

# MTX *User-Club Deutschland*

Info 32  
09.06.1989

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

**Telefon-Sprechzeiten**

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

Inhaltsverzeichnis

<b>C l u b:</b>	
Korrektur & Nachtrag	Seite 2
Fragen & Antworten	Seite 3
Diverses	Seite 4
<b>T u r b o - P a s c a l:</b>	
Info's	Seite 5
<b>A s s e m b l e r:</b>	
DS versus EQU	Seite 6
SLR-Systems	Seite 14
<b>N e w W o r d:</b>	
Tips	Seite 8
<b>S u p e r C a l c:</b>	
Patch	Seite 11
<b>C P / M:</b>	
Daten von einer Diskette holen	Seite 12
<b>d B A S E:</b>	
SuperBusiness	Seite 14
<b>S o f t w a r e:</b>	
Dieses und Jenes	Seite 13
CONFIG5 und FORM5	Seite 16
E45	Seite 18
Wortmanager	Seite 23
<b>L e s e r b r i e f:</b>	
Hans Gras, NL 1506	Seite 20
Claudio Romanazzi, 3070	Seite 21
<b>H a r d w a r e:</b>	
FDX-ECB-Adapter	Seite 4
Umbau auf 8 MHz	Seite 25
Tips	Seite 28
Sprachausgabe	Seite 29
80-Zeichenkarte	Seite 31
Schaltplan 80-Zeichen-Karte original	Seite 36
Schaltplan 80-Zeichen-Karte HGR	Seite 41
Diskettenwechsel erkennen	Seite 43
c't SRAM-Floppy	Seite 45
Maus	Seite 47
DART per Interrupt für die Maus	Seite 51
Hard-Disc	Seite 53
Hard-Disc und RAM x.x	Seite 60
IBM-Tastatur	Seite 61
<b>K o m i k:</b>	
Fremde Sprak is sich schwäre Sprak	Seite 63

Preis für dieses Info: DM 12,-

Redaktionsschluß für's nächste Info: Anfang August

**Anmerkung zu Info-Beiträgen**

(Herbert zur Nedden, 2000)

Aus technischen Gründen muß unten auf jeder Seite der Druckvorlage mindestens ein Centimeter unbedruckter Rand bleiben - dort 'fäßt' die Druckmaschine das Papier an. Außerdem gehen oben auf jeder Seite noch die beiden Kopfzeilen samt der trennenden Leerzeile vom nutzbaren Platz ab. Bitte trägt diesem Umstand Rechnung, insbesondere, falls Ihr Euren Beitrag ausgedruckt an mich schickt. Für Euch ist ein geeignetes .PL einfacher als für mich die Aktion mit Schere und Klebstoff.

Liebe Leserin, lieber Leser,

stimmt, dieses Info hat lange auf sich warten lassen! Hier meine (fadenscheinige ?) Begründung: Ich wollte zuvor das Clubtreffen abwarten, zumal ich die Zeit bis dahin auch gut brauchen konnte um das eine oder andere Programm zu erstellen (siehe weiter hinten) + Familie + Firma = wenig Zeit. Ich hoffe Euch durch den Inhalt des Infos und die anderen Ankündigungen und Angebote für Eure Geduld zu entschädigen.

Unser nordisches Clubtreffen fand am Wochendende 26/27. Mai bei Hemmingen (südlich von Hannover) in einem Landgasthof statt. Von den ca. 20 Teilnehmern, die aus dem Bereich Rendsburg, Puchheim/Fürstfeldbruck, B-Erembodegem incl. dieser genannten Orte kamen, haben 11 vor Ort übernachtet. Es war richtig schön, zusammen sitzen und Gedanken austauschen (PIP a:=b:?.Tip) zu können, ohne auf die Uhr zu schauen, und an 2-? Stunden Fahrt denken zu müssen.

Es war schon erstaunlich zu sehen, was aus unserer meist noch schwarzen Kiste herauszuholen ist. Sei es nun Horst Kupka's 80-Zeichen-Grafik, für die er auch schon einiges an Software erstellt hat, oder die Festplatte an Claudio Romanazzi's Memotech. Kurt-Bernd Rohloff hatte seine Literaturverwaltung mitgebracht, und so gezeigt, daß man mit dBASE einiges auf die Beine stellen kann; insbesondere die möglichen Suchanfragen waren erstaunlich! Aber auch 'Kleinigkeiten' gab es zu sehen, wie eine an meiner Mühle laufende Maus, mit der zwar auch der Randausgleich unter NewWord zurückgelehnt gemacht werden kann, die jedoch sicherlich eher für Grafik-Anwendungen einsetzbar ist. Olaf Krumnow wartete gleich mit zwei Klick-Overlays auf, die leider noch als Prototypen zu bezeichnen sind - immerhin war sein Rechner zur Umbau/Erweiterungs-Kur bei mir, so daß er nicht so recht ans Programmieren kam. Nichtsdestotrotz war der Betrieb mit dem Cache (genauerer siehe meinen Artikel zum Disc Change Signal), mit dem die Diskettenzugriffe erheblich beschleunigt werden können, oder auch sein neuer DiJey, den jetzt keine BDOS-Errors mehr aus den Angeln heben, und der auch DIR-Listings und gar .SUB-Dateien auf Diskette schreiben kann. Wem sein Klick-Heap immer noch nicht voll genug ist, dem möchte ich meinen Non-Doc Editor MTX-Edit in der .KLX-Version ans Herz legen, der für 12k bis 48k Bernd Preusing's Note ersetzt, und mehr kann, als seine CP/M-Brüder. Mein Terminkalender war zwar auch schon zu sehen, d.h. die Oberfläche und Terminverwaltung im Klick laufen, aber ... Anwendungen, die es eilig haben, ihren Senf auf die Glotze zu bringen, sollten sich Olaf's Bildschirmtreiber FastScreen bedienen, der logischerweise kompatibel zum MTX+++-Bildschirmtreiber von RAM 4.x ist (d.h. gemischter Betrieb ist möglich), jedoch nur die allerwichtigsten Funktionen bietend direkt auf die Ports der 80-Zeichen-Karte zugreift. Wie unter Olaf's Programm LIFE zu sehen, sieht man eigentlich kaum etwas davon, wie die 28x96 oder gar 56x96 Zeichen neu ausgegeben werden. (Wie heißt es auch so schön: Wie sie sehen, sehen Sie nichts. Und warum Sie nichts sehen, werden Sie gleich sehen.) Gerhard Witzel hat mir Arbeit, und Olaf Platz auf seinem Schreibtisch beschert: Er hat eine FDX-Einschubkarte mitgebracht, die sich durch ein großes Loch auszeichnet, in welches eine ECB-Karte eingesteckt werden kann, und - schließlich ist eine ECB-Karte deutlich kleiner als eine FDX-Karte - gleich noch die Leiterbahnen und den Platz für eine Hardware-Uhr samt der erforderlichen IC's hat. Manfred Flume bombardierte uns gleich mit 2 Disketten à 790k voller Turbo-Pascal-Routinen/Programme, die Holger Göbel gleich zwecks Untersuchung eintütete. Hat der die Doku fertig, gibt's neue PD's. Holger stellte sein Statistik-Paket (Prototyp) vor, welches er mit einem Spreadsheet für die Dateneingabe und Grafik-Ausgabe versehen hat.

Die Reihenfolge und (un)vollständigkeit meiner o.g. Ausführungen ergab sich völlig zufällig beim schreiben - bitte seht es mir nach. Ich habe auf dem Clubtreffen das eine oder andere nur aus dem Augenwinkel mitbekommen - es war einfach so viel, daß ich keine Chance hatte, alles in Ruhe anzusehen. Auf jeden Fall war das Treffen nicht nur sehr informativ (böswillige Stimmen sprechen von Copy-Fete). Es war auch toll, einfach mal den einen oder anderen wiederzusehen und zu klönen (= norddeutsch für schnacken, = norddeutsch für unterhalten). Meinen Dank an alle, die dazu beigetragen haben, daß es ein so schönes Wochenende wurde - insbesondere an die, die nach Übernachtung vor Ort 'geschrieben' haben.

