

MTX *User-Club Deutschland*

Info 26
01.04.1988

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

Telefon-Sprechzeiten

Herbert zur Nedden: Do 18 - 22 Uhr, Sa 13 - 16 Uhr

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

Inhaltsverzeichnis**C L U B**

Lesenswertes	Seite 1
Korrektur & Nachtrag	Seite 2
Fragen & Antworten	Seite 2
Ganz zum Schluß noch etwas / Forts	Seite 47

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

Esc A	Seite 3
-------	---------

N e w W o r d

Zeichendefinitionen für den Drucker	Seite 4
Automation	Seite 5
Seitenaufbau	Seite 7

S o f t w a r e

Neuigkeiten	Seite 10
-------------	----------

d B A S E

d B A S E	Seite 11
Errors ...	Seite 17

L e s e r b r i e f

Hans Gras, NL 1506	Seite 16
Herbert Oppmann, 8520	Seite 22
Jan Bredereke, 2000	Seite 27
Anton Reiser, 8977	Seite 28

H a r d w a r e

Diverses	Seite 29
VS4-Probleme	Seite 30
Port 7	Seite 31
Akku für c't Statik-RAM-Floppy	Seite 39
512 kB auf c't Statik-RAM-Floppy	Seite 39

O C R

Klarschriftleser	Seite 32
------------------	----------

L o g i k

Karnaugh-Tafeln	Seite 34
-----------------	----------

S t a t i s t i k

S t a t i s t i k	Seite 44
-------------------	----------

A s s e m b l e r

BLOCK-SWAP/MOVE	Seite 45
-----------------	----------

Preis für dieses Info: DM 10,-

Redaktionsschluß für Info 27: 01. Mai 1988

Liebe Leserin, lieber Leser,

entschuldigt bitte meine Inkonsistenz im letzten Info von wegen dem Termin für die Bestellaktion. Einsendeschluß ist natürlich der Tag der Arbeit, d.h. der 1. Mai 1988.

Im Info 25 habe ich einen Wettbewerb angeboten, bei dem es darum geht ein Steine-Schiebe-Puzzle zu knacken. Ich bin einigermaßen über- rascht, daß sich bislang nur drei Bewerber versuchen - d.h. im Topf sind jetzt DM 28,-. Und diese drei sind auch noch aus meinem engeren Freundeskreis, was ich schon merkwürdig finde. Geknackt wurde es üb- rigens noch nicht - Ihr habt also noch eine Chance.

Just erhielt ich von Jan Bredereke, der sich außer Konkurrenz mit dem Spiel befaßt hat, da ich ihm das Programm mal so gab, eine Lösung. Wenn ich bis 15. Juni 1988 (Datum des Poststempels) keine andere Lö- sung erhalte geht der Topf an Jan.

Die im letzten Info angekündigte Patch-PD CLUB.030 wird vermutlich noch etwas auf sich warten lassen! Alleine der NewWord-Patch mit allem Drum und Dran hat mich einiges an Zeit gekostet - schließlich ist das Patchprogramm so umfangreich, daß es sogar mit Overlays arbeitet. Auf der Patchdiskette belegt der ganze NewWord-Patch-Krempel mit allen Sources knapp 100 kB! (Dafür ist auch alles mögliche und unmögliche drin.) Die anderer Patches sind zwar nicht so kritisch (SuperCalc ist fertig), wird aber halt noch 'ne Weile dauern.

In den Kleinanzeigen biete ich u.a. einige funktionstüchtige Gerä- te/Schaltungen aus meiner früheren Hobby-Elektroniker-Zeit an. Mangels Anfragen habe ich die Preise nocheinmal etwas herabgesetzt - irgend- jemand muß damit doch eigentlich etwas anfangen können! 'Ruf doch mal an!'

In München wird voraussichtlich am 17.4.88 ein Clubtreffen statt- finden. Clubmitglieder aus dem Postleitzahlenbereich 8 erhalten dazu noch eine schriftliche Einladung (sofern das Konto gedeckt ist), ande- re Interessenten sollten sich vorher mit Kurt-Bernd (089 - 6374815) oder Martin (089 - 1231001) in Verbindung setzen.

Ich finde es recht betrüblich, daß immer wieder Mitglieder ihr Kon- to einfach im Sande verlaufen lassen. Immerhin sind ca. 1/3 der Konto- stände so niedrig, daß es nicht für's Info langt. Klar, so ist es ein- facher: Einfach einschlafen lassen. Bitte, wenn Ihr schon dem Memotech und dem Club ade sagen wollt, spendiert eine Postkarte. Kürzlich habe ich alle 'Nicht-Info25-Bezeiher' mittels einer Postkarte auf diesen Umstand hingewiesen, und bis 15. Juni 1988 um eine Antwort gebeten: Das waren über 110 Karten!!!

Immer wieder höre ich von Problemen mit MS.COM, dem Tool zu RAM4, mit dem MSDOS-Disketten gelesen/beschrieben werden können. Bernd hat sich diese Sache angesehen und nach langem Suchen auch die Ursache ge- funden: Die Systemspuren (auch auf Original RAM4-Disc!). Um diesem Übel beizukommen müßt Ihr mit WRITESYS (ist bei RAM4 dabei) Euch neue Systemspuren erzeugen! Horst Kupka hat einen weiteren Kinken entdeckt: MS gibt gelegentlich falsche Dateigrößen an - trotzdem sind die Datei- en in Ordnung, und unter MSDOS wird die Größe der Datei richtig ange- geben.

Hallo EDICTA-Grafiker, mit dem letzten Update der Grafik-Software habe ich einen Fragebogen verschickt, der sogar mit Eurem Absender versehen und an mich (für einen Fensterumschlag) adressiert war. Unter 50% Rückläufer finde ich sehr mager!

Das nächste nordische Clubtreffen, vermutlich in Hemmingen, organi- siert von Christian Löhrmann, wird im August/September 1988 stattfin- den. Leider geht das nicht früher! Wer von Euch Lust hat, daran teil- zunehmen, jedoch in diesen beiden Monaten zeitliche Einschränkungen hat, möge es mir bitte mitteilen! Wir wollen wieder versuchen einen Raum incl. Kaffee und Mittagessen für ca. DM 25-30.- pro Person zu be- kommen.

C L U B: Korrektur & Nachtrag / Fragen & Antworten**Korrektur & Nachtrag**NJW-Leitsatzkartei / Klarschriftleser

Im letzten Info habe ich aus Versehen diese Anfrage unter dem Namen von Uwe Sanders abgedruckt. Der Interessent dieses Themas ist Hermann Salomon, Birkenstr. 32, 2800 Bremen 1, 0421 - 14520

OCR

Diese im Zusammenhang mit o.g. Artikel (NJW-LsK) verwendete Abkürzung bedeutet 'Optical Character Recognition', d.h. optische Zeichenerkennung - auch Klarschriftlesen genannt.

dBFLAG - Bitverarbeitung unter dBASE (Kurt-Bernd Rohloff, 8000)

Leider hat sich trotz ausgiebigen Testens in DBFLAG ein ganzer kleiner dummer Fehler eingeschlichen. Bei A4BA muß natürlich ein RL E hin statt eines RR E! Wenn man schon nur zwei Möglichkeiten hat, erwischt man doch glatt die falsche (Gesetz von Murphy). In der HEX Datei sieht das so aus (zum Glück geht das Patchen da sehr einfach):

statt

```
:10A4B00030FE02DABBA4FE0F2047AFCB1B1FCB1137
```

muß es richtig(er) heißen

```
:10A4B00030FE02DABBA4FE0F2047AFCB131FCB1137
```

Der Fehler tritt beim "G" Aktionscode auf, wenn Wildcards benutzt werden. Das Testergebnis ist dann nicht zuverlässig gewesen. Blöde Sache, tut mir auch echt leid, aber ist nun mal passiert.

Anm.d.HzN.: Auf CLUB.024 ist mittlerweile die fehlerhafte Version ersetzt worden.

PASCAL.PAS von CLUB.028 (Holger Hansen, 3300)

In dieser Datei zwischen den Zeilen 71 und 72 END; (* WHILE *) und START_SUB_PROCESS die folgende Zeile einfügen:

```
FOR I:=1 TO 80 DO CMD(.I.):=#00;
```

Fragen & Antworten

F: Herbert zur Nedden, 2000

Wer kann mit nebenstehender Unterschrift etwas anfangen.

Der Brief stammt von einem 'Ehemaligen' aus 5307 Wachtberg (evtl. umgezogen!)



F: Hat jemand ein kommentiertes NewWord-Listing ?

A: Herbert zur Nedden, 2000 & Uwe Grass, 3300

Es gibt für NewWord ein User-Area-Listing (Vers. 2.02 oder 2.16/17, hat HzN), in dem die Bereiche, in denen u.a. NWINSTAL herumpatchen dokumentiert sind. Für eine Abdruck im INFO ist es viel zu lang.

A: Uwe Grass, 3300

Uhrenanzeige abschalten mit: ?<ESC>AIO

Uhrenanzeige einschalten mit: ?<ESC>AI1

(siehe auch Leserbrief von Jan Brederke)

A: Uwe Grass, 3300

RAM4 Funktionstasten funktionieren mit dBaseII, Vers. 2.41 von Schneider. Die Unterschiede zur englischen Version müßte jemand mit viel Zeit mal auskniften.

F: Uwe Grass, 3300

Wollen wir nicht auch die Telefonnummer in der Klammer beim Namen über einem Artikel mit angeben?

C L U B: Fragen & Antworten / T U R B O: Esc A

A: Uwe Grass, 3300

XDIR3 hat Schwierigkeiten bei Dateien über 512k!

Ein mit LU erstelltes File überschritt diese Grenze und in der Anzeige mit XDIR3 blieb das File bei 512k Größe. Die verbleibende Speicherkapazität der Diskette wurde richtig berechnet, das File selbst war i.O. Wer weiß, wo die mathematische Unbegabtheit des Programms liegt (und behebt sie)?

F: Hans Friz, 6700

Wie kann man den Cursor dauerhaft, oder zumindest in Turbo Pascal und NewWord groß machen, damit man ihn auch findet ?

A: Herbert zur Nedden, 2000

Turbo-Pascal: An den Terminal-Init-String mit TINST folgende Sequenz anhängen: Esc Ä C E oder Esc Ä C < je nach dem, ob der Cursor groß oder klein sein soll.

NewWord: Das kann mein NewWord-Patch-Programm von der Patch-PD.

F: Peter Würfel, 7262

Warum wird die Turbo-Pascal-Zeile:

```
writeln(#27,#91,#88,#27,#91,#71,#27,#91,#84,#49);
```

nicht korrekt ausgewertet, warum klappt das Umschalten nur, wenn ich den Befehl in zwei Zeilen schreibe?

A: Olaf Krumnow, 2050 & Herbert zur Nedden

Die Ausgabe in Klartext:

```
writeln (Esc, 'ÄX', Esc, 'ÄG', Esc, 'ÄT1');
```

Die Bildschirmsteuersequenz Esc ÄG schaltet nicht nur auf das Bildschirmformat G, sondern stellt auch noch eine Funktionstastentabelle ein. Die Nummer der Tabelle muß nach dem 'G' folgen.

Damit bedeutet die o.g. Ausgabe:

Esc Ä X = Bildschirm-Reset

Esc Ä G Esc = Bildschirmformat-G, Fkey-Tabelle lassen, da Esc ungültige Tabellen-Nr. ist

Ä T 1 = Erscheint halt

CR LF = Cursor Spalte 1, Zeile +1

Wird die Ausgabe hingegen in Zwei Zeilen zerlegt, folgt dem Esc ÄG das CR/LF der Zeilenschaltung, kommt folgendes am Bildschirm an:

Esc Ä X = Bildschirm-Reset

Esc Ä G CR = Bildschirmformat-G, Fkey-Tabelle lassen, da s.o.

LF = Cursor Zeile +1

Esc Ä T 1 = Fkey-Tabelle 1

CR LF = Cursor Spalte 1, Zeile +1

Richtige und einfachere Lösung:

```
writeln (#27,'ÄX',#27,'ÄG1');
```

Oder auch in Peters Notation:

```
writeln(#27,#91,#88,#27,#91,#71,#49);
```

Turbo-Pascal alternate Zeichensatz

(Herbert zur Nedden, 2000)

(Damit sich der Einsatz des neuen 80-Zeichen-Satzes auch lohnt)

Natürlich kann man einen der Strings (mit DDT, MONI, DU), die beim Aufruf von Turbo-Pascal ausgegeben werden, überschreiben. Erfreulicherweise geht das auch eleganter, da schließlich TINST, das Installationsprogramm das Installieren eines Terminal-Initialisierungs- und einen Terminal-Uninitialisierungs-String unterstützt.

Man wähle in TINST die Option Screen aus, selektiere einen Schirm (am geeignetesten ist wohl der MTX-Schirm), und modifiziere die Definition (wird schließlich angeboten). Dann nur noch den Terminal-(Un)Init-String durch Esc A (bzw. Esc S) ergänzen!

Selbstverständlich kann hier auch gleich ein Esc Ä T 2 eine andere Funktionstastentabelle selektiert werden, u.s.w.

New Word: Zeichendefinitionen für den Drucker

(Uwe Grass, 3300, 0531/343167)

Zeichen automatisch an den Drucker geben

Es gibt in NewWord eine Möglichkeit den Drucker zu initialisieren (im Patch-Menü des NWINSTAL die Funktion ULINI). Dabei kann es sich um ein einfaches RESET für den Drucker handeln (schwache Leistung des Programmierers, wenn ihm da nichts besseres einfällt), oder es werden ein paar Zeichen neu definiert, damit man sie besser lesen kann (aha, schon besser). Nur reicht der Platz an dieser Stelle nicht für große Texte, man muß also in eine Patch-Area springen. Das geht in NewWord mit der Hex-Eingabe FF im Patch-Menü, anschließend wird die Adresse eingegeben, wohin gesprungen werden soll. Ich habe ein nettes Plätzchen an der Stelle 024Dh gefunden. Also lautet der Sprungbefehl in Hex: FF 4D 02.

An der Stelle habe ich folgendes eingegeben:

```

024D                                     98 1B 40
0250    1B 50 00 00 1B 59 7E 3F 40 84 12 80 12 60 0C 00
0260    1B 59 5C BC 00 42 00 42 00 42 00 BC 1B 59 A1 00
0270    60 90 00 90 60 00 00 00 1B 59 5B 8E 10 28 40 8B
0280    40 28 10 8E 1B 59 A2 3F 00 08 04 00 04 08 00 3E
0290    1B 59 6C 00 00 00 FC 00 02 00 00 00 00 00 1B 59
02A0    A5 44 A8 54 2A 44 80 04 0A 04 1B 59 A4 00 48 90
02B0    0B A0 48 00 00 00 1B 59 A3 00 50 8B 00 AB 50 00
02C0    00 00 1B 59 A7 32 48 06 80 00 80 06 48 32 1B 59
02D0    24 3C 42 1B A5 00 A5 00 42 3C 1B 59 A6 00 00 D0
02E0    20 C0 00 00 00 00

```

Die ersten vier Zahlen links am Zeilenanfang sind die Adressen, an denen im Programm die Bytes stehen. Die erste Zahl im Block des Patches, 98h, gibt die Anzahl der Zeichen an, die den Patch bilden, hier also $9 * 16 + 8 = 152$ Bytes. Die folgende Eingabe, 1B 40, ist ein Drucker-RESET. 1B 50 00 ist die Einstellung auf Elite-Schriftart. Die folgende 00 trennt diese Voreinstellungen von der Definition neuer Schriftzeichen ab, erleichtert mir also das Aufsuchen dieser Stelle, ist für das Funktionieren dieses Patches aber völlig überflüssig. 1B 59 ZZ XX XX XX XX XX XX ist der Code für ein neudefiniertes Zeichen. Dabei bedeuten: ZZ = Hex-Code des neudefinierten Zeichens

XX = neuer Pixelcode des Zeichens (wie man den errechnet stand in einem alten INFO).

Ich habe also der Reihe nach folgende Zeichen neu definiert: ß, Ø, !-kursiv, Ä, "-kursiv, l, %-kursiv, Dollar-kursiv, #-kursiv, '-kursiv, Dollar-Zeichen, &-kursiv. Mit diesen Zeichen habe ich die Ausdrücke mit dem DMX80 und der Olympia Carrera mit den gleichen Zeichen auf dem Papier. Beim Korrekturlesen muß ich keine Abweichungen mehr erwarten, wenn ich den fertigen Text dann auf der Typenradschreibmaschine ausgeben.

Die Kurssiv-Zeichen in der Reihenfolge A1h = ' , A2h = n, A3h = # , A4h = ¢, A5h = %, A6h = ' , A7h = Ω.

New Word: Automation

Die ganzen Angaben beziehen sich auf einen DMX80, für andere Drucker muß es aber ähnlich funktionieren. Allerdings lassen sich diese Zeichen nicht mit <^PY><Zeichen> erreichen, sondern nur über special Customs. Um das Zeichen A1 zu drucken, definiert man das Custom Q mit .xQa108 und gibt im Text an der entsprechenden Stelle <^P>Q<Leerzeichen> ein. Das Leerzeichen ist wichtig, sonst blockt NW falsch und das mit <^P>Q definierte Zeichen wird überschrieben!

Freie Speicherplätze im NewWord sind die sogenannten Patch-Areas, man erkennt sie daran, daß dort Zeilenweise Nullen stehen. Adressen gebe ich nur ohne Gewähr an: altes NW: ab 591h, (ab 4B1h), ab 431h.

neues NW: ab 5B6h, (ab 4B1h), ab 431h.

Die Adressen in Klammern nur dann, wenn das NW nicht wieder mit NWINSTAL installiert werden soll. Außerdem ist der Platz frei, an dem die Funktionstastenbelegung stand, wenn RAM4 benutzt wird.

Die angegebenen Adressen kann man erreichen, wenn man im NWINSTAL-Patch-Menu eine Adresse anspringt, die vor der gewählten Adresse liegt und dann immer mit RET weiterläuft.

UG

Anm.d.HzN.: Auch hier schlägt natürlich mein NewWord-Patch-Programm der Patch-PD (CLUB.030) voll zu. Bis zu ca. 170 Zeichen Drucker-Init-String können damit leicht eingebaut werden.

NewWord-Automation

(Uwe Grass, 3300)

Auch in Newword kann man mit einer Batch-Datei immer wiederkehrende Aufgaben erledigen. Dazu muß man sich von der ORIGINAL-Memotech-CP/M Diskette das File XSUB.COM holen.

Dann nimmt man Herberts MTX-Edit und schreibt eine Batch-Datei. Wenn ein Zeichen zugleich mit der Control-Taste gedrückt werden soll, setze ich die Tastenkombination in <>. **Sonst** müssen **zwei** Tasten gedrückt werden! <RET> = Returntaste, <ESC> = Escapetaste. Kommentare sind kursiv gedruckt.

TEST.SUB sollte so aussehen:

```
xsub<RET>           die .COM-Dateien werden auch auf dem
nw<RET>             Path gesucht.
dtest.txt<RET>     zwischen d und TEST.TXT darf kein <RET>
<^P><^Q>a"         (Eingabe: Ctrl-P Ctrl-P Ctrl-P Ctrl-Q a ")
^ö<RET>           (Eingabe Dach ö)
nr<RET>
<^P><ESC><^K>dx
^c                 entspricht 03H, damit steigt man auch aus
                   ZEX wieder aus
```

(Siehe auch den Leserbrief von Jan Brederke)

New Word: Automation

Antwort zur Frage von Peter Würfel zu ZEX in Info 25-10 (Jan Brederke, 2000)

Peter wollte mit einer ZEX-Batch-Datei NW aufrufen, Anführungszeichen durch nichts ersetzen lassen, die Daten abspeichern und NW wieder verlassen, dies alles natürlich automatisch.

