten im Memotech-Format.

H a r d w a r e: Booten von 09 und 1A und ...

Und nun zum Preis:

Im Austausch gegen das Original-Eprom kostet das neue Eprom DM 35,-- incl. Diskette, Verpackung und Porto.

Wer nicht austauschen will, zahlt für ein 2764 Eprom DM 45,-- incl. Diskette, Verpackung und Porto.

Ein 2564 Eprom kostet DM 35,-- zuzügl. Marktpreis eines 2564, derzeit also ca. 50,-- bis 55,-- DM incl. Diskette, Verpackung und Porto.

Interessierte CMOS-FLOPPY Besitzer mögen sich wegen einer noch zu erstellenden Version mit CMOS-Unterstützung direkt an mich wenden.

Bestellungen mit Angabe des gewünschten Suchpfades bitte an:
Michael Keßler, Röntgenweg 6, 5600 Wuppertal 1, Tel: 0202/436649
Bankverbindung: Postgiroamt Saarbrücken, BLZ 590 100 66
Kto.Nr.: 1451 58-667

Nun aber näheres zum Eprom:

Ich habe durch Vorverlegung des TOAM auf EC00H Platz geschaffen, um einen Sektorpuffer von 1K Größe unterzubringen. Diese interne Änderung des TOAM macht zumindest unter ZCPR2/P2DOS überhaupt nichts, da dieses System den TOAM sogar auf E900H setzt. Unter normalem CP/M läuft nur RAM3 nicht mehr, da dieses scheinbar den TOAM bei F00H erwartet.

Nun gab es bei der praktischen Realisierung dieses Vorgangs die folgende Schwierigkeit: War eine Diskette im Laufwerk, die zwar ein falsches Format hatte, deren erster Sektor aber dennoch einlesbar war (kommt bei verwandten Formaten vor), so führte dies immer zum Absturz des Rechners. Beispiel: Ich will auf 03 booten und eine 07-Disk ist drin. Der erste Sektor kann gelesen werden, aber dann erfolgt der Absturz weil ja eine 80Tr. Disk vorliegt. Dieses Problem habe ich gelöst, indem ich **bereits beim SYSCOPY das jeweilige Format der Zieldiskette auf der Diskette selbst vermerke**. Mir ist aufgefallen, daß im Bootsektor einer Diskette das Byte 10BH noch unbenutzt ist (es ist das Byte hinter den 8 Byte Laufwerksconfigs). An dieser Stelle wird jetzt das Diskettenformat abgelegt, so daß bei erfolgreichem Lesen des ersten Sektors ein Testen dieser Stelle über Booten oder Fehlerausgabe entscheidet.

Hardware: IM2-Interrupts und DMA für den ECB-Bus**IM2-Interrupts und DMA-Zugriff für den ECB-Bus** (Herbert zur Nedden, 2000)

Es ist schon ein Krampf, was so alles gemacht werden muß, um den ECB-Bus an und für sich zum Laufen zu bekommen. Und als dann noch die IM2-Interrupts dazu kamen wurde es spannend. Was aber nun, wenn auch noch ein ECB-Bus-Adapter mit eigenen Bustreibern hinzu soll, oder auch ein DMA-Controller auf den ECB-Bus soll (z.B. für eine Harddisk oder schnelleren Zugriff auf das Video-RAM der Edicta-Grafik-Karte)? Tja, da kommen dann die eigentlichen Probleme.

Meine Lösung besteht aus einem PAL auf der Bus-Interface-Karte in der FDX; die ECB-Option hingegen wurde beseitigt. Und die läuft auch mit 8 MHz problemlos.

Also, wie war das noch mit der Bussteuerung in der guten alten reinen MTX-FDX-Umgebung? Eigentlich richtig simpel:

Wenn der MTX sendet empfängt die FDX, und umgekehrt.

Klingt eigentlich banal. Bleibt nur zu klären, wann der MTX sendet und wann empfängt, und wie das funktioniert, wenn der Floppy-Controller per DMA auf das RAM der Z80 zugreift. Um das zu erläutern, und auch die später von mir entwickelte Lösung der ganzen Chose transparenter zu machen, möchte ich in Sachen Z80-Logik etwas ausholen:

Es geht um folgende Signale:

A0 - A15,	die Adreß-Signale
D0 - D7,	die Daten-Signale
MREQ	Low signalisiert Speicher-Zugriff
IOREQ	Low signalisiert Port-Zugriff
RD	Low signalisiert Lese-Zugriff auf Speicher oder Port
WR	Low signalisiert Schreib-Zugriff auf Speicher oder Port
M1	Low signalisiert bei Speicher-Zugriff: Befehls-Hol-Zyklus Port-Zugriff: Interrupt-Antwort
INT	Low signalisiert: Interrupt-Anforderung
BUSREQ	Low signalisiert: Bus-Anforderung
BUSACK	Low signalisiert: Bus-Freigabe

Bei Speicher-Zugriffen von der CPU liefert die CPU mit A0-A15 die Adresse, und signalisiert mit MREQ = Low und RD oder WR = Low was sie von der Außenwelt wünscht. Zusätzlich signalisiert sie mittels M1, ob Sie Daten oder einen Befehl holt (nur Lese-Zugriffe).

Bei Port-Zugriffen von der CPU (außer Interrupt-Antwort), liefert die CPU mit A0-A7 die Adresse, und signalisiert mit IOREQ = Low, M1 = High und RD oder WR = Low, was sie von der Außenwelt wünscht.

Bei der Interrupt-Antwort sieht das etwas anders aus: die CPU signalisiert mittels IOREQ = Low, M1 = Low und RD = Low, was sie von der Außenwelt wünscht: einen Port-Zugriff besonderer Art. Aber im Gegensatz zu den üblichen Port-Zugriffen liefert sie nicht die Adresse, da sie schließlich nicht wissen kann, wer den Interrupt angefordert hat. Hierbei muß der Interrupt-Anfordernde Baustein selbstständig reagieren, d.h. wissen, daß er gemeint ist.

Aus dem obigen Verhalten der CPU bei Port-Zugriffen ist ersichtlich, warum bei der Port-Dekodierung immer das Signal M1 mit einbezogen werden muß, da bei der Interrupt-Antwort auf dem Adreßbus recht willkürliche Signale anliegen können.

Hardware: IM2-Interrupts und DMA für den ECB-Bus

Bei DMA-Zugriffen ist es so, daß ein externer Baustein die Kontrolle über den Bus vorübergehend übernimmt, d.h. daß dieser Baustein an Stelle der CPU die Signale A0-A15, MREQ, IOREQ, RD und WR erzeugt. Der DMA-Baustein signalisiert mittels BUSREQ = Low, daß er die Kontrolle über den Bus haben will, was die CPU mit dem Signal BUSACK = Low gewährt.

Es stellt sich natürlich die Frage, was tut die Hardware, wenn mehr als ein Baustein Interrupts erzeugen darf, und zwei gleichzeitig wollen?

Die Z80-Hardware unterstützt die sog. Daisy-Chain:

Die Z80-Interrupt-Erzeuger haben jeder einen Interrupt-Enable-Eingang (IEI) und einen Interrupt-Enable-Ausgang (IEO). Die Interrupteure werden in einer Prioritäten-Kette hintereinandergeschaltet, d.h. der IEO des ersten wird mit dem IEI des zweiten, der IEO des zweiten mit dem IEI des dritten, u.s.w. verbunden. Jeder dieser Bausteine gibt die Information seines IEI-Eingangs an seinen IEO-Ausgang, d.h. an seinen Hintermann weiter.

Will nun einer interrupten, dann prüft er seinen IEI, d.h. schaut nach, ob er darf. Wenn nein, dann wartet er. Wenn ja, dann signalisiert er an seinem IEO, daß er dran ist, d.h. keiner hinter ihm darf, und legt los.

D.h. wenn ein Baustein einen Interrupt laufen hat, dann kann kein anderer Baustein weiter hinten in der Kette einen Interrupt auslösen. Ein Baustein, der weiter vorne in der Kette ist, also eine höhere Priorität hat, kann dazwischenfunken - d.h. den anderen Interrupt unterbrechen.

Ann.d.HzN.: Wer weitergehende Informationen zu den Z80-Interrupts an und für sich lesen möchte, dem empfehle ich Info 16 ab Seite 6.

Für DMA-Zugriffe gibt es natürlich eine ebensolche Daisy-Chain, was wir allerdings im Falle des MTX unter den Tisch fallen lassen können.

Um die Bussteuerung des MTX zu beschreiben benötige ich folgende Signale:

MREQ, IOREQ, RD, M1, A0-A7, BUSACK, BTROM und DTIED

Außer BTROM (Low = Zugriff auf Boot-ROM in der FDX) und DTIED (Interrupt-Enable-Ausgang des DART) sind dies die o.g. Signale der CPU.

Wenn die CPU die Buskontrolle hat, d.h. BUSACK = High, dann werden Daten von der FDX zum MTX transportiert, falls

- aus dem Boot-ROM gelesen wird, oder
- ein Port-Lesezugriff oberhalb von 20H stattfinden, oder
- eine Interrupt-Antwort-Zyklus läuft, der weder vom CTC noch vom DART ausgelöst wurde (d.h. DTIED = High).

Hat hingegen die CPU die Buskontrolle abgegeben (was bei MTX/FDX bedeutet, daß der Floppycontroller an der Reihe ist, dann werden Daten von der FDX zum MTX bei Schreibzugriffen transportiert.

Um das nun in PAL-Notation zu sagen:

```
/MTXDIR = BUSACK * /BTROM * /RD
          + BUSACK * /IOREQ * M1 * /RD * (A7 + A6 + A5)
          + BUSACK * /IOREQ * /M1 * /RD * DTIED
          + /BUSACK * RD
```


Hardware: IM2-Interrupts und DMA für den ECB-Bus

Das bedeutet, daß MTXDIR = Low ist (d.h. Transport zum MTX), falls

- BUSACK = High und IOREQ = Low und M1 = High und RD = Low
und (A7 oder A6 oder A5 = High)
- oder BUSACK = High und BTROM = Low und RD = Low
- oder BUSACK = High und IOREQ = Low und M1 = Low und RD = Low
und DTIED = High
- oder BUSACK = Low und RD = High

Bei dieser Notation bedeutet

'+' = ODER, '*' = UND, '/' vor einem Signal NICHT (d.h. Low).

Also ist eigentlich auf der MTX-Seite alles bestens.

Aber dummerweise ist die Bus-Steuerung seitens der FDX rudimentär:

1. Wenn MTX empfängt, sendet FDX
2. Wenn CPU den Bus freigibt (BUSACK=Low) übernimmt die FDX die Kontrolle.

Vergessen wir die Anwesenheit eines ECB-Busses ist das auch gut so, **ABER!**

Also habe ich mich entschlossen, erst einmal der FDX eine saubere Bus-Steuerung zu verpassen, die regelt, wann die FDX Daten auf den Bus geben darf, und wann sie sogar die Kontrolle über den Bus erlangen darf.

Eigentlich ist das simpel:

1. Sie bekommt die Buskontrolle, wenn Sie selbst diese verlangt hat, und die CPU diese gewährte.
2. Sie darf senden, falls sie keine Buskontrolle hat, und
 - aus dem Boot-ROM gelesen wird, oder
 - Port-Lesezugriffe auf Ports zwischen 20H und 7FH gemacht werden, (d.h. Port-Zugriff und A7 = Low und A5 oder A6 = High), oder
 sie die Buskontrolle hat, und
 - Schreibzugriffe ausführt

Um die PAL-Notation zu schreiben, benötige ich zusätzlich die Signale:

FDXBREQ = BUSREQ auf der FDX-Seite
FDXRD = RD auf der FDX-Seite

FDXBACK = BUSACK für FDX
FDXDIR = Datentransfer-Richtung für FDX (Low = FDX sendet).

$/\text{FDXBACK} = /\text{FDXBREQ} * /\text{BUSACK}$

$/\text{FDXDIR} = /\text{BTROM} * /\text{RD}$
 $+ /\text{IOREQ} * /\text{RD} * \text{M1} * /\text{A7} + (\text{A6} * \text{A5})$
 $+ \text{FDXRD} * /\text{FDXBREQ} * /\text{BUSACK}$

Zu bemerken ist, daß die FDX keine Interrupts auslöst, und daher die Interrupt-Geschichte, d.h. das Signal DTIED, hier unter den Tisch fallen kann, und es auch tut!