Euer Herbert

C l u b: Korrektur & Nachtrag / Fragen & Antworten**Korrektur und Nachtrag**Info 27-52: Spritverbrauchsrechnung (Kurt-Bernd Rohloff, 8000)

Leider habe ich erst nach Veröffentlichung des Beitrages folgendes bemerkt: Wenn SC eine Zahl im "\*" Format darstellen soll (so wie in der Spalte J in dem Verbrauchs-Spreadsheet), benutzt es grundsätzlich nicht die erste Stelle der Zelle. Vermutlich soll hier Platz für ein Minuszeichen reserviert werden, was jedoch im "\*" Format keinen Sinn macht. Die auf S. 53 unten angegebene Formel  $(I10-a)/(b-a)*c$  stellt die Werte in der I-Spalte im Bereich von a bis b deshalb erst richtig dar, wenn die J-Spalte c+1 Stellen breit gemacht wird.

RAM 4.5-Diskette: (Herbert zur Nedden, 2000)

Auf dieser Scheibe ist eigentlich nichts neues. Sie soll lediglich helfen, mit all dem Wust an RAM 4.x-en und Zubehör klar zu kommen. Auf dieser Diskette ist daher die Installation zusammen mit einigen Informationen menügeführt sowie die m.E. wichtigsten Programme für den Einsatz von RAM 4.5 zu finden.

Maus-Treiber (KLICK.006): (Herbert zur Nedden, 2000)

Weiter hinten im Info findet Ihr mehr zur Maus - hier nur ein Nachtrag, damit die, die die Maus schon anzuschließen versuchen auf diesen Punkt stoßen. Manfred Flume (5600) hat beim Anschließen seiner Maus festgestellt, daß seine V24-Treiber 1488 von Texas Instruments die gewünschten -12 Volt nicht liefern kann - was wohl daran liegt, daß die Maus daraus Saft zieht. Nachdem er einen Widerstand von 2,2 bis 2,7 Ohm zwischen -12 Volt (Pin 1 des 1488) und den Pin, an den RTS/DTR/TxD (= 1 Pin) kommen, gelötet hat, lief die Maus an der RS232-A. Mir ist dieses Problem nicht aufgestoßen, da meine nicht-Texas-Treiber keine Mucken machen. Also erst Testen, dann messen, dann löten!

MTX-Edit im KLICK (KLICK.006): (Herbert zur Nedden, 2000)

Ja, es gibt meinen Editor MTX-Edit nun im KLICK - und er kann mehr als das Original und fängt BDOS-Errors ab. Nichtsdestotrotz vermißte Jan Bredereke (2000) bei der Installation die Voreinstellung, ob Zeilenschaltungen (CR/LF) durch Dreiecke angezeigt werden sollen, oder nicht. Hier nun die Ergänzungen für die Dateien E45.PAT und E45.INS, um dieses nachzuholen:

E45.PAT (die folgenden zwei Zeilen)

00135 1b ; CR/LF als Dreiecke beim Starten  
; 1b = ja, 00 = nein

E45.INS (die folgenden vier Zeilen)

Sollen CR/LF's als Dreiecke dargestellt werden. Kann bei laufendem MTX-Edit mit ^O^C umgeschaltet werden.

135 F1b/00

Anmerkungen: E45.KLX heißt der MTX-Edit für's KLICK, daher die E45. E45.PAT ist für ThePatch, E45.INS ist für TheInst - TheInst ist Olaf's Fortführung der Idee von TheInst. Doku zu all diesen Teilen ist auch auf KLICK.006

HINWEIS: Die neueste Version von MTX-Edit für's KLICK ist auf KLICK.006. Alle anderen Exemplare hiervon, die ich vorher schon als Prototypen weitergegeben habe sind unvollständiger. Wenn Dein E45.KLX mit dem Befehl ^K^P nichts anfangen kann ist es nicht die neueste Version!

POST.KLX (diverse KLICK.00?) (Herbert zur Nedden, 2000)

Jan Bredereke (2000) stellte fest, daß die Auslands-Portokosten dieser Anzeige die Unterscheidung nach bestimmten Ländern, wie insbesondere CEPT-Ländern nicht beinhaltet.

Das stimmt ebenso wie der Umstand, daß Berlin-Sonderbestimmungen nicht mit drin sind. Das ist so! Ich weiß momentan nicht, wie ich das alles so unterbringen kann, daß mir das Panel noch gefällt.

Club: Korrektur & Nachtrag / Fragen & Antworten**Fragen und Antworten**

- A: (Andreas Fischer, CH-4303)  
Auf die Anfrage aus Info 31-3 zum Thema Sprachausgabe.  
--> Siehe Artikel weiter hinten.
- F: (Kurt Meitzner, 8760)  
Wer kann mir die 12DITVM auf Kassette liefern ?
- F: (Frank Schäfers, 4350)  
Wie steuere ich den Drucker STAR LC-10 unter NewWord an (Druckertreiber ?)
- F: (Andreas Fischer, CH-4303)  
Mir ist die Bedeutung der Angaben über code, free, data unklar. Trotz Reserven unter der Bezeichnung "free" erhalte ich Fehlermeldungen 99 (=Compiler-überlauf).
- A: Siehe weiter hinten: Turbo-Pascal Info's
- F: (Andreas Fischer, CH-4303)  
Mein Drucker übernimmt keine Leerzeichen aus dem NW, sondern übergeht sie ohne Zwischenraum. Ich kann mir nur mit der Einfügung von ", " am Anfang der Zeile helfen. Wer weiss Rat ?
- A/F(Herbert zur Nedden, 2000)  
A: Evtl Punkt-Kommando '.uj off' oder ein einfacherer Druckertreiber.  
F: Was für'n Drucker und was für'n Druckertreiber hast/benutzt Du ?
- F: (Andreas Fischer, CH-4303)  
Wo gibt es CP/M-Software zu kaufen, die auf unserem Rechner läuft. Ich denke z.B. an Flight-Simulator.
- A: (Herbert zur Nedden, 2000)  
Richtige CP/M-Software können wir in der Regel finden. Programme, wie Flight-Simulator benötigen doch wohl Grafik, die kein Bestandteil von CP/M ist. Daher laufen solche Programme zwar teilweise unter CP/M, aber dann mit wenig Grafik oder nur auf bestimmten Kisten.
- F: (Andreas Fischer, CH-4303)  
Das Gehämmer auf dem Rechner geht meinen Familienangehörigen auf die Nerven. Wer hat Erfahrung mit dem Einsatz von umweltfreundlicheren Keyboards ? Primär sollten die Hackertöne verschwinden. Vielleicht können auch die einzelnen Tasten gedämpft werden.
- A: (Herbert zur Nedden, 2000)  
Unter RAM 4.x gibt es die Möglichkeit bestimmte IBM-Tastaturen anzuschließen, die allerdings die Memotech-Tastatur nicht ersetzen können - da die Einbindung in dieser ersten Version noch etwas unvollständig ist.
- F: (Andreas Fischer, CH-4303)  
Ich besitze einen MTX 512 mit je einem 3.5"-Disk-Driver und RAM-Disk. Nun möchte ich einen zusätzlichen Driver und zusätzliche RAM-Disks anschließen. Da der 3.5"-Driver auf der linken Seite aber keinen Anschluß aufweist, sehe ich keine Möglichkeit. Wer hat bereits ähnliche Probleme gelöst.
- A: (Herbert zur Nedden, 2000)  
Soweit ich weiß, unterstützt der Disk-Driver zwei Laufwerke, die beide über ein gemeinsames Fachbandkabel an diesen angeschlossen werden. Beim ersten Laufwerk steckt der Jumper DS0 auf dem Laufwerk, beim zweiten DS1. Damit wären A:=B: und C: gegeben. Für eine RAM-Floppy ist evtl. ein ECB-Adapter eine geeignete Lösung.
- F: (Peter Lang, 8502)  
If KeyPressed klappt nicht so recht unter Turbo-Pascal. Auch BIOS 6 hilft nicht.
- A: Siehe unten: Turbo-Pascal Info's
- F: (Mario Hilgemeier, 2800)  
Wie lese ich MTX-Disketten auf MsDos-System ?
- A: (Herbert zur Nedden, 2000)  
Garnicht - hat noch keiner geschafft. Ggf. kann ich für Dich konvertieren.
- F: (Andreas Fischer, CH-4303)  
Wie kann das Modem-Prog. M1.COM (CLUB.001) auf 7 Bit umprogrammiert werden ?