Im Prinzip ist das ganz einfach, die Batch-Datei muß folgendes enthalten:

```
^#
nw test.txt
^qa"öör
^Å^kx
```

Das kleine ö ist je nach Darstellung auch der senkrechte Strich und erzeugt ein Return. ^# verhindert, daß ZEX ständig sagt, daß es noch da ist. ^Å erzeugt ein Escape. ZEX erklärt dies alles übrigens mit dem Kommando

ZEX //

Der accent circonflex ^ ist hier direkt einzugeben, es ist zur Abwechslung kein Control-Code gemeint.

Die oben beschriebene Batch-Datei funktioniert auch (fast) wunderbar, nur speichert Newword unter ZEX bei ^kx, ^kd und ^ks nicht ab, sondern macht sofort einen Warmboot. Warum, weiß ich nicht. ^kq und ^kj gehen, aber auf die Nachfrage, ob man es ernst meint, kommt von ZEX kein Zeichen, sondern man muß selbst Y tippen. Abspeichern kann man nur durch Markieren des gesamten Textes als Block und anschließendes Block-schreiben, aber dabei darf man nicht den Namen einer existierenden Datei angeben, und löschen kann man mit ^kj auch nicht automatisch, wie oben schon gesagt.

Wenn die geänderte Datei also unter demselben Namen abgespeichert werden soll, sehe ich nur folgende, umständliche Möglichkeit:

```
^#
era block.#####
nw test.txt
^qa"öör
^Å^qr^kb^qc^kk^kwblock.#####
^kx
?^ÅAT1
era *.##a##
era *.##b##
era test.bak
ren test.bak=test.txt
ren test.txt=block.#####
```

Falls ihr euch über die Dollar-Inflation wundert: ZEX ersetzt ## durch \$, da \$ immer ein Sonderzeichen einleitet.

Bei dieser Batch-Datei wird Newword unsanft durch den beschriebenen Warmboot nach ^kx verlassen, dabei wird auch die Funktionstastentabelle nicht wieder zurückgestellt, falls dies Newword normalerweise bei euch macht. Dafür habe ich die Zeile mit ^ÅAT1 eingebaut.

Man kann dem Batch-Programm auch den Textdateinamen als Parameter mitgeben, dann muß man in der Batch-Datei "test" durch "\$1" ersetzen.

New Word: Seitenaufbau**NewWord Seitenaufbau**

(Herbert zur Nedden, 2000)

Bislang hatte ich nie Probleme mit meinen Kopfzeilen (Headers) im Info. Der Aufbau war wie folgt:

- | | | |
|-------------|----------------------------|---------------|
| 1. Zeile | 'MTX User-Club Deutschland | Info ## - ##' |
| 2. Zeile | Leerzeile | |
| 3. Zeile | Themen-Überschrift | |
| 4. Zeile | Leerzeile | |
| ab 5. Zeile | Text | |

Nachdem ich an meinem NewWord beim Erstellen des Patch-Programmes herumgebastelt habe, und dabei auch sichergestellt habe, daß nicht fünf verschiedene NewWord-.COM's durch die Gegend geistern fehlte im Ausdruck plötzlich die 4. Zeile. Die Ursache mußte gefunden werden.

Da ich nun in der Firma im Bereich Bürokommunikation, also u.a. Textverarbeitung tätig bin, und dort auch mit solchen Thematiken konfrontiert werde wollte ich dem ganzen richtig auf den Grund gehen.

In NewWord ist ein meines Erachtens sehr eigenwilliges Zusammenspiel der einzelnen Teile einer Seite festzustellen, die ich hier einmal versuchen will aufzuzeigen:

Vorab möchte ich bemerken, daß mit .PL nicht unbedingt die Seitenlänge festgelegt wird!!!!!!

Im folgenden gehe ich also auf das Zusammenspiel der Punkt-Kommandos .PL, .MT, .HM, .MB, .FM sowie .H1-3, .F1-3 ein.

Die ersten 5 dieser Kommandos erhalten einen Zahlenwert (z.B. .PL10), auf den es hier ankommt. Um mir das Leben einfacher zu machen, bezeichne ich im weiteren die beim .XX-Kommando genannte Zahl einfach mit XX, also bei .PL10 ist PL=10. Da wir uns auch für die Anzahl von Text-, Kopf- und Fußzeilen auf einer Seite interessieren wird diese Anzahl mit TX, HR bzw. FR bezeichnet, womit wir haben:

PL	Seitenlänge	Page-Length
MT	Rand oben	Margin at Top
HM	Rand unter Kopfzeilen	Header Margin
MB	Rand unten	Margin at Bottom
FM	Rand vor Fußzeilen	Footer Margin
TX	Anzahl Textzeilen	
HR	Anzahl Kopfzeilen	(werden mittels .H1-3 angegeben)
FR	Anzahl Fußzeilen	(werden mittels .F1-3 angegeben)

Es können bis zu drei Kopf- und Fußzeilen auf einer Seite definiert werden, die mittels der Punkt-Kommandos

.H1, .H2, .H3 bzw. .F1, .F2, .F3 angegeben werden können. (.H0 ist ein Synonym für .H1, .F0 für .F1).

Sollen (wie im Info) zwei Kopfzeilen mit einer Leerzeile dazwischen definiert werden, so muß die mittlere Kopfzeile (.H2) als Leere Zeile angegeben werden:

```
.H1^BMTX User-Club Deutschland^B           Info ## - ##
.H2
.H3^ST h e m a^S
```

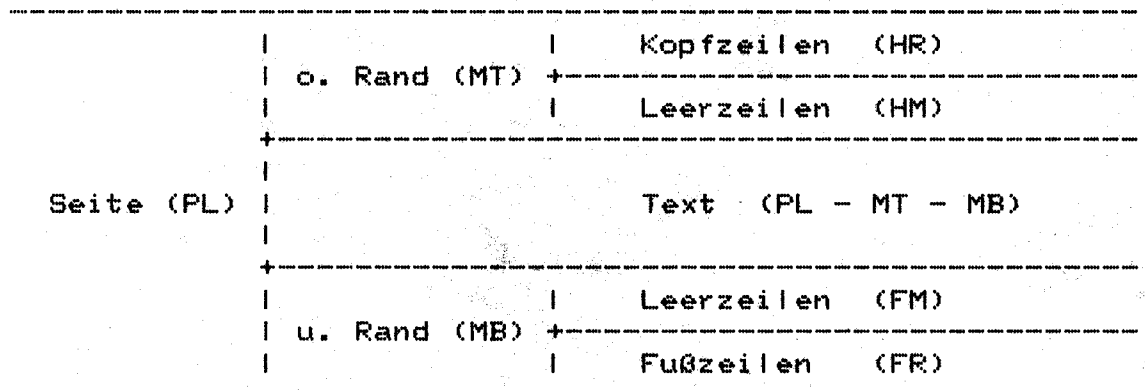
New Word: Seitenaufbau

Bei einer 'pingelig' aufgebauten Seite gilt:

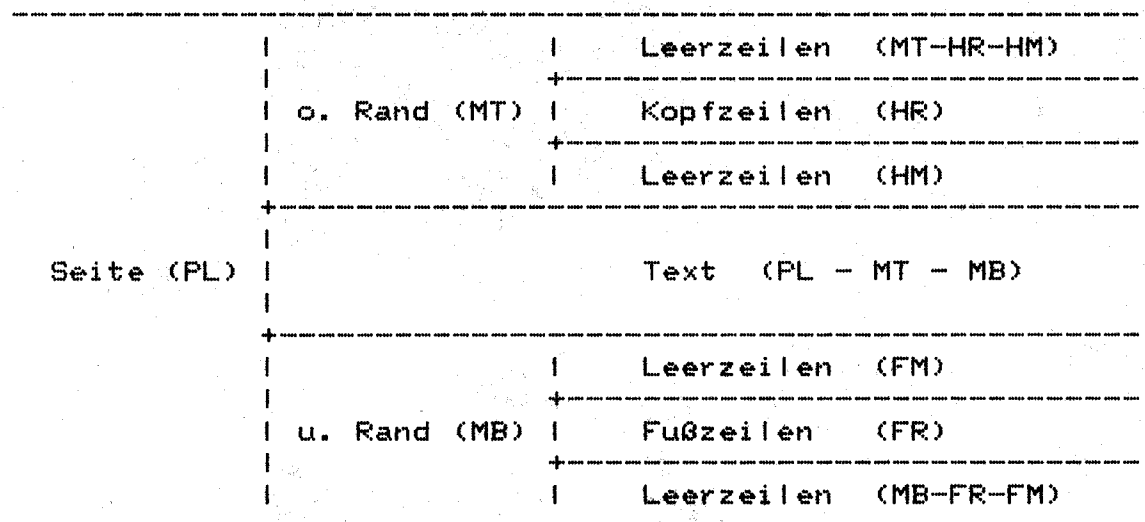
1. $MT = HR + HM$
2. $MB = FR + FM$
3. $TX = PL - MT + MB$

Das bedeutet, daß die Kopf(Fuß)-Zeilen innerhalb des oberen (unteren) Randes liegen, und zwischen Ihnen und dem Text soviele Zeilen frei bleiben, wie durch HM (FM) angegeben.

Das sieht dann so aus:



Wenn nun der Bereich für den oberen (unteren) Rand größer angegeben wurde, als für den Kopf(Fuß)-Zeilen samt Abstand zum Text benötigt, dann werden einfach Leerzeilen am Rand der Seite eingefügt, um diesen Freiraum zu überbrücken:



Es stellt sich die berechtigte Frage, was nun passiert, wenn z.B. die Kopfzeilen samt ihrem Abstand nicht mehr in den oberen Rand passen. Im Prinzip kann man das einfach sagen:

Erst fliegen die Leerzeilen, die den Abstand zum Text sicherstellen raus, und, falls das nicht langt, wird die Seite länger.

Software: Neuigkeiten**BACKUP mit COPY**

(Holger Hansen, 3300)

BACKUP ist ein Kopierprogramm, das RAM4.x voraussetzt. Es besteht aus dem Analyseprogramm BACKUP.COM bzw. BACKUP.CHN und dem Kopierprogramm BACKCOPY.COM.

BACKUP ist so konzipiert, daß es nur dann kopiert, wenn der Zeiteintrag auf dem Ziellaufwerk aktueller ist als der auf dem Quellaufwerk oder das Quellfile auf dem Ziellaufwerk noch nicht existiert.

BACKUP kann man auf zwei Arten aufrufen:

1. die Kommandozeilenversion oder
2. die interaktive Version, die alle Eingaben vom Benutzer abrufen.

COPY ist eine abgemagerte Version von Backup, die sich nicht mehr an Zeiteinträgen orientiert, sondern nur die spezifizierten Dateien von der Quelle nach dem angegebenen Ziel kopiert, wobei auch hiermit zwischen unterschiedlichen Userbereichen kopiert werden kann.

BACKUP und COPY analysieren die Directories und die Benutzereingaben und legen das Ergebnis oberhalb BFF8 im Speicher ab. Die eigentliche Kopierarbeit erledigt ein kleines Maschinenprogramm.

ZEIG - ein verbessertes TYPE

(Kurt-Bernd Rohloff, B000)

Den Ärger mit 'TYPE DATEI.DOC' kennt ja wohl inzwischen jeder (DATEI.DOC soll ein NewWord Dokument sein). Wenn man Pech hat, darf man nach dem Bildschirmgitter im grafischen Zeichensatz und ohne den Cursor weitermachen. Das hat nun ein Ende! Mit 'ZEIG DATEI.DOC' erhält man eine (wenigstens) lesbare Ausgabe von Dokumenten und anderen 'schwer verdaulichen' Dateien. Das können, wenn du willst, auch Programmdateien sein (z. B. wenn man nach Copyright Vermerken suchen will o. ä.). ZEIG verdaut alles, weil es alle Kontrollzeichen wahlweise unterdrückt oder in lesbare Form (^A, ^B usw.) umwandelt. Für NewWord Dokumente ist ein spezieller Modus vorgesehen, der einige wenige NewWord Features unterstützt (z. B. Fettdruck und Unterstreichen). Die Ausgabe kann auch mit auf den Drucker geschickt werden, so daß man damit auch ein primitives Druckprogramm hat (um z. B. mal schnell einen Arbeitsausdruck eines Dokuments anzufertigen oder um mal ein Dokument mit allen Punktkommandos auszudrucken). Als 'Bonbon' hält ZEIG die Ausgabe alle 20 Zeilen an (abschaltbar), so daß man nicht ständig mit ^S auf der Lauer liegen muß. Was ZEIG sonst noch kann, kannst du aus dem mitgelieferten DOC ersehen, wenn du mir eine Diskette (03) mit frankiertem Rückumschlag schickst. ZEIG gibt es auch auf CLUB.024.

Super Tape

(Herbert zur Nedden, 2000)

Wolfgang hat die Programme STLZ, STSZ, STBACKUP und STSZAUTO ausprobiert (CLUB.025) und konnte keinen Fehler finden. Auch Kombinationen wie mit STSZ abspeichern und mit STLMTX laden, oder mit STSMTX sichern und mit STLZ laden brachten das gewünschte Ergebnis.

Er hat die Tests mit meinem MTX/FDX unter RAM4.2 mit einem Philips Datenrecorder D6600/30P sowie Cassetten von AGFA (Ferro I) und BASF Chromdioxid mit Dateien von kleiner 1k bis über 30k Größe durchgeführt. Mit 3600 Baud lief alles glatt. Bei 7200 Baud gibt es Probleme. Auch mit STLZ gelingt das LESEN bei 7200 Baud am MTX nicht, wohl aber an einem alten ZXB1.

Auf CLUB.025 ist von Wolfgang eine etwas andere 'Benutzeroberfläche' zu SuperTape - d.h. CLUB.025 wurde ergänzt.

d B A S E

dBASE Tips

(Kurt-Bernd Rohloff, 8000)

Mittlerweile habe ich schon etliche dBASE Programme geschrieben. Was mir dabei aufgestoßen ist, möchte ich hier in willkürlicher Reihenfolge darlegen.

1) Vergleiche von Strings

dBASE kennt bekanntlich zwei Vergleichsmodi: EXACT ON und EXACT OFF. Bei ON ist alles normal, das folgende bezieht sich nur auf OFF. Dieser Modus kann Verwirrung stiften. So ist "antonia" = "anton" TRUE, jedoch "anton" = "antonia" FALSE! Linke und rechte Seite werden hier also nicht mehr gleichwertig behandelt. Was macht dBASE da? Anscheinend wird der rechte String linksbündig 'über' den linken gelegt und dann über die ganze Länge des rechten Strings zeichenweise verglichen. Wenn sich dabei immer Übereinstimmung zeigt (und evtl. der linke noch nicht zu Ende ist), ist das Ergebnis TRUE, sonst FALSE. Der längere String (oftmals ein Datenfeld) sollte in diesem Fall also links stehen und der kürzere (oftmals eine Variable) rechts. Bei gleich langen Strings tritt dieses Problem natürlich nicht auf.

2) Wozu ist TYPE() gut?

TYPE(<Ausdruck>) liefert den Typ des Ausdrucks als eines der Zeichen "C" (Character), "N" (numerisch), "L" (logisch) oder "U" (undefiniert). Was kann man damit anfangen? Stellen wir uns mal vor, wir hätten für ein Programmpaket, bestehend aus den Programmen (Dateien) A1, A2, A3 ..., eine Universalroutine SUBR geschrieben (z. B. formatierte Anzeige eines Datensatzes). Nun schreiben wir ein neues Unterprogramm A50, das sich auch die Dienste von SUBR zunutzen machen könnte, wenn ... ja wenn man SUBR an einer Stelle etwas erweitern würde. Diese Erweiterung darf aber bei allen bisherigen Aufrufern nicht zum Tragen kommen. Was wir brauchen, ist eine neue logische Variable NEU, mit deren Hilfe wir in SUBR entscheiden können, ob der Aufruf von A50 erfolgte (T) oder nicht (F). Schematisch sieht das in SUBR dann so aus:

```
IF NEU
  * zusätzliche Anweisungen für A50
  ...
ENDIF
```

Das Problem ist jetzt, das wir alle 'alten' Aufrufer von SUBR ausfindig machen müssen und dort vor dem Aufruf die Zeile

```
STORE F TO NEU
```

einfügen müssen. Das riecht nach Arbeit. Dabei geht es viel eleganter und sicherer. Wir setzen die Variable NEU am Anfang von SUBR einfach auf F, wenn sie undefiniert ist! In SUBR:

```
IF TYPE(NEU) = "U"
  STORE F TO NEU
ENDIF
...
IF NEU
  * zusätzliche Anweisungen für A50
  ...
ENDIF
```

d B A S E

3) Schleifen mit LOOPings

Mit LOOP kann man sofort an den Anfang einer DO WHILE Schleife zurückspringen. Ich verwende ihn gern, um nach einer Benutzereingabe die eingegebenen Daten zu prüfen. Dafür kann man folgendes Schema verwenden:

```

STORE F TO INPUT:OK
DO WHILE .NOT. INPUT:OK
  * Maske ausgeben
  $ 5,1 SAY "... " GET VAR1
  ...
  READ

  * Fehleranzeige löschen:
  $ 22,1
  * Prüfungen: VAR1 soll nicht negativ sein (als Beispiel)
  IF VAR1 < 0
    $ 22,1 SAY "FEHLEINGABE"
  LOOP
ENDIF

  * nächste Prüfung usw.
  ...
  * alles gut gegangen:
  STORE T TO INPUT:OK
ENDDO

```

Die IF Abfragen werden dann immer so formuliert, daß der Negativfall TRUE ergibt (oder .NOT. (Positivfall)). Wenn dann der LOOP ausgeführt wird, ist INPUT:OK immer noch F, so daß die Schleife nochmal durchlaufen wird. Sind am Ende alle Prüfungen positiv verlaufen, wird INPUT:OK auf T gesetzt, um die Schleife zu verlassen. Der Nachteil bei dieser Methode ist, daß der Benutzer bei mehreren Fehlern die verschiedenen Fehlermeldungen nacheinander bekommt.

4) TOO MANY MEMORY VARIABLES

Leider ist die Anzahl der Speichervariablen in dBASE auf 64 beschränkt. Das kann bei großen Programmpaketen zu Problemen führen. Dagegen kann man zweierlei tun: rationeller Gebrauch und kompakte Speicherung logischer Variablen. Letzteres werde ich weiter unten näher ausführen. Was meine ich nun mit rationellem Gebrauch? Betrachten wir ein Unterprogramm, so lassen sich seine Variablen in drei Klassen einteilen: globale Variablen (bestimmte Systemparameter, die in vielen UP's gebraucht werden o. ä.), Übergabevariablen (zur Steuerung des UP's) und lokale Variablen. Letztere werden nur von dem UP benötigt. Dort kann man nun ansetzen und diese vor dem RETURN mit dem RELEASE Befehl wieder freigeben. Ich gehe dazu konkret folgendermaßen vor:

d B A S E

Innerhalb eines (logisch zusammenhängenden) Unterprogrammabbaues fangen alle alle UP Namen mit der gleichen Buchstabenfolge an (z. B. MTBA). Danach werden sie durch Ziffern unterschieden (MTBA1, MTBA2, MTBA3 usw.) Die Ziffer ist meistens identisch mit der Auswahlziffer aus einem Menü des aufrufenden Programms. Im Beispiel gibt MTBA ein Menü mit den Auswahlmöglichkeiten 1, 2 und 3 aus und ruft daraufhin eines der UP's MTBA1, MTBA2 oder MTBA3 auf. Diese können Folgemenus ausgeben, woraus sich dann die nächste Ziffer bestimmt (MTBA11, MTBA12 usw.). Jede Ziffer steht (in der Regel) für eine UP-Ebene, so daß man nebenbei im Directory leicht die Verschachtelung (den Baum) erkennt. Nun zu den Variablen. Alle (lokalen) Variablennamen beginnen mit dem Namen des Programms. In MTBA fangen alle lokalen Variablen mit den Buchstaben MTBA an. Danach folgt mindestens ein weiterer Buchstabe (z. B. MTBAOK, MTBALETT usw.). Gleiches gilt für seine UP's. In MTBA1 möge es beispielsweise die Variablen MTBA1LOE, MTBA1NEU usw. geben. In MTBA11 dann entsprechend MTBA11OPT, MTBA11ALT usw. Am Ende von MTBA11 werden dessen Variable durch den Befehl 'RELEASE ALL LIKE MTBA11*' wieder freigegeben. Durch die Konstruktion der Namen vermeidet man 'unfreiwillige' Freigaben! Entsprechend endet MTBA1 mit 'RELEASE ALL LIKE MTBA1*'. Damit gibt es alle seine Variablen und, falls bis jetzt noch nicht geschehen, die seiner Unterprogramme frei. Schließlich endet MTBA mit 'RELEASE ALL LIKE MTBA*', womit dann alle Variablen aus dem MTBA-Baum weg sind. Wenn man sich an obige Namensregeln hält, kann man auch bedenkenlos in einem UP neue Variablen hinzufügen oder nicht mehr benötigte entfernen, ohne daß man den RELEASE Befehl ändern muß.