Hardware: IM2-Interrupts und DMA für den ECB-Bus

Falls auch der ECB-Adapter mit Bustreibern ausgestattet werden soll, müssen entsprechende Signale ECBBACK und ECBDIR erzeugt werden:

1. Der ECB-Bus bekommt die Buskontrolle, wenn Er diese verlangt hat (d.h. wenn der Bus von der CPU angefordert wurde, und die FDX es nicht war), und die CPU diese gewährte.
2. Er darf senden, falls er keine Buskontrolle hat, und
 - Port-Lesezugriffe auf Ports oberhalb von 7FH gemacht werden, (d.h Port-Zugriff und A7 = High), oder
 - ein Interrupt-Antwort-Zyklus aktiv ist, und weder der DART noch der CTC vom MTX diesen verursachten, d.h. DTIED = High ist, oder er die Buskontrolle hat, und
 - Schreibzugriffe ausführt:

$\text{/ECBBACK} = \text{FDXBREQ} * \text{/BUSACK}$

$\text{/ECBDIR} = \text{/IOREQ} * \text{/RD} * \text{M1} * \text{A7}$
 $+ \text{/IOREQ} * \text{/M1} * \text{DTIED}$
 $+ \text{/ECBRD} * \text{FDXBREQ} * \text{/BUSACK}$

Die beiden Signale FDXBACK und FDXDIR müssen wie folgt verwendet werden:

1. FDXDIR steuert die Übertragungs-Richtung, d.h. den DIR-Pin des entsprechenden 74 LS 245.
2. FDXBACK steuert die Übertragungs-Richtung der Treiber für den Adreßbus sowie die Signale MREQ, IOREQ, RD und WR.
 Außerdem wird das BUSACK-Signal in der FDX nicht vom Bus genommen, sondern durch FDXBACK ersetzt.

Die Signale ECBDIR, ECBBACK müssen bei einem eventuell vorhandenen Bustreiber-satz auf dem ECB-Interface entsprechend verwendet werden.

Es ist jedoch auf jeden Fall sicherzustellen, daß Diskettenzugriffe und andere DMA-Zugriffe seitens des ECB-Bus nicht gleichzeitig auftreten können.

Ich habe ein PAL gebrannt, und auf meiner Bus-Interface-Karte in der FDX eingesetzt, welches die o.g. Signale FDXDIR, FDXBACK, ECBDIR und ECBBACK erzeugt, und meine Kiste funktioniert einwandfrei. Meine ECB-Option habe ich natürlich entfernt.

Hier die Pin-Belegung des PAL's:

Input	IOREQ	-	1	20	-	VCC	
Input	RD	-	2	19	-	DTIED	Input
Input	M1	-	3	18	-	BTROM	Input
Input	FDXRD	-	4	17	-	ECBBACK	Output
Input	ECBRD	-	5	16	-	ECBDIR	Output
Input	A4	-	6	15	-	FDXDIR	Output
Input	A5	-	7	14	-	FDXBACK	Output
Input	A6	-	8	13	-	BUSACK	Input
Input	A7	-	9	12	-	FDXBREQ	Input
	GND	-	10	11	-	NC	Input

Hardware: IM2-Interrupts und DMA für den ECB-BusDie Signale:

1. Vom MTX-FDX-ECB-Flachbandkabel, also vom Bus, abgegriffen:

IOREQ, RD, M1, A4-A7, BUSACK	CPU-Signale
BTROM	BTROM
DTIED	DTIED
2. Vom FDX-BUS abgegriffen:

FDXRD	RD
FDXBREQ	BUSREQ
3. Vom ECB-BUS abgegriffen:

ECBRD	RD (falls <u>keine</u> Treiber auf ECB-Adapter, nimm CPU-RD vom Bus)
-------	----------------------------------------------------------------------
4. Ausgänge des PAL's:

FDXBACK	BUSACK fuer FDX-BUS
FDXDIR	Richtungssteuerung fuer FDX-Interface welches an Stelle des BUSACK verwendet werden muß!
ECBDIR	dto. für ggf. entsprechendes ECB-Interface
ECBBACK	dto. für ggf. entsprechendes ECB-Interface

Die Gleichungen:

$$/FDXBACK = /FDXBREQ * /BUSACK$$

$$/ECBBACK = FDXBREQ * /BUSACK$$

$$\begin{aligned}
 /FDXDIR &= /BTROM * /RD \\
 &+ /IOREQ * /RD * M1 * /A7 * A6 \\
 &+ /IOREQ * /RD * M1 * /A7 * A5 \\
 &+ FDXRD * /FDXBREQ * /BUSACK
 \end{aligned}$$

$$\begin{aligned}
 /ECBDIR &= /IOREQ * /RD * M1 * A7 \\
 &+ /IOREQ * /M1 * DTIED \\
 &+ /ECBRD * FDXBREQ * /BUSACK
 \end{aligned}$$

Logik / Idee in Kürze:

BUSACK an FDX nur, wenn BUSACK von CPU + BUSREQ von FDX!!! Damit besteht die Chance, dem ECB auch BUSREQ zu gestatten (DMA). Da keinerlei Interrupts von der FDX stammen, schlägt sich dies nicht bei der Dekodierung der FDX-Steuersignale nieder.

- Vorteil: 1. ECB-Option und ähnliche Konstruktionen sind überflüssig!
 2. ECB wird so DMA-fähig (und natürlich Interrupt-fähig)!

Für Einsatz einer Z280-Karte auf dem ECB-Bus muß eine etwas aufwendigere Logik auf der ECB- und MTX-Seite her! Jedenfalls arbeitet meine FDX (BOZ / FDC) jetzt völlig autonom ohne bei Interrupts oder DMA irgendein Fehlverhalten zu zeigen.

Einbau:

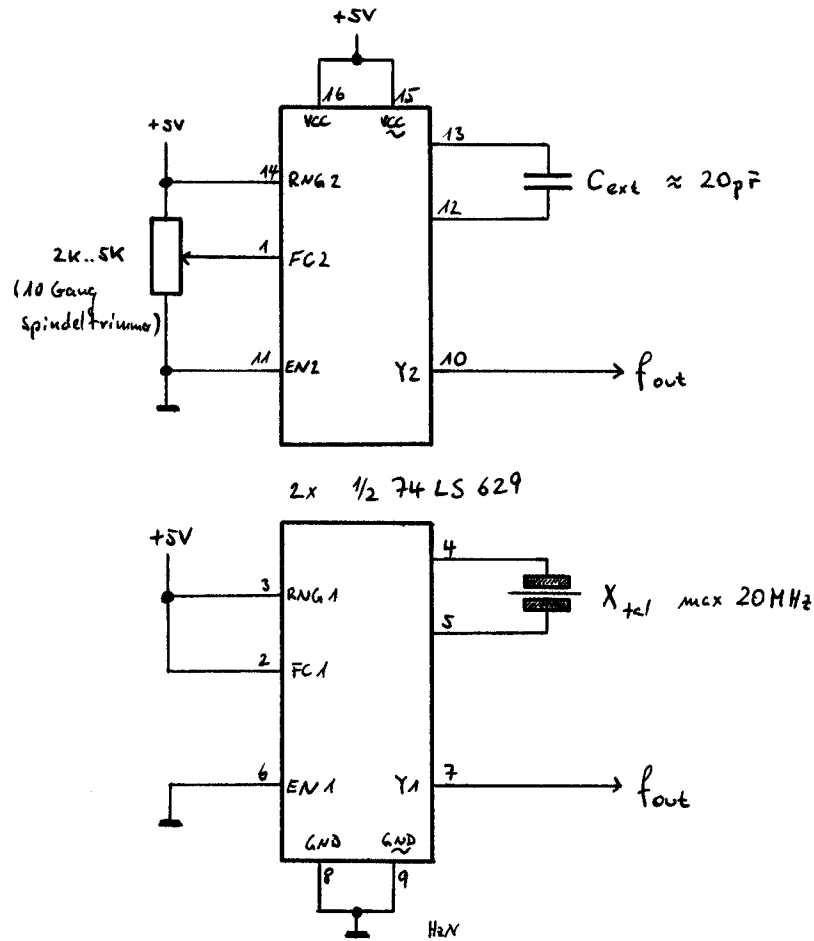
Ich habe auf der FDX-Interface-Karte 20 Löcher gebohrt, (mußte daher 3 Leiterbahnen per Kabel nachverdrahten) und zwar direkt eben dem 20-poligen IC, welches dem Boot-EPROM am nächsten ist, und dort mein PAL (14L4) eingesetzt. Die Chose läuft 100%-ig mit 8 MHz bei mir seit einiger Zeit. Vom einen LS244, der das BUSACK in die FDX weitergibt muß die FDX-Seite abgeknipst werden, da dieses Signal für die FDX aus dem PAL kommt.

Hardware: Variable Taktfrequenz

CPU-Frequenz-Feineinstellung

(Manfred Flume, 5600)

Mit einem VCO (Voltage Controlled Oscillator = Spannungsgesteuerter Oszillator) vom Typ 74 LS 629 können leicht zwei Taktgeneratoren aufgebaut werden. Dabei ist es sogar möglich, diese regelbar (über ein Poti) oder auch mit einem Quartz aufzubauen. Am besten nimmt man einen regelbaren für die CPU-Frequenz, und lößt den anderen mit 4 MHz für die RS 232 quartzen.



Dies ist ein Schaltungs-Beispiel für je 1x Kondensator- und 1x Quartz-Betrieb, beides in einem IC. 74 LS 629. Daher die GND und VCC nur einmal. Die Oszillatoren können selbstverständlich getauscht werden. Wird RNG2 ebenso wie FC2 über ein Poti gesteuert, erhöht sich der Frequenzbereich (in für unsere Zwecke unnötiger Weise) drastisch.

CP / M: Nsweep-Macke / Leserbrief: Kurt-Bernd Rohloff, 8000

Was ist, wenn dich deine Leibwache erschießt?

(Peter Würfel, 7262)

Die NSWEEP- (oder WASAUCHSONSTIMMER-) MACKE:

Das Verify in nsweep erzeugt unter bestimmten Umständen Schrott auf einer Diskette im Format 09: die Diskette kann auf bestimmten Spuren nicht mehr gelesen werden (d.h. nsweep bricht ab mit blinkendem Fenster genauso bei beim späteren 'rcheck' mit Angabe: 'Disk-FehlerLESEN', '....', 'CRC-Fehler') (alle folgenden 'k'-Größenangaben sind durch Versuch gefundene Werte, keine Grenzwerte!)

Die Bedingungen unter denen der Fehler auf- bzw nicht tritt

(Ich glaub mich tritt...)

- Kopieren von RAM-Disc A: nach C:(09)
- lange Datei (150k)
- Fehler tritt auf je nach Lage der Datei auf A: !!!
 - wenn die 150k-Datei an erster Stelle auf A: abgelegt ist, (die Datei muß 'am Stück' abgelegt sein, es reicht nicht, wenn sie nur den ersten directory-Eintrag belegt), dann tritt der Fehler nicht auf.
 - Der Fehler tritt auch nicht auf, wenn 1 bis 2 Dateien davor sind. Diese Dateien dürfen jedoch nicht zu groß sein.
- Die Lage der Datei auf der Zieldiskette scheint keine Rolle zu spielen.

Uwe Grass hat auf seiner Maschine und auf einer anderen, die er gerade zur Reparatur hatte, mit den Dateien und in der Konstellation, in der bei mir der Fehler auftrat, das ganze auch ausprobiert: auf seiner Maschine trat der Fehler nicht auf, auf der anderen Maschine nicht immer und auch nicht zu den von mir ausprobierten Bedingungen!

Kopieren mit PIP (Würg) und Verfiy klappt! Übrigens, meine Maschine läuft mit 5 MHz, ist aber nicht mehr auf 4 MHz umschaltbar.

Leserbrief von Kurt-Bernd Rohloff

Lieber Herbert!

Clubtreffen:

Das "münchner" Clubtreffen fand diesmal in Burghausen statt. Wir waren in den Räumlichkeiten eines Ingenieurbüros, bei dem Gunter Ebert arbeitet. Von daher bildeten die dort herumstehenden CAD Workstations natürlich einen reizvollen Blickfang. Was den MTX betraf, so war der Hit des Tages wohl die ECB Bus Adapterkarte von Gerhard Witzel aus Erlangen. Ebenso fand seine Hardware-Uhr großes Interesse. Oliver Knapp zeigte uns seine umgebaute FDX. Sie nimmt bei ihm auf einem stabilen Metallrahmen die Hauptplatine auf (oben über den Laufwerken). In Sachen Software demonstrierte uns Holger Göbel aus Coburg die Vorteile von RAM4. Unser "Neu-Münchner" Ulli Hönisch führte uns NWPATCH vor. Daß das Programm was geändert hat, war offensichtlich, aber keiner wußte so recht, was! Henning Herder zeigte uns noch ein MS-DOS Programm (aus Chip Special MS-DOS 1), daß CP/M Disketten lesen, schreiben und formatieren kann (letzteres allerdings noch nicht für unser MTX 03 Format). Schließlich habe ich noch meine neue dBASE Anwendung zur Verwaltung von Tonaufzeichnungen (im Moment nur Tonbänder, wird aber erweitert auf LP's und CD's) vorgeführt.

Assemblerkurs:

Gähnende Leere! Ich soll aber etwa gegen Pfingsten einen Beitrag von Andreas Nickel bekommen. Wenn alles klappt, könnte er bis Ende Mai bei Dir sein.