D i v e r s e s / H a r d w a r e: FDX-ECB-Adapter**Kleinkram**Wieder eine dSM: RS 232-Platine:

Beim IC 3 auf der RS232-Platine sollten die Pins 9 und 10 verbunden sein, was anscheinend nicht immer der Fall ist. Lauft Deine Serielle nicht, kann dies der Grund sein.

Disketten-Format ELZET80:

(Hartmut Traber, 5270)

Michael Keler hat dieses Format geknackt: Format DC mit Tracktyp 5.

dBASE und RESET:

(Hartmut Traber, 5270)

Aus einem Help zu dBASE:

## 1. RESET :&lt;Laufwerk&gt;:

nach einem Diskettenwechsel ist dieser Befehl auszufuhren. Dadurch konnen wahrend der Arbeit die dBASE Disketten ausgetauscht werden. Bevor jedoch der Befehl RESET ergeht, sind unbedingt alle Dateien auf der zu wechselnden Diskette abzumelden.

## 2. CLEAR

schliet alle eroffneten Datenbanken und loscht alle erzeugten temporaren Variablen und Arbeitsbereiche.

Evtl. vor RESET ein CLEAR ???

**ECB-Adapter fur die FDX**

(Herbert zur Nedden, 2000)

Ja, es gibt ihn. Gerhard Witzel hat eine FDX-Einschubkarte entworfen, die er auch atzt, bohrt, lotet und deren Kontakte veredelt. In der Grundversion bietet diese Karte einen ECB-Steckplatz, bei dem die Leitungen D<sub>0-7</sub>, A<sub>0-7</sub> und die wichtigsten Steuerleitungen sowie Masse und +5 Volt durchgefuhrt sind. Vorgesehen, d.h. durch Leiterbahnen unterstutzt, ist die Verdrahtung der restlichen Leitungen vom FDX-Bus zum ECB-Bus (z.B. A<sub>8-15</sub>, die aber in der Regel eh nicht benotigt werden). Auerdem ist auf der Platine das Layout fur eine Hardware-Uhr an Port-Adresse 3Eh vorgesehen. Es kann auf dieser Platine wahlweise ein Port 7-Ersatz fur die Uhr aufgebaut oder die Bauteile der Uhr direkt auf die Platine gelotet werden. Den fur den Wechsel des Uhren-Ports benotigten Patch in RAM 4.3 und DATE4 habe ich noch nicht herausgesucht - Anruf genugt. Damit kann immerhin Port 7 wieder frei gemacht werden. In der Nicht-Grundversion ist alles fertig gelotet.

**Wo gibt's Turbo-Pascal zu kaufen**

(Herbert zur Nedden, 2000)

Heimsoeth liefert Turbo-Pascal nimmer fur CP/M. Borland U.S.A. hat den Vertrieb dieses Programmes an

Alpha Systems Corporation  
711 Chatsworth Place  
San Jose, CA 95128  
U.S.A.

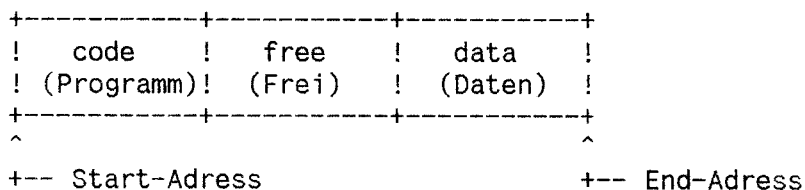
ubergeben. Ich habe um eine Preisliste gebeten. Wenn ich genaueres wei, melde ich mich wieder.

Turbo-Pascal: Info's

**Turbo-Pascal - Speicheraufteilung**

(Herbert zur Nedden, 2000)

Turbo-Pascal verteilt Programm und Daten in .COM's und .CHN's wie folgt:



Vor dem Code liegt Turbo's Runtime-Library (daher ist die StartAdress auch nicht 100h), und hinter den Data kommt das BDOS. Wenn ein Programm unter einem 58k-System einfach als .COM bzw. .CHN umgewandelt wird, ohne die End-Adress herabzusetzen, läuft es nicht unter einem 54k-System, da Turbo die End-Adress automatisch dicht unter das BDOS packt - und zwar fest. Wird nun das Programm unter einem kleineren CP/M-System aufgerufen, liegt die End-Adress, und damit die Datenbereiche im BDOS - und das mag Turbo nicht, meldet als soetwas wie NoSpace und beendet sich, was auch gut so ist. Wird ein Turbo-Programm unter ZEX gestartet, treten die gleichen Platzprobleme zu Tage, sich ZEX resident unterhalb des BDOS (genauer sogar unterhalb des CCP = ZCPR2) packt. Daher sollte die End-Adress stets kleiner gesetzt werden, z.B. auf meinen Lieblings-Wert von A000, was bislang keine Probleme bereitete - auch nicht unter ZEX!!

Nun zur Frage, woher Compiler-Überlauf: Wenn Turbo compiliert, hat es das Programm, jedoch nicht die Include-Dateien im Speicher. Ist nun das Programm groß, bleibt wenig Restspeicher für die bei der Umwandlung benötigten Tabellen und sonstigen Informationen. Die Lösung ist recht simpel: Will ich das riesige TEST.PAS umwandeln, wandle ich einfach folgendes einzeliliges Programm um:

```
(*$I TEST.PAS*)
```

Turbo hat damit nur wenige Bytes Programm-Quelltext im Speicher, da es die Include-Dateien nur durchgeht und nicht ganz lädt. So klappt die Compilation für's .COM bzw. .CHN. Klappt das immer noch nicht, sollten evtl. mal die Bezeichner für die diversen Variablen, Procedures etc. gekürzt werden.

Wenn Du jedoch die Umwandlung im Speicher machst, d.h. das Programm nicht als .COM oder .CHN erzeugst, ist der Platz wesentlich kleiner, da dann Turbo + Source + ganzes Programm + alle Daten im Speicher stehen müssen.

**KeyPressed**

(Herbert zur Nedden, 2000)

Da habe ich so eine Schleife

```
repeat
    TueDieseUndJenes;
until KeyPressed;
```

und das blöde Programm ist nur mit Hilfe von LINE FEED als Repeat-Taste oder auch durch wildes Gehämmere zu überzeugen, daß KeyPressed true ist.

Der Grund für dieses Problem ist der Compiler-Schalter C+, der es ermöglicht, das Programm mit ^C zu unterbrechen. C+ stellt sicher, daß Turbo selbst alles von der Tastatur abholt und nachschaut, ob es denn ein ^C sei. Nach C- verhielt sich dieses Programm deutlich besser.

A s s e m b l e r: DS versus EQU

## DS vs. EQU

(Herbert zur Nedden, 2000)

Häufig benötigt ein Assembler-Programmierer in seinem Programm Datenbereiche, die nicht allzu klein sind, weshalb eine statische Definition mittels Define-Space (DS) nicht in Frage kommt. Benötige ich z.B. folgende Bereiche

1. LesePuffer 1024 Bytes
2. SchreibPuffer 2048 Bytes
3. SpielPlatz 500 Bytes
3. Stack 100 Bytes

so kann ich diese so mit Define-Spaces

```
LesePuffer:    ds 1024
SchreibPuffer: ds 2048
SpielPlatz:    ds 500
StackGrenze:   ds 100
Stack:
```

oder so mit Equates

```
LesePuffer     equ $
SchreibPuffer  equ LesePuffer + 1024
SpielPlatz     equ SchreibPuffer + 2048
StackGrenze    equ SpielPlatz + 500
Stack          equ StackGrenze + 100
```

definieren. Diese seltsam anmutende Definition für den Stack liegt daran, daß der Stack von oben nach unten gefüllt wird, d.h. PUSH HL wird intern in die Befehle DEC SP; DEC SP; LD (SP),HL umgesetzt.