5) Kompakte Speicherung logischer Variablen

Für logische Variablen legt dBASE ein Byte an, obwohl eigentlich ein Bit genügen würde. Da ich in einem meiner UP's etliche logische Variablen benötigte, sann ich auf Abhilfe. Die Lösung ist die Assembler Schnittstelle von dBASE! Man kann in dBASE eigene Assembler Programme einbinden. Der Speicherbereich ab A400h aufwärts wird nämlich nur vom SORT Befehl belegt. Sofern man ihn nicht braucht, kann man in diesen Bereich eigene Routinen einlagern. Dazu übersetzt man zunächst die Assembler Quelle in eine HEX Datei (ich benutze dazu ZBOASM). Die ORG Adresse wird dabei auf A400h gelegt. dBASE kann diese Hex Datei laden. Das geht mit dem Befehl

```
LOAD <datei>
```

wobei <datei> der Dateiname ohne den Suffix '.HEX' ist. **Dieser Befehl muß nach jedem SORT wiederholt werden!** Danach setzt man die CALL Adresse auf 41984 dezimal = A400h:

```
SET CALL TO 41984
```

Nun kann man die Routine aufrufen mit

```
CALL <variable>
```

wobei <variable> eine Speichervariable vom Typ C ist. <variable> ist optional. Das Assemblerprogramm erhält in HL die Adresse von <variable>. Vor dem eigentlichen String setzt dBASE in einem Byte die Länge des Strings. Auf dieses Längenbyte zeigt HL. Auf den höheren Adressen folgen die Zeichen des Strings. Die Assembleroutine darf den Stringinhalt verändern, nicht jedoch seine Länge! Der modifizierte String steht danach in dBASE unter dem Variablennamen zur Verfügung. Das Assemblerprogramm muß sich mit REI beenden.

d B A S E

Nun habe ich mich an die Arbeit gemacht und ein Assemblerprogramm DBFLAG geschrieben, mit dem man bequem einzelne Bits eines Bytes manipulieren kann. Damit kann ich jetzt bis zu 8 logische Entscheidungen in einer einzigen Variablen speichern! Neben setzen, löschen und testen eines Bits kann das Programm auch alle 8 Bits mit einer Bitmaske verknüpfen, u.a. logisch UND, ODER und EX-ODER. Nähere Einzelheiten sind dem beigelegten DOC File zu entnehmen (der versehentlich schon im vorigen Info abgedruckt wurde). Das Assembler Quellprogramm kann nur bei mir gegen Einsendung einer Diskette (Typ 03, formatiert!) und eines frankierten Rückumschlags bezogen werden. Es läßt sich mit dem ZBOASM von H.-H. Herder assemblieren. Die HEX Datei, die Herbert auch hat, gibt es als Zugabe gratis (sie ist auch auf CLUB.024).

Zu den vermeintlichen dBASE Macken

(Kurt-Bernd Rohloff, B000)

Ich beziehe mich auf die Fragen von Peter Würfel zu dBASE im Info 25)

- a) Das angesprochene Problem mit der Zeile 0 ist keine Macke. dBASE benutzt in den Bildschirmmasken die Zeile 0 bereits als Statuszeile. Dort erscheint "INSERT", wenn man mit CTRL-V in den Einfügemodus schaltet und "DELETED", wenn man mit CTRL-U einen Datensatz löscht. Daher sollte die Zeile 0 in Bildschirmmasken nicht benutzt werden.
- b) Diese Absonderlichkeit war mir neu, sie ist aber leicht erklärlich und aus dBASE Sicht auch konsequent. Das Semikolon wird in dBASE in zwei verwandten, aber doch leicht unterschiedlichen Arten interpretiert. Zum einen läßt es sich verwenden, um eine Ausgabe auf mehrere Zeilen aufzuteilen. Dies ist in Peters Beispiel geschehen. (Bei RAW ON wäre dieser Effekt übrigens nicht aufgetreten. Dort wird das Semikolon "wörtlich" ausgegeben.) Häufiger benutzt man es in diesem Sinne jedoch beim REPORT Befehl, um mehrzeilige Spaltenüberschriften zu erzeugen. Jedes Semikolon in der Spaltenüberschrift wird als Zeilenwechsel interpretiert. Dabei werden die Überschriften in Bezug auf die Spaltenbreite zentriert. Dies ist in Peters Beispiel auch geschehen, jedoch in Bezug auf die Gesamtlänge des Textes (also wenn man Semikolon wörtlich nimmt und nicht als Zeilenwechsel). Beim REPORT Befehl kann man durch voranstellen von '<' oder '>' den Text auch links- bzw. rechtsbündig stellen. Diese Ausrichtung gilt dann auch für die folgenden durch ein ';' erzeugten Zeilen. (Beim '?' Befehl ist mir das nicht gelungen). Beispiele:

```

Anzahl;in Stück      <Anzahl;in Stück      >Anzahl;in Stück
erzeugt die Überschriften
+-----+ +-----+ +-----+
      Anzahl      Anzahl      Anzahl
      in Stück    in Stück    in Stück
    
```

Zum zweiten kann man das Semikolon in Programmen benutzen, um eine Anweisung auf mehrere Zeilen zu verteilen. Ist das letzte Zeichen einer Zeile (keine Kommentarzeile!) ein Semikolon, so gilt die nächste Zeile als Fortsetzung der vorigen. Beispiel:

```

DO WHILE (bedingung1 .AND. bedingung2) ;
OR (bedingung3 .AND. .NOT. bedingung4)
    
```

d B A S E

Bei dieser Gelegenheit möchte ich noch erwähnen, daß man speziell den '?' Befehl auch anders fortsetzen kann. Dazu gibt es in dBASE den (wohl nicht so bekannten) '??' Befehl. Er beginnt seine Ausgabe im Gegensatz zu '?' nicht mit einer CR/LF Sequenz. Er eignet sich daher besonders gut dazu, wenn eine Ausgabe einen gemeinsamen Anfang, jedoch ein von einer Bedingung abhängiges Ende haben soll. Im folgenden Beispiel nehme ich an, daß die logische Variable FILEINUSE den Wert TRUE hat, wenn gerade die Datei ANTON eröffnet ist und FALSE, wenn die Datei BERTA eröffnet ist. Um den Benutzer über die eröffnete Datei zu informieren, kann man so vorgehen:

```
? "Aktuelle Datei ist"
```

```
IF FILEINUSE
  ?? "ANTON"
```

```
ELSE
  ?? "BERTA"
```

```
ENDIF
```

Dabei schließt dBASE die Ausgabe in der RAW OFF Betriebsart mit einem Blank ab, so daß die Ausgabe (bei TRUE)

```
Aktuelle Datei ist ANTON
```

```
resultiert. Bei RAW ON wäre dagegen
```

```
Aktuelle Datei istANTON
```

herausgekommen. Man kann das übrigens sehr schön analysieren, wenn man die Ausgabe in einer ALTERNATE Datei mitprotokollieren läßt und sich diese danach mit ZEIG filename AI_Ü anzeigen läßt.

dBaseII, Vers. 2.41

(Uwe Grass, 3300, 0531/343167)

Die dBaseII Version von Schneider kommt mit deutschen Texten. Unter anderem wird bei der Erstellung eines Reports von dem Programm Fragen gestellt, die mit J oder N zu beantworten sind. Nun ist aber nur der Text in dem Programm eingedeutscht, die Antworten müssen aber mit Yes oder No beantwortet werden, sonst läuft es nicht so, wie es sein soll (bei der Erstellung des Reports wird J wie Nein gewertet!).

Der Vorteil dieser Version ist, daß die Funktionstasten unter RAM4 ausgewertet werden.

UG

Leserbrief: Hans Gras, NL 1506

Wegen dem minimalen Interesse der deutschen Mitglieder, habe ich mich entschlossen meine BASICODE Programme in Public Domain zu geben. Leider hat nur 1 Person reagiert (für einen ATARI !!!!!). Ich habe selbstverständlich keine deutsche Betriebsanleitung geschrieben. Das macht mir zu wenig Spaß! Aber, eine holländische Anleitung ist dabei für Leute die diesmal interessiert sind. Bitte erst nochmal Info 21, Seite 9 lesen.

Anm.d.HzN.: Kommt auf eine der nächsten CLUB.xxx-PD's

dBASE II

Ich habe das dBINSTAL programm benutzt um anzugeben welches Laufwerk das default Laufwerk ist. Es klappt auch für das .TXT file. Warum die Vorschläge von Uwe Grass?

Und dann, die (hoffentliche) gute Antwort auf ihre Frage: Warum gehen bei ihren Ausdrucken von Mitglieder-Listen einige Sachen schief? Ihre Ideen von .NDX Problemen sind wahrscheinlich richtig!!! Ich habe auf der Diskette das Programm PUTEOF.COM anbei (kein Programm von mir). Das kann eine (unschöne) Lösung sein.

Anm.d.HzN.: Kommt auf eine der nächsten CLUB.xxx-PD's (nach TEST)

Antworten:

Auf Frage von Dietrich Jerg, Info 25 Seite 7:

Meine Tastaturbelegung für dBASE sieht aus wie folgt:

Normal:

PAGE = ^S	EOL = ^Y	BRK = ^C	F1 = ^W	F5 = ^W
TAB = ^I	UP = ^E	DEL = ^G	F2 = ^W	F6 = ^Q
LEFT = ^H	HOME= 20x^E	RIGHT= ^D		F7 = 8x^E
INS = ^V	DOWN= ^X	CLS = ^U		F8 = 8x^X

Shift

PAGE = ^P		DEL = ^T	
LEFT = ^Z	HOME= 20x^X	RIGHT= ^B	F3 = ^T
		CLS = ^N	

Ich habe das so gemacht, das es ähnlich arbeitet wie Newword. Prüfen mal nach in Manual was alles bedeutet.

Auf Frage von Rainer Tews, Info 25 Seite 8:

Mit nur einen niederohmigen Widerstand in Reihe mit dem Ventil. Der Spannungsabfall über diesen Widerstand ist linear zum Strom.

Auf Frage von Joachim Höft, Info 25 Seite 8:

Das RING-RomPack ist mit Kabel bei mir für DM 25.00 (incl. P+V) zu bekommen. Verrechnung über's Club-Konto. Es gibt beim MTX-Computer TIME\$ in der Ecke rechts oben aus, genau wie FDX mit RAM 4.2, und ist Interrupt-gesteuert.

mit freundlichem Gruß,

Hans Gras,
Statenhoek 49,
NL 1506 VM ZAANDAM.
tel. 0031-75174991.

P.S.: So eben in einer (Holländische) Mailbox gefunden:

DEBUGSII.TXT, mit einige Fehler u.s.w.

Es is natürlich in Holländisch, aber die wichtige Sachen müßten klar sein. Siehe folgende Seiten:

d B A S E: Errors ...

dBASE II is in de loop der jaren tamelijk berucht geworden door het grote aantal zogenaamde bugs. Dat zijn zwakke plekken in de programmacode. Die komen vaak alleen in bijzondere situaties aan het licht, bijvoorbeeld bij een ongebruikelijke combinatie van instructies, of na een foutieve invoer van een commando of gegevens. De fabrikant, Ashton-Tate, spreekt in die gevallen van een "anomaly", ofwel "onregelmatigheid", maar in feite gaat het gewoon om ontwerpfouten. En die kunnen ons heel wat tijd en ergernis kosten, en soms ook gegevens, of zelfs een heel bestand.

Een voorbeeld. Als een bedrag in een N-rubriek precies 100.00 bedraagt raakt er wel eens een 0 van zoek - slechts 10.00 is er dan overgebleven.

Een ander voorbeeld. Typen we na opening van de correctie-dialoog per ongeluk ^P zonder dat er een printer is ingeschakeld, dan veroorzaakt het indrukken van een volgende toets, ook van de RETURN- of de ESC-toets, een systeem-foutmelding. dBASE II is dan al uitgeschakeld, en alle in het geheugen opgeslagen gegevens zijn we onherroepelijk kwijt.

En een laatste voorbeeld. Laten we eens aannemen dat we beschikken over twee aan elkaar verwante bestanden, waarvan het eerste 1020 entrees telt, en het tweede 1160. Trachten we nu deze twee bestanden met behulp van een APPEND FROM-commando te verenigen, dan behoort dBASE II, na de verwerking van het commando, vanzelfsprekend mede te delen dat er aan het ontvangende bestand 1160 entrees zijn toegevoegd. Maar in plaats daarvan is er slechts van 1147 entrees sprake. Nader onderzoek leert dat er inderdaad 13 entrees spoorloos verdwenen zijn - en wonderlijk genoeg niet uit de staart van de collectie, maar ergens uit het midden.

Van de bugs in de oerversie van dBASE II - Wayne Ratliff, de auteur van het programma, hield het in een openhartige bui op zo'n driehonderd stuks - werden er telkens bij het uitbrengen van een nieuwe versie (2.3 in 1982, 2.4 in 1983, 2.41 en 2.43 in 1984, en 2.43* in 1985) een aantal weggewerkt of onschadelijk gemaakt. Heel goed. Maar verdwenen zijn ze nog lang niet allemaal. En dus moeten we, of we nu met een oude dan wel met een recente versie van dBASE II te maken hebben, in allerlei situaties en met tal van commando's heel voorzichtig zijn. Dat zal het onderstaande overzicht wel duidelijk maken.

* * *

?? of @ ... SAY CHR(7) - heeft geen piepgeluid tot gevolg. Rechtgezet met ingang van versie 2.4.

@ ... SAY - invulvensters waarbij de symbolen 9 en/of # zijn gebruikt functioneren niet goed. Rechtgezet in de 2.4-versie.

- heeft als de printer aan staat geen effect in het .TXT-bestand dat ten gevolge van een SET ALTERNATE-commando ontstaat.
- na een typfout, bijvoorbeeld SAT, blijft de correctie-dialoog uit. In plaats daarvan loopt de computer vast en is een RESET-procedure onvermijdelijk - ten koste van de gegevens in het computergeheugen.

APPEND - heeft het bestand een breedte van 24 of meer rubrieken en staat de cursor op de eerste positie van rubriek 24, dan heeft een druk op de RETURN-toets het afsluiten van de procedure tot gevolg, in plaats van het verspringen van de cursor. Rechtgezet met ingang van versie 2.4.

APPEND FROM - functioneert als een bestand meer dan 300 entrees omvat niet betrouwbaar. Entrees verdwijnen spoorloos (zie de voorgaande pagina), of er komen zomaar blanco entrees bij. Nog steeds niet rechtgezet.

- overdracht van gegevens naar een leeg bestand terwijl een SET CARRY ON-commando van kracht is veroorzaakt afval.

Rechtgezet met ingang van versie 2.4.

d B A S E: Errors ...

- afbreken via een druk op de ESC-toets veroorzaakt schade aan het bestand van bestemming. Rechtgezet met ingang van versie 2.41.
 - als de bestanden van herkomst en bestemming beide een omvang van 1000 tekens per entree hebben, en dezelfde structuur, veroorzaakt de procedure de toevoeging van blanco entrees. Rechtgezet met ingang van versie 2.41.
 - geeft geen gelegenheid tot corrigeren als de toevoeging SDF achterwege is gebleven. Met ingang van versie 2.41 rechtgezet.
- BROWSE** - functioneert niet goed bij rubrieken met een breedte van 80, of een veelvoud daarvan. Rechtgezet met ingang van 2.4.
- de instructies ^N en ^T functioneren niet goed. Rechtgezet met ingang van versie 2.4.
- CANCEL** - soms worden niet alle programmabestanden afgesloten. Rechtgezet met ingang van versie 2.4.
- CREATE** - drukken we in antwoord op het verzoek ENTER FILENAME de ESC-toets in, dan loopt dBASE II vast. Rechtgezet met ingang van versie 2.41.
- DELETE FILE** - het aangegeven bestand wordt weliswaar gewist maar blijft open, zodat er fragmenten uit andere bestanden in terecht kunnen komen en de systeem-administratie op de diskette een warboel wordt.
- DISPLAY FILES LIKE *.*** - commando is onverenigbaar met ALTERNATE-bestand. Rechtgezet met ingang van versie 2.4.
- DISPLAY MEMORY** - geeft soms geen juiste omvang van variabelen. Rechtgezet met ingang van versie 2.4.
- DISPLAY STRUCTURE** - geeft geen betrouwbare informatie als een bestand nog (of weer) leeg is. Rechtgezet met ingang van versie 2.4.
- geeft na een COPY-procedure, ook als er in het aantal rubrieken een wijziging gekomen is, en ook onder een andere naam, als datum van laatste bewerking die van het oorspronkelijke bestand weer.
 - blijft het oorspronkelijke aantal entrees aangeven, ook al zijn er met INSERT of INSERT BEFORE een of meer entrees aan het bestand in kwestie toegevoegd.
- DO** - aanspreken van een .PRG-bestand dat niet via MODIFY COMMAND tot stand is gekomen maar bijvoorbeeld via een tekstverwerkingsprogramma, mislukt soms. Gewoonlijk functioneert het commando wel als we het achtervoegsel van het programmabestand noemen. Rechtgezet met ingang van versie 2.41.
- DO CASE** - een genestelde CASE-opdracht functioneert niet altijd correct. Rechtgezet met ingang van versie 2.4.
- macro's binnen een CASE-opdracht functioneren niet altijd goed. Rechtgezet met ingang van versie 2.4.
- EJECT** - heeft soms het uitdraaien van meer dan een enkel vel papier tot gevolg. Rechtgezet met ingang van versie 2.4.

d B A S E: Errors ...