Mit freundlichen Grüßen

Kurt-Bernd

Leserbrief: Hans Gras, NL-1506

Bester Club-Coordinator,

Anbei einige Programmen für die -CLUB.03? diskette. Es sind Catalogprogrammen, weil ich auf meine Frage in Info 23, Seite 5, keine Reaktion bekam. Die programmen **NEWCAT4** und **CAT4** sind meine neueste Versionen von **NEWCAT** bzw. **CAT**. Ich habe dafür die alte Programmen disassembliert und erweitert mit neue (klare) Fehlermeldungen und ein **ZCPR 2** Suchpfad. Das Suchpfad is normalerweise extern aber es kan intern installiert werden (mit **GENINS**). Beide Programmen suchen das benützte **MAST.CAT** file mit alle Daten via das Suchpfad. Ich habe meine Privat Exemplar, z.B. folgendes installiert:

Externes Suchpfad **A0: E0:** und meine Catalog Programmen sind auf **E0:**. Wann ich eingelogd bin auf **D5:** und eingebe **NEWCAT4 <RET>**, dann wird **NEWCAT4** über **D5:**, **A0:** und **E0:** auf **E0:** gefunden und gestartet. **NEWCAT4** sucht dan über seines Internes pfad (nur auf **E0:**) und findet dort **MAST.CAT**. Dieses **MAST.CAT** wird dan Updated mit die Daten aus **D5:**. Das gleiche gilt für **CAT4**. Wann ich wieder ins **D5:** bin und ich gerne ein Übersicht habe wollte von meiner **MBASIC** Programmen, dann type ich **CAT4 *.BAS <RET>** und via **D5: A0: E0:** wird **CAT4** gestartet, sucht intern über **E0:** das Datenfile **MAST.CAT** und gibt alle ***.BAS** Files aus, mit ein angabe auf welche Diskette ich das habe.

Weitere Informationen mit **NEWCAT4 //** bzw. **CAT4 //** (in Englisch). Was ein bißchen extra Info braucht ist die Name einer Diskette. Eine Name fangt immer an mit eine "-". z.B. **-NAME.001**. Ich habe eine Vorschlag für einige Klarheit:

-DSKNAME.100

| \ /

| \ _ Angabe in welche USER wir sind auf die Diskette.

| \ _ _ Angabe Reihenummer, wann wir z.B. 3 (M)BASIC Disketten haben.

Das heißt: **-GAMES.203** ist 2^e **GAMES** Diskette und Files sind ins **USERbereich 03**.

Weitere Programmen:

PUTCAT: ist einer der original Programmen. Es gibt die daten aus auf **CRT:**, **LST:** und/oder **DiSk** aus. Versuch mal (ohne Suchpfad).

PUTCAT26: ist ein meiner compiliertes **MBASIC** Programmen. Es braucht **BRUN.COM** (auch dabei) und **MAST.CAT**, versteht kein Suchpfad, d.h. ich muß ins **USER E0:** sein und dan das Programm starten. Es gibt **RUN-time Options:** **PUTCAT260 <RET>** gibt Microline OKI84 Control Codes aus, **PUTCAT26D <RET>** gibt (hoffe ich) die DMX-80 Control Codes aus, **PUTCAT26P <RET>** gibt Control Codes für meine Panasonic KX-P1092. Das Default ist (selbstverständlich) die "P". Bitte mir anrufen oder anschreiben wann ihre Printer nicht dabei ist (die **DMX-80** ist nicht getestet worden!). Die Printer Daten werden dann sofort in meine **Sorce-Programm** eingetragen und zur verfügung gestellt.

CAT, CAT2: sind auch einige der original Programmen. Versuch es auch mal.

OLDCAT: ist auch ein meiner compiliertes **MBASIC** Programmen. Es braucht auch **BRUN.COM** und **MAST.CAT**, versteht kein Suchpfad, d.h. ich muß in das **USER E0:** sein und dan das Programm starten. Es braucht ein weile Zeit, aber es gibt ein neues File "**OLDCAT.CAT**" aus, mit ein sortiertes Übersicht jeder Diskette aus.

MASTSTAT: auch **MBASIC**. Es gibt ein kleines Übersicht von Zahl der Einträge jedes Buchstabe (ins File "**MASTSTAT.CAT**") und aufs Schirm.

P.S.1 Die original Programmen laufen auch ohne **ZCPR 2 / P2DOS / RAM 4.2**.

P.S.2 Die original Programmen sind von **Ward Cristensen**.

Leserbrief: Hans Gras, NL-1506

Neu!

Speziel geschrieben, weil auch hier keine Reaktion kamm auf meine Frage (Info 23, Seite 5) für Leute mit Ideen für 40-Zeichen VDP benützung mit MBASIC.

Programm: **VS4.BAS**.

Eine kleines MBASIC Programm. Mal starten mit ein normales FDX oder RAM-System (ohne ein laufende CRT40, sehe unten). Dies war ein erster Versuch. Es läuft, aber ist siemlich langsam.

Fürs KLICK.00? habe ich danach auch etwas gebastelt. Dafür anbei meine 40-Zeichen-Treiber, welche ich geschrieben habe für (M)BASIC(ode) zwecke.

Es wird auf folgende Weise gestartet:

<u>CRT40</u> <RET>	(Ins Startup vielleicht?)
shift esc	(Schöne Tasten Combination)
F taste für CRT40	(16 möglichkeiten)
<u>IOLOADER</u> CRT40	(Tolles Programm)
<u>DEVICE</u> CON:=CRT40 <RET>	(Brauche ich sehr oft für LST:)
	(mal ansehen mit DEVICE DISPLAY ALL)

Zürück schalten mit DEVICE CON:=CRT.

Wie es funktioniert:

Beim starten von CRT40 (dank an Olaf Krumnow für seine Hilfsprogrammen zur erzeugung von KLICK Programmen) wird das Treiber ergendwo in HEAP eingelagert. Wann es Initialisiert wird (shift esc, Fx), sucht das Programm Raum zwischen (FREE) und (TOAM), hängt sich darein, stelt (FREE) auf ein neue (hohere) wert ein, initialisiert das 40-Zeichen VDP genau wie das MTX das macht, Cleart das Schirm und gibt einige Texten aus. Als die freie Speicherplatz nicht vorhanden ist, folgt eine Fehlermeldung und passiert nichts weiter (ausser die VDP initialisierung).

Danach muß ein neue IO-Treiber geladen werden (IOLOADER CRT40). Die IO-Treiber testet ob CRT40 seine BSR direct unten (!) die neue (FREE) gehängt hat, und gibt die Daten über an die Bank-Switch Routine (BSR). Wann kein BSR vorhanden ist, dann werden die Daten auf normaler Weise ans 80-Zeichen Karte übergeben, egal was mit DEVICE CON:=CRT oder CON:=CRT40 eingestellt ist.

Ich hoffe das diese Programmen einwandfrei läufen, aber es ist mir klar das es Problemen gibt wann es zusammen gebraucht wird mit andere 40-Zeichen Routinen, z.B. vonaus TurboPascal. Es ist nicht speziel für Turbo geschrieben! Meine Routine fangt auch die meiste (alle?) RAM 4.2 Schirm-SteuerCodes ab. Das bedeutet das eine Sequenze wie ESC Ä W x,y,b,h kein Mühl bringt auf das 40-Zeichen Schirm. Die Sequenze SHIFT-ESC Fx bringt ein zerstörtes 40-Zeichen Schirm wieder auf die Beinen (ja, nochmal starten!).

Für wünschen (oder Problemen) bitte zu mir wenden.

P.S. CRT40 hat seine eigene Stack unterhalb (TOAM)! So, nicht weiter benützen.

Dann habe ich noch etwas. Wegend die minimale Interesse von die Deutsche Mitglieder, habe ich entschlossen um meine BASICODE Programmen ins Public Domain zu geben. Leider hat nur 1 Person reagiert (für eine ATARI !!!!!). Ich habe selbstverstandich keine Deutsche Betriebsanleitung geschrieben. Es gab mir zu wenig spaß! Aber, eine Holländische Anleitung ist dabei fürs Leute die diesmal intressiert sind. Bitte erst nochmal Info 21, Seite 9 lesen.

Leserbrief: Hans Gras, NL-1506 / Michael Kessler, 5600

dBASE II

Und dann, das (hoffentliche) gute Antwort auf ihre Frage: Warum gehen bei ihre Ausdrücke von Mitglieder Listen einige Sachen falsch? Ihre Idee von .NDX Problemen sind wahrscheinlich richtig!!! Ich habe aufs Diskette das Programm PUTEOF.COM anbei (kein Programm von mir). Das muß die (unschöne) Lösung sein.

Ann.d.HzN.: PutEOF stellt sicher, daß am Ende der Datei mindestens ein ^Z (1AH) steht.

mit freundlicher gruß, Hans Gras

Leserbrief von Michael Kessler

Lieber Herbert!

Jetzt nochmal kurz zu meinem letzten Brief: Es ging da z.B. um die fehlenden **WD-Chips** 1691 und 2143. Diese IC's können nicht nur die Datensicherheit bei DD verbessern, sie tun das tatsächlich. Manfred Flume hat bei allen DD-Formaten weniger Ausfälle als früher (beim 80-Spur Laufwerk). Er will die Precompensation jetzt auf alle seine Laufwerke ausdehnen. Und die TDK-HD Disks sind wie gesagt echt **MIST**, sie funktionieren weder bei mir noch bei Manfred ohne Precompensation.

Thema **Formstar**: Bei mir lautet die Versionsnummer jetzt 1.16, nachdem ich erstens Fehler bei den Formaten 11 und 13 beseitigt und zweitens nochmal einige Formate eingebaut habe, als da sind 1C, 1D, 6F, 93, AB, FA. 1C und 1D sind die Mammutformate von Herbert Oppmann jeweils ohne und mit Systemspuren. 6F ist das bereits erwähnte halbe #19 (einseitig), 93, AB und FA sind ZDOS-Formate, wobei das FA (ZDOS 80 Tr. DD/DS) neu ins FORMATE.DAT aufgenommen wurde.

In diesem Zusammenhang gleich eine große Bitte:

Du solltest im Info eine neue Rubrik schaffen, ähnlich derjenigen der bereits vergebenen Portadressen, in der die neu zugewiesenen Formatnummern bekanntgegeben werden. Da einige davon jetzt in einem BOOT-EPROM eingebaut sind, wird diese Liste zwingend nötig. **Insbesondere die Formate 1D und 6F müssen allgemein geschützt werden.** Hier jetzt eine kurze Liste der Formate, die hier in Wuppertal derzeit in Verwendung sind:

- 1C = MaxHD 8" 80T ohne sys
- 1D = MaxHD 8" 80T 2sys
- 6F = MaxBP 8" 77T ss 2sys
- 70 = MaxBP 8" 77T ss ohne sys
- B5 = MaxBP 5" 80T ohne sys
- FA = HKM-ZDOS 5" 80T DS

Nun will ich auch eine Bitte wiederholen (Manfred denkt übrigens genauso): Bitte mehr **Programmbesprechungen** ins Info !!! Egal ob Lob oder Kritik, jedes Clubmitglied sollte wissen, was sich hinter diversen Angeboten verbirgt und wie Leute urteilen, die es getestet haben. Das gilt z.B. für Formstar und für Manfreds Plan und Klassenarbeitsaufgabenzusammenstellprogramm.

Das war's wieder mal aus Wuppertal. Tschüß !

Michael

Leserbrief: Dieter Ritter, 8520

Dieter Ritter, 8520 _____ 210488

In letzter Zeit hat sich meine FDX so oft aufgehängt, daß ich mir ein paar Gedanken ob ihrer Zukunft machte. Die Schrottplatzidee siegte nicht, sodaß ich mich an einen Umbau machte. Hier also die Erlanger Umbaustory, die nur ergänzende Tips bringen soll, dazu einen Vorschlag zur Rechnerkopplung mit 8085/280 Systemen.