Die erste Variante der Definition der Datenbereiche hat den enormen Vorteil, daß es ein Kinderspiel ist, einen weiteren Datenbereich einzufügen. Die entsprechende Define-Space-Zeile wird einfach an der gewünschten Stelle eingefügt. Dummerweise wird der so mit DS definierte Platz fest zugewiesen, was bedeutet, daß die reservierten Bytes auch im erzeugten .COM zu finden sind, d.h. dieses mit obigen Datenbereichen um schlappe 3672 Bytes zu groß ist.

Die zweite Variante ist die sinnvollere, da lediglich die Adressen der Datenbereiche definiert werden, jedoch der damit angesprochene Platz im .COM nicht drin ist. Aufwendig ist es nur, wenn ein weiterer Datenbereich eingefügt werden muß, da für die Definition einer der Adressen der Label und die Größe des davor liegenden Datenbereiches herangezogen werden. Soll vor den SpielPlatz noch ein 300-Byte-Bereich hin, müssen alle im folgenden Listing unterstrichenen Dinge angepackt werden:

```
LesePuffer     equ $
SchreibPuffer  equ LesePuffer + 1024
Bereich      equ SchreibPuffer + 2048
SpielPlatz    equ Bereich + 300
StackGrenze    equ SpielPlatz + 500
Stack          equ StackGrenze + 100
```

Es stellte sich mir daher die Frage, warum die Assembler keinen Befehl kennen, der von der Syntax her wie Define-Space aussieht, d.h. daß in einer Zeile Name und Länge des betroffenen Datenbereiches definiert wird - ohne andere Stellen ändern zu müssen, jedoch auf der anderen Seite wie Equates wirkt, d.h. mein .COM nicht aufbläht.



A s s e m b l e r: DS versus EQU

Nach dem Motto 'selbst ist der Programmierer' habe ich mir ein Macro gebastelt, was eben dieses leistet:

```

1: Platz          MACRO Lab,Len
2: IFNDef         Off
3: Off           aset  0
4: EndIF
5: Lab           equ   $+Off
6: Off           aset  Off+Len
7:              ENDM

```

Die Ziffern in Spalte 1 sind nicht bestandteil des Macros, sondern sollen helfen, dieses zu beschreiben. Zuvor will ich jedoch aufzeigen, wie ich mit diesem Macro meine o.g. Datenbereiche definieren kann:

```

Platz LesePuffer,1024
Platz SchreibPuffer,2048
Platz SpielPlatz,500
Platz StackGrenze,100
Platz Stack,0

```

Dabei hat das Macro zwei Einschränkungen: Zum einen darfst Du in Deinem Programm die Bezeichner 'Platz' und 'Off' nicht schon verwendet haben, und sie so definierten Datenbereiche müssen hintereinander hinter dem Ende des Programm-Codes liegen, d.h. ihnen darf nichts anderes folgen, auch keine Module aus einer Library - es sei denn das wird gezielt an den Eingenhheiten des Macros orientiert vorgenommen.

Nun zur Beschreibung des Macros:

Zeile 1: Definitions-Kopf mit dem dem Macro sein Name 'Platz' und seine beiden Parameter 'Lab' = Label, d.h. Datenbereichs-Name und 'Len' = Länge, d.h. Datenbereichs-Länge vorgegeben werden.

Zeilen 2-4: Die Variable 'Off' wird, falls sie noch nicht definiert, wurde auf den Starwert von 0 gesetzt. Der Opcode ASET läßt eine spätere Wertänderung zu!

Zeile 5: Der Datenbereichs-Name wird per Equate auf die Adresse '\$+Off' definiert. '\$' ist die aktuelle Programm-Adresse, die durch das Macro 'Platz' auch nicht verändert wird - das war ja auch Sinn & Zweck der Aktion.

Zeile 6: Der Offset wird um die Länge des Datenbereiches erhöht (was mit dem Opcode ASET geht), damit so das evtl. folgende 'Platz'-Equate mit dem richtigen Wert versorgt wird.

Die o.g. Macro-Aufrufe werden also in folgende Equates umgesetzt:

```

LesePuffer      equ   $
SchreibPuffer   equ   $ + 1024
SpielPlatz      equ   $ + 3072
StackGrenze     equ   $ + 3572
Stack           equ   $ + 3672

```

Und just das sollte auch herauskommen!

HINWEIS: Folgende Opcodes haben die selbe Wirkung:

- DEFS und DS
- DEFL und ASET

New Word: Tips**Einrücken**

(Herbert zur Nedden, 2000)

Kürzlich erhielt ich einen Text, der so aussehen sollte:  
(und natürlich auch so gedruckt aussah!)

1. Hier geht der Text los, wobei der Übersichtlichkeit halber natürlich alles außer der '1.' eingerückt sein sollte - und trotzdem Blocksatz angesagt ist, was auch die erste Zeile betrifft.

Bemerkenswerterweise sah das, was NewWord drucken sollte wie folgt aus:

1. Hier geht der Text los, wobei der Übersichtlichkeit halber natürlich alles  
.lm4  
außer der '1.' eingerückt sein sollte - und trotzdem Blocksatz angesagt ist,  
was auch die erste Zeile betrifft.  
.lm0

Was nun, wenn sich die erste Zeile ändert? Nichts als Probleme. Dabei ist New-Word da besser, als der Autor der o.g. Mimik glaubte. Um einen solchen Absatz zu formatieren (also ^B abzuschicken), wird der Absatz erst einmal so eingeben:

1. Hier geht der Text los, wobei der Übersichtlichkeit halber natürlich alles außer der '1.' eingerückt sein sollte - und trotzdem Blocksatz angesagt ist, was auch die erste Zeile betrifft.

Und jetzt geht's los:

Cursor auf das 'H' von 'Hier'

<BS>

*'H' in richtige Spalte packen, falls es wie in diesem Beispiel nicht in Spalte 4 steht.*

^O^I <ESC>

*Tabulator in Spalte 4 (da sind wir jetzt) setzen, falls da nicht schon einer ist*

^O^G

*Linken Rand vorübergehend auf Tab-Position.*

*In der Zeile L---!----- ... R erscheint ein 'V' in der ersten Tab-Position, vermutlich in Spalte 4. Sonst nochmal ^O^G*

^B

*Randausgleich durchführen, und zur großen Freude passiert hierbei just das, was gewünscht war.*

Wird der Absatz geändert, wird der Randausgleich ebenso wiederholt.

Nun nocheinmal die Wirkungsweise von ^O^G im einzelnen:

Mit diesem Kommando wird der linke Rand vorübergehend auf eine Tabulator-Position (in der Zeile L---!---!---!---R durch die Ausrufezeichen '!' gekennzeichnet) gesetzt, was durch ein 'V' an Stelle des entsprechenden '!' in der L---!---!---!---R-Zeile angezeigt wird. Mit jedem ^O^G wandert das 'V' eine Tabulatorposition nach rechts. Im Unterschied zu ^O^L oder dem Punkt-Kommando '.lm' ist der ^O^G-Rand nur vorübergehend gültig, und zwar bis zum nächsten Drücken von <RET>, also dem Beenden eines Absatzes oder zum nächsten ^B, also Randausgleich. Beim ^B werden die Zeichen, die in der Zeile stehen, in der der Cursor bei der Eingabe des ^B vor dem Cursor stehen nicht eingerückt (im o.g. Fall war das die '1.') - der Rest des Absatzes (d.h. bis zur nächsten festen Zeilenschaltung) wird eingerückt formatiert.

Vor jedem ^B muß daher mit ^O^G der linke Rand erneut angepaßt werden, sonst wirkt das ^B halt ohne Einrücken. Soll anders eingerückt werden: Tab entsprechend setzen, ^O^G so oft wie nötig (d.h. bis das 'V' da ist, wo es hin soll), ^B (Cursor vorher an die richtige Stelle) und fertig ist die Lauge.

New Word: Tips**Punkte in Spalte 1 ausgeben**

(Herbert zur Nedden, 2000)

Null-Problemo: Einfach  $\wedge P \wedge B \wedge P \wedge B$  vor den Punkt, der nicht als Punkt-Kommando wirken, sondern in Spalte eins stehend auch gedruckt werden soll eingeben. D.h. vor dem . einfach Fettdruck an/aus (oder falls er an war aus/an).