- GO of GOTO n - functioneert niet goed bij een geïndexeerd bestand. Met ingang van versie 2.4 rechtgezet.
- functioneert soms niet goed als n het nummer van de laatste entree is. Rechtgezet met ingang van versie 2.4.
 - weigert als entree n van het afleg-teken voorzien is - ook als tevoren een SET DELETE(D) ON-commando gegeven is. Met ingang van versie 2.41 rechtgezet.
- IF ... ENDIF - als onmiddellijk na de F van ENDIF, dus zonder afscheidende spatie, tekst (commentaar of wat dan ook) volgt werkt ENDIF niet. Met ingang van versie 2.4 rechtgezet.
- INDEX TO - het indexeren van een bestand lukt niet altijd. Soms volgt, na het oproepen van een geïndexeerd bestand, een foutmelding. Meestal is het nogmaals indexeren van het bestand afdoende.
- het indexeren op een rubriek van het L-type functioneert vaak niet goed. Nog steeds niet rechtgezet.
- INSERT - openen we een INSERT-procedure en voegen we geen gegevens in, dan kan het bestand onklaar raken. Rechtgezet met ingang van versie 2.4.
- werken we met een .NDX-bestand, dan worden er op de aangegeven positie drie entrees ingevoegd. Rechtgezet met ingang van versie 2.4.
- JOIN - schakelt de werking van de ESC-toets uit, zodat het indrukken geen enkel gevolg heeft. Rechtgezet met ingang van versie 2.43*.
- JOIN ... FOR P.<sleutel>=S.<sleutel> - heeft een nieuw bestand tot gevolg, met twee identieke rubrieken. Rechtgezet met ingang van versie 2.41.
- LIST - beschadigt bestanden met een breedte van ongeveer 1000 tekens en een L-rubriek die onmiddellijk op een N-rubriek volgt. Rechtgezet met ingang van versie 2.41.
- LOCATE - typen we WHILE in plaats van FOR, dan zien we geen foutmelding, en de computer loopt vast. Rechtgezet met ingang van versie 2.41.
- LOOP - reageert soms op de verkeerde DO WHILE, zodat niet de bedoelde lus tot stand komt. Rechtgezet met ingang van versie 2.4.
- MODIFY COMMAND - typfout MODIFY COMMAD scheidt een bestand genaamd AD.PRG. Rechtgezet met ingang van versie 2.4.
- proberen we met de instructie ^W een nieuwe of verbeterde programmatekst vast te leggen, dan krijgen we soms de systeem-prompt in plaats van de stip te zien, terwijl de zojuist vervaardigde tekst of de ingevoerde wijzigingen verdwenen zijn. Rechtgezet met ingang van versie 2.4, al doen we er verstandig aan om geen bestanden te vervaardigen van meer dan 4000 bytes (4 Kb).
 - breekt regels met een breedte van meer dan 80 tekens af en knoopt ze aan de volgende vast. We doen er verstandig aan om regels niet breder dan 77 tekens te maken - dan is er nog ruimte voor een line feed en/of carriage return-tekens.
- MODIFY STRUCTURE - werkt niet als dit commando onmiddellijk na een MODIFY COMMAND-opdracht gegeven wordt. Er verschijnt afval in beeld. We moeten eerst een willekeurig ander commando geven, bijvoorbeeld ERASE.

d B A S E: Errors ...

- PACK - bij geïndexeerde bestanden van enige omvang raken entrees in het ongereede na een PACK-opdracht. Rechtgezet met ingang van versie 2.4 in zoverre, dat een PACK-procedure mogelijk is bij het niet-geïndexeerde bestand, dat daarna weer moet worden geïndexeerd.
- maakt geen ruimte vrij na het wissen van een aantal of van alle entrees, wat blijkt uit een vergelijking van de omvang van het bestand voor en na de opdracht. Een oplossing is kopiëren na de PACK-opdracht.
- QUIT - na werkzaamheden met een secundair bestand wordt het niet correct afgesloten, zodat eventuele wijzigingen niet (of niet alle) op de diskette terechtkomen. We dienen eerst het commando CLEAR te geven.
- READ - bestaat de inhoud van een L-rubriek uit de kwalificatie .T., dan verandert die onmiddellijk na het indrukken van de RETURN-toets in .F. Rechtgezet met ingang van versie 2.41.
- RENAME - na een omdoop-procedure kan de vorige bestandsnaam nog onverwacht opduiken. Rechtgezet met ingang van versie 2.4.
- REPLACE ... WITH - proberen we per ongeluk de inhoud van een N-rubriek te vervangen door getallen van 50 tekens of meer, dan wordt het bestand van bestemming vernietigd. Rechtgezet met ingang van versie 2.41.
- REPORT FORM - optellen van 24 rubrieken met behulp van TOTAL of SUBTOTAL werkt niet goed. Rechtgezet met ingang van versie 2.4.
- verbeteren van een tyfout via de correctie-dialogoog leidt niet tot het gewenste resultaat: we krijgen het bericht FILE IS CURRENTLY OPEN te zien. Rechtgezet met ingang van versie 2.43*.
 - werkt foutief als de inhoud van een rubriek een puntkomma bevat.
- RESET - kan bestanden beschadigen die twee of meer diskettes omspannen. Rechtgezet met ingang van versie 2.4 in die zin, dat het commando in het vervolg door de letter van de drive in kwestie en een dubbele punt gevolgd moet worden.
- SAVE - de vorm SAVE ALL EXCEPT ... werkt niet. Rechtgezet met ingang van versie 2.43*.
- SET ALTERNATE TO - wordt dit commando gevolgd door een spatie, dan komt er een leeg .TXT-bestand op de aangegeven diskette. Rechtgezet met ingang van versie 2.4.
- SET CONFIRM ON - het invoeren van gegevens van het N-type verloopt niet altijd vlekkeloos. Rechtgezet met ingang van versie 2.4.
- SET ESCAPE ON - functioneert niet bij ACCEPT en INPUT. Rechtgezet met ingang van versie 2.4.
- SET EXACT ON - functioneert soms niet goed als de inhoud van een rubriek overtollige spaties bevat.
- SET INDEX TO - werkt uitsluitend bij het primaire bestand. Rechtgezet met ingang van versie 2.4.
- het bijwerken van .NDX-bestanden tijdens APPEND-, EDIT- en READ-procedures duurt erg lang. Rechtgezet met ingang van versie 2.41.

d B A S E: Errors ...

- functioneert niet als de sleutelrubrieken van verschillend type zijn. In zoverre rechtgezet met ingang van versie 2.4, dat we een N-rubriek, zoals gedemonstreerd op pagina 135, eerst via de STR-functie in C-informatie moeten omzetten.
- SET LINKAGE ON - functioneert niet goed als commando's met een andere term dan NEXT gegeven worden. Rechtgezet met ingang van versie 2.4.
- functioneert niet goed bij koppeling van een geïndexeerd aan een niet-geïndexeerd bestand. Rechtgezet met ingang van versie 2.4.
 - functioneert niet goed als het commando pas wordt gegeven nadat de te koppelen bestanden zijn opgeroepen. Niet rechtgezet - het commando dient tevoren gegeven te worden.
 - de commando's GOTO en SKIP werken niet, of niet goed.
- SKIP - reageert op een variabele met als inhoud een negatieve waarde, en op de uitdrukking VAL("-1"), als op positieve getallen. Het minteken blijft dus buiten beschouwing. Rechtgezet met ingang van versie 2.41.
- SORT - reageert niet op het indrukken van de ESC-toets. Rechtgezet met ingang van versie 2.43*.
- een bestand van enige omvang (meer dan 257 entrees) raakt soms in het ongereede. Zoals elders in deze handleiding al is opgemerkt verdient INDEX de voorkeur boven SORT, alleen al vanwege de tijdwinst.
 - functioneert niet onmiddellijk na een APPEND-procedure. Het bestand moet eerst (met USE) gesloten en heropend worden.
- STORE - plaatsen we een combinatie van twee variabelen in een derde, en is de omvang van die combinatie groter dan 254 tekens, dan zien we geen foutmelding. Commando's waarin deze variabele een rol speelt hebben een foutief resultaat. Rechtgezet met ingang van versie 2.41.
- SUM - accepteert de naam van een variabele niet als die precies 10 tekens omvat. Rechtgezet met ingang van versie 2.4.
- TOTAL - functioneert niet goed als de term FIELDS in de opdracht voorkomt. Rechtgezet met ingang van versie 2.4.
- functioneert niet goed als met FOR een voorwaarde gesteld is waar niet aan voldaan kan worden. Rechtgezet met ingang van versie 2.4.
 - functioneert niet goed als de twee N-rubrieken in de bestanden van herkomst en van bestemming een verschillende breedte hebben. Rechtgezet met ingang van versie 2.41.
 - als het bestand van bestemming niet bestaat en de N-rubriek in het bestand van herkomst niet breed genoeg is voor de resulterende telling, dan klopt de uitkomst niet.
 - reageert niet op het indrukken van de ESC-toets. Rechtgezet met ingang van versie 2.43*.
- USE - sluit een .NDX-bestand niet correct af. Deels rechtgezet met ingang van versie 2.4, maar bij een DELETE FILE-commando gaat dBASE II niet - zoals voordien - na of het bestand wel afgesloten is.

Deze opsomming is vermoedelijk nog niet volledig. Wie er nadere informatie aan toe kan voegen nodig ik vriendelijk uit met mij contact op te nemen. Adres: Ruysdaelstraat 68, 1071 XG Amsterdam.

Leserbrief: Herbert Oppmann, 8520

1. Auf gewissen Seiten ändert sich kaum etwas. Vielleicht könnte man die ja, wie die Namensliste und das Inhaltsverzeichnis, nur alle 2 bis 3 mal in das Info nehmen. Ich denke da an die Angebotsseiten (Hard- und Software) sowie an die Seite 4.

Anm.d.HzN.: Ja & Nein! (Dies mal hat sich etwas geändert!)

In diesem Zusammenhang würde mich auch mal interessieren, was die so lange angekündigte HD64180 Subprozessorkarte tut. Bauen werde ich mir das Ding sowieso nicht, da ich ja bereits durch den c't180 einen eigenständigen HD64180-Rechner habe, der den MTX als Subprozessor benutzt (haha ...). Für so niedrige Tätigkeiten wie Tastatureingabe und Bildschirmausgabe ...

Nochwas: ist eine Prozessor-Austausch-Karte mit einem Z280 geplant? Und wo kriegt man im Moment in Deutschland einen Z280 und zu welchem Preis? In den USA soll er für 40\$ zu haben sein. Gegen den Z280 hat sogar der HD64180 keine Chance und wird ihm langfristig wohl den Platz räumen müssen.

Anm.d.HzN.: Planung: Zweitprozessorkarte mit Z280, ≥ 1 MB auf ECB.

2. Nix wie Ärger mit den Nicht-Panasonic-Farbbändern! Nachdem das mitgelieferte Panasonic-Farbband ca. 2500 Blatt (mit Wiederauffüllen natürlich) schaffte, stammte das nächste Farbband aus einer Club-Sammelbestellung. Nach 200 (!) Seiten war es am Ende!!! Wiederauffüllen ging nicht, da die Farbe in einer Kartusche statt in einem Schwämmchen war und außerdem sich die Farbe mit meiner Nachtränkefarbe (Pelikan: Stempelfarbe ohne Öl) nicht vertrug (ließ sich nicht benetzen, Tropfen blieb darauf stehen). Da ich dringend was drucken musste, blieb mir nur die Notschlachtung übrig: aus der Kassette des Panasonic-Farbbands das Band heraus und weggeschmissen (es war zerschissen) und das Band aus diesem Versager implantiert. Damit konnte ich wieder drucken. Dann habe ich mir wieder ein Farbband besorgt (wieder eines von einer Club-Bestellung). Das sah auch ganz professionell aus: Schwämmchen drin, Blechstreifen um den An- druck zu erhöhen wenn die Druckstärke nachläßt, ... ABER: wieder nach 200 (!) Seiten: weiße Flecke im Ausdruck! Eine Untersuchung des Farbbandes zeigte: das Band ist nicht endlos gewebt, sondern an einer Stelle zusammengeschweißt. Und genau da klappt jetzt ein Loch. Also drucke ich wieder mit meiner zusammengebastelten Farb- bandkassette und weiß eines: das nächste ist ein Panasonic!!

Anm. d.HzN.: Bei schaffte übrigens eines der Farbbänder aus der letzten Sammelbestellung (war von INMAC) kaum 40 Seiten As- ssembler-Listing! Auch die von GeHa sind nicht gerade das Non- PlusUltra - aber deutlich besser als die INMAC-NoName!

3. Du fragst im Info nach einem "Turbo-Pascal-Listing". Damit meinst du wohl Turbo Pascal (TURBO.COM, TURBO.OVR) disassembliert. Ich ha- be dazu einen (zugegebenerweise vagen) Hinweis: In einer alten c't (muss wohl Jahrgang 85 oder 86 gewesen sein) las ich mal eine klei- ne Anzeige von jemandem (eine Privatperson), der das Overlay disas- sembliert hat und es als Listing auf Papier anbietet. Ich glaube er wollte DM 30,- dafür. Ich hab damals an ihn geschrieben aber keine Antwort bekommen. Ich weiß nicht mehr, ob die Anzeige in den Leser- briefen, den Clubnachrichten oder den Kleinanzeigen stand (Hotline gab es damals noch nicht). Ich konnte es dir leider nicht raussu- chen, da ich in meinen alten c'ts damals Werbung und Kleinanzeigen herausriß und wegwarf (aus heutiger Sicht: leider!).

Anm.d.HzN.: Das TURBO.OVR ist fast banal und recht uninteressant.

Leserbrief: Herbert Oppmann, B520

4. Viele Hardware-Tips hast du im neuen Info zusammengetragen, auch ich habe einen. Den kenne ich schon länger, habe ihn aber bei den letzten Briefen vergessen zu erwähnen. Vielleicht erinnerst du dich daran, daß ich mal Probleme mit der Hardware-Uhr an Port 7 hatte. Der Rechner wollte damals mit eingesteckter Uhr nicht booten, wenn er warm war. Ich hatte mich damit schon fast abgefunden, als folgendes auftrat: nach dem Umbau der Stromversorgung (alle Netzteile, Lüfter, Travos, -> RAUS. Dafür ein Schaltnetzteil von Völkner) wurde es in der FDX wärmer als bisher (ist aber nicht bedenklich). Daraufhin baute die 80 Zeichenkarte Mist: die Zeichen wurden immer "dünner", d.h. nur wenn mehrere Punkte nebeneinander lagen, sah man sie. Senkrechte Striche, die nur aus einer Punktreihe bestehen, verschwanden dagegen. Ich öffnete die FDX, steckte die 80 Zeichenkarte ganz nach oben, legte ein Handtuch zwecks Aufheizung drauf, schaltete den Rechner ein und wartete. Nach Auftreten der Störungen machte ich mich dann mit Kältespray auf die Suche. Zunächst sah es so aus, als ob der 74LS266 (das Schieberegister) der Täter sei. Aber Auslöten, Sockel und Ersatz-IC rein brachten keine Besserung. Dann deutete alles auf den Taktgenerator 74S04 hin. Auslöten, Sockel rein, anderen 74S04 rein, und: nix geht mehr! Andere IC's (74HC04, 74ALS04 ...) tuns auch nicht. Fummel, fummel,.. da plötzlich: es geht! Und wieder nicht. WO HAT MICHAEL GERADE DIE FINGER GEHABT? Eilige Untersuchungen zeigen: Finger auf Pin 11 des 74S04 -> geht. Da ich Michael nicht dazu überreden konnte, seinen Finger längere Zeit zur Verfügung zu stellen, wurde mit verschiedenen Kondensatoren und Widerständen gespielt, bis feststand: ein Widerstand 1 KOhm gegen Masse tut es (dadurch wird der Arbeitspunkt des "Verstärkers" 74S04 richtig eingestellt). Nach ca. 1 Woche klingelte es bei mir und ich erkannte, daß genau das selbe auch im Grundgerät passiert. 1 KOhm von Pin 3 gegen Masse und - ich kann problemlos mit eingesteckter Uhr booten, auch wenn der Rechner warm ist. Der Taktgenerator ist nämlich damals einfach nicht losgelaufen.
5. Es ist mir manchmal passiert, daß ein Programm in die Binsen geht, und zwar so, daß der Warmstartvektor und Teile des BIOS überschrieben wurden, KLICK sich aber noch meldet. Nun kann ich leider keinen WarmBoot auslösen, sonst ist der Rechner ganz abgestürzt. Natürlich kann ich Reset geben, aber was mache ich, wenn ich den Inhalt der RamDisc noch brauche, aber meine RamDisc nicht Reset-fest ist? Für diesen Fall habe ich folgenden Soft-Reset entwickelt, der sich jederzeit in KLICK eingeben läßt: (/ steht für Return-Taste)

```

<Shift>+<Esc>  Klick kommt
M               in den Monitor
SO/            Warmstartadresse neu setzen
C3/00/80/./    JP B000H (Adresse ist egal, muß über 4000H liegen)

SB000/         jetzt dort eine Routine hinsetzen
F3/           DI
3E/40/        LD  A,40H      ;Boot-EPR0M einblenden
D3/00/        OUT (0),A
21/00/21/     LD  HL,2100H   ;Inhalt in den Speicher kopieren
11/00/EC/     LD  DE,0EC00H
01/00/20/     LD  BC,2000H
ED/80/        LDIR
3E/80/        LD  A,80H     ;wieder auf RAM-Betrieb
D3/00/        OUT (0),A
C3/B8/EC/./   JP  0ECB8H    ;Booten ohne Speichertest

```

Leserbrief: Herbert Oppmann, 8520

Q Monitor verlassen
 W/ WarmBoot auslösen (springt nach 0000H)

Die Routine funktioniert natürlich nur, wenn am Boot-EEPROM nicht allzuviel verändert wurde.

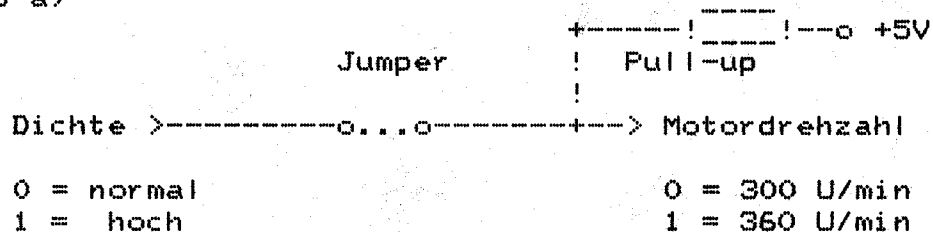
6. Siehe: neue c't, Seite 162 "Randvolle Scheiben"
Anm.d.HzN.: Ich habe einen Leserbrief an c't geschickt, in dem ich andeutete, daß dem Titel das Fragezeichen fehlen würde!

Bevor ich oben angegebenen Artikel gelesen habe, wußte ich nicht, wie IBMs PS/2-Geräte 1.44 MByte auf eine Diskette bringen. Dabei ist alles ganz einfach: man nehme HD-Disketten (die sind mit Chromdioxid- statt Eisenoxid-Partikeln beschichtet) und ein HD-Laufwerk (das die für CrO2 erforderliche höhere Magnetisierung zustande bringt). Nun schreibt man wie bei 8"-Laufwerken mit 500 KBit/sec, also der doppelte Geschwindigkeit. Das beschleunigt das Lesen und Schreiben und erhöht die Dichte, also die Kapazität der Diskette. Bisherige HD-Laufwerke schalten nun bei High-Density die Motordrehzahl von 300 U/min hoch auf 360 U/min, wie das bei 8"-Laufwerken üblich ist. Der Trick beim neuen PS/2-Format ist nun, die Motordrehzahl nicht zu erhöhen, und damit genau die doppelte Dichte wie bei Normalformaten zu haben.

Soweit sogut, aber weshalb braucht man dann dazu spezielle Laufwerke? HD-Laufwerke können doch bereits alles: normale und hohe Dichte, sowie 300 und 360 Umdrehungen pro Minute! Schwierigkeiten ergeben sich nur dabei, dem Laufwerk zu verklickern, daß es hohe Dichte, aber niedrige Motordrehzahl nehmen soll. Das wurde bei HD-Laufwerken nicht vorgesehen, also ist auch kein Jumper dafür da. Mit einem kleinen Trick, der auch in dem oben angegebenen Artikel verraten wird, gehts aber doch!