- 1) **WARNUNG!!** Aus Kosten- und Bequemlichkeitsgründen habe ich statt mit einer Reinigungsdiskette (es kam sehr oft und beliebig Schrott) mit einem Spiritus-Wattestäbchen den Schreib-/Lesekopf gereinigt. Beim Zusammenbauen des Kopfes merkte ich, daß selbiger nicht fest justiert ist; man braucht viel Glück, um die Spuren wieder einlesen zu können. (Epson, 40-Spur)
- 2) Im Wesentlichen habe ich die FDX so umgebaut, wie es der Umbau von Michael Kessler, Info 21/33, beschreibt. Ich kann nur alle ermutigen, dasselbe zu tun, denn erstens ist es eine Wohltat, die FDX bis auf den letzten Stecker zerpfückt vor sich zu sehen, und zweitens hinterher ein schöneres Arbeiten. Es sollten sich aber wirklich nur Bastler ranwagen, die ein wenig Erfahrung haben, denn beim Löten einer 60-pol. Leiste kann ein Fehler das AUS bedeuten, für den absoluten Laien ist zumindest die Fehlersuche ein großes Übel. Es sei nochmals drauf hingewiesen, daß beim Entfernen des Netzteils auf der MTX-Platine der R60 und die ZD3 (Handbuch S.199) unbedingt drauf bleiben müssen, sie machen aus den -12V die -5V, die für die VS4-Videorams lebenswichtig sind. Eine sehr gute Idee war, die jetzt am FDX-Deckel hängende MTX-Platinen mittels Gewindeschneider ohne Durchbohrungen zu befestigen. Mit der Schraubengröße M4 hatte ich Schwierigkeiten, den Schneider kann man sich sparen, indem man in die U-Schiene 3,5mm bohrt, und eine M4 Schraube mit Kraft reindreht. Den FDX-Slot habe ich nicht versetzt, sondern die obersten Zentimeter abgesägt. Somit habe ich möglichst wenig verändert. Es empfiehlt sich (für eigene Erweiterungen) die restlichen Kontaktleisten gleich mit einzulöten. Mittels Relais wird zwischen VS4 und 80Z umgeschaltet, mit einem Poti in der 80Z-Leitung können gleiche Helligkeiten eingestellt werden. Das Relais kann jetzt über einen Port oder über einen Schalter an der Tastatur (das Kabel ist 2 lang) betätigt werden. Die Umbauphilosophie lag in einer möglichst großen Modularisierung, um ein universelles System zu bekommen. So gibt es jetzt eine Centronics, den FDX-Bus, Port 7, einen Key-Port etc. als Einbaubuchsen (dafür wimmelt es im Kasten nur so von Leitungen). Sollte dereinst ein größerer Rechner mein Zimmer zieren, kann ich den FDX immer noch z.B. via RS 232 als Terminal oder Subrechner verwenden.
- 3) In diesem Sinne ist auch die Anbindung des MOPPEL-Systems der Elo zu verstehen. Es sind sicher viele Clubmitglieder, die von solchen Selbstbau-Platinen-Computern (8085A) auf den Memotech gewechselt haben. Diese Computer haben viele interessante Schnittstellen, die man noch einfach (falls es Ports sind) nutzen kann. Einfach Rechnerplatine raus und MTX-Bus rein (ganz so naiv natürlich nicht). Für Interessenten habe ich eine Kopplungsplatine MTX --> MOPPEL entworfen. Noch besser ist es, mittels des BUSREQ-Signals die CPU auszuschalten (sie tut dann so, als wäre sie nicht vorhanden, also Ausgänge Tri-State). Das kann so aussehen, daß, sobald der Bus-Stecker an die FDX kommt, der Platinencomputer als Hardwareerweiterung benutzt wird (z.B. I/O immer aktiv, A8-A15 gesperrt), ansonsten (ohne Umbau) als eigenständiger Rechner. So etwas werde ich demnächst für den MOPPEL fertigstellen.

Leserbrief: Andreas Fischer, CH 4303

MTX als Wettercomputer

Eine sinnvolle und interessante Einsatzmöglichkeit für Computer bieten die Erfassung, die statistische Auswertung und Prognose des Wetterablaufes. Auch bei dieser Gelegenheit zeigt es sich wieder, dass unser altehrwürdige MTX noch voll aktuell sein kann. Hardwaremässig kommen bei mir der Wettercomputer der Firma ELV aus D-2950 Leer und der MTX 512 mit 3,5"-Floppy und RAM-Disk zur Anwendung. Im gegenwärtigen Stand habe ich nun alle relevanten Daten in der Kiste. Jetzt geht es an die Software-Entwicklung. Doch darüber später. Vorerst möchte ich als Beispiel zeigen, wie ich die Datentransfers hardwaremässig organisiert habe. Vielleicht dienen diese Beschreibungen als Anregung, es auch zu versuchen. Bei der ganzen Entwicklungsarbeit profitierte ich viel von den Club-Infos! Rückblickend kann ich festhalten, dass dieser Anwendungsbereich fast unbeschränkte Hobby-Möglichkeiten bietet, sowohl aus der Sicht der Wetterkunde, wie auch Mathematik, Software und Elektronik.

Wettercomputer WS 7000 der Firma ELV am geeignetsten.

Die Prüfung verschiedener Angebote führte rasch zum Produkt der ELV (Siehe Beschrieb in der Zeitschrift ELV Nr.42-44). Für ca. 1000 DM erwarb ich einen Bausatz des Wettercomputers. Nach 15 stündiger Lötarbeit (nein-nicht am Stück!) reduzierte sich die riesige Transportschachtel auf eine 29x15x8 cm grosse Kiste mit kompakter und hochwertiger Elektronik.

- Temperaturerfassung aussen und innen
- Feuchtigkeitserfassung aussen und innen
- Luftdruck auf millibar genau(!)
- Sonnenscheindauer
- Windgeschwindigkeit
- Windrichtung

Der Bausatz ist gut beschrieben und eigentlich recht einfach. Eine gewisse Routine für Lötarbeiten ist allerdings erforderlich. Die wenigen Schwachstellen wie fehlende Stecker und Sockel etc. sind bald vergessen. Erwähnenswert ist noch die absolut professionelle Konstruktion der Windgeschwindigkeits- und Windrichtungserfassung, die seither ihre Runden hoch über dem Dach dreht. Absoluter Hit bezüglich berührungsloser Erfassung und Zuverlässigkeit!

Problemlose Inbetriebnahme

Nach der erwähnten Lötarbeit konnte bereits der erste Probelauf stattfinden. Die 24 7-Displays(!) leuchteten auf Anhieb und liefern seither -nach einer aufwendigen Eichphase- laufend Informationen über das Wettergeschehen. Es handelt sich hier um eine absolute Supermaschine die jedem weiterempfohlen werden kann! Das mitgelieferte Interface überträgt die 32 Zahlen auf eine 8-Bit-Schnittstelle mit den notwendigen Übertragungsleitungen wie Strobe, Busy, PE und GND. Der Wettercomputer arbeitet nun schon ein Jahr ohne nennenswerte Probleme.

Anschluss an dem MTX

Leserbrief: Andreas Fischer, CH 4303

ist der freie Steckplatz DIL 20 in der schwarzen Kiste). Das gab mir allerdings später ein paar Probleme mit der Hardware-Uhr, aber darauf später. Die Signale des Wettercomputers liegen auf einer 8-Bit-Leitung an. Da ich noch weitere Wettererfassungsgeräte an dem Port angeschlossen habe, liegt nach Port 7 ein Europakarten-Träger für 24 Einschubkarten. Mit einer Decodierung über den Baustein 74154 schalte ich jeweils eine Platine ein, die dort anstehenden Signale werden dann über Port 7 gelesen und ausgewertet. So habe ich neben dem Wettercomputer noch Erfassungen für Blitz und Donner, analoge Sonnenscheinintensität, einen Regendetektor und noch in Arbeit eine Erfassung der Regenmenge über eine elektronische Waage der Firma Soehnle (Abgriff der 7-Segment-Displays mit dem Baustein 54915). Natürlich mit dazwischengeschalteten Treibern 4050. Am Anfang stellten sich Probleme mit der zeitlichen Abfolge der Signale. Seit aber das Signal IN vom Baustein 74373 des MTX (links unten, Anschluss 1) über ein 7474 auf das BUSY des Wettercomputers geführt wird (das STROBE des Wettercomputers geht auf das Clock des 7474), funktioniert die Uebertragung auf den MTX absolut problemlos. Falls jemand ähnliche Probleme hat, sende ich gerne die entsprechenden Skizzen.

Probleme mit der Hardware-Uhr

Anfänglich habe ich mit mit der ROM-Pascal-Platine gearbeitet. Die zeitliche Erfassung der Wetterdaten erfolgte über den eingebauten Timer. Mit der Erweiterung auf CP/M fiel diese Zeitquelle weg, was mir anfänglich grosse Probleme bereitete. Für die Wettererfassung war ich absolut auf Zeitangaben angewiesen. Die Hardware-Uhr von Herr zur Nedden benötigte alle OUT's des Port 7. Was tun? Für einen neuen Port mit dem Baustein 8255 war ich zu faul, da ich zudem alle Stecker mit dem Floppy und der 80-Z-Karte verlötet habe (auch hier konnte ich dank dem Club weitere Abstürze verhindern). Wer ähnliche Probleme hat, soll das RPB Taschenbuch Requardt: Neue Aufgaben für der HC konsultieren. Das write-Signal wird ja nur für das Stellen der Uhr benötigt. Deshalb habe ich dieses Signal abgehängt und leite es nun auf einen Umschalter. Wenn der Schalter auf lesen steht (fast immer, da die Uhr recht genau geht), brauche ich das Signal out5 um meine Platine einzuschalten (OUT #20). Gleichzeitig verhindere ich, dass die Uhr unbeabsichtigt verstellt wird. Wenn ich die Uhr stelle, lege ich den Schalter um und das Signal liegt wieder auf write. Im Lesezustand versorgt der Umschalter das Signal write mit GND. Die Uhr-Ausgänge entkopple ich mit dem Baustein 4066 von den andern Leitungen.

noch ungeahnte Entwicklungsmöglichkeiten

Diese Hardware-Erweiterung bietet viele weitere Möglichkeiten für die Entwicklung von elektronischen Schaltungen und Programmen. Zugleich profitiere ich von den neu erworbenen Kenntnissen über die Meteorologie (der nächste Sommer kommt bestimmt..). Ich möchte mit meinem Beitrag zeigen, dass unser Memo noch für vieles zu gebrauchen ist. Und dass auch Laien auf der relativ unempfindlichen Anlage Eingriffe vornehmen können.

Leserbrief: Stefan Hößler, 1000

Lieber Herbert!

nach so langer Zeit also endlich mal wieder ein Leserbrief von mir. Das Lob an den Club spare ich mir, denn das unser Memotech ohne den Club noch in der Steinzeit wäre, weiß eh jeder.

Nun möchte ich zuerst mal meine Erfahrungen zu dem Thema **NewWord und der DMX 80** niederlegen, obwohl dieses Thema für die meisten Leute im Club ja bereits erledigt zu sein scheint. Ich bin jedoch jetzt erst auf die Lösung gekommen und vielleicht gibt es ja irgendjemanden, der immernoch danach sucht.

NewWord (V 2.02) bietet ja für den DMX zwei Druckertreiber an:

- den Draft-Treiber: dieser verarbeitet die Drucker Customs (^W,^Q...) gut, weil er sie bereits innerhalb einer Zeile auswertet. Er unterstützt jedoch keine .cw,Superscript,Subscript.
- den Epson-Treiber (Epson MX80/100 & FX80/100): Er unterstützt die .cw,Superscript,Subscript. Bei den Customs versagt er jedoch, weil diese erst nach einem Zeilenende zur Wirkung kommen.

Nun gibt es auf der CLUB.009 einen Druckertreiber DMX 80. Diesen habe ich bereits vor längerer Zeit ausprobiert, hatte jedoch ein neues Problem: Ich habe meinen Drucker über den DIP-Schalter auf Elite-Schrift eingestellt und bei den obengenannten Treibern (draft,Epson) wird auch Elite gedruckt. Wenn ich aber den DMX 80-Treiber benutze druckt er PICA!

Die Lösung habe ich nun endlich im NWINSTAL gefunden:

Im Menü Special Patches, Untermenü 3 gibt es zwei Unterpunkte die da heißen: Width for normal pitch

Width for alternate pitch

Der erste ist eingestellt auf Hex 0C, d.h. Dezimal 12, also Pica.

Der zweite ist eingestellt auf Hex 0A, d.h. Dezimal 10, also Elite.

Wenn man nun diese beiden Werte vertauscht druckt der DMX 80-Treiber endlich in Elite. Bei den anderen Treibern scheinen diese Voreinstellungen jedoch nicht beachtet zu werden. Wenn man den Treiber aber erstmal richtig eingebaut hat, kann man sehen, daß er die Vorteile der anderen beiden Treiber beinhaltet und obendrein auch vernünftig Unterstreichen kann.

Dazu noch eine Frage: zu irgendwelchen Druckertreibern ist einmal die Bemerkung gefallen, diese seien Grafikfähig. Wie druckt man denn mit NewWord eine Grafik ??

Nun komme ich zu einer anderen Sache, nämlich **NWPATCH von KCLICK.003**. Zunächst einmal finde ich ein Programm mit solchen Ambitionen sehr begrüßenswert, also habe ich es einmal auf mein NewWord losgelassen (bereits installiert, bereits gepatcht für RAM4-Funktionstasten/Spaltenblöcke mit Hilfe deiner Patchprogramme).

Folgende Dinge funktionieren bei mir nicht:

- Der Patch für NWSCHIRM. Wenn ich NWSCHIRM benutzt habe, und in den 80 * 25 Modus zurückgekehrt bin, dann sind folgende Fehler aufgetreten:
 - 1) erscheinen beim Blättern (^C/^R) zwischendurch immer irgendwelche Textfetzen, die erst nach Druck auf die <ESC>-Taste verschwinden.
 - 2) Wenn ich im Text ein Punkt-Kommando habe, so steht der Punkt ~~bei 191 in der ersten Spalte der betreffenden Zeile, sondern~~ ~~der letzte Spalte der vorhergehenden Zeile.~~

Leserbrief Stefan Höfler, 1000

- Der Patch mit dem Drucker-Init-String zeigt überhaupt keine Wirkung.