**Hyphen-Help mit Tücken**

(Herbert zur Nedden, 2000)

Da ändert man ein Bißchen im Text, drückt  $\wedge B$  einfach mal der Optik wegen, wobei natürlich die Trenn-Hilfe (Hyphen-Help) aktiv ist, und der blöde Hund (damit meine ich NewWord) weigert sich statthaft, ein 16 Buchstaben langes Wort, von dem immerhin 10 Zeichen noch in die Zeile passen zur Trennung anzubieten. Das mutete mir doch recht befremdlich an. Also kontrolliert, ob die Trenn-Hilfe an ist:  $\wedge O$  gedrückt und kurz danach erscheint u.a. 'H turn hyphen help off', d.h. die ist an. Naja, um 1:30 nachts merkt man wohl nicht mehr alles: vielleicht hatte ich gleich 2x  $\wedge B$  eingegeben (Prellen oder zitternder Finger?). Frisch auf einen neuen Versuch unternommen - wieder dieser Mist.

Hallo Murphy, Herbert läßt grüßen. Ich gehe also mit dem Cursor ans Ende der Zeile, in der die besagten 10 Zeichen meines Wortes Platz haben und hole das Wort mit  $\wedge G \wedge G$  ( $\wedge G = \langle \text{DEL} \rangle$ ) hoch in die Zeile. Man weiß ja nie, ob NewWord das vielleicht einfach nicht probieren wollte.  $\wedge B$  - und wieder nichts Neues.

Dann kam mir ein - wie sich herausstellte genialer - Gedanke:  $\wedge O \wedge D$ . Damit habe ich erreicht, daß die Print-Controls wie  $\wedge B$  für Fettdruck,  $\wedge S$  für Unterstreichen und eben das Gleichheitszeichen für die erlaubten Trennstellen (die von früheren  $\wedge B$ -Läufen her stammten) angezeigt wurden. Ei gucke da, mein 16-Buchstaben-Wort enthielt einen solchen in der 11. Stelle. Diesen habe ich dort mal weggelöscht - da NewWord da eh nicht trennen sollte. Wieder  $\wedge B$ : Es klappte so wie es sollte.

Nun das Warum, d.h. der Hintergrund einer dSN (dunkle Stunde von NewStar, dem NewWord-Hersteller): für NewWord war das Wort an Stelle 11 quasi zu Ende. Da so aber nur 1 Buchstabe über die Kante hinausragte mochte es wohl keine Trennhilfe anbieten. (Dies habe ich nur geraten ...)

Wer nun glaubt, daß sei die einzige Macke der Hyphen-Help, der/die irrt. Wenn ich bei der Trennhilfe eine Trennung an einer Stelle angebe, wo schon einmal eine angegeben wurde hat das Wort dort gleich zwei 'erlaubte Trennstellen' hintereinander, was sich dadurch zeigt, daß die Zeile mit dem ersten Teil des Wortes ein Zeichen zu kurz ist.

**FAZIT:** Am besten alle 'erlaubten jedoch ungenutzten Trennstellen' entfernen:

$\wedge Q \wedge A$	<i>Suchen und Ersetzen</i>
$\wedge P \wedge \langle \text{RET} \rangle$	<i>suche Ctrl-<math>\wedge</math> (wird als '=' angezeigt)</i>
	<i>Eingabe Ctrl-P Ctrl-<math>\wedge</math></i>
$\langle \text{RET} \rangle$	<i>ersetze durch nichts</i>
gn	<i>überall ohne zu fragen</i>

Die an Zeilenenden stehenden Trennstellen, die durch '-' angezeigt werden, überleben diesen Such/Ersetz-Durchlauf, da sie anders intern codiert sind. Ich stelle jedenfalls sicher, daß die Print-Controls während des  $\wedge B$ -Durchlaufes immer angezeigt werden.

NewWord: Tips**NewWord-Laufwerke und Olaf's DiJey**

(Herbert zur Nedden, 2000)

Ich bin in NewWord, Opening Menü, auf F:, Directory an. Shift-Esc F1 (=DiJey), und kurz mit DiJey die Dokumente von Diskette nach F: gewuppt, 2x Esc, und schon ist das Opening Menü wieder da. Kurz 'LF:<Ret>', um F neu einzuloggen. Hä, wirklich kopiert, wo sind meine Dateien ??? Shift-Esc F1: Ja, alles da. Ägypten ?

Hier erst mal die Therapie, die Erklärung folgt weiter hinten:

Bei der Installation von NewWord gibt es eine Stelle, wo die gültigen Laufwerke angegeben werden können. Das soll dazu dienen, daß NewWord keine BDOS-Errors produziert, wenn man ein ungültiges Laufwerk angibt.

Hier die Installation (Menüs gekürzt, Kommentare in *kursiv*):

A&gt;NWINSTAL NW

*Aufruf NWINSTAL*

## MAIN MENU

C Computer related items

What is your choice? C

*Auswahl treffen*

## SYSTEM INSTALLATION MENU

A Install legal drives

What is your choice? A

*Auswahl treffen*

The drives that Newword can currently use are...

Fixed: H A F G I

Removable: B C D E

Do you want to change this? Y/N Y

*so ist es jetzt noch**aber paßt mir nicht*

Enter the letters (A-P) that are valid drives on your system. The first drive you specify will be used as the default drive. You will also be asked if each drive has a removable disk (as opposed to fixed).

Drive? (A-P, or RETURN when done) H

*1. Laufwerk ist H, da NW dort seine Overlays sucht  
wechselbar*

Is this disk removable? Y/N Y

Drive? (A-P, or RETURN when done) A

*2. Laufwerk ist A  
wechselbar*

Is this disk removable? Y/N Y

*u.s.w.*

Drive? (A-P, or RETURN when done)

The drives that Newword can currently use are...

Fixed:

Removable: H A B C D E F G I

Do you want to change this? Y/N N

*gut**nun NWINSTAL verlassen*

SuperCalc: Patch

Warum habe ich alle Laufwerke als wechselbar (Removable) gekennzeichnet, werdet Ihr jetzt fragen. Nun wie zu sehen, war das ursprünglich anders. Wenn ein Laufwerk als nicht wechselbar (Fixed) gilt, dann wird dieses durch NewWord nach dem Einloggen (L vom Opening Menü oder ^K^L im Text) dem BDOS nicht neu angemeldet, was beim Schreiben zu dem BDOS Err R/O führt, wenn die Diskette doch gewechselt wurde. Also kann ich meine RAM-Floppies wohl gefahrlos als Fixed deklarieren, da diese bekanntlich nicht einfach gewechselt werden.

Das war auch richtig, bis Olaf's DiJey kam.

Zurück zu meinem Phänomen, welches ich oben auf der vorherigen Seite schilderte. Mit DiJey habe ich den Inhalt der Diskette geändert, was, da ich NewWord nicht verlassen habe, es aber auch nicht mit NewWord selbst machte, aus Sicht von NewWord einem Diskettenwechsel gleichkommt. Schließlich ist DiJey kein CP/M-Programm - CP/M ist immerin noch nur Single-Task-Single-User. Seit ich alle Laufwerke mittels NWINSTAL als Removable erklärt habe, ist dies gegessen. Nach DiJey kurz L bzw. ^K^L tut nun so, wie gewünscht.

ACHTUNG: Wenn Du einen Text in Arbeit hast, bloß nicht mit DiJey auf das Laufwerk kopieren, auf dem dieser Text steht, sonst gibt's leicht Ärger. Warum hat Olaf in seinem DiJey.DOC erläutert.

**Printer-Overlays**

(Herbert zur Nedden, 2000)

Die 8/9-Nadel-Matrixdrucker arbeiten beim Vorschub (also auch bei Zeilenhöhen) auf der Basis von 1/72" oder 1/216" (216 = 3\*72, "=Zoll). Wenn nun ein 24-Nadler zum Einsatz kommt, gehen die Probleme los, da diese Drucker mehr für 1/60" bzw. 1/360" zu haben sind. Abhilfe schafft nur ein geeigneter Druckertreiber wie z.B. der LQ1500-Treiber, der für einen 24-Nadel-EPSON-Drucker gedacht ist. Es gibt aber noch weitere.