Man muß hierbei 2 Typen unterscheiden:

Typ a)



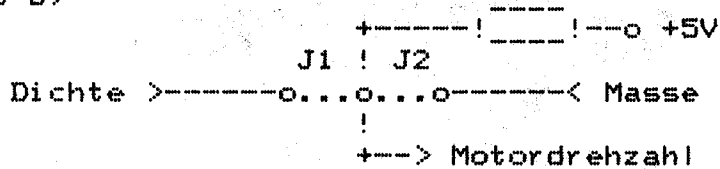
In diesem Fall bedeutet
 Jumper gesteckt: Motordrehzahl wird mit der Dichte umgeschaltet und
 Jumper offen: Motordrehzahl ständig hoch (für IBM AT)

Hier muß man den Jumper weglassen und einen Schalter zwischen den Pin für die Motordrehzahl und Masse (ev. über einen Widerstand) legen. Mit diesem Schalter läßt sich nun die Motordrehzahl unabhängig von der Dichte wählen:

Schalter zu: Motordrehzahl 300 U/min
 Schalter offen: Motordrehzahl 360 U/min

Leserbrief: Herbert Oppmann, 8520

Typ b)



0 = hoch 0 = 360 U/min
1 = niedrig 1 = 300 U/min

In diesem Fall bedeutet

J1 gesteckt: Motordrehzahl wird mit der Dichte umgeschaltet und
J2 gesteckt: Motordrehzahl ständig hoch (für IBM AT)

Hier muß man beide Jumper weglassen und zwischen dem mittleren und
(in dieser Zeichnung) rechten Pin einen Schalter legen, mit dem
sich nun die Drehzahl unabhängig von der Dichte wählen läßt:

Schalter zu: Motordrehzahl 360 U/min

Schalter offen: Motordrehzahl 300 U/min

Anm.d.HzN.: Für die neuen TEAC FD55 GFR ist die Lösung noch in Ar-
beit!

Soweit zur Hardware. War doch gar nicht schwer, oder? Jetzt zur
Software. Da es für den Memotech noch kein Standardformat für 2
MByte unformatiert gibt, mußte ich eines erfinden. Wenn es gefällt,
kann man das ja zum neuen Standard erheben. Ich hab den noch freien
Config-Code 1C genommen. Mit FORMATE4 muß jetzt das Format bekannt
gemacht werden:

FORMATE4/

```

E                    Formate edieren
1C/                  Config-Code
j                    Ja, ich möchte ändern
MaxHD 8" 80T ohne Sys/    Name
88/                  Sektoren pro Spur
2/                   Blockgröße
880/                 Anzahl Blöcke
0/                   EXM
384/                 Anzahl Dir-Einträge
FC/                  ALO$
0/                   AL1$
96/                  CKS
0/                   Offset
1024/                Reclen
1/                   First
80/                  Track-Count
2/                   Seiten
2/                   Dichte (Density)
5/                   Zoll
0/                   VR/RR
0/                   invertiert
0/                   Tracktyp
1/                   Skew
11/                  phys. Sektoren pro Spur
n                    nichts ändern
FF/                  raus
  
```

Leserbrief: Herbert Oppmann, 8520

B	beenden
J	abspeichern
/	alter Name
C	Configs einbauen
E	einbauen
1C	den neuen
Q	raus
Q	raus

So, das war die erste Hälfte. Wir können damit zwar solch ein Format lesen und schreiben, aber noch nicht formatieren. Deshalb noch folgende Patches an FORM4:

ZSID FORM4.COM/	oder DDT oder DDTZ
S4FA/	
1C../	Format 10 fliegt raus
S4FE/	
64../	Interleave-Tabelle für Rückseite
M5F6 607 5F5/	Format 10 fliegt raus
S607/	Format 1C hat selben Sektoraufbau wie
1C../	z.B. die Formate 1A und 1B
S558/	nun die beiden Interleave-Tabellen
1/5/9/2/6/0A/3/7/0B/4/8/0/	
0A/3/7/0B/4/8/1/5/9/2/6/0../	
^C	Debugger verlassen
SAVE 8 FORM4X.COM/	Neues Programm abspeichern

Auf zum Test:

Schalter einstellen auf hohe Schreibdichte, niedrige Motordrehzahl.
Dann: (d ist durch den Bezeichner des HD-Laufwerks zu ersetzen)

CFIG4 d:1C/	Laufwerk auf das neue Format stellen
FORM4X d:/	HD-Diskette rein und los gehts!
/	
RCHECK4 d:/	brauchte nicht gepatcht zu werden. Wir haben
	so viele Sektoren pro Spur, daß RCHECK4 mit
	der Bildschirmausgabe durcheinander kommt!

Anm.d.HzN.: Wenn's tut, sollte auch noch mit INST42 das neue Format dauerhaft in RAM4 eingebaut werden, und ggf. mit INSTCFIG auch CFIG4 ...

Fertig! Viel Spaß! Herbert.

Anm.d.HzN.: Format 1C:

```
A>STAT D:DSK:
  D: Drive Characteristics
 14080: 128 Byte Record Capacity
  1760: Kilobyte Drive Capacity
  384: 32 Byte Directory Entries
  384: Checked Directory Entries
  128: Records/ Extent
   16: Records/ Block
   88: Sectors/ Track
    0: Reserved Tracks
```

Leserbrief: Jan Bredereke, 2000

Korrektur

INFO 25 - 31 Hardware-Reparatur (Jan Bredereke, 2000)

Uwe hat meine Änderungen, um meinen ECB-Bus zum Laufen zu bringen, nicht vollständig beschrieben. Ich habe nicht nur bei A8 bis A15 die Buserminierung entfernt, sondern auch bei A0 bis A2. Nötig waren A0 bis A2 und A13 bis A15, damit der Rechner lief. Zur Sicherheit habe ich dann auch noch A8 bis A12 abgetrennt, da diese von der CMOS-Floppy nicht benötigt werden.

Antwort zur Frage von Peter Würfel nach der Uhrumschaltung in Info 25-10
(Jan Bredereke, 2000)

Peter wollte die Uhr unter Klix mit dem <ESC>AS-Befehl von der Kommandozeile aus ein- und ausschalten. Das gleiche habe ich auch einmal versucht, und zwar auf Funktionstasten gelegt.

Dabei ergibt sich das Problem, daß der Befehl Leerzeichen enthält. Und der Kommandozeilenprozessor druckt nur die Buchstaben bis vor das erste Leerzeichen, weil er diese als den Namen des nicht vorhandenen Programms ansieht, den er mit einem Fragezeichen kommentiert. Die "Parameter" will er nicht mitdrucken.

So geht es also nicht. Ich habe mir daher das Miniprogramm ECHO2 geschrieben, das solange Zeichen von der Tastatur liest und ausdruckt, bis es ein , also 07Fh, findet. Ich habe auf einer Funktionstaste z.B.

```
! :echo2<RET><ESC>ASh .(<DEL>
```

Dies schaltet die Uhr ein. Auf einer anderen Taste ist die öffnende Klammer durch das Leerzeichen ersetzt, diese Taste schaltet die Uhr aus.

Hier nun das Programm:

```
CONIOF = 6
INPUT  = 0FFH
BDOS   = 0005H
WBOOT  = 0000H
CHRDEL = 07FH
```

```
START: LD      C,CONIOF
        LD      E,INPUT
        CALL   BDOS
        CP      CHRDEL
        JR     Z,ENDE
        LD      C,CONIOF
        LD      E,A
        CALL   BDOS
        JR     START
ENDE:  JP      WBOOT
```

Diese Version ist allerdings nur für Funktionstasten geeignet, da "keine Taste" als ^\$ interpretiert wird. Dafür darf die Funktionstaste auch ^\$ enthalten. Ist die Statusabfrage bei der benutzten Funktionstastentabelle abgeschaltet, wird der Befehl allerdings erst ausgeführt, wenn man eine beliebige weitere Taste nach der Funktionstaste drückt.

Das Programm heißt übrigens ECHO2, da es schon ein Programm ECHO bei den P2DOS-Programmen gibt. Dieses druckt seine Parameter einfach auf den Schirm.

Leserbrief: Anton Reiser, 8977

Zuerst ein ganz dickes Lob an alle Info-Macher. Ohne Infos wäre meine schwarze Kiste schon längst auf dem schrott gelandet. Und seit RAM4.2 läuft alles noch besser und schöner. Seit kurzem ist die Hauptplatine auch noch ins FDX-Gehäuse gewandert, ähnlich der Beschreibung von Michael Kessler aus Info 21-33, mit Stromversorgung vom FDX-Netzteil (ohne Nachregelung, bisher keine Probleme). Dazu gleich eine Frage: Uwe Grass schreibt in Info 18-37 vage: "Jetzt werden alle Teile, die der Spannungsversorgung dienen, aus dem Grundgerät entfernt." Welche sind das denn konkret? Bei manchen Kondensatoren und Dioden bin ich mir nicht sicher.

Was jetzt noch fehlt, ist eine Festplatte. Stichwort: DMTI-Controller an fremden Bussen, c't 4/87 und 9/87, jeweils Seite 138. Wer hat dazu eine Meinung, oder gar Erfahrung darin, dieses Ding an den FDX-Bus anzuschließen? Ich meine ca. 1000 DM für 20 MB Festplatte sind besser als ca. 500 DM für 1 MB RAM-Floppy. Zumal nach dem Booten das Laden der Programme entfällt. Diese Harddisk mit Controller könnte vielleicht auch noch irgendwann an einem MS-DOS-Rechner verwendet werden. Übrigens, MS.COM hat noch ein paar Macken: Leere Sub-Directories können nicht mehr gelöscht werden, und nachdem im Root-Directory Dateien gelöscht wurden, passen Sub-Directories nicht mehr in den Puffer. Wünschenswert wäre auch eine Option, um die Umlaute von ASCII nach IBM-Code zu konvertieren. Zumal man MS.COM meist nur für Text-Files sinnvoll einsetzen kann.

Noch ein paar formale Verbesserungsvorschläge fürs Info: Bringt doch beim Erscheinen einer Club-PD-Diskette eine ausführliche Beschreibung vom Inhalt. Der lange Vorspann im Info würde 1/2-jährlich reichen, wie die Mitgliederlisten und das Register; also im neuen Info jedesmal nur die Updates und Änderungen. Der Druck des Infos könnte auch wieder besser werden - ich erhielt teilweise Seiten, auf denen der Druck nur hellgrau statt satt schwarz war.

Zum Schluß ein NewWord-Tip für 3-spaltige Listen: Erste Spalte eine Ordnungsnummer, zweite Spalte ein mehrzeiliger Text, dann auf Höhe der letzten Zeile in der dritten Spalte ein Betrag. Meine Lösung:

Mit .RR L-----! die zweite Spalte definieren, ganz links ^OX Rand lösen, Ordnungsnummer schreiben, ^P<RET>, dann der Fließtext, ^P<RET>, <TAB>, ein paar Blanks, um die Kommas untereinander zu plazieren, den Betrag und zum Schluß <RET>.

Dann nächster Absatz. Auf dem Bildschirm sieht das so aus:

```

_____BSP.LST_____P01_L01_C01_Einfüg_____----00:06:21
L-----!
.RR -----!
1.2.
    Das ist ein Beispiel für einen
    mehrzeiligen Fließtext, der in
    die Spalte Zwei kommen soll. Und
    zum Schluß ein Betrag:
                                     1.234,56

```

Und auf dem Drucker so:

```

1.2. Das ist ein Beispiel für einen
    mehrzeiligen Fließtext, der in
    die Spalte Zwei kommen soll. Und
    zum Schluß ein Betrag:
                                     1.234,56

```

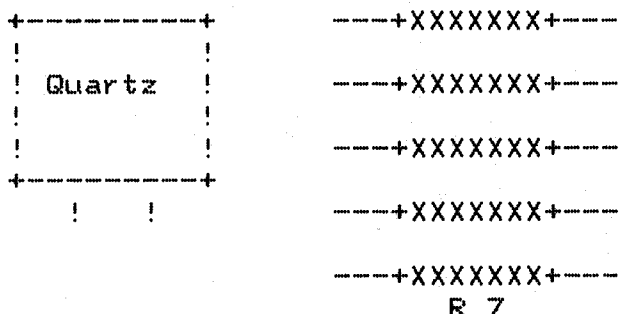
Nachteile: Zum Betrag eingeben kann der Dezimal-Tabulator nicht verwendet werden, und den kompletten Text überrechnen lassen ist auch nicht möglich, da sonst die erste Spalte auf die zweite eingerückt wird (also mit Cursor auf Fließtextanfang und nur Absatz rechnen).

Hardware: Diverses

SDX-80-Zeichen-Kontrast

(Herbert zur Nedden, 2000)

Auf der Karte SDX ist ein Widerstand namens R7, der einen Teil eines Spannungsteilers bildet, durch den das Monitor-Signal läuft. Wird dieser Widerstand entfernt, ist das Bild richtig knorke!



Automatische Umschaltung für Laufwerke

(Michael Kessler, 5600)

Folgendes ist für TEAC FD 55 GFR realisiert worden: HEAD-LOAD (Pin 33 am 5"-Stecker des Floppycontrollers) wird von Masse abgetrennt und statt dessen an Pin 12 des IC 5A (74 LS 237) des Floppy-Controllers angeschlossen. Da der kleine Transistor unterhalb von Pin 33 jetzt keine Masse mehr hat, muß diese ebenfalls neu gelegt werden. Das sind auch schon alle Änderungen. Nur falls TEAC FD 55 GFR: statt auf I auf IS jumpern.

FDX-Floppy-Controller aufrüsten

(Michael Kessler, 5600)

Wer die beiden fehlenden WD-Chips 1691 und 2143 auf den Controller setzen kann nicht schaden. Obendrein kann die sog. Precompensation mit dem Poti eingestellt werden, was schon bei minimaler Einstellung bei DD-Formaten die Datensicherheit erhöhen kann.

TIP 2955 im MTX defekt ?

(Helge Siegloff, 2370)

Digital-Multimeter Minus-Pol an Pin 16 eines Video-RAMs, Plus-Pol an Pin 9 eines solchen IC's. Werden hier zwischen 4.75 und 5.25 Volt gemessen, so ist alles o.k.

Sind hier nur ca. 3 Volt, und ist die Spannung am Widerstand R62 größer als 1 Volt, so ist der TIP 2955 defekt. Sofort ausschalten, denn nach ca. 30 Sekunden konnte ich diesen Widerstand herausziehen. Das lag an der Überhitzung, da die zu verbratene Leistung auf das 10-fache Anstieg.

Beim Ersetzen des TIP darauf achten, daß die isolierende Glimmerscheibe und Plastik-Scheiben (zwei!) so sitzen, daß das Gehäuse des TIP nicht mit dem Kühlblech in Kontakt steht!

Übrigens muß am Pin 8 der Video-RAMs ca. 12 Volt, an Pin 1 -5 Volt anliegen!

Floppy-Controller unter 8 MHz

(Herbert zur Nedden, 2000)

Horst Kupka hat im Info 25, Seite 23 eine pffiffige Schaltung veröffentlicht, mittels derer der Zugriff auf den Floppy-Controller durch WAITs an die 8 MHz angepaßt wird. Die Dioden, die am Pin 6 des Binärdekoders 74 LS 138 hängen sind nicht gerade das Optimum der Digital-elektronik.

Die Pins 1-6 des 74 LS 138 können wie folgt beschaltet werden:

Pin des LS 138	1	2	3	4 & 5	6
Verbinden mit	IC 7A, Pin 8 (LS27)	A 3	A 5	A 7	A 6

Allerdings muß nun statt des Pins 15 der Pin 14 ausgangseitig genommen werden.

(Bemerkung: IC 7A, Pin 8 ist high, falls IOREQ, M1 und A4 low sind).

Hardware: VS4-Probleme / Port 7

Probleme mit VS 4

(Holger Fischer, CH-4303)

Mein Bildschirm hat seit einiger Zeit Pocken! Das drückt sich so aus, indem auf dem ganzen VS 4 willkürliche Zeichen entstehen (ASCII-Zeichen und undefinierte Zeichen, manchmal auch ein Punktraster über das gesamte Bildschirmfeld). Die normale Schrift wird dadurch fast unleserlich. Möglicherweise stammt dieser Fehler aus einigen Überfütterungen des Ports 7 (INSTB) mit kurzfristigen Spannungsspitzen von ca. 9 Volt. (MTX + 32k + Pascal-Karte)

Anm.d.HzN.: Zumindest ist mir bekannt, das ein MTX/FDX dank unpfleglicher Behandlung des Port 7 den Betrieb nicht so recht aufnehmen wollte - was durch Austausch der beiden Port-Bausteine (74 LS 373 bzw. 374) rechts und links neben dem Port behoben werden konnte. Außerdem sind die Video-RAMs sehr empfindlich gegen fehlende -5 Volt und ähnliche Spirenzchen! Vielleicht liegt's daran!

Port 7

(Herbert zur Nedden, 2000)

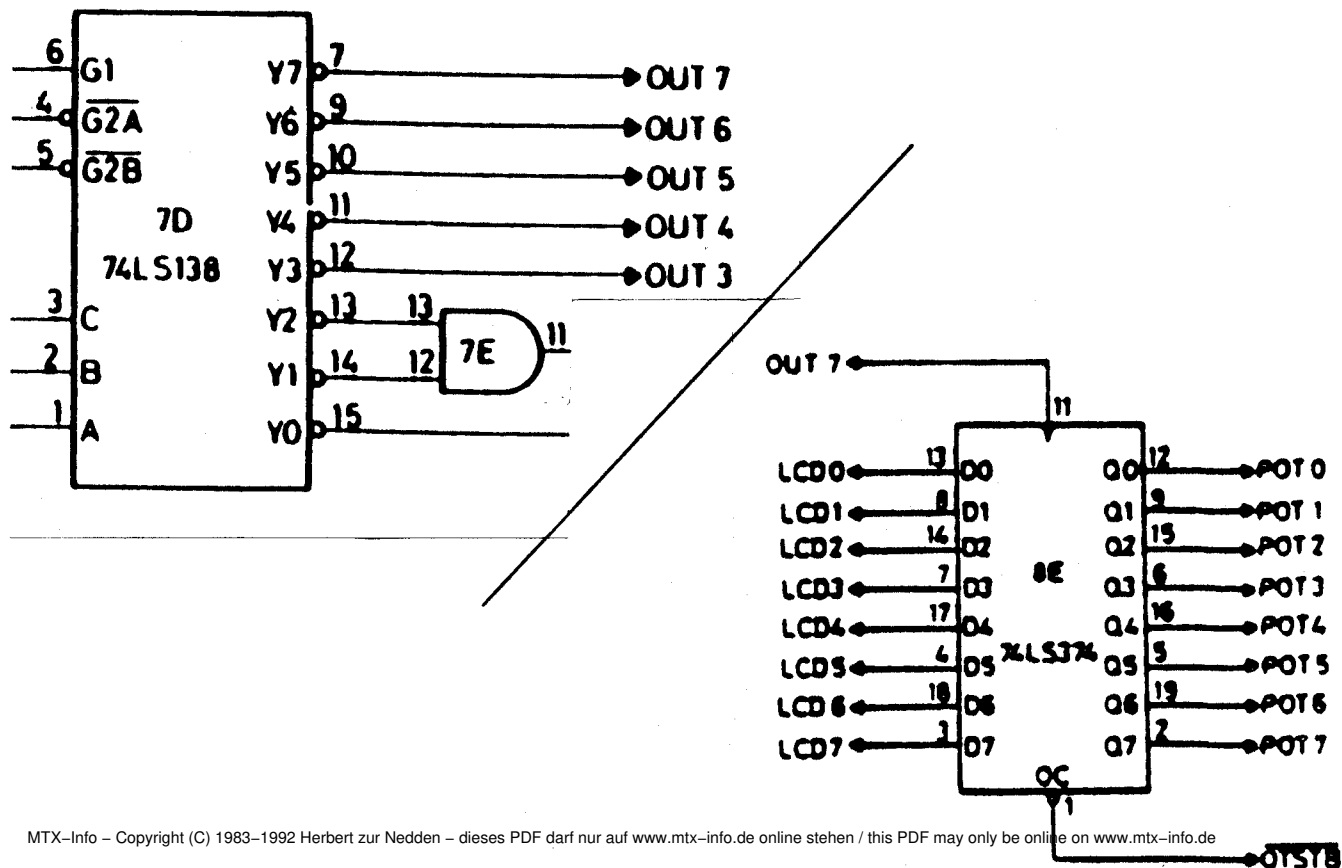
Immer wieder möchte jemand Port 7 trotz Hardware-Uhr nutzen können. Hier sind nun zwei Lösungen:

1. (Herbert zur Nedden, 2000)

Wer keinen Kassettenbetrieb benötigt, kann zumindest OUT (3) nutzen. Dazu muß nur ein 74 LS 374 so wie für den echten Port 7 verdrahtet werden, d.h. am besten huckepack auf den von Port 7 (IC 8E), wobei folgende Pins hochgebogen werden: Pin's 12,9,15,6,16,5, 19,2 und 1. Letzterer wird mit Pin 12 des IC 7D (74 LS 138) verbunden, die anderen ergeben die vom Port 7 her bekannten Signale OTSTB und POTO-7.