Da Du in der Beschreibung zu NWPATCH geschrieben hast, das diese Patches bei bereits verpanschten NewWords mißlingen können habe ich Probeweise mal ein uninstalliertes, ungepatchtes NewWord durch NWPATCH bearbeiten lassen. Die Fehler waren genau die gleichen !

Ich habe noch eine Idee für einen weiteren Patch, welchen NWPATCH gleich mit erledigen könnte: Im Menü Print Controls gibt es eine Spalte der folgendes steht:

Q custom 1
W custom 2
E custom 3
R custom 4

Praktisch wäre es nun wenn ein Patch die Texte 'custom 1' bis 'custom 4' durch eigene Texte ersetzen würde, die angeben was diese Zeichen bewirken.

Sehr gut gefällt mir das Programm **BRADFORD** (CLUB.027) das endlich NLQ-Druck auf unseren El-Cheapo-Memotech Druckern erlaubt. Mich stört jedoch, daß man die vielen Voreinstellungen bei jedem Druck von Hand eintippen muß. Hier behelfe ich mir mit einer kleinen ZEX-Datei.

```
PRINT.ZEX:
bradford
#180818A213LNYNNNN.
```

Aufruf: zex print Dateiname

Jetzt werden alle Voreinstellung automatisch gemacht und man muß nur noch die allerletzte Frage 'Is all this correct' mit einem Y beantworten. Dazu auch eine Frage: Hat bereits irgendjemand eine Möglichkeit gefunden mehrere Schriftbreiten innerhalb einer Zeile zu mischen ?

Zu **MS/MSFORM4** folgendes:

Das ganze funktioniert, wenn ich das Format 40 Spuren, 9 Sektoren, 2 Seiten benutze und ist sehr benutzerfreundlich.

Nicht funktioniert hat bei mir jedoch folgendes:

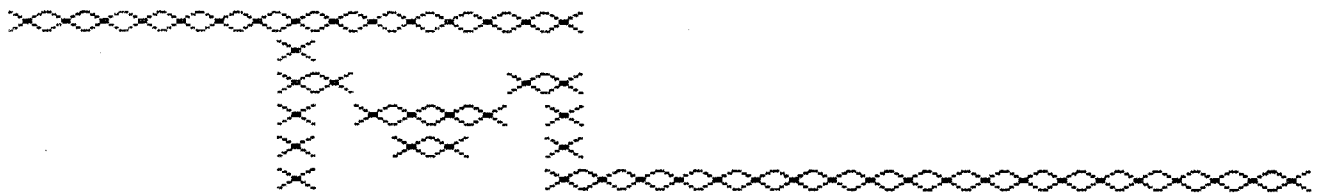
Ich habe mit MSFORM4 eine Diskette im MS-DOS-Format 40 Spuren, 8 Sektoren, 2 Seiten formatiert und anschliessend (mit MS) eine Datei auf diese kopiert. Auf andern Rechnern (u.a. IBM PC-XT), die dieses Format lesen können war diese Diskette jedoch nicht lesbar. Außerdem wird die Größe einer MS-DOS-Datei nicht immer richtig angezeigt: Eine Datei, die unter MS-DOS 83 KByte lang ist wird von MS mit 1042 KByte angezeigt.

Probleme bereiten bei mir die Klick-Programme von Jan Brederekes **Wecker** (KLICK.003). Nach dem Aufruf von KWecker oder Weckaus erhalte ich nur die Meldung 'Klix-Loader (c) 1987 Jan Bredereke' und dann tut sich garnichts mehr. Hier hilft dann nur noch der Klick-Warmboot. Was muß ich tun, damit diese Programm sich laden lassen ?

Anbei findest Du eine Immatrikulationsbescheinigung für die Info-ermäßigung und ich möchte Dich bitten, mir diese wieder zurückzusenden, da ich sie vielleicht noch anderweitig brauche. Das wars nun, was ich loswerden wollte, und ich hoffe, das bald jemand Antworten für meine Fragen findet.

Stefan

Leserbrief: Thomas Mäurer, 4200



Thomas Mäurer
 Duisburger Str. 296 - 4200 Oberhausen I

Tel. 0208/20883 - Konto Nr. 10.086.668
 bei Stadtparkasse Oberhausen (Blz. 365.500.00)

Firma / Herrn / Frau / Frä.
 MTX User Club
 Herbert zur Nedden
 Sonnenau 2
 2000 Hamburg 76

Ihre Nachricht vom 17.3.1988
 Oberhausen, den 24.3.1988

Lieber Herbert, liebe MTX-ler!

Seit einem Jahr liegt mein MTX-Computer - abgesehen von Zwischenzeiten, in denen ich das Gerät verliehen hatte - unter dem Bett, da ich auf den Atari-ST umgestiegen bin. Ich hatte vor dem Kauf sehr viel darüber nachgedacht, ob es sich wohl lohnen würde, ein Floppy-System für den MTX zu erwerben, und bin zu dem Schluß gekommen, daß das zu teuer geworden wäre, ganz abgesehen davon, daß Ihr mittlerweile mit erheblichem Aufwand versucht, die schwarze Kiste lediglich um geringe Prozentsätze zu beschleunigen. Sicher, das Atari-Standard-BASIC ist kaum schneller als das des MTX, aber es gibt ja auch noch andere Betriebssysteme und Programmiersprachen. Ich habe z.B. ein Kalah-Programm in MTX-Basic, bei dem jeder Computerzug ca. 1 Minute braucht, in PEARL auf dem ST ist die benötigte Zeit kaum meßbar, der Computer ist hauptsächlich mit dem Bildschirmaufbau beschäftigt!

Natürlich hat auch der ST seine Macken: Ein äußerst dürftiges "Bedienungshandbuch" (von der MTX-Dokumentation könnte sich wirklich manch ein Computer-Hersteller eine dicke Scheibe abschneiden), das die Anschaffung von Zusatzliteratur zwingend erforderlich macht und ein Betriebssystem, das um einige Nummern zu groß geraten (mehr als 190 KB, weil es nicht in Assembler sondern in C geschrieben wurde) und nicht allzu durchschaubar ist. Hinzu kommt der Supervisor-Modus der CPU, bei dem ich bis heute nicht weiß, wozu der gut sein soll, der aber die Assembler-Programmierung erschwert, da natürlich alle bedeutenden Adressen nur im Supervisor-Modus zugänglich sind und dieser quasi bei jedem Assembler-Programm eingeschaltet werden muß (mehrere Befehle sind hierzu erforderlich). Ansonsten allerdings ist die Programmierung des 68000 recht einfach, hat man sich erst einmal an die oft recht abweichende 68000-Nomenklatur gewöhnt.

Mit dem ST bin ich bis jetzt recht zufrieden, was aber nicht bedeutet, daß ich den MTX verkaufe (da steckt doch etwas zuviel Geld drin, um ihn für'n Appel und 'n Ei zu verschauern!). Wenn ich genug Platz hätte, stände mit Sicherheit auch noch der MTX auf irgend einem Tisch.

Da ich den MTX auch weiterhin haben werde, ist mein Interesse am Club natürlich nicht auf den absoluten Nullpunkt gesunken, ich bin aber z.Z. weder an Bestellaktionen (die ich eh selbst billiger abwickle) noch an Infos (die mittlerweile sowieso nur noch CP/M behandeln und mir nichts bringen) interessiert. Du (Herbert) kannst meinen Namen aus der Adreßdatei ruhig löschen, aber ich stehe weiterhin etwaigen Anfragen (bzgl. meiner Programme) zur Verfügung.

Tschüs

Thomas

dBASE: Tips

dBASE-Literaturtip

(Peter Würfel, 7262)

Dr. Peter Albrecht, Das Datenbanksystem dBASEII, Markt&Technik, DM 68,-

Wer das Markt&Technik dBASE besitzt, kann sich dieses Buch eigentlich sparen. Im ersten Teil ist es ähnlich wie das dBASE-Handbuch aufgebaut: Es werden an einer Beispieldatenbank die grundlegenden dBASE-Befehle eingeführt. Nur der REPORT-Befehl ist etwas ausführlicher als im Markt&Technik-dBASE-Handbuch beschrieben. Wer nur eine englische Version hat und mit dieser Sprache auf Kriegsfuß steht, kann mit diesem Buch sich die Grundlagen der Programmiersprache dBASE aneignen. Im zweiten Teil des Buches stellt Albrecht ein paar praktische Routinen vor, doch wer etwas Programmiererfahrung in einer anderen Sprache hat, wird auch diesen Teil kaum brauchen. Die einzige Routine, die mir diesen Teil lesenswert erscheinen ließ, stellte sich dann leider als falsch heraus! (s.u. 'Zweistufiges Suchen')

Luis Castro u.a., Das dBASE-Kompodium, Markt&Technik, DM 79,-

Ein stolzer Preis, aber dafür doppelt so viel Seiten wie obiges Buch. Der Untertitel: 'Das umfassende Handbuch für den fortgeschrittenen Programmierer' zeigt die Zielgruppe.

Es eignet sich nicht zum Erlernen von dBASE, sondern kann eher als Nachschlagewerk für Probleme dienen bzw. den Übergang vom spontanen Programmieren zum strukturell durchdachten Aufbau eines Datenbanksystems erleichtern.

Wer schon Grundlagen in dBASE hat und auf der Suche nach weiterer Literatur ist, sollte auf jeden Fall mal in dieses Buch schauen. (Es gibt ja auch Büchereien) Viele der folgenden Tips basieren auf diesem Buch.

Kompakte Speicherung von (logischen) Variablen

Das, was Kurt-Bernd Rohloff im letzten Info (26-12ff) über die Probleme mit der begrenzten Anzahl von Speichervariablen geschrieben hat, kann ich nur unterstreichen. Die Tips zur Strukturierung der Variablen durch geschickte Wahl der Variablennamen fand ich ganz duft!

Da das Problem weniger das des verfügbaren Speicherplatzes ist, sondern eher die Begrenzung auf 64 Variable, hab ich mir für 'logische' Variable ('logisch' vom Sinn, nicht vom Variablentyp her, deshalb die Anführungszeichen) folgendes Verfahren einfallen lassen:

Wenn ich z.B. 6 'logische' Variablen habe, wobei die drei ersten auf TRUE und die drei folgenden auf FALSE stehen sollen, dann packe ich diese in folgende Variable 'mzeiger':

```
STOR '111000' TO mzeiger
```

Die dritte Variable kann ich nun z.B. mit

```
IF $(mzeiger,3,1)='1'
```

auf TRUE abfragen.

Eine dieser 'logischen' Variablen kann ich nun mit folgendem Dreizeiler ändern (hier z.B. die zweite Variable auf FALSE setzen):

```
STOR 2 TO mpos
```

```
STOR '0' TO mneu
```

```
STOR $(mzeiger,1,mpos-1)+mneu+$(mzeiger,mpos+1)
```

Dieses Verfahren mag zwar etwas umständlich erscheinen, doch es bietet mir noch weitere Möglichkeiten: Da ich ja nicht mit 'echten' logischen Variablen arbeite, kann ich auch die Ziffern 2 bis 9 verwenden und dadurch weitere Informationen speichern.

Im obigen Beispiel zeigt sich außerdem eine Syntax-Variante der Teilstring-Funktion '\$(...)', die nicht im Handbuch steht:

Wenn keine Länge angegeben wird, wird der Teilstring vom angegebenen Anfang bis zum Ende übernommen.

d B A S E: Tips**Dateitypen selbst wählen**

Bei dBASE-Dateien gibt es ja festgelegte Dateitypen (z.B. DBF, CMD usw.) Doch diese Dateitypen sind nicht zwingend. Wenn ich z.B. ein Unterprogramm mit 'DO test' aufrufe, sucht das Hauptprogramm nach der Datei 'test.cmd'. Doch ich hätte dem Programm 'test' auch den Typ 'xyz' zuweisen können. Dann müßte ich es mit 'DO test.xyz' aufrufen. Dies gilt für alle Dateitypen, nicht nur für Befehlsdateien.