Auf der CLUB.040 ist in dieser Hinsicht sicherlich reichlich Auswahl zu finden. Da sind immerhin 4 NewWord-Printeroverlays. Welcher Treiber am besten paßt ist eigentlich nur durch probieren zu ermitteln.

**Patch zu Supercalc**

(Uwe Grass, 7000)

Alle Supercalc-Benutzer wissen, daß sie das HELPFILe zu SC immer mitschleppen müssen, wenn sie mit dem Programm arbeiten wollen. Das ist besonders ärgerlich, wenn die Hilfestellungen nicht mehr benötigt werden und das Programm auf die EPROM- oder SRAM-Floppy geschrieben werden soll, zumal das Programm 9K groß ist. Abhilfe schafft ein kleiner operativer Eingriff in SC.COM. An Adresse 5060h steht "HLP", dies muß in "OVL" umgewandelt werden und schon sucht SC beim Aufruf nicht mehr nach den Hilfen. Werden in der Bearbeitung Hilfestellungen angefordert (mit "?"), so erscheint die Meldung "SC.HLP not installed" und man kann danach einfach weiterarbeiten. Wenn die Hilfen dennoch vorhanden sind, so werden sie auch gegeben. Die Abfrage im Aufruf, ob SC.HLP da ist, ist also absolut überflüssig.

UG

**Anm.d.HzN.:** Uwe's Patch bringt SC bei dessen Aufruf an Stelle von SC.HLP sein sowieso erforderliches Overlay SC.OVL zu suchen - was es also auch findet.

CP / M: Daten von einer Diskette holen**Daten und kein Direktory**

(Herbert zur Nedden, 2000)

Ich erhielt eine Diskette mit Daten irgendwo mitten drauf, aber ohne jegliches Direktory. Ja, auf exotischen Systemen, derer es viele unter CP/M gibt, gibt es anscheinend auch mindestens ein beknacktes Programm, welches die Daten ohne Direktory-Einträge an selbstverwaltete Stellen ablegt. Was aber nun, wenn die Daten ohne dieses Programm zwecks Übernahme auf ein anderes System dort runtergeholt werden sollen ?

Naja, COPYD grast bekanntlich die Disketten sektorweise ab. (Kennst Du COPYD nicht ? Siehe Info 27.) Die Lösung meines Problems war durch diesen Ansatz zu lösen: Die Diskette sektorweise lesen, und alle nicht-leeren Sektoren einfach in eine Datei stellen. Das folgende Turbo-Pascal-Programm tut das nämliche.

(Da ich zu faul war viel zu programmieren, müssen dem Programm mitgeteilt werden: Anzahl Spuren der Diskette und Anzahl Sektoren je Spur.)

```

var emptybfr,buffer:          array[0..127] of byte;
    datei:                   file;
    ok:                       boolean;
    i,track,sector,NoOfTracks,NoOfSectors: integer;

begin
  for i:=0 to 127 do emptybfr[i]:=$e5;          { leerer Puffer }
  NoOfTracks:=40;                               { Anzahl Spuren }
  NoOfSectors:=64;                              { Sektoren je Spur }
  Sector:=1; Track:=0;                          { da geht's los }
  assign(datei,'F:Daten');                      { da kommen die Daten hin }
  rewrite(datei);

  repeat
    BDOS(14,1);                                 { Selektiere Laufwerk B: }
    BIOS(11,addr(buffer));                      { Setze DMA-Adresse }
    BIOS(09,Track);                             { Setze Spur }
    BIOS(10,Sector);                            { Setze Sektor }
    writeln(track,' ',sector);                  { Meldung }
    if BIOS(12) <> 0 then                       { Sektor Lesen }
      begin write('Lesefehler'); halt; end;     { Abbruch bei Problemen }
    ok:=false;                                  { Glaube Puffer-Leer }
    for i:=0 to 127 do
      if buffer[i] <> emptybfr[i] then         { Puffer wirklich leer }
        ok:=true;                             { Setze Puffer-Nichtleer }
    if ok then
      blockwrite(datei,buffer,1,i);            { Datei fortschreiben }
      Sector:=succ(Sector);                    { Naechster Sektor }
      if Sector = NoOfSectors then             { Ende der Spur ? }
        begin
          Sector:=1; Track:=succ(Track);
        end;
      if Track > NoOfTracks then                { Letzte Spur }
        begin
          close(datei);                        { Datei schliessen }
          write('Fertig'); halt;               { Zurueck zum CP/M }
        end;
    until false;                               { Loop forever }

end.
```

S o f t w a r e: Dieses und Jenes

**BRADFORD**

(Herbert Oppmann, 8520)

Bradford (Version 1.20, CLUB 027) hat einen Fehler: die Soft-Hyphens von Word-Star bzw. NewWord werden immer unterdrückt, auch wenn sie am Zeilenende stehen!

**MS**

(Herbert Oppmann, 8520)

In Info 30, Seite 22 schreibt Hans Gras, daß er sich ein verbessertes MS.COM wünscht. Das tu ich auch schon lange. Da das Original-MS von Bernd Preusing stammt, und dieser aber daran nichts mehr tut, biete ich mich hiermit an, MS zu debuggen und zu erweitern, sofern ich die Source der jetzigen MS-Version kriege, denn ich möchte nicht unbedingt wieder bei Null anfangen müssen.

**TDIFF**

(Herbert Oppmann, 8520)

Ich habe ein Turbo Pascal Programm geschrieben, mit dem man Unterschiede in Textdateien herausfinden kann. Wenn man DIFF auf Textdateien losläßt, meckert das die ganze Datei an, auch wenn nur ein einziger Buchstabe am Anfang eingefügt wurde. Mein TDIFF schaltet auf breiten Bildschirm (wo möglich) und zeigt beide Dateien parallel nebeneinander an. Man kann dann eine Zeile oder einen Bildschirm scrollen sowie nur einseitig scrollen (falls in den Textdateien Weglassungen/Einschübe vorgenommen wurden).

--> CLUB.042

**RD**

(Herbert Oppmann, 8520)

Mein READTRK Programm, auf das Hans Gras in Info 29 Seite 21 ein Loblied sang, war nicht so toll:

ein kapitaler Fehler zerschoß manchmal das System --> Absturz!

Als ich den Fehler jetzt bemerkte und beseitigte, hab ich das Programm auch gleich noch ein wenig aufgeblasen und das Ergebnis liegt bei HZN vor (wird wohl auf eine CLUB-Diskette kommen).

--> CLUB.043

(oder wenn Du's eilig hast, von mir, da CLUB.043 noch nicht fertig ist).

**Übrigens Hans - zu meiner Ehrenrettung:**

(Herbert Oppmann, 8520)

Die Angabe 5" statt 8" im Info 26 ist in meinem Brief, den ich an HZN schickte, richtig (zumindest auf meinem Kontrollausdruck)! Ich weiß nicht, wer hier "korrigierte".

d B A S E: SuperBusiness / A s s e m b l e r: SLR-Systems

### Super Business

(Herbert zur Nedden, 2000)

Dirk Leßner hat eine dBASE II-Anwendung unter dem Namen SB = Super Business erstellt, die er wie folgt anbot:

"Ich hätte da noch eine Anwendung in dBASE II anzubieten. Diese, - in zweijähriger mühevoller Kleinarbeit geschriebenen und optimierten Programme für den Kleinbetrieb -, enthalten folgende Funktionen:

- komfortable Adreßverwaltung
- Rechnungsausgangsbuch
- Wareneingangsbuch
- Kassenbuch
- Mahnwesen
- diverse Statistik- und Übersichtsfunktionen"

Benötigt wird eine FDX mit zwei Laufwerken - ich vermute eine SDX tut es auch. Ursprünglich wollte Dirk diese Anwendung für DM 40.- verkaufen, hat sich dann jedoch dazu entschlossen, sie als Public-Domain freizugeben, da er nicht garantieren kann, daß nicht der eine oder andere Bug (z.B. DIV/0-Fehler bei Leerer Datenbank) noch drin ist, und er keine Zeit hat, eine Dokumentation zu schreiben.