Wer partout will, kann auch noch mit ähnlicher Mimik In (4) verwenden, wobei hier nur Bit 4-7 genommen werden dürfen, da Bit 0-3 für den Drucker-Status verbraucht sind!

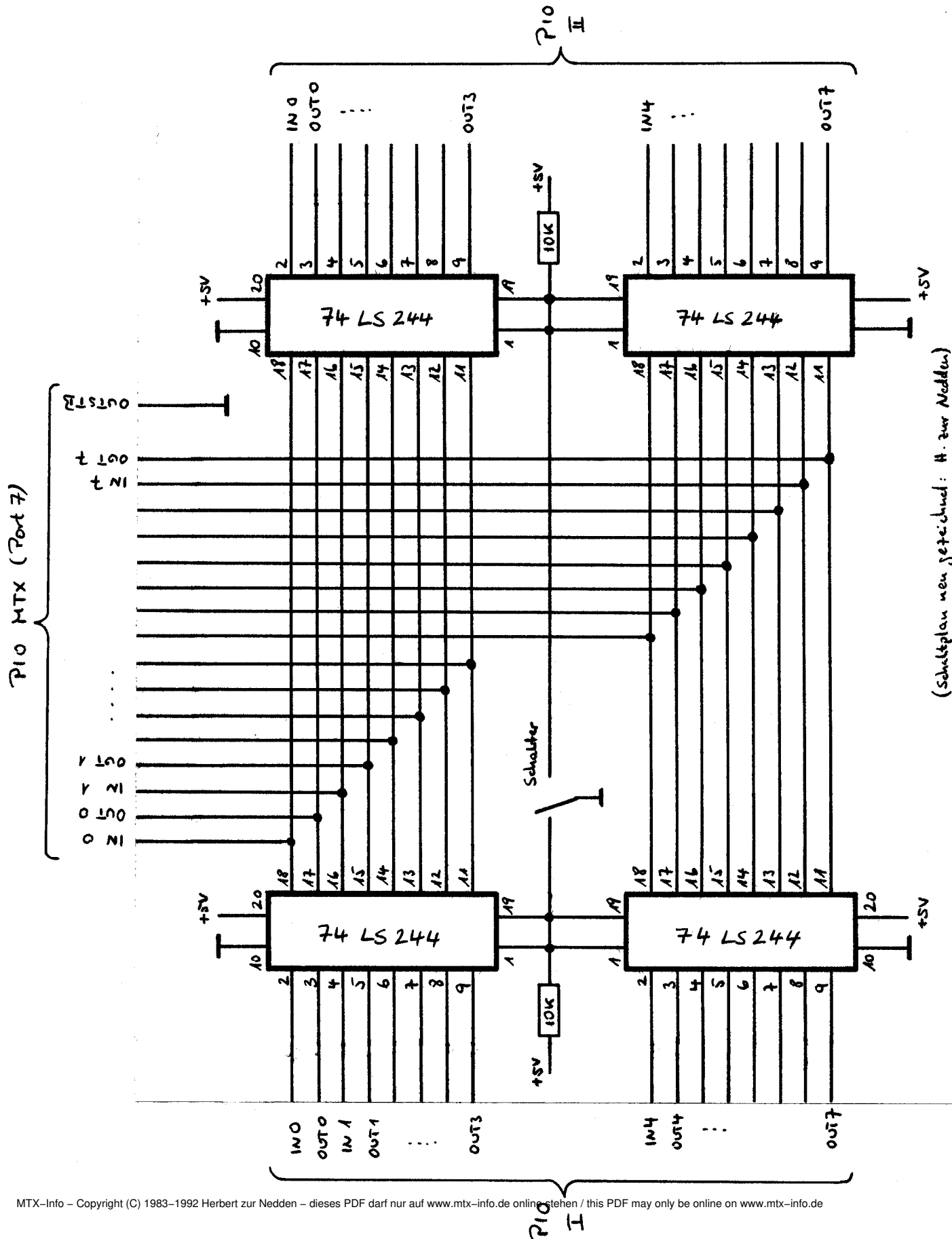
Zur Anschauung mögen diese Ausschnitte des MTX-Handbuchs dienen:



Hardware: Port 7

2. (Erik d'Hondt, B-9440)

Um den vollen Funktionsumfang zu erreichen kann man sich mit einigen IC's und einem Schalter behelfen.



O C R: Klarschriftleser

(Rainer Zangs, B000)

Klarschriftleser OMNI-READER

Der OMNI-READER (OR) von Oberon International ist ein Klarschriftleser (Preis 395,- DM), der verschiedene gängige Schrifttypen wie Courier 10/12, Gothic 12 und Elite 12 vom Blatt erkennt und über eine RS232-Schnittstelle überträgt. Die Erkennung und ASCII-Umsetzung der Buchstaben findet im OR statt, sodaß der angeschlossene Computer im Prinzip nur ein Modem-Programm benötigt.

Für den MTX existiert da eine angepaßte Version von MDM712, nämlich M1 (auf einer der Club-Disketten). Damit habe ich den OR getestet und verschiedene Texte in den MTX übertragen.

Der niedrige Preis wird dadurch möglich, daß die Abtastbewegung des OR vom Benutzer ausgeführt wird. Man muß den Lesekopf des OR zeilenweise über eine eingespannte Seite schieben (ca. 1s je Zeile) und den Zeilenvorschub durch Drücken einer Taste auf dem Lesekopf signalisieren. Der Lesekopf wird auf einer Schiene mit einem Fenster geführt, durch das man die abzutastende Zeile sieht. Ob das Abtasten erfolgreich war, wird durch einen Ton angezeigt.

Die Qualität der Vorlage ist natürlich entscheidend für die Erfolgsquote. Eine mit Schreibmaschine geschriebene Seite macht keine Schwierigkeiten, eine gute Kopie auch nicht. Das Schriftbild, das Matrixdrucker auf's Papier bringen, ist normalerweise nicht ausreichend, ausgenommen NLQ-Drucker. Die Grenze die hier gezogen wird, kann man beim OR durch eine Einstellung POORCOPY ein wenig hinausschieben. Die Erkennung wird langsamer (ca. 1s je Zeile), aber sicherer.

Während der Abtastung wird noch nichts zum Computer übertragen. Erst nach dem O.K.-Ton sieht man auf dem Schirm des Computers die Zeichenfolge ankommen. Ich war überrascht, wie zuverlässig der OR bei einer guten Vorlage arbeitet. Zwar kann man bei dem Preis keine Wunder erwarten, aber schneller und genauer geht's per Hand sicher nicht, wenn man nicht gerade im Maschinenschreiben sehr fit ist.

Der OR kann z.B. dann eingesetzt werden, wenn man größere, fremde Textteile in einen eigenen Text übernehmen will, wenn man einen Text mit Hilfe eines Programms übersetzen lassen will und den Text nicht eintippen möchte oder wenn man einen Text zum automatischen Korrekturlesen, zum Erstellen eines Index oder zum erneuten Formatieren in eine Textverarbeitung übernehmen will.

Hier ein Beispiel, das sehr deutlich auch eine weitere Grenze zeigt. Der OR kommt schließlich aus den USA und dort werden bekanntlich ä,ö,ü,Ä,Ö,Ü und ß nicht verwendet.

Eine den juristischen Personen nahekommende rechtliche Selbständigkeit weist, wie bereits oben erwjhnt, die OHG und die KG auf. Jedoch handelt es sich hierbei um Personengesellschaften, der die umfassende Rechtspersgnlichkeit fehlt.

Die Rechtsf&higkeit ausl&ndischer Personen wird geregelt durch Art. 7 des EGBGB, so da& die Rechtsf&higkeit und die Gesch&ftsfjhgigkeit einer Person dem Recht des Staates unterliegen, dem die Person angehrt.

Logik: Karnaugh-Tafeln

Die Rechtsfähigkeit erlangt eine juristische Person durch die gesetzlich bestimmten Voraussetzungen und der Mensch mit der Vollendung der Geburt. Im Laufe seines Lebens kann er in unterschiedlichem Maße am Rechtsleben teilnehmen. Wie bereits gesagt beginnt die Rechtsfähigkeit mit der Vollendung der Geburt. Mit der Vollendung des 7. Lebens-

In Programm-Listings sind ä, ö, ü, Ä, Ö, Ü und ß nicht enthalten, dafür sind sie leider selten in einer der oben genannten Schriftarten abgedruckt. Mit vollständig englisch geschriebenen Texten gibt's natürlich keine Schwierigkeiten.

Für das Einlesen des Textbeispiels bei 1200 Baud brauchte ich 26 Sekunden. Eine ganze Seite Text kann man ohne Probleme in ungefähr 1 Minute einlesen. Das Eintippen eines englischen Textes ist sicherlich schwieriger als das Eintippen eines deutschen Textes und damit auch langsamer. Bei englischen Texten entfällt natürlich auch jede Korrektur der falschen Buchstaben ä, ö, ü, Ä, Ö, Ü und ß.

Karnaugh-Tafeln

(Herbert zur Nedden, 2000)

Als ich den folgenden Artikel von Ulrich Reichart las, überlegte ich, ob es nicht eine Anwendung der Sache gäbe, und ward fündig!

Ein PAL ist intern so aufgebaut, wie die Schaltungen, die Ulrich in seinem Artikel herausbekommt: Alle Eingangssignale eines PAL's stehen im Original und invertiert zur Verfügung. Diese Signale gehen an die Eingänge einiger entsprechend breiter Und-Gatter. Die Ausgänge dieser Und-Gatter werden gruppiert, und dann über Oder-Gatter auf die Ausgänge gegeben.

Das PAL 14L4, welches im Memotech verschiedentlich zu finden ist, hat z.B. 14 Eingänge. Im PAL stehen diese 14 Signale zuzüglich deren 14 inversen, also 28 Signale zur Verfügung. Diese werden alle an insgesamt 16 28-fach Und-Gatter geführt. Jeweils 4 dieser Und-Gatter werden auf ein 4-fach Nor-Gatter (d.h. Oder-Gatter mit invertiertem Ausgang) gegeben, und die Ausgänge der Nor-Gatter sind die 4 Ausgangssignale des PAL's.

Das PAL wird dadurch programmiert, daß nun einige der Eingänge der einzelnen Und-Gatter von deren Eingangssignal abgetrennt werden, wodurch die gewünschte Logik-Funktion erreicht wird.

Programm ist auf CLUB.029.

Es ist am Beginn etwas schwierig den Sinn für die Existenz einer schon vom Namen exotisch vorkommenden Sache zu verstehen. Vielleicht werden Sie anschließend immer noch nicht überzeugt sein und darum verstehe ich es voll und ganz wenn Sie der Meinung sind, es sei sinnlos, altmodisch und zu keinem Zweck nutzbringend.

Natürlich muß auch hier eine bestimmtes Grundlagenwissen über digitale Schaltungsverknüpfungen vorhanden sein, sonst müßte ich das Rad wieder neu erfinden. Bestimmt wissen Sie was logische Verknüpfungen sind, also AND, OR, NOT, NAND, NOR, XOR und was es da sonst noch an Junktoren (=Verknüpfungen) gibt. Sollte Sie das alles schon einmal interessiert haben liegen Sie auf dem richtigen Weg sich weiterzubilden. Es dürfte auch bekannt sein, daß diese Grundverknüpfungen untereinander verknüpft werden können und auch aus einer Verknüpfung eine andere gebaut werden kann. Allerdings gibt es eine Einschränkung die da lautet: Eben gesagtes gilt nur für Verknüpfungsglieder Sorte NAND und NOR. Das hat einen bestimmten Grund. Aus einem AND-Glied lässt sich kein NOT-Glied bilden weil es nicht elementar in ihm enthalten ist. Jetzt werden Experten sagen, vorher wollte er nur NOR und NAND zulassen und plötzlich faselt er was von NOT-Glied - ja dann müsste das XNOR auch zugelassen werden. Im Prinzip ist das Richtig, jedoch eignet sich diese Spezialverknüpfung nicht zur Normierung (jenes Verfahren zur Erstellung eines Verknüpfungsnetzes mittels nur einer Art von Verknüpfungsgrundgliedern, welches durch die Verwendung einer Karnaughtafel vereinfacht werden soll). Klingt ja ganz nett kompliziert, und das bringt's? Na ja ich muß zugeben, etwas Arbeit ist damit verbunden, aber ein kleines Programm hilft Ihnen ein bißchen dabei - Sie werden sehen.

Nun aber einmal die Frage wie funktioniert und wie wendet man sowas an? Als erstes muß ich nochmal in die Trickkiste greifen und zwei neue Zauberworte hervorholen (zwecks näheren Erläuterungen). Man kann Verknüpfungsschaltungen auch ganz trocken theoretisch betrachten und von dem gehe ich aus. Man präzisierete das ganze noch in der formalen Aussagenlogik, die Ihnen ja auch nicht ganz fremd sein dürfte als IF-Programmierer. Solche Dinge wie: Wenn Aussage A richtig AND Aussage B richtig Dann ist Aussage C falsch. Hier gerät man leicht ins Philosophieren, da jede Sprache (ausgenommen die der Mathematik) verschiedene Auslegungen und Formulierungen zulässt. Also kehren wir hier um und schreiten nur noch auf unsere Terme, die den Grundverknüpfungen zugeordnet sind zu. Es gibt zum Einen die Minterme, die der Grundverknüpfung AND, zum anderen die Maxterme, die der Grundverknüpfung OR zugeordnet sind. Zur Darstellung dieser Terme vereinbare ich jetzt noch die Schreibweise dieser Formalsprache:

Für AND gilt statt $Q = A \text{ AND } B$: $Q = AB$ (wobei Q das Ergebnis

Für OR gilt statt $Q = A \text{ OR } B$: $Q = A \vee B$ (wobei Q das Ergebnis der Aussage ist)

Für Not gilt statt $Q = \text{NOT } A$: $Q = !A$

Hierzu muß ich anmerken, daß diese Schreibweisen allgemeine Gültigkeiten haben bis auf die der NOT Verknüpfung. Die NOT verknüpften Variablen werden auch als negierte bezeichnet und werden mit einem Querstrich über dem Buchstaben gekennzeichnet (was mit dem Computer nicht ohne weiteres möglich ist).

Um zu unseren MIN und MAX Termen zurückzukehren Betrachten wir im folgenden eine AND-Verknüpfung mit drei Eingängen wobei alle Kombinationen negierter Eingänge durchlaufen werden sollen.

Logik: Karnaugh-Tafeln

A	B	C	ABC	AB!C	A!BC	A!B!C	!ABC	!AB!C	!A!BC	!A!B!C
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

Es sind somit alle möglichen Kombinationen bei 3 Variablen gleich 2^3 - somit 8 Minterme. Allgemein gilt für MIN und MAX Terme:

Bei N Schaltvariablen ist die mögliche Anzahl der Terme 2^N

Wobei jede Schaltvariable einmal normal oder negiert enthalten sein muß. Ein Beispiel:

- !AB!C!D ist ein Minterm bei 4 Schaltvariablen
- A!B ist kein Minterm bei 3 Schaltvariablen
- AB!C!DE ist kein Minterm bei 4 Schaltvariablen (richtig wäre 5)

Eine Besonderheit wird Ihnen vielleicht schon an der Wahrheitstabelle aufgefallen sein. Bei jeweils einer bestimmten Eingangskombination liefert nur jeweils ein MIN-Term das Aussageergebnis 1. Das ist eine sehr wichtige Feststellung, die man sich zunutze macht. Nehmen wir an sie möchten zu einer gegebenen Wahrheitstabelle ein Schaltnetz entwickeln. Wie gehen Sie vor? Nun könnten Sie hergehen und experimentiell solange versuchen bis Ihre Schaltung die gewünschten Werte liefert. Das ist natürlich unkonventionell und zeitraubend zugleich. Besser wäre eine systematisch Möglichkeit. Ich möchte Ihnen das anhand eines Beispiels mittels der Minterme aufzeigen:

Wahrheitstabelle:

A	B	C	Z	Minterme
0	0	0	0	
0	0	1	1	!A!BC
0	1	0	1	!AB!C
0	1	1	0	
1	0	0	1	A!B!C
1	0	1	0	
1	1	0	0	
1	1	1	1	ABC

Sie sehen das in den Zeilen in denen die Ausgangsvariable Z=1 ist die Minterme eingetragen sind. Es ist nun einfach möglich ein Schaltnetz aufzubauen. Die Minterme sind ja nichts anderes als formale Verknüpfungsschaltungen. Man schaltet also diese vier speziellen AND-

Logik: Karnaugh-Tafeln

Verknüpfungen über eine OR-Verknüpfung zusammen und erhält als Ergebnis die gewünschte Schaltung. Formal würde für das Schaltnetz gelten:

$$Z = (!A!BC) \vee (!AB!C) \vee (A!B!C) \vee (ABC)$$

Dieser Ausdruck wird auch als vollständige disjunktive Normalform (abgeleitet von der OR-Funktion die auch als Disjunktion bezeichnet wird) bezeichnet.

Um die zweite Möglichkeit der Maxterme anzusprechen, möchte ich nur kurz darauf eingehen. Im Prinzip handelt es sich um das gleiche Verfahren lediglich unter anderen Vorzeichen. Es werden OR-Verknüpfungen gesucht bei deren Eingangskombinationen der Ausgang den Wert 0 annimmt. Ein Beispiel für Maxterme wären:

$$\begin{array}{ll} !A \vee B \vee !C \vee !D & \text{bei 4 Schaltvariablen} \\ !A \vee !B \vee C & \text{bei 3 Schaltvariablen} \\ A \vee B \vee C \vee D \vee !E & \text{bei 5 Schaltvariablen} \end{array}$$

Bei der Lösung der Wahrheitstabelle werden also alle Maxterme für deren Wert die Ausgangsvariable Z den Wert 0 hat festgehalten und über eine AND-Verknüpfung verbunden. Die vollständige konjunktive Normalform (abgeleitet von der AND-Funktion die auch als Konjunktion bezeichnet wird) für die vorige Wahrheitstabelle würde so aussehen:

$$Z = (A \vee B \vee C)(A \vee !B \vee !C)(!A \vee B \vee !C)(!A \vee !B \vee C)$$

Beide Terme liefern das gleiche Ergebnis sind somit vom Wahrheitswert gesehen identisch, erfordern jedoch unterschiedliche Verknüpfungsglieder. Es genügt wie Sie noch sehen werden, wenn man eine der beiden Formen verwendet.