Diese dBASE-Möglichkeit habe ich wie folgt ausgenutzt: Nur die Befehlsdatei, mit der das Programm gestartet wird, erhält die Erweiterung CMD, alle Unterprogramme erhalten die Erweiterung .000 oder .001 usw. So sehe ich sofort, mit welchem Programm ich ein Programmpaket starten muß. (Daß alle Programmdateien eines Programmpakets die gleichen ersten zwei oder drei Buchstaben tragen, versteht sich bei mir von selbst)

(M)eine Abfrageroutine

Wenn ich mit 'WAIT TO...' nur bestimmte Eingaben zulassen möchte, verwende ich folgende Routine (im folgenden Beispiel sollen nur die Eingaben 'j' oder 'J' oder 'z' oder 'p' erlaubt sein):

```
STOR ' ' TO mantw
DO WHIL S(mantw,'jJzp')=0
    WAIT TO mantw
ENDD
```

Kompatibel programmieren

Diese Idee kam mir, als ich für einen Freund ein dBASE-Programm für seinen Schneider-Joyce programmierte. dBASE-Programme, die auf unserer Maschine laufen, laufen ja auch auf anderen Computern, wenn sie keine Steuerzeichen beinhalten, die z.B. Bildschirmattribute setzen. Auch bei all den Steuerzeichen, die den Drucker ansteuern, bin ich sehr hardware-abhängig, auch wenn ich davon ausgehe, daß unser DMX80 ja 'EPSON-kompatibel' ist. Aber wer denkt nicht daran, wenn er ein Programm schreibt, daß es doch sein könnte, daß er sich mal nen anderen Drucker oder gar (Gott behüte) nen anderen Computer (MSDOS ick hör dir trapsen) anschaffen könnte. Und dann all die gewohnten Programme nach Steuerzeichen untersuchen? Also deshalb folgendes:

Ich möchte z.B ein Wort in Fettdruck auf unserem Bildschirm ausgeben. Bei mir lautet die Zeile dann:

```
? &mfett+'wort'+&mnorm
```

Den beiden Variablen 'fett' und 'norm' muß ich dann natürlich Werte zugewiesen haben:

```
STOR 'CHR(20)' TO fett
STOR 'CHR(18)' TO norm
```

Diese Variable kann ich nun z.B. durch ein kleines Installationsprogramm erfragen, in eine .MEM-Datei ablegen und aus dieser Datei zu Beginn des Hauptprogramms RESToren (wow, welch Wortschöpfung!)

Läuft das Programm auf einem anderen Computer, muß ich nur für die entsprechende Initialisierung der Variablen sorgen.

Druckersteuerzeichen können auf die gleiche Art und Weise übergeben werden.

CREATE FROM...

Ihr kennt sicher alle die Möglichkeit, mit CREATE <Dateiname> FROM <Quelldatei> innerhalb eines dBASE-Programms eine neue Datei zu eröffnen. Bei dem Programmpaket 'dbt...' (s.u.) kam dies immer wieder vor, und im Handumdrehn hatte ich eine ganze Reihe von <Quelldateien> auf der Diskette. Jede Datei nahm nun, obwohl sie ja ganz kurz war, mindestens 2k Speicherplatz ein (vom Bildschirmverbrauch im directory ganz zu schweigen). Was tun (Trotzki)?

dBASE: Tips

Laßt's mich an einem Beispiel erklären:

Ich habe zwei Quelldateien mit 'COPY TO <namen> STRU EXTE' erzeugt. Für unser Beispiel sollen sie 'quelle1' und 'quelle2' heißen. 'quelle1' soll 3 Datensätze und 'quelle2' soll 5 Datensätze umfassen.

Dann:

```
USE quelle1
APPE FROM quelle2
```

Jetzt hat meine Datei 'quelle1' 8 Datensätze und ist für das CREATen einer neuen Datei nicht mehr zu gebrauchen. Was tun (Murphy)?

Wenn ich Quelldatei2 brauche, rufe ich folgendes auf:

```
USE quelle1
COPY TO temp FOR #>3 .AND. #<9
```

(wobei ich mir natürlich in diesem Fall das .AND..... sparen könnte, aber wenn es noch eine quelle3 gäbe, bräucht ich's)

```
USE
CREA <Dateiname> FROM temp
DELE FILE temp
```

Die letzte Zeile könnte ich mir zwar sparen, aber wer will schon überflüssiges auf seiner Diskette?

dBASE-Druckeransteuerung:

Wenn man von dBASE was an den Drucker schickt, kann man ja auch Steuerzeichen zur Druckersteuerung übergeben. Theoretisch müßte die Sequenz

```
? CHR(27)+'W'+CHR(1)+'test'+CHR(27)+'W'+CHR(0)
```

den Drucker auf Breitschrift umschalten, das Wort 'test' ausgeben und dann wieder in normale Schriftbreite zurückschalten. Doch letzteres funktioniert nicht. Der Drucker bleibt auf Breitschrift! dBASE wertet nämlich CHR(0) nicht korrekt aus. Was tun?

Irgendwann muß an den Drucker der Steuerbefehl für MSB Operation, Resetting (chr(27)+"=") geschickt werden (einmal reicht) und dann nicht CHR(0) sondern CHR(128).

Also obiges Beispiel:

```
? CHR(27)+'='+CHR(27)+'W'+ 'test'+CHR(27)+"W"+CHR(128)
```

das geht!

Die EOF-Funktion

Die EOF-Funktion hat ja in dBASE zwei mögliche Aufgaben.

- EOF steht auf .T., wenn der letzte Datensatz erreicht ist.
- EOF steht auf .F., wenn mit LOCa oder CONT ein Datensatz gesucht und gefunden wird.

Was aber nun, wenn der letzte Datensatz (durch LOCA oder CONT) erreicht wird, aber dieser Datensatz den gesuchten Kriterien entspricht. Die dBASE eigene EOF-Markierung ist direkt nach der Suche auch in diesem Fall auf .F.(was auf den Suchvorgang bezogen richtig ist), wird dann aber durch einen weiteren Befehl (z.B. durch ein nachfolgendes ENDI) auf .T. gestellt (was bezogen auf den Datensatzzeiger korrekt ist)

Das auf das Suchen bezogene .T. oder .F. muß also sofort ausgewertet werden!

DELETE, PACK und Speicherplatz

(oder: Was dBASE mal hat, gibt es nicht wieder los)

Eine Datenbank mit 500 Datensätzen belegt z.B. 50 kByte. Nun lösche ich 50% der Datensätze und führe PACK durch. Zurück auf der CP/M-Ebene muß ich beim Blick ins Directory feststellen, daß die Datenbank immer noch 50 kByte belegt, obwohl ja nur noch 250 Datensätze vorhanden sind. Die Datei scheint also nicht kleiner geworden zu sein. dBASE behält den frei gewordenen Speicherplatz quasi als Reserve:

d B A S E: Tips

In diesem Beispiel kann ich nämlich nun bis zu 250 neue Datensätze eingeben, bevor die Datenbank über 50 kByte anwächst. Der im Directory angegebene Umfang einer Datenbank sagt also nicht unbedingt etwas darüber aus, wie 'gefüllt' sie ist.

Assembler-Programme in dBASE

Das Markt&Technik-Handbuch behandelt ja die Einbindung von Assembler-Programmen in dBASE recht knapp. Kurt-Bernd Rohloff hat diese Lücke mit seinen Hinweisen in den letzten beiden Infos (25-54 und 26-13) ja schon weitgehend geschlossen. Hier nun noch folgende Ergänzungen:

a) Das, was mit POKE oder LOAD in den Speicher geladen wird, darf ja dBASE selbst nicht überschreiben. Die von Kurt-Bernd angegebene Grenze von A400h gilt jedoch nicht für jede dBASE-Version. Lt. dBASE-Kompendium gibt es folgende Grenzen:

dBASE-Version	(Hex)	(Dez)
2.3, 2.3A, 2.3B	A400	41984
2.4	B000	45056
2.41	C000	49152

b) Für Assembler-PUSHs und -POPs steht ein Pufferbereich von ca. 254 Byte zur Verfügung.

c) Zum Zurückkehren zu dBASE dient der Assemblerbefehl RETI. In dBASE werden zwar alle Register gesichert, es ist trotzdem zu empfehlen, das HL-Registerpaar beim Aufrufen einer Assembleroutine zu sichern (SHLD) und vor dem Zurückkehren zu dBASE zurückzustellen (LHLD).

Zweistufiges Suchen

Im Albrecht-Buch (s.o) fand ich eine Suchroutine, die ich ganz gut fand, nur mußte ich feststellen, daß der Typ es doch nicht ganz geblickt hat.

Was es mit der 'zweistufigen Suche' auf sich hat, möchte ich am Beispiel einer Adressendatei aufzeigen. Wenn ich in meiner Datei den Eintrag 'Kohl' (Datenbankfeld: Nachname), 'Helmut' (Datenbankfeld: Vorname) suchen möchte, meine Datei aber nur nach Nachnamen indiziert ist, finde ich mit FIND den ersten Eintrag 'Kohl'. Wenn's der Zufall will, mag das 'Kohl, Helmut' sein, was aber, wenn es 'Kohl, Albert' oder 'Kohl, Berta' ist?

Hier setzt die Idee von Albrecht ein: Er sucht 'Kohl' mit FIND auf der nach Nachnamen indizierten Datenbank und dann schließt er die Suche nach 'Kohl, Helmut' mit LOCATE FOR... an. Dabei geht er davon aus (und das ischt der falsche Fähler), daß LOCATE mit der Suche an dem Datensatz beginnt, auf dem der Datensatzzeiger steht. Doch LOCATE beginnt bei seiner Suche genau wie FIND immer beim ersten Datensatz einer Datenbank; Albrecht hätte sich also das Suchen mit FIND ersparen können: kein Zeitgewinn, sondern -verschwendung (probieren die Typen denn nicht aus, was sie verbraten und verkaufen?).

Hätte er folgendes gemacht, dann hätte seine Theorie gestimmt:

```
FIND 'Kohl'
```

```
LOCA NEXT 65535 FOR vorname='Helmut'
```

Mit FIND wäre (sehr schnell) der erste 'Kohl' gefunden worden und LOCATE hätte dann, durch den Zusatz NEXT... an diesem ersten 'Kohl'-Datensatz die Suche fortgesetzt nach dem Datensatz mit dem Vornamen 'Helmut'. Ob es wirklich noch 65535 weitere Datensätze gibt oder nicht, ist dBASE völlig egal, wenn's weniger sind, hört halt die Suche vorher auf.

dBASE: Tips**Der RESET-Befehl klappt nicht**

Lt. Markt&Technik-Handbuch müßte es möglich sein, mit dem Befehl RESET eine neue Diskette anzumelden. Doch das klappt nicht. Wenn ich eine Datenbank auf einem Laufwerk eröffne, später wieder schließe und ohne RESET-Befehl die Diskette wechsele, dann von der neuen Diskette eine Datenbank eröffnen will, bricht dBASE mit Fehlermeldung 'R/D' ab. Wenn ich nun jedoch vor dem Diskettenwechsel RESET eingebe, scheint sich das Versprechen des Handbuchs zu erfüllen, ich kann eine Datenbank auf der gewechselten Diskette öffnen und Daten lesen. Wenn ich aber nun Daten in die Datenbank schreiben will? Is nicht! Abbruch! R/D!

Übrigens: unter CP/M 3.0 klappt das ganze, habe auf diese Art und Weise ein dBASE-Programm für einen Joyce mit seinem jämmerlichen 170k Laufwerk geschrieben.

Ja und nun hätte ich da folgenden **W U N S C H** (es wäre doch gelacht, wenn CP/M 3.0 besser wäre als unser ... wie heißt das doch nochmal?):

Wer schafft es, den Diskettenwechsel durch ein Assemblerprogramm einzubauen? Ich hab's versucht (mit: `ld c,13 / call 0005`), aber es hat nicht geklappt! (Aber mit Assembler klappt bei mir öfter nicht (eigentlich fast nie)) Ich vermute, dBASE legt eine eigene 'Diskettentabelle' an und greift nicht auf die Informationen des Betriebssystems zurück!

Wers schafft, kriegt von mir eine Flasche Schampanjer!

Speicherplatzbedarf einer INDEX-Datei

dBASE arbeitet bei der Indizierung mit der sog. B+ Baum-Struktur (fragt mich nicht, was das ist, da sollen die Informatixen antworten). Für den Juser (Juser??) bedeutet das, daß nicht genau festgelegt werden kann, wie groß eine Indexdatei wirklich ist, das hängt vom Inhalt (nicht der Struktur) einer Datenbank ab. Es lassen sich nur die maximale und minimale Größe einer Indexdatei errechnen.

Und das geht so:

Die Länge des Schlüsselfeldes sei: `schllg`

Die max. Zahl der Datensätze sei: `saetze`

Die min. Größe in kByte ist dann:

`mink = INT(512*(INT(saetze/INT(509/(schllg+4)))+1)/1024)`

Die max. Größe ist dann `mink * 2`

Der STORE-Befehl

Als Schlußtop eine klitzekleine Kleinigkeit: der STORE-Befehl akzeptiert auch folgende Syntax:

`STOR irgendwas TO var1,var2.....`

dBASE-Tools auf CLUB-Public-Domain

Im Info 8 hat Frank Bueschler schon mal was über dBASE geschrieben und dabei darauf hingewiesen, daß der dBASE-Interpreter alles liest, also strukturierende führende Leerzeichen und Kommentarzeilen sowie Schlüsselwörter länger als vier Zeichen aus Gründen der Geschwindigkeit vermieden werden sollten. So ein Programm ist natürlich nur noch sehr schwer zu lesen. Also hab ich mich drangemacht, ein Programm zu schreiben, das überflüssige Leerzeichen entfernt.