Ich rief das Teil mit

A>DBASE SB

auf, und fand ein Menü mit u.a. dem Punkt 5 (Info) vor. Dank der Menüführung wird es sicherlich nicht schwer sein, sich in dieser Anwendung zurechtzufinden.

Falls Du SB wirklich zufrieden arbeiten kannst, und es erfolgreich nutzt, kannst Du ja evtl. Dirk Leßner, Zum Frankenholtz 20, 3525 Obw.-Heisebeck, ein kleines Dankeschön ...

### SLR-Assembler und Linker

(Herbert zur Nedden, 2000)

Die amerikanische Firma SLR-Systems hat einen Assembler und einen Linker auf den Markt gebracht, der mehr als ein Ersatz für das Microsoft-Gespann M80/L80 ist.

Wer kennt es nicht: L80 aufgerufen, und nach einiger Zeit ist der Linker auch mit seiner Arbeit fertig. Nicht so mit dem SLR-Linker. Aufgerufen - fertig. Der Linker ist UNGLAUBLICH schnell. Der Assembler natürlich auch.

Hier kurz ein Steckbrief:

#### SLR-Assembler: SLR180

- Kann alles was der M80 kann, außer 8080-Sources, dafür aber HD64180
- Erzeugt wahlweise .COM, .HEX, Microsoft-.REL, SLR-.REL
- Kann mittels .ACCEPT Werte interaktiv einlesen
- geschachtelte INCLUDES und IF's
- Kann mit EXTERNALS rechnen
- Cross-Reference und Symbol-Tabelle
- Labels mit 16 signifikanten Zeichen
- Datum/Uhrzeit im Listing
- Assembler-Befehle können durch Macros redefiniert werden
- Sucht INCLUDES auf anderen User-Bereichen
- Unterstützt ZCPR3-Error-Flag bzw. bricht .SUB-Datei ab
- Umfangreich installierbar
- Schafft fast alle Sources mit EINEM Pass
- Initialisiert DS-Bereiche mit 00h oder Offh bei .COM



d B A S E: SuperBusiness / A s s e m b l e r: SLR-Systems

SLR-Linker: SLRNK

- Kann alles was L80 kann
- Kann Microsoft- und SLR-.REL's linken
- Erzeugt wahlweise .COM oder .HEX
- Erzeugt Lade-Übersicht (Load-Map)
- Kann Befehle aus Datei einlesen
- Erzeugt Symbol-Tabelle
- Kann mit EXTERNALs rechnen (bei SLR-.REL)
- Unterstützt ZCPR3-Error-Flag bzw. bricht .SUB-Datei ab
- Initialisiert DS-Bereich mit 00h oder 0ffh
- External-Labels mit 16 signifikanten Zeichen
- Erzählt gerne genauer, was er tut (Verbose-Mode)
- Hat Librarian dabei

Beide sind SUUUUUPERRR-Schnell!!!

Wo ist der Haken ? NIRGENDS!

Zu kaufen ist das Teil am besten direkt aus den U.S.A., da das richtig Kohle spart. Folgendes habe ich getan:

1. Bei der Deutschen Bundespost eine Auslandsüberweisung an 'SLR-Systems, Butler, PA 16001, U.S.A' über 116.40 US-Dollar angefordert, mit der Bitte, den Scheck an mich zu senden (da die Auslandsüberweisungen der DBP per Scheck erfolgen).

2. Bestellung an SLR-Systems

Quantity	ITEM-No.	Description	Price
1	103	SLR180-Assembler	49.95
1	101	SLRNK-Linker	49.95
		Both on 5 1/4"-Disc	
		Format KAYPRO SS/DD or DS/DD	
1	SHIPPING	UPS Red Label	16.50
			=====
			116.40

Diese Bestellung mit diesem Scheck an

SLR Systems  
 1622, N. Main Street  
 Butler, PA 16001  
 U.S.A.

Ich erhielt das Zeugs per Luftpost!

- Wartezeiten: 1. 7 Tage
- 2. 1 Monat

Alternativer Bestell-Weg:

Ich bin bereit, auf einen Schlag die Bestellung für mehrere Interessenten in die USA zu senden, was sicherlich einiges an Porto sparen wird. Daher folgendes Angebot meinerseits:

Schick mir bis zum 15. August Deine Bestellung sowie das Geld als Scheck. Ich löse den Scheck erst dann ein, wenn's ernst wird.

S o f t w a r e: CFIG5 und FORM5

**cfig5 von Michael Kessler**

(Uwe Grass, 7000)

Die Idee von Michael finde ich sehr gut. Er baut in eine leere Speicherstelle des Systems, die sonst nie benutzt wird, ein Kennbyte ein, das das Format identifiziert. Mit dem Aufruf "cfig5 b:?" wird das Laufwerk auf das Format der eingelegten Diskette umcfig5t. Die Überlegung, welches Format nun die Diskette hat, entfällt. Soweit so gut, was ist aber, wenn das Kennbyte nicht im System ist? Dann soll das Programm nach der Formatstruktur auf der Diskette auch noch das richtige Format finden, natürlich kann das nicht immer 100%ig richtig sein, dazu sind sich die Formate teilweise zu ähnlich.

Da das Programm PD ist, habe inzwischen auch eine Version davon. Wenn ich eine Diskette mit Kennbyte benutze funktioniert es auch ganz ordentlich, lege ich aber eine der Original-MTX-Format-Disketten ein, die noch nicht vorbereitet wurde, so wird beim ersten cfig5-Versuch das Laufwerk lediglich auscfig5t, beim zweiten oder dritten Versuch klappt es dann, das richtige Format ist dann drin. Warum ist das so? Liegt das an der Geschwindigkeit der Laufwerke (TEAC 3ms, 6MHz)?

Zweite Frage: Wie werden Formate erkannt, die eine Geschwindigkeitsumschaltung benötigen (77-Spur), wird die Geschwindigkeit bei umgerüsteter FDX-Platine automatisch umgeschaltet?

UG

**Anm.d.HzN.:** Zur zweiten Frage: JA, da bei mir die Erkennung von z.B. 0A und 1B richtig automatisch funktioniert.

**form5.com/klx**

(Jan Bredereke, 2000)

Form5 schreibt in den ersten Sektor ein JP 0 und eine Formatkennung. Dies macht mir einige Probleme. Ich habe ein eigenes Diskettenformat FB entworfen und damals auch bei BP angemeldet. Es sieht genauso aus 09 wie und hat nur keine Systemspuren, denn die brauche ich unter RAM ja definitiv nicht. Booten tue ich ja von der SRAM-Floppy. Da ich ausschließlich Format FB verwende, besteht auch keine Verwechslungsgefahr.

Daß mir jemand die Format-Nummer geklaut hat, stört mich ja nicht besonders, aber zum Formatieren muß ich auf Format 09 wechseln, und wenn dann in den ersten Direktory-Eintrag des Format FB unverdauliche Daten geschrieben werden, ist das nicht so schön. Da muß ich bei FORM4.COM bleiben.

**cfig5 und form5 von Michael Kessler**

(Herbert zur Nedden, 2000)

Wie Uwe bemerkt, hat Michael Kessler uns hier wirklich eine schöne Sache an die Hand gegeben. Und seine Formaterkennung bei disketten ohne Format-ID ist auch noch schneller als meine mit CFIG43.COM.

Leider muß ich Euch vor dem Einsatz von CFIG5.KLX warnen! Wenn ich als Formatangabe B:??,C:?? eingebe, erscheint nur 'B:??C:??', also gibt's kein Komma. Nehme ich das Blank statt des Kommas, so meint CFIG5, daß es das Format bei B nicht in RAM 4 vorfindet. Bei anschließendem Aufruf stand mein Name in der Form 'RBERT ZUR NEDDEN' in der Eingabezeile. Anscheinend hat Michael da den Zeileneditor selbst verbastelt ...