Einige Nachteile besitzt dieses Verfahren dennoch, es wird für jeden Wert $Z=1$ ein eigenes Schaltglied benötigt, das zudem mit unterschiedlicher Anzahl invertierter Eingänge ausgestattet sein muß. Diesen Nachteil behebt man dadurch, daß alle Signale grundsätzlich normal und invertiert zur Verfügung gestellt werden und die Anzahl der Schaltglieder verringert man durch die Anwendung der Karnaughtafel die ich im folgenden beschreibe.

Das Verfahren von Karnaugh besteht nun darin, die Eingangskombinationen grafisch so aufzuzeichnen, daß alle benachbarten Eingangskombinationen zusammenliegen. Das bedeutet, daß von einer Kombination zur anderen sich nur jeweils 1 Bit verändert. Das beigefügte Programm nimmt eine Wahrheitstabelle mit 16 Werten auf und zeichnet die zugehörige Karnaughtafel. Um damit etwas anfangen zu können erläutere ich kurz das Prinzip. Auf der nächsten Seite sehen Sie eine Karnaughtafel für 16 Werte. Das Programm nimmt Ihnen die Arbeit des Eintragens der Werte in die Tafel ab. Den Nutzen den Sie daraus haben ist etwas Schwieriger, denn Sie müssen jetzt 1,2,4 oder 8 Werte horizontal oder vertikal zusammenfassen. Unten ist ein Beispiel eingezeichnet. Die so zusammengefassten Werte werden wiederum in Minterme umgesetzt. Der Clou der Sache ist, daß Sie jetzt weniger Terme benötigen als vorher bei der Wahrheitstabelle. Die Terme ermitteln Sie mit Hilfe der Vorgegebenen Teilterme. (Übrigens können Sie auch selbst einmal versuchen aus einer Wahrheitstabelle eine Karnaughtafel zu erarbeiten. Sie müssen dazu lediglich den jeweiligen Eingangszustand in der Reihenfolge Spalte, Reihe ablesen und den zugehörigen Ausgangszustand im Kreuzungsfeld eintragen.)

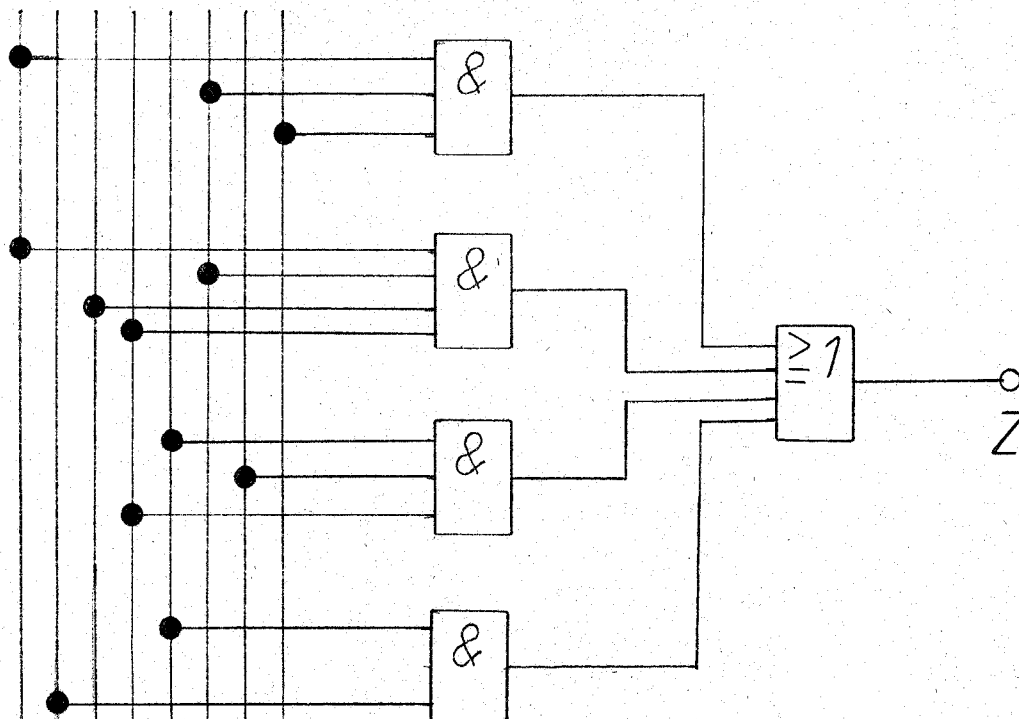
Logik: Karnaugh-Tafeln

	A B	!A!B 0 0	!AB 0 1	AB 1 1	A!B 1 0	
C D	!C!D 0 0	0	1	0	1	
	!CD 0 1	1	1	0	0	
	CD 1 1	0	1	0	1	
	C!D 1 0	0	1	0	1	

Dabei können zusammenfassungen auch über den Rand nach links, rechts, oben, unten gemacht werden. Variablen die dabei einmal normal und einmal negiert vorkommen entfallen dabei. Ein Beispiel: Die Vier 1 Werte der zweiten Spalte ergeben den Minterm !AB, da die Variablen C und D zweimal normal und zweimal invertiert vorkommen können sie entfallen (verknüpft man nämlich ein Variable über ein AND-Glied normal und invertiert ergibt sich für Q=0; !A AND A = 0; da in dieser Form der Karnaughtafel für Minterme 0-Werte nicht berücksichtigt werden müssen dürfen Variable deren Wert 0 ergibt weggelassen werden)

Ein 1-Wert der nicht zusammengefasst werden kann ergibt einen vollständigen Minterm z.B. die 1 in der dritten Reihe, letzte Spalte ergibt den Term A!BCD. Der Unterschied ist sehr deutlich : bei 8 Ausgangswerten Z=1 ergäben sich mit der Wahrheitstabelle 8 Minterme -- so erhalten wir durch zusammenfassen lediglich 4 Terme und somit haben wir 4 Schaltglieder eingespart. Die zugehörige Schaltung zeigt nachstehendes Bild.

$$Z = (A!B!D) \vee (A!BCD) \vee (!A!CD) \vee (!AB)$$



Logik: Karnaugh-Tafeln

Bei der Normierung unserer Schaltverknüpfung kommt ein algebraisches Gesetz der formalen Aussagenlogik zur Anwendung. Es ist das De-Morgansche Gesetz und besagt, daß ein an allen Eingängen negiertes OR-Glied einem ausgangsseitig negierten AND-Glied entspricht und umgekehrt ebenso (also ein an allen Eingängen negiertes AND-Glied einem ausgangsseitig negierten OR-Glied entspricht). Die folgenden Schaltbilder zeigen in Einzelschritten das Verfahren zur Normierung (zum besseren Verständnis). Bild 1 zeigt ein Schaltbild das aus der disjunktiven Normalform hervorgegangen ist. In Bild 2 ist das OR-Glied durch ein an Ausgang und allen Eingängen negiertes AND-Glied ersetzt worden. Wenn wir uns nochmal auf das eben vorgestellte Gesetz besinnen, so stellen wir fest, daß wir ja kein eingangsseitig negiertes OR-Glied hatten - so müssen wir also jetzt, um die gleiche Funktion beizubehalten - zusätzlich auch die Eingänge des AND-Gliedes negieren (daraus folgt, daß man ein OR-Glied durch ein an allen Eingängen und am Ausgang negiertes AND-Glied ersetzen kann - ebenso umgekehrt). Nun ist es einfach aus den (in der 1.Schaltstufe) AND-Gliedern noch zur Normierung benötigte NAND (Sie erinnern sich doch noch ?) umzuwandeln. Es werden einfach die Negierungen, die ja nichts anderes als NOT-Glieder sind, von den Eingängen der 2.Schaltstufe auf die Ausgänge der 1.Schaltstufe vorgezogen. (Sollte bei einer Schaltung ein Eingang direkt am OR-Glied liegen, das wir durch ein NAND ersetzen so muß dieser ebenfalls negiert werden). Das fertig normierte Schaltnetz sehen Sie in Bild 3. Jetzt heißt es nur noch auf in den Elektronik-Shop und 2 Pfund NAND-Glieder eingekauft, Lötkolben startbereit und fröhliches experimentieren.

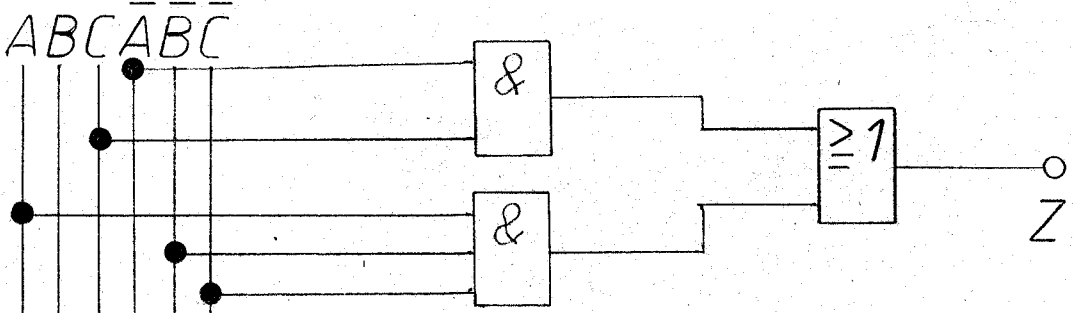


Bild 1

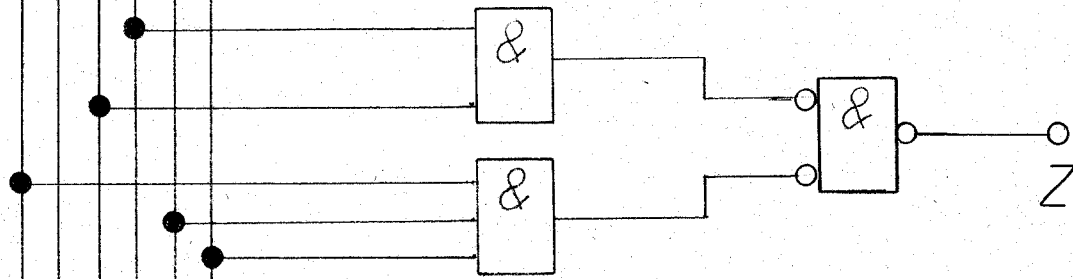


Bild 2

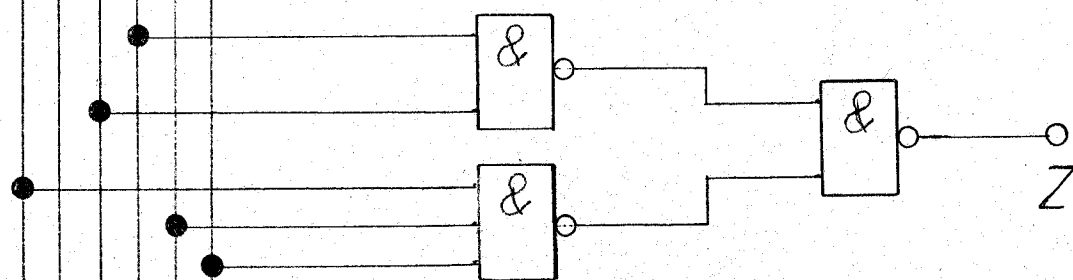


Bild 3

Hardware: Akku für und 512 kB auf c't Statik-RAM-Floppy**Batterie oder Akku für c't-SRAM-Floppy ?** (Hartmut Traber, 5270)

Eine neue Batterie - sogar noch in Folie geschweißt -, die angeblich zehn Jahre halten soll gab schon nach 1 1/2 Jahren klein bei! Die kleinsten Akkus (30 mAh), davon 3 Stück sind hier sicherlich die sicherere Lösung.

Anm.d.HzN.: beim Einsetzen vom Akku muß evtl. noch der Ladewiderstand hinzu.

Erweiterung der c't 256k-CMOS-RAM-Floppy auf 512 kB

(Herbert zur Nedden, 2000)

Da meine Boot-Floppy mit 256 kB zu klein wurde (oh diese vielen KLICK-Overlays) nahm ich mir mal den Schaltplan für die Karte zur Hand. Die Schaltung ist eigentlich recht einfach:

Mit einem Binärdekoder (74 HC 138) werden die einzelnen RAM's selektiert. Als Eingabe für den Dekoder dienen 3 Bit des Spur-Registers (=Track-Registers). Bei einem Zugriff (Schreiben oder Lesen) auf die Karte wird ein Aktivierungs-Signal (Output-Enable) an den Binärdekoder gegeben, damit dieser den an seinem Eingang anliegenden 3-Bit-Wert dekodiert, um den entsprechenden seiner 8 Ausgänge zu aktivieren (d.h. auf low zu legen). Diese 8 Ausgänge sind mit ein Chip-Enable-Eingängen der 8 RAM's auf der Karte verbunden, d.h. dieser Binärdekoder dient zum selektieren des gewünschten RAM's.

Bemerke, daß ein 3-Bit-Wert als Binärzahl interpretiert die Werte von null bis sieben, also acht verschiedene Werte annehmen kann.

Um acht weitere RAM's verkräften zu können ist logischerweise ein weiterer Binärdekoder gut geeignet. Jetzt muß lediglich noch irgendwie gesteuert werden, welcher der nun zwei Binärdekoder jeweils aktiviert werden soll.

Da auch noch erfreulicherweise zwei der Aktivierungseingänge des alten Binärdekoders (er hat sogar drei) miteinander verbunden sind, kann ich also einen dieser beiden für meine Zwecke nehmen, da diese beiden Aktivierungs-Eingänge gleichzeitig auf low liegen müssen, damit sich etwas tut. Bisläng hieß es also,

'Wenn ich Karte es sage, bist DU aktiv'.

Nun lautet die Formel:

'Wenn ich Karte es sage, und Herbert's Zusatzlogik auch der Meinung ist, bist DU aktiv'.

Es muß allerdings sichergestellt werden, daß immer nur einer der beiden Binärdekoder aktiv sein kann. Zusätzlich legt ja die Logik der Karte fest, ob überhaupt einer aktiviert werden soll.

Das ist mit beiden Ausgängen eines Flip-Flop (Q und der dazu inverse Q-quer) leicht machbar. In das Flip-Flop muß nur noch das nächste freie Bit des Spur-Registers der Karte gespeichert werden, und fertig ist die Chose - bis auf die Anpassung der Software!

Da nun 128 an Stelle von 64 Spuren zur Verfügung stehen muß die Treibersoftware (in RAM4 und ggf. im Boot-EPROM) angepaßt werden. Außerdem muß das Format 56 an die 'neue' Karte angepaßt werden. Das Format 56 kann ruhig verändert werden, da die SRAM-Floppy i.a. nicht den Rechner wechselt.

Hardware: 512 kB auf c't Statik-RAM-Floppy

Hardware-Änderungen

Gundlage für diese Aktion ist der Schaltplan von c't 5/86, Seite 102.

1. IC 11 (74 HC 138) Pin 5 hochbiegen
2. IC 11' (74 HC 138) auf IC 11 huckepack auflöten, dabei die Pins 5 sowie Pins 7, 9 bis 15 hochbiegen, die restlichen Pin's mit dem unteren 74 HC 138 verlöten.
3. Auf die RAM's (ICs 20-27) huckepack einen weiteren Satz gleicher RAM's löten, wobei von den oberen jeweils Pin 20 hochgebogen wird.
4. Die 8x Pin 20 der Huckepack-RAM's mit den Pins 7, 9 bis 15 des IC 11' (74 HC 138 huckepack) verbinden, d.h. jedes RAM erhält einen Pin des IC 11'
5. Auf IC 1 (74 LS 175) huckepack IC 1' (74 LS 74) auflöten, wobei lediglich IC 1' Pin 7 an IC 1 Pin 8 und IC 1' Pin 14 an IC 1 Pin 16 gelötet werden, d.h. nur Masse und +5 Volt werden verbunden. Pins 1 bis 6 des IC 1' werden abgekniffen.
6. Die anderen Pins des IC 1' (74 LS 74 huckepack) werden wie folgt verdrahtet:

Pin	Bedeutung	an
8	invertierter Output	Pin 5 des IC 11' (oberer 74 HC 138)
9	uninvertierter Output	Pin 5 des IC 11 (unterer 74 HC 138)
10	Set für FlipFlop	+5 Volt, d.h. Pin 14 des IC 1'
11	Takt	Pin 9 des IC 1 (74 LS 174 drunter)
12	Dateneingang	Pin 8 des IC 2 (74 LS 373)
13	Reset für FlipFlop	+5 Volt, d.h. Pin 14 des IC 1'

Funktionsweise:

Das Track-Register der Karte wird um das Bit D3 erweitert, da die derzeitige Dekodierungsmimik nur 256k ansteuern kann. Die verschiedenen RAMs werden durch das IC 11 (74 HC 138) selektiert. Daher muß ein zweiter 74 HC 138 für den 2. Satz RAMs her, und über D3 des Track-Registers jeweils einer der 74 HC 138 aktiviert werden. Dies wird dadurch erreicht, daß D3/Track auf ein Flip-Flop (74 LS 74) gegeben wird, dessen beiden Ausgänge (nicht-invertiert und invertiert) als Enable-Signal für die beiden 74 HC 138 dienen. Erfreulicherweise hat der 74 HC 138 drei Enable-Eingänge, von denen einer für diesen Zweck genommen werden konnte.

Anmerkung zur 1 MB-c't-Eprom-Floppy

Hierbei sind solche Klimmzüge nicht erforderlich, da die auf dieser Karte bei einer Kapazität von 1 MB eingesetzten EPROM's ein internes Paging haben, sich jedoch davon abgesehen nach Außen wie die für die 256k-Floppy verhalten. Das Paging wird durch Schreibzugriffe auf die EPROM's erreicht.

Hardware: 512 kB auf c't Statik-RAM-Floppy

Änderungen in FORMATE.DAT

CfCode	#56
Name	c't-CMOSRAM 512K
SPT	32
BLOCKSIZE in K	2
Blöcke	252
EXM	1
Dir-Einträge	128
ALO \$	C0
AL1 \$	00
CKS	0
OFF	2
RECLEN	128
FIRST	0
TRACKCNT	128
Seiten	1
Density	1
5 oder 8 Zoll	8
VR/RR	0
Invertiert	0
TrackTyp	1
SKEW	0
phys. Sektoren pro Spur	32

Ändern von CFIG4

Wer will - es ist aber nur kosmetischer Natur -, kann mittels INSCFIG dafür sorgen, daß die nun neue Beschreibung des Formates 56 auch von CFIG4 ausgegeben wird.

Änderungen in RAM4

Das nun um-definierte Format 56 muß auch in RAM4 eingebaut werden, was mittels INST4 oder INST42 vorgenommen werden muß:

0. Die CMOS-Floppy darf nicht konfiguriert sein!!!
Ggf. CFIG4 I:FF o.ä. bzw. neues Booten ohne CFIG4 für CMOS-Floppy!
1. Aufrufen des INST4 bzw. INST42 und angeben des Namens von RAM4.COM
2. Auswählen der Option (F)ormate
3. Merken der Formate die jetzt alle drin sind!!
4. (L)öschen der Formate
5. Wieder (E)inbauen der just gelöschten Formate (hierbei werden die neuen Informationen über das Format 56 herangezogen!)
6. Verlassen des Installationsprogrammes und dabei sichern von RAM4.

Hardware: 512 kB auf c't Statik-RAM-Floppy

Weitere Änderungen in RAM4 (so heißt mein RAM42)
 (Andere Versionen von RAM4 evtl. andere Adresse)
 Eingaben **Fett**, Wichtiges unterstrichen, Hinweise *kursiv*.