Dann fand ich im 'dBASE-Kompendium' ein Programm Generierung von Referenzlisten von Variablen. Nur, dieses Programm war in der Form, in der es abgedruckt war, nicht lauffähig (und nicht nur wegen Druckfehlern). Also überarbeitete ich dieses Programm und verwendete einige Ideen daraus für weitere Programme: Das Ergebnis war ein Programmpaket, das ich hiermit dem Club als PD übergeben möchte.

dBASE: Tips

Die einzelnen Programme leisten (wenn auch langsam) folgendes:

<1> SCHLÜSSELWORTE in einer dBASE-Befehlsdatei BEARBEITEN:

- Schlüsselworte werden in Großbuchstaben umgewandelt und wenn nötig auf 4 Zeichen gekürzt. Gleichzeitig werden Leerzeilen aus dem Programm entfernt.

<2> ZEILENUMMERN in dBASE-Befehlsdatei EINFÜGEN

Eine dBASE-Programm-Datei wird so bearbeitet, daß die ausgegebene Textdatei mit NW ausgedruckt werden kann.

Dazu wird die dBASE-Programm-Datei wie folgt bearbeitet:

- Die Programmzeilen werden entsprechend der Programmlogik eingerückt,
- Die Programmstruktur wird durch Punkte verdeutlicht.
- Die Datei wird mit Zeilennummern versehen.

<3> STRAFFEN einer dBASE-Befehlsdatei:

- Es werden in einer dBASE-Programm-Datei sämtliche Leerzeichen entfernt, die beim Einrücken einer Programmzeile geschrieben wurden, außerdem werden alle Kommentarzeilen, die mit einem '*' beginnen entfernt

<4> FORMATIEREN einer dBASE-Befehlsdatei:

- Dieses Teilprogramm ist die teilweise Umkehrung des Programms <3>: Programmzeilen werden gemäß der Programmlogik eingerückt.

<5> Referenzliste der VARIABLEN von dBASE-Befehlsdateien

Dieses Programm untersucht eine oder mehrere Programmdateien nach Variablen und listet diese in einer Übersicht auf, wobei angegeben wird, in welcher Programmzeile die Variable verwendet wird.

Die Satznummer bezieht sich auf die Nummerierung im Listing, das mit <2> erzeugt wurde.

<6> Referenzliste der UNTERPROGRAMME von dBASE-Befehlsdateien

Dieses Programm untersucht eine oder mehrere Programmdateien nach Unterprogrammen, die mit DO <Datei> aufgerufen werden und listet diese in einer Übersicht (mit Zeilennummern) auf.

<7> Referenzliste der DATENBANK-DATEIEN von dBASE-Befehlsdateien

Dieses Programm untersucht eine oder mehrere Programmdateien nach Datenbank- und Textdateien (nicht Unterprogramm-Dateien) und listet diese (mit Zeilennummern) in einer Übersicht auf.

Das Programm untersucht weiter, ob die Datenbanken vom Typ .DBF vorhanden sind; wenn ja, wird Ihre Struktur ausgegeben.

d B A S E: Vergleiche / T u r b o - P a s c a l: Binominalkoeffizienten**Vergleiche in dBASE - nochmal betrachtet**

(Kurt-Bernd Rohloff, 8000)

In Info 26-11 hatte ich ein paar Bemerkungen über Vergleiche von Strings in dBASE gemacht. Etwas eilfertig hatte ich dort bemerkt, daß im EXACT ON Modus alles normal sei. Mitnichten! Ich habe mir die Sache noch mal angeschaut. Die Ergebnisse will ich zunächst als Matrix angeben. Ein Matrixelement stellt das Ergebnis eines Vergleichs der Form OP1 = OP2 dar, u. z. vor dem Schrägstrich im EXACT OFF Modus, nach dem Schrägstrich im EXACT ON Modus.

OP1	OP2 -->				
	"An"	"Antonia"	"Anton "	"An "	" "
"An"		F/F	F/F	F/T	
"Antonia"	T/F		F/F	F/F	
↓ "Anton "	T/F	F/F		F/F	
"An "	T/T	F/F	F/F		

(In der Hauptdiagonalen müßte immer T/T stehen, ich habe dies nicht geprüft.) Bisher dachte ich immer, daß man im EXACT OFF Modus mehr Treffer erreichen würde. Wie das letzte Element in Zeile 1 zeigt, war das ein Irrtum. Im EXACT OFF Modus wird anscheinend der gesamte OP2 zum Vergleich herangezogen (so wie im letzten Beitrag geschildert), bei EXACT ON werden dagegen Leerzeichen am Ende **nicht** mit bewertet, so daß das Ergebnis dann TRUE wird! Dies gilt offenbar auch bei vertauschten Operanden (s. letzte Zeile, 1. Spalte).

Binomialkoeffizienten - quick and dirty

(Kurt-Bernd Rohloff, 8000)

Die BK's (erspart es mir, dieses ekelige Wort noch einmal auszuschreiben) treten in vielen Formeln aus dem Bereich Statistik und Kombinatorik auf. Die folgende TURBO-Pascal Funktion berechnet diese Dinger sehr flott, da sie nur INTEGER Arithmetik benutzt. Aus diesem Grunde funktioniert sie aber nur bis einschl. n=14. **Dies muß vom Aufrufer sichergestellt werden!** Aus Effizienzgründen wurde in die Funktion keine Prüfung eingebaut.

```

FUNCTION BinKo(n,k: INTEGER): INTEGER;
( Berechnet den Binomialkoeffizient n ueber k
  geht bis n=14, keine Fehlerpruefung!!!
)
VAR
  a          : INTEGER;
  j,smaller,larger : BYTE;
BEGIN
  smaller:=k; larger:=n-k;
  IF smaller > larger
  THEN BEGIN
    j:=smaller; smaller:=larger; larger:=j;
  END;
  ( hier ist smaller der kleinere der beiden Werte k und (n-k),
    larger der groessere (Spiegelungseigenschaft der Bin-Koeff.)
  )
  IF smaller = 0
  THEN BinKo:=1
  ELSE BEGIN
    a:=Succ(larger);
    FOR j:=2 to smaller DO
      a:=(larger+j) * a DIV j;
    BinKo:=a;
  END;
END;

```

SuperCalc: Spritverbrauch

SuperCalc berechnet Spritverbrauch

Da offenbar viele von euch mit SuperCalc nicht so recht etwas anfangen können, möchte ich hier anhand eines Beispiels zeigen, was ich damit mache. Das weiter unten abgebildete Spreadsheet ermittelt mir den Kraftstoffverbrauch meines Autos. (Die Zahlen sind nicht gezinkt, man muß dazu aber wissen, daß ich einen Diesel fahre.)

	A	B	C	D	E	F	G	H	I	J	K	
1	Verbrauchsberechnung											
2	VERBR											
3	km	Stand bei Übernahme:			98000							
4												
5	Durchschnittl. Verbrauch:	5.22										
6												
7	<----Datum---->	km Diff.			km seit	Ltr.	Vbr.					
8	tt	mm	jj	Stand	Übn.							
9	-----											
10	3	5	88	110164	578	12164	29.90	5.17	*****			
11	5	4	88	109586	413	11586	20.99	5.08	*****			
12	4	4	88	109173	454	11173	30.15	6.64	*****			
13	2	4	88	108719	730	10719	37.00	5.07	*****			
14	31	3	88	107989	412	9989	23.65	5.74	*****			
15	27	3	88	107577	563	9577	29.47	5.23	*****			
16	19	12	87	107014	688	9014	37.41	5.44	*****			
17	29	8	87	106326	626	8326	30.14	4.81	***			
18	6	8	87	105700	607	7700	30.41	5.01	*****			
19	5	8	87	105093	233	7093	11.70	5.02	*****			
20	31	7	87	104860	659	6860	34.90	5.30	*****			
21	28	7	87	104201	536	6201	28.31	5.28	*****			
22	9	6	87	103665	602	5665	29.80	4.95	***			
23	8	6	87	103063	533	5063	27.00	5.07	*****			
24	28	5	87	102530	735	4530	33.00	4.49	**			
25	22	5	87	101795	193	3795	9.80	5.08	*****			
26	10	5	87	101602	335	3602	17.14	5.12	*****			
27	3	5	87	101267	408	3267	18.61	4.56	**			
28	17	4	87	100859	939	2859	50.00	5.32	*****			
29	15	3	87	99920	527	1920	34.86	6.61	*****			
30	20	12	86	99393	478	1393	25.87	5.41	*****			
31	1	11	86	98915	484	915	23.70	4.90	***			
32	14	9	86	98431	408	431	20.16	4.94	***			
33	12	9	86	98023		23	22.21					
34	-----											

In Zelle A1 trage ich immer ein kurzes Stichwort zum Zweck des Spreadsheets ein. Das ist insofern günstig, als daß SuperCalc diesen Text dann anzeigt, wenn man sich ein Verzeichnis der SuperCalc Dateien ausgeben läßt (z. B. beim /L(oad) Befehl). Ansonsten benutze ich die Zeile 1 nur als Statuszeile, die SC ja leider nicht zur Verfügung stellt. Daher steht in G1 der Name der SC Datei. Die Spalte A bleibt als Heftrand frei (Breite 6). Da ich mir keinen Neuwagen leisten kann, habe ich in Zelle G3 den km-Stand eingetragen, bei dem ich den Wagen gekauft habe.

In Zelle I5 wird der durchschnittliche Verbrauch (in Liter/100 km) angezeigt. Er wird berechnet, indem alle getankten Liter (Spalte H) aufaddiert werden, davon wird jedoch der letzte Wert in H33 abgezogen, denn der geht ja noch auf Kosten meines Vorgängers. Das Ergebnis wird mit 100 multipliziert. Diese Zahl wird dann durch die von mir gefahrene Strecke, also E10-E33, dividiert. In I5 ist daher folgende Formel eingetragen:

SuperCalc: Spritverbrauch

(SUM(H9:H34)-H33)*100/MAX(E9:E34)-E33)

Daß im Nenner nicht direkt E10-E33 steht, liegt daran, daß ich beim nächsten Tanken ja wieder eine neue Zeile 10 einfügen muß (das Spreadsheet wächst also im Laufe der Zeit nach oben), und dann würde die Formel in E11-E34 geändert werden, was dann falsch wäre. Im Zähler müßte jedoch auch SUM(H9:H32) gehen.

Bei den folgenden Kopfzeilen 7 und 8 wurde der Text rechtsbündig gestellt (mit /F und TR). In der "Diff" Spalte wird die Wegdifferenz zum vorigen Tanken angegeben, in der nächsten Spalte erscheint die gesamte Laufleistung seit Übernahme des Fahrzeugs, in der H Spalte die jeweils getankten Liter (immer voll!) und in den Spalten I und J der Verbrauch, einmal als Zahl und einmal als Balkendiagramm (mit /FJ*). Die J-Spalte ist 25 Zeichen breit (/FJ25).

Die Zeilen 10 bis 33 bilden den eigentlichen Datenbereich. In den Spalten B, C, D und E werden die Werte direkt von Hand eingetragen. In Zelle F10 steht die Formel

E10-E11

Deswegen wird der Bereich F9:F34 geschützt. In Zelle G10 steht die Differenz zum Übernahme-km-Stand:

E10-G3

Der Bereich G9:G34 wird ebenfalls geschützt. In der Spalte "Ltr" werden dann die getankten Liter eingetippt. In I10 wird dann der Verbrauch wie folgt berechnet:

IF(F10>0,H10*100/F10,0)

Die Abfrage auf F10>0 ist eine kleine Absicherung gegen versehentliche Fehleingaben des km-Stands (sog. Plausi-Check). Denn da ich normalerweise ja vorwärts fahre, muß der neue km-Stand größer als der vorige, die Differenz in F10 also positiv sein. In J10 steht die Formel

(I10-4)*5

Die Spalte J ist, wie gesagt, 25 Zeichen breit. Der größte Wert für den Verbrauch, der J gerade ganz ausfüllt, ist also 9 L/100 km. Andererseits erhalte ich bei 4 L/100 km das Resultat Null, also kein Sternchen mehr. Diese Grenzwerte wurden zu Anfang einmal geschätzt. Sie können bei dir natürlich anders liegen. Wenn du beispielsweise meinst, daß dein Verbrauch zwischen 8 und 15 Litern variiert, nimmst du die Formel (I10-8)*4 und machst die J Spalte 28 Zeichen breit. Allgemein gesprochen: Wenn Spalte J eine Breite von c Stellen hat und du nimmst an daß dein Verbrauch zwischen a und b (L/100 km) schwanken kann, trägst du in J10 die Formel

(I10-a)/(b-a)*c

ein. Natürlich wird auch der Bereich I9:J34 geschützt.

Das Spreadsheet wird im TAB Modus und mit manueller Berechnung betrieben (einstellen mit /G). Neue Daten werden immer in einer neu einzufügenden Zeile 10 eingetragen. Dafür habe ich mir das folgende kleine Programm geschrieben (VERBR.XQT):

SuperCalc: Spritverbrauch / Hardware: 8 MegaHertz

```

/GT
=A10
/IR
/FR
TR
>NA
NA
NA
NA
/RF11

>/RG11
,AYN
NA
/RI11

>/RJ11

=B10
/GT

```

<-- dies ist die letzte Zeile

Die Leerzeilen im Programm sind zwei aufeinanderfolgende RET's. Am Schluß steht kein RET! Die Eingabezeilen werden mit NA vorbelegt. Alle Formeln werden aus der darunterliegenden Zeile 11 (die alte Zeile 10) kopiert. Sie brauchen also nur einmal eingetippt zu werden. Zum Schluß springt der Cursor auf B10 und der TAB Modus wird wieder eingeschaltet. Dann kann man die neuen Werte eingeben und anschließend die Berechnung mit "!" starten.

Zu Anfang, wenn man das Spreadsheet aufbaut, wird nur in G10 (das ist jetzt G33) die oben angegebene Formel eingetragen. Die getankten Liter hat noch der Vorgänger verfahren. Beim nächsten volltanken, wenn man also erstmals für die selbst gefahrene Strecke tankt, wird Zeile 10 per Hand eingefügt und die Werte bzw. Formeln eingetragen. Dann werden die oben genannten Spalten geschützt und der TAB Mode eingeschaltet. Danach kann man dann mit dem Programm weiterarbeiten. So hat man immer einen Überblick, was das Töff-Töff so schluckt.

Ein möglicher Hemmschuh für den Erfolg

(Herbert zur Nedden, 2000)

Seit ich mein Netzteil aus der FDX verbannte, und damit auch mein Ventilator seine Dienste quittieren durfte zeigte sich das Phänomen, daß meine Kiste die 8 MHz nicht mehr so recht mochte: Ewig lange automatisch ablaufende Autostart-Sequenzen (auch über Stunden - so als Test) liefen, aber sowie die Tastatur ins Spiel kam tat sich nichts. Allerdings schien etwas Kühlung für den Z80 B CTC etwas Linderung zu schaffen.

Ich vermute, daß bei 8 MHz der CTC mit dem Interrupt-Acknowledge Probleme hat, und es vielleicht etwas nützt, statt auf alle M1-Zyklen lediglich auf die M1-IOREQ-Zyklen einen Wait zu spendieren, da schließlich die Gurke mit 8 MHz und einem M1-Wait tut. Übrigens habe nicht nur ich derartige Erfahrungen gemacht. Bei Manfred Flume scheint der CTC bis max. 6.8 MHz, jedoch der Floppy Controller über 7 MHz ohne Wait zu akzeptieren.

Wenn ich dahintergekommen bin, schreibe ich einen ausführlicheren Artikel als diesen - mie diesem möchte ich eigentlich mal die Expterten anregen und bitten über diese Probleme und meine Idee nachzudenken, und mir evtl. zu helfen.

Hardware: 8 MegaHertz-Kochbuch

8 MHz - Kochbuch

(Dr. Holger Göbel, 8630; 09561/15131)

Es ist schon vermaledeit mit diesen 8 MHz. Wie diffizil die Sache ist, merkt man allein an der stattlichen Zahl von Beiträgen über dieses Thema. Mein Rechner lief ja schon eine geraume Zeit mit 8 MHz - jetzt hat Horst Kupka diese elegante Schaltung entworfen (25/19), die dem Floppy-Kontroller ein WAIT aufzwingt, mit dem Erfolg, daß mir meine Schaltung nicht mehr gut genug erschien und ich seine nachbauen wollte. Damit vielleicht noch einige andere davon profitieren könnten, habe ich zusammen mit Gerhard Witzel eine Platine geätzt, die einfach an den Platz des ICs 9D in einen Sockel gesteckt wird. Da alle Chips nach unten hängen, geht die Tastatur auch wieder zu (diese Platine könnt ihr zum Selbstkostenpreis getestet von mir haben; ich denke nicht, daß sie viel teurer als DM 20,- werden kann).

Leider war alles wieder mal nicht so einfach (Danke für Deinen Rat, Horst!): Die Datenübertragung von der Disk in den MTX per DMA war mit 8 MHz vollkommen unsicher. Ich habe lange herumprobiert, bis mir einfiel, daß Jan Brederke und Uwe Grass empfohlen haben, von der Terminierung des ECB-Busses einige Signale wegzunehmen (25/31, 26/27). Bei Horst Kupka lief die Schaltung möglicherweise deshalb, weil er gar keinen ECB-Bus besitzt!

Ja, und dann war ich so frevlerisch, die Terminierung ganz herauszuziehen - und siehe da, seitdem läuft mein Rechner ohne Probleme. Die Signale sehen zwar auf dem Oszi viel grauenvoller aus - aber Schöneres muß eben nicht immer besser sein (jeder Verheiratete weiß das!) (im Übrigen werden die Signale etwas sauberer, wenn man auf der RS232-Karte und auf der CP/M-Karte in der FDX die Treiber gegen HC-Typen austauscht. Bei mir jedenfalls stecken wieder die LS-Treiber drinnen, da sie ja funktionieren, während es die HC-Typen eigentlich wegen ihrer geringen Ausgangsleistung nicht dürften).

Weil mir aber Herbert und Uwe versichert haben, daß einige ECB-Systeme es übernehmen, wenn man sie so ganz ohne Terminierung arbeiten läßt, habe ich auf der besagten 8 MHz-Platine noch einen elektronischen Schalter vorgesehen, der bei Floppy-Zugriffen automatisch auf 4 MHz umschaltet. Damit er das nicht so furchtbar lange tut, habe ich die von Herbert vorgesehene Zeitkonstante verkürzt, so daß er nur noch etwa 100 ms die 4 MHz hält. Als Schaltsignal habe ich BUSRQ genommen, das immer low aktiv wird, wenn ein DMA-Zugriff erfolgt (wenn erstmal Herberts Z280-Karte fertig ist, muß man evtl. ein anderes Signal nehmen). Da das Längste an Diskettenzugriffen ja der Schreib-Lesevorgang von der Diskette ist, ergibt ein Umschalten auf 4 MHz beim DMA-Vorgang einen nur unwesentlichen Zeitverlust.

Die WAIT-Schaltung hat deshalb noch einen zweiten Eingang gefunden, weil alle mir bekannten c't-EPROM-Disks bei 8 MHz nur mit einem WAIT laufen.

Am Stecker der Platine habe ich auch 5V, Masse und nochmal 5V über einen Pull-up-Widerstand herausgeführt. Das hat folgenden Hintergedanken: Um beim Booten auf 4 MHz zu schalten, wird das RE/CPM Signal ausgewertet. Dieses ist auch aktiv (low), wenn der MTX im ROM-Modus, also ohne FDX, arbeitet. Wenn ich hier auch die 8 MHz haben will, dann sollte das RE/CPM-Signal über einen Schalter geführt werden, der entweder dieses Signal (also im Normalfall) oder +5V über einen Pull-up-Widerstand der 8 MHz-Platine anbietet.

Weil die 8 MHz wohl die Grenze des Erreichbaren und zudem bei manchen Systemen wirklich sehr kritisch sind, habe ich mal alle bisher beschriebenen und von mir nachvollzogenen Kunstgriffe zusammengefaßt. Da sie nicht immer besonders einleuchtend sind, ist diese Tabelle eher eine Art Kochbuch geworden. Ich habe mir eigentlich gedacht, daß dieses Kochbuch nach und nach erweitert und aktualisiert wird (auch von anderen, die sich damit beschäftigen).

Alle unbedingt notwendigen Maßnahmen habe ich mit einem "*" gekennzeichnet. Die anderen sind mehr Probiererei, aber in der Not nützlich.

Hardware: 8 MegaHertz-Kochbuch

Bauteil	Maßnahme	Begründung	INFO
1. R24	Widerstand verändern (Poti) oder stärkeres Netzteil	A) Allgemeine Maßnahmen (FDX-Netzteil) Netzteil hochregeln, weil 8 MHz mehr Strom brauchen evtl. Gleichspannungen auf die MTX-Platine führen, um der Hitzeentwicklung im MTX Herr zu werden (dicke Verbindungskabel nehmen!)	18/37,21/34 25/29-30
* 1. XTAL1	8 MHz-Quarz	B) Hauptplatine klar	21/30
* 2. R32, R33	3,3 kOhm	damit der Quarz den Schwingkreis optimal anregt	21/30
* IC 9D	74 HC 04	falls nicht die 8 MHz-Platine zum Einsatz kommt	21/30
* 3. IC 10A	Z80 H CPU	ist für diese Taktfrequenz ausgelegt	21/30
* 4. IC 8A	Z80 B CTC	eigentlich bis 6 MHz, funktioniert aber (?)	21/30
* 5. IC 2C-5C, 2D-5D	RAMs höchstens 150 ns	schneller, beim MTX 500 genügen 3732 (gibt's die?), sonst wie beim MTX 512: 4164	21/30
* 6. IC 9D	raus, gedrehter Sockel rein	zur Not evtl. statische RAMs verwenden	25/19 27/?
* 7. IC 9E	P4 abkneifen, hochbiegen	da wird die 8 MHz-Platine reingesteckt	18/19,21/30
8. R35, R36	besser drinlassen	damit die Zeitkonstanten des CTC gleich bleiben können	21/30
* 9. C5, C6	auslöten	laut ZILDG macht sonst die CPU evtl. Schwierigkeiten	21/30,24/9
10. R14, R15	evtl. durch andere ersetzen	damit das CAS-Signal für die RAMs eher kommt	24/9
* 11. 8 MHz-Platine	evtl. überbrücken	gerade diese RAS-/CAS-Signale sind sehr zeitempfindlich	25/21
	reinstecken, Verb. löten	CAS noch eher (nur wenn 9. nicht genügt)	25/19,27/?
		ist von mir getestet, Anschlußplan anbei	
1. Platinenstecker	evtl. fest verlöten	C) 512k-Karte keine Wackelkontakte, Kapazitäten vermindert	22/19
2. C1, C2	evtl. durch andere ersetzen	RAS/MPX/CAS-Signalfolge für die RAMs sind manchmal auch hier kritisch (RAM von BP läuft dann nicht). Bei mir: C1 raus, C=47 pF	24/10,25/20 24/10,25/20
3. R1, R2	evtl. andere Werte	s. "B) Hauptplatine: 10."	
1. Platinenstecker	evtl. fest verlöten	D) RS232-Platine s. C) I.	22/19
* 2. IC 1	Z80 B DART (oder SIO)	für die schnellere Frequenz nötig	21/30
3. IC 5	evtl. 74HC244	"sauberere" Steuersignale (z.B. RFSH, M1, PHI)	25/20
4. IC 6	evtl. 74HC241 oder 74ALS241	dto.	25/26
* 1. IC 1C	P5 abkneifen, unter der Platine P5 mit P8 verbinden	E) 80-Zeichen-Karte der Video-Kontroller bekommt dadurch die Daten schneller	21/30
1.		F) Floppy-Kontroller-Karte läuft gut	
1.		G) Interface-Karte (kleine Karte in der FDX mit dem CP/M-EPROM)	
2. CP/M-EPROM	Treiber evtl. HC oder ALS kann bleiben	schnellere Signale (hat bei mir keinen Vorteil gebracht) gebootet wird mit 4 MHz	24/13
* 1. Anordnung	MTX zwischen FDX und ECB	H) ECB-Bus Abschluß nach beiden Seiten	25/31,26/27 27/?
2. Terminierung	evtl. bestimmte Signale oder ganze Terminierung raus	bestimmte Steuersignale werden zeitlich verschoben nur nötig, wenn Disk-MTX-Transfer wirklich mit 8 Mhz laufen soll, evtl. hier auf 4 MHz umschalten (Anschluß auf der 8-MHz-Platine vorgesehen), da nur geringer Zeitverlust	
a) c't-CMOS-RAM-Floppy		I) ECB-Karten diese ICs machen beim Schreiben mit 8 MHz evtl. Probleme, der Blockkondensator stabilisiert die Schaltströme	24/11
1. IC1, IC11	je einen Blockkondensator (74LS174, 100 nF) von +5V auf Masse, d.h. P8->P16, direkt einlöten		
b) Conitec-DISI-CMOS-RAM-Floppy		läuft gut	
1.			
c) c't-EPROM-Floppy		an WAIT-Logik anschließen, da die EPROM-Floppy mit reinen 8 MHz nicht mitkommt (warum?). Dieses Signal geht auf low, wenn ein IORQ in Verbindung mit M1 an die EPROM-Karte geht.	24/11
* 1. IC6	von P19 Kabel zur 8MHz-Pl. (74LS68B)		
d) EDICTA-Graphik-Karte		läuft gut	
1.			
e) c't-IO-Karte		in Erprobung (Information folgt)	28/?
1.			