A>MONI RAM4

----- neuer Bildschirm -----
 MONI Version 2.03 vom 12.02.87 (C) 1985,1986 Olaf Krumnow

Geladen wurde RAM4 COM
 Geht bis Hex 5900
 Kommando>S Von>*0100 Bis>*5900 Wonach>E6 3F *Suche AND 3FH*
 1283
 Kommando>1

----- neuer Bildschirm -----
 MONI Version 2.03 vom 12.02.87 (C) 1985,1986 Olaf Krumnow

AF	BC	DE	HL	IX	SP
0000 C3	0000 C3	0000 C3	0000 C3	0000 C3	0100 C3

AF'	BC'	DE'	HL'	IY	PC
0000 C3	0000 C3	0000 C3	0000 C3	0000 C3	0100 C3
			JP	0157H	

Kommando>L Von>1270

```

FFA0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FFB0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FFC0 40 52 41 4D 34 32 80 F4 03 5F 1B 1B 50 57 54 56
FFD0 C3 DB E6 C3 00 F4 C3 E3 E6 C9 00 00 C9 00 00 80
FFE0 C9 00 00 00 21 20 20 20 04 56 3F 00 20 00 80 00
FFF0 C3 B9 F6 C3 F8 F5 C3 F5 F5 C3 A2 F5 C3 B4 F6 67
0000 C3 03 E1 AB 00 C3 FC A6 00 00 00 00 00 00 00
0010 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0020 C5 00 00 00 00 00 00 00 00 00 00 00 00 00 04
0030 C3 B2 C8 FF FF FF FF FF FF FF FF FF FF FF FF F0
0040 7C E5 A1 E2 05 0A 00 E6 0E 03 58 26 2B 08 00 00
0050 EF FF FF FF FF FF FF FF FF FF FF FF FF 09 52 41 4D
    
```

----- neuer Bildschirm -----
 MONI Version 2.03 vom 12.02.87 (C) 1985,1986 Olaf Krumnow

1270 DJNZ 126AH	AF	BC	DE	HL	IX	SP
1272 EI	0000 C3	0000 C3	0000 C3	0000 C3	0000 C3	0100 C3

1273 RET						
1274 LD C,0A4H						
1276 BIT 0,(IX-01H)	AF'	BC'	DE'	HL'	IY	PC
127A JR NZ,127EH	0000 C3	0000 C3	0000 C3	0000 C3	0000 C3	0100 C3
127C LD C,0ABH				JP	0157H	

Kommando>A 1283 **AND 7F**

```

1280 LD A,(OFFEAH) FFA0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
1283 AND 3FH FFB0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
1285 LD D,A FFC0 40 52 41 4D 34 32 80 F4 03 5F 1B 1B 50 57 54 56
1286 LD A,(OFFECH) FFD0 C3 DB E6 C3 00 F4 C3 E3 E6 C9 00 00 C9 00 00 80
1289 DEC A FFE0 C9 00 00 00 21 20 20 20 04 56 3F 00 20 00 80 00
128A ADD A,A FFF0 C3 B9 F6 C3 F8 F5 C3 F5 F5 C3 A2 F5 C3 B4 F6 67
128B ADD A,A 0000 C3 03 E1 AB 00 C3 FC A6 00 00 00 00 00 00 00
128C ADD A,A 0010 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
128D SRL D 0020 C5 00 00 00 00 00 00 00 00 00 00 00 00 00 04
128F RRA 0030 C3 B2 C8 FF FF FF FF FF FF FF FF FF FF FF FF F0
1290 SRL D 0040 7C E5 A1 E2 05 0A 00 E6 0E 03 58 26 2B 08 00 00
1292 RRA 0050 EF FF FF FF FF FF FF FF FF FF FF FF FF 09 52 41 4D
    
```

Hardware: 512 kB auf c't Statik-RAM-Floppy

Nach einer wiederholung der Eingabe Kommando>L Von>1270

----- neuer Bildschirm -----
 MONI Version 2.03 vom 12.02.87 (C) 1985,1986 Olaf Krumnow

1270	DJNZ	126AH	AF	BC	DE	HL	IX	SP
1272	EI		0000 C3	0000 C3	0000 C3	0000 C3	0000 C3	0100 C3
1273	RET							
1274	LD	C,0A4H						
1276	BIT	0,(IX-01H)	AF'	BC'	DE'	HL'	IY	PC
127A	JR	NZ,127EH	0000 C3	0000 C3	0000 C3	0000 C3	0000 C3	0100 C3
127C	LD	C,0A8H				JP	0157H	
127E	LD	B,80H	Kommando>					
1280	LD	A,(OFFEAH)	FFA0	FF FF FF FF	FF FF FF FF	FF FF FF FF	FF FF FF FF	FF FF FF FF
1283	AND	7FH	FFB0	FF FF FF FF	FF FF FF FF	FF FF FF FF	FF FF FF FF	FF FF FF FF
1285	LD	D,A	FFC0	40 52 41 4D	34 32 80 F4	03 5F 1B 1B	50 57 54 56	
1286	LD	A,(OFFECH)	FFD0	C3 DB E6 C3	00 F4 C3 E3	E6 C9 00 00	C9 00 00 80	
1289	DEC	A	FFE0	C9 00 00 00	21 20 20 20	04 56 3F 00	20 00 80 00	
128A	ADD	A,A	FFF0	C3 B9 F6 C3	F8 F5 C3 F5	F5 C3 A2 F5	C3 B4 F6 67	
128B	ADD	A,A	0000	C3 03 E1 AB	00 C3 FC A6	00 00 00 00	00 00 00 00	
128C	ADD	A,A	0010	FF FF FF FF	FF FF FF FF	FF FF FF FF	FF FF FF FF	
128D	SRL	D	0020	C5 00 00 00	00 00 00 00	00 00 00 00	00 00 00 04	
128F	RRA		0030	C3 B2 C8 FF	FF FF FF FF	FF FF FF FF	FF FF FF F0	
1290	SRL	D	0040	7C E5 A1 E2	05 0A 00 E6	0E 03 58 26	2B 08 00 00	
1292	RRA		0050	EF FF FF FF	FF FF FF FF	FF FF FF FF	FF FF 09 52 41 4D	

Die Eingabe Kommando>O bewirkt:

----- neuer Bildschirm -----
 MONI Version 2.03 vom 12.02.87 (C) 1985,1986 Olaf Krumnow

Kommando>W Von>0100 Bis>*5900 Name>*RAM4 COM
 File existiert. Überschreiben (J/N) ? J

Nun noch ESC drücken - fertig

ÜBRIGENS: Diese Änderungen können auch mit DDT gemacht werden, nur weiß ich nicht, wie ich mit diesem Gerät Hex-Folgen suchen kann - ach eigentlich habe ich überhaupt kaum eine Ahnung, wie man mit DDT irgend etwas macht, da ich nur mit MONI von Olaf arbeite.

Änderungen des BOOT-EPROMs

Wer von dieser Platine booten möchte, oder es schon tut muß nach dieser Aufrüstung das Boot-EPROM Version 1.6, welches auch BOOT I:56/512k meldet einsetzen, welches es bei mir im Tausch gegen das vorhandene Boot-Eprom für DM 10,- gibt.

Wird das EPROM nicht ausgewechselt kann das System nicht von dieser aufgerüsteten Platine booten!

Statistik**Anspruchsvolle Statistik - in Planung**

(Dr. Holger Göbel, 8630)

In den letzten Jahren bin ich immer wieder in Verlegenheit gekommen, innerhalb eines Meßprogrammes auch statistische Auswertungen einzubauen. Nachdem die Schulzeit etwa Pi-Quadrat Jahre her ist, habe ich mir dann immer mühevoll irgendwelche Formeln an Land gezogen und mit denen dann Wurstel-Programme konstruiert.

Jetzt bin ich in der c't 3/88 auf eine Anzeige des Heise-Verlages gestoßen, die Remittenden (also vom Händler an den Verlag zurückgegebene Bücher) billig für DM 9,80 anbot, u.a. 2 Bücher von Dr. Oskar Hoffmann, nämlich:

"Biomedizinische Statistik mit dem Microcomputer" und
"Statistik mit BASIC".

Beide sind im Luther Verlag, Gensingen, erschienen, zwar schon 1984 und 1985, aber Gutes wird ja bekanntlich nicht schlechter, wenn es älter wird.

Folgendes ist dann passiert: Als Nichtmathematiker habe ich das erste Mal Statistik begriffen! Das erstgenannte Buch habe ich an einem Tag "aufgesaugt" und war einfach begeistert davon, wie gut komplexe Sachverhalte didaktisch aufgearbeitet werden können. Zudem sind alle statistischen Tests in BASIC-Programmen ausgearbeitet und sehr gut strukturiert.

Aus dem Inhalt des erstgenannten Buches:

Matrizenoperationen, gängige deskriptive Statistik (MW, SD, SEM, Boxer- und Whiskerdarstellung usw.), t-Test, Wilcoxon-Test, Kontingenztafeln, Chi-Quadrat-Test (mit YATES-Korrektur, exakter Test nach Fisher), Regression und Korrelation, mehrdimensionale Varianzanalyse (!), Faktorenanalyse (!).

Das zweite Buch bietet zusätzlich noch:

Kruskal-Wallis-Test, Friedman-Test, Analyse nicht linearer Zusammenhänge (!).

Nun habe ich folgendes vor:

Ich möchte die Programme für unseren MTX nach TURBO-Pascal transkribieren, u.a. deswegen, weil sie dadurch schneller werden, in andere Programme mittels INCLUDE einbindbar sind und evtl. auf andere Rechner (über den Source-Code) übertragbar werden (Kollegen von mir haben sich schon angemeldet). Die Programme werde ich dann zu Public Domain machen, so daß sie jeder beziehen kann. Dringend zu raten wäre jedoch dann, zumindest sich das erstgenannte Buch anzuschaffen, und zwar möglichst bald (denn Remittenden gibt es nicht unbegrenzt). Ihr könnt das selber tun (Verlag Heinz Heise, Pf 61 04 07, 3000 Hannover 61) oder mir eine Nachricht zukommen lassen, daß Ihr interessiert seid. Für den Notfall habe ich schon 5 Reserve-Exemplare bestellt.

Wie lange ich für das Umschreiben brauche, weiß ich nicht, jedenfalls werde ich das bittere Ende im INFO bekanntgeben.

Noch etwas: Ich kann mir vorstellen, daß bei einer Faktorenanalyse die Aufrüstung auf 8 MHz eine willkommene Konzession an die Geduld ist.

A s s e m b l e r : B L O C K - S W A P / M O V E

Ein besonderer Block-Schiebe-Algorithmus (Dr. Holger Göbel, 8630)
(oder: Tausch zweier benachbarter, ungleich großer Blöcke ohne Puffer)

In Z80-Assembler gibt es ja die mächtigen LDIR/LDDR- Blockladebefehle, d.h.: Gibt man Quell-, Zieladresse und Anzahl der zu verschiebenden Bytes an, so geschieht der folgende Byte-Transfer außerordentlich bequem und schnell.

Folgendes Problem aber (mit Beispielzahlen):

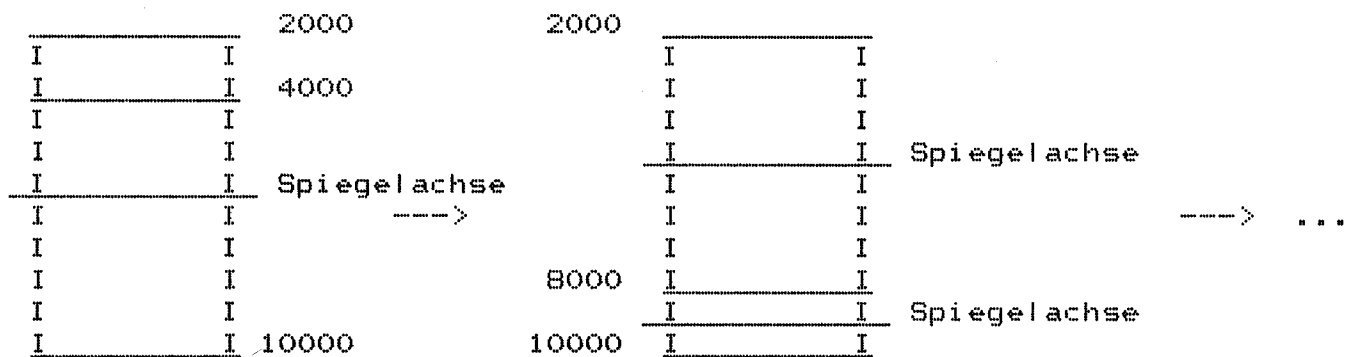
Ich habe einen Datenblock von der Adresse 0-20000.
 Den Block 2000-4000 möchte ich nach der Adresse 8000 verschieben, der Block von 4000-10000 muß natürlich dabei nach unten gesetzt werden.

Ganz einfach, denkt man sich, ich schreibe den Block 2000-4000 in einen Puffer, verschiebe den Block 4000-10000 nach unten und füge dann den Puffer nach 8000 ein.

Das ist sicher auch der schnellste Weg. Was aber, wenn ich gar nicht genügend Puffer habe (was bei unseren etwas bescheidenen 64 k gar nicht so abwegig ist)? Dann könnte ich z.B. Unter-Blöcke "kleinpufferweise" (z.B. immer 256 Bytes) nacheinander übertragen. Das wird dann aber ziemlich aufwendig und langsam, außerdem möchte ich partout einen Algorithmus ohne Puffer.

Sicherlich existieren solche Algorithmen schon. Mir war jedenfalls keiner bekannt, so daß ich mir folgendes ausgedacht habe:

Ich definiere einen großen Block, der von der Anfangsadresse (2000) bis zur späteren Endadresse (10000) reicht. Den spiegle ich an der Mittellinie (6000). Anschließend muß ich noch die beiden Einzelblöcke (2000-8000 und 8000-10000) einzeln für sich zurückspiegeln. Das ist gleichbedeutend mit einem Tausch zweier benachbarter, auch ungleich großer Blöcke. Folgende Zeichnung soll das verdeutlichen:



A s s e m b l e r: BLOCK-SWAP/MOVE

Nun das Programm: Übergeben werden der Beginn des 1. Blockes in DE, das Ende des 1. Blockes in BC und das Ende des 2. Blockes in HL. Zurückgegeben wird in BC das Ende des vormals zweiten Blockes (also die Trennlinie der Blöcke):

```

PUSH HL          ;Ende des 2. Blockes
PUSH DE          ;Beginn des 1. Blockes
CALL SPIEGEL    ;Gesamtblock spiegeln
POP DE
POP HL
PUSH HL
AND A            ;CARRY löschen
SBC HL,BC       ;neue Trenngrenze =
ADC HL,DE       ;END2 - (END1 - BEG1) -1 =
DEC HL          ;END2 - END1 + BEG1 -1
LD B,H
LD C,L          ;Ergebnis nach BC
PUSH HL         ;und auf den STACK
CALL SPIEGEL    ;Spiegeln des vormals 2. Blockes
POP DE
INC DE          ;neuer Beginn des vormals 1. Blockes
POP HL          ;Ende des Gesamtblockes
CALL SPIEGEL    ;Spiegeln des vormals 1. Blockes
RET
;
;Spiegelsubroutine: HL = obere, DE = untere Grenze
;
SPIEGEL: PUSH BC      ;BC retten
PUSH HL         ;obere Grenze retten
AND A          ;CARRY löschen
SBC HL,DE      ;so viele Bytes hat der Block
INC HL         ;(egal, ob gerade oder ungerade)
SRL H          ;
RR L           ;Zähler = INT((Anzahl+1)/2)
LD B,H
LD C,L         ;Zähler nach BC
POP HL        ;obere Grenze zurück
LD A,B
OR C
JP Z,GOHOME   ;so wenig ???
;
HALLO: LD A,(DE) ;(sehr vielsagendes Label)
LDI    ;(HL) --> (DE), HL+1, DE+1, BC-1
DEC HL  ;HL+1 rückgängig machen
LD (HL),A ;(DE) --> (HL)
DEC HL  ;HL-1
JP FE,HALLO ;bis BC=0
;
GOHOME: POP BC   ;BC zurück
RET

```

C L U B: Ganz zum Schluß noch etwas / Ports

Wenn jemand einen besseren Algorithmus kennt, möge er bitte an mich (und evtl. auch an die anderen Clubmitglieder) denken. Zur Zeit nämlich kopple ich meinen alten Spectrum mit dem MTX. Dazu habe ich den ECB-Adapter aus der c't 6/86 an den Spectrum angeschlossen, der als Bonbon die Verwendung eines eigenen EPROMs erlaubt. Da im Spectrum noch reichlich Platz im Betriebssystem ist, baue ich da verschiedene Verbesserungen ein, u.a. auch eine MOVE-Routine (dafür also obiges Programm) mit RENUM-Funktion und die nackten SUPERTAPE-Routinen. Wie ich im letzten INFO beschrieben habe, klappt die Kopplung der Rechner mit 7200 Baud über ein kleines Interface sehr gut. Wenn ich also Daten aus dem Spectrum speichern will, gebe ich sie über SUPERTAPE an den MTX, der weiter mit der Floppy kommuniziert. Das ist zwar etwas umständlich, aber immer noch weit besser als ein Kassettenrekorder und billiger als eine Floppykarte für den Spectrum (z.B. die IFC aus der c't), für die dann auch noch ein BDOS zu schreiben wäre. (Der Spectrum soll in meiner Kopplung momentan als billiger Speicheroszillograph für Signale im kHz-Bereich, die der MTX aufgearbeitet hat, fungieren).

Ganz zum Schluß noch etwas:

(Dr. Holger Göbel, 8630)

Ich wäre dankbar für eine regere Veröffentlichung von Routinen, die man in einer Programmbibliothek gut gebrauchen kann (z.B. in Assembler, Pascal, BASIC, C), wie es ja auch schon geschehen ist (z.B. die VS4-Routinen von Olaf Krumnow für TURBO-Pascal oder viele nützliche Routinen von Kurt-Bernd Rohloff und HzN). Besonders würde mich interessieren eine **Floating-Point-Arithmetik** in Assembler (die Grundrechenarten würden schon fürs erste genügen).

Ports

(Herbert zur Nedden, 2000)

<u>Bereich</u>	<u>Port</u>	<u>Verwendung</u>
MTX	00 - 0F	Grundgerät
	10 - 14	SDX-Floppy-Controller!
	18 - 1B	8255-PIO-Box, H. zur Nedden
	1F	vorgesehen für Cassettenmotorsteuerung
FDX	30 - 33	80-Zeichen-Karte
	38 - 39	6845-Controller der 80-Zeichen-Karte
	40 - 47	FDX-Floppy-Controller
	50 - 5F	4x Memotech SiliconDisc
	70 - 73	EPROM/SRAM-Floppy von J. Marquart und F. Cröll
ECB	80 - 83	EDICTA Grafik-Karte
	88 - 8B	HardDisk (noch nicht Realisiert)
	90 - 97	2x 8255-PIO
	98 - 9B	c't RAM-Floppy
	A0 - A3	EDICTA RAM-Floppy
	A4 - A7	c't EPROM-Floppy bis 256 kB
	AB - AB	c't SRAM-Floppy
	B8 - BB	Conitec-Floppy DISI
	BC - BF	Conitec-Floppy DISI
	C0 - C4	Reserviert für Testzwecke
	CC - CF	Janich & Klass Programmier
	<u>D0 - DF</u>	c't I/O-Karte mit CTC, SIO und 2x PIO
<u>E4 - E8</u>	c't EPROM-Floppy 1 MB	
F8 - FB	HD 64180 Sub-Prozessor-Karte	

Anm. d. HzN.: Ob für EPROM-Floppy 1 MB E4 - E8 stimmt kläre ich noch!