Und dann kommt noch ein grundsätzliches Problem hinzu, welches die CP/M-Version von CFIG5 und natürlich auch CFIG4 dadurch lösen, daß sie mit einem Warm-Boot enden, und so dem BDOS klar machen, daß alle Laufwerke neu zu selektieren sind.

S o f t w a r e: CFIG5 und FORM5

Nun zur Begründung: Das BDOS benötigt für jedes Laufwerk die sogenannte Allocation-Map, bei der es sich um einen Speicherbereich handelt, in dem für jeden Block auf der Diskette ein Bit reserviert ist. Wenn das BDOS ein Laufwerk neu einloggt, liest es das Directory ein und ermittelt daraus, welche Blöcke auf der Diskette durch Dateien belegt sind. Diese Information speichert es sich in der Allocation-Map, damit es beim Anlegen neuer Dateien nicht erst auf der Diskette nach freiem Platz suchen muß - sonst wären die 0A-Laufwerke vermutlich langsamer als die 03-er!

Da dieser Platz von der Größe der Diskette - genauer von der Anzahl der Blöcke auf eben dieser abhängt, sich also mit jedem umCONFIGurieren ändern kann, wird dieser Platz auch je nach bedarf reserviert, und zwar am unteren Ende des Bereichs zwischen FREE und TOAM, d.h. in Richtung FREE.

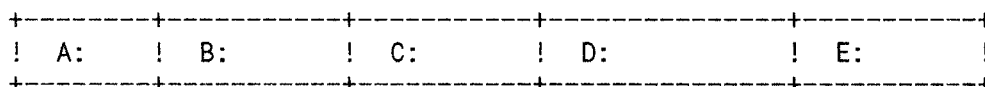
Und die Reservierung dieses Speichers macht nicht etwa das BDOS und (noch nicht) das BIOS, sondern CFIG.

CFIG macht beim konfigurieren eines neuen Laufwerkes folgendes:

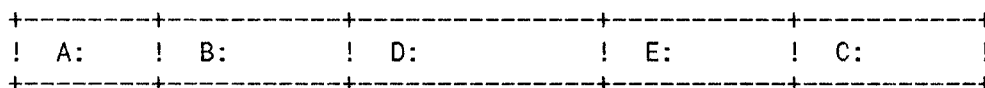
AllocMap-Adresse = FREE  
 FREE = FREE + AllocMap-Länge.

So weit so gut. Was aber, wenn ein Laufwerk umconfiguriert wird, und die neue AllocMap länger oder kürzer als die alte ist. Dann müssen die AllocMaps halt ggf. verschoben werden, um nicht unnötig Platz, der eh schon knapp ist, zu verschwenden.

Gehen wir von folgender Aufteilung der AllocMaps aus:



Wird C: umconfiguriert, wird erst der Platz von C: freigegeben, d.h. die AllocMaps von D: und E: werden an die von B: herangeschoben, und anschließend die neue von C: hinten angebäpft. Damit erhalten wir:



Nur leider ist CFIG **faul**!!! Statt in diesem Fall die AllocMaps von D: und E: wirklich nach vorne zu verschieben, werden nur die Adressen der AllocMaps im jeweiligen Disc-Parameter-Header (auch DPH genannt) neu eingetragen. D.h. es steht nach dieser Änderung Müll in den AllocMaps von D: und E: - und natürlich auch von C:!. Da das CP/M-CFIG immer mit einem WarmBoot endet, stellt es sicher, daß das BDOS die Laufwerke neu einloggt, wenn auf sie zugegriffen wird, d.h. die AllocMaps allesamt für ungültig erklärt.

CFIG5.KLX ist zwar genauso faul, macht aber (leider ??) keinen WarmBoot. Auch ein DiskReset darf es nicht an das BDOS absetzen, da es sonst evtl. laufenden Programmen ins Handwerk pfuscht! Es muß definitiv diese AllocMaps samt Inhalt verschieben!!

Für Insider und Interessierte: Zu jeder AllocMap wird auch noch eine CheckSum Map auf die selbe Art und Weise reserviert, die das BDOS benützt, um seinen BDOS Err R/O zu erzeugen, d.h. unangemeldete Diskettenwechsel zu erkennen, die daher genauso zu verschieben ist.

Software: E45

---

 Peter Lang, Banderbacher Str.24, 8502 Zirndorf
 

---

E 45

Pflanzenschutzmittel im Info ?

---

Wer mault kriegt was auf die ...

Na, na! Auf die Diskette natürlich!! Um dem Rätselraten ein Ende zu bereiten, will ich hier flugs erklärend einschreiten.

Im letzten Info hat sich einer (war das ich???) über das leider etwas umständliche Handling von "NOTE" ausgelassen. Prompt liegt ein paar Tage später eine Diskette im Briefkasten - Absender HzN (wer ist denn das nu schon wieder??). Im üblichen "DOC" der Hinweis, daß es sich bei dem gelieferten Programm um soetwas wie ein neues Byte-Verbrechen eben jenes Herrn handelt. Und das ganze verbunden mit der Bitte um einen eingehenden Test (soll er haben, hä hä).

Ganz schnell mal die restlichen "DOC's" überflogen - bäh, der Name klingt ja wie Umweltverseuchungsmittel - und dank meines überdurchschnittlichen EiQu festgestellt: Ein neuer Editor für's Klick! Na Super. Gleich mal laden und - denkste! Fängt das Ding doch an zu meckern und fragt mich via Bernsteinglotze ob das mit dem nur 1k freien Platz im Heap '..wohl ein Witz sein..' soll! Na sauber, geht ja mal wieder gut los.

Um nicht die ganze Story meiner Probierereien darzulegen - wozu wurden eigentlich "DOC's" geschrieben, wenn der Peter zu faul zum lesen ist? - komme ich gleich zu dem, was dabei herauskam (was für ein Ekel-Deutsch - Pfui).

Bei "E45" handelt es sich um die Klick-fähig gemachte und modifizierte Version des guten alten "MTX-Edit" des Herrn HzN. Wie nicht anders zu erwarten war, sind Klick-spezifische Einstellungen (Speicher, F-Taste etc.) mit dem neuen "KI" (Klick-Install) installierbar. Als Zugabe gibt's zu "E45" auch noch ein "HELP" als Klicker. Keine schlechte Idee. Falls man nicht (wegen fehlender Penunze o.ä.) mit RAM-Platz geizen muß, sollte der zur Verfügung gestellte RAM-Bereich so groß wie möglich sein (ab etwa 8k). Nur so kann "MTX-Edit" seine universellen Fähigkeiten als Allround-Editor voll entfalten. Obwohl hauptsächlich als Non-Doc-Editor für ".MAC" konzipiert, sollten auch 'Nichtassemblerer' dieses Ding nützen. Schriftstücke (in Verbindung mit z.B. "BRADFORD"), Tabellen, Help-Files, Notizzettel usw. lassen sich mit "E45" genausogut / besser erstellen wie mit Newword (Non-Doc). Besser, weil "E45" nicht erst wiederholt geladen werden muß. Besser, weil "E45" so verteufelt schnell ist! Bei Drive A, einem ca.120-Zeilen-Text versuchte ich anfangs mehrmals fluchend per '^K^S' mein Werk zu sichern - weil ich nicht schnallte, daß das kurze Bildschirmzucken ein komplettes Absaven inclusive Anlegen eines '.BAK'-Files beinhaltet! Oh Mann, das nenne ich Tempo. Besser auch, weil "E45" (außer automatischem Zeilenumbruch und Hyphen-Help) sogar noch mehr Nützlichkeiten bietet als NW. Ohne Anspruch auf Vollständigkeit hier ein paar davon:

Wechsel der Glotzen-Formate per '^01/^02'. Zusätzlich mit '^0^F' (per Dialog!) Eingabe eines beliebigen Formates! Einstellungen bleiben auch bei wiederholtem Aufruf erhalten. Mit '^K0 - ^K9' kann der geneigte Tastentompteur 'Marken' in seinen Text setzen, die er via '^Q0 - ^Q9' jederzeit wieder anhopsen kann. Selbstmurmeln beherrscht "E45" alle Cursor-, Block-, Lösch- und Such/Ersetz-Optionen. '^OS,^OA,^OG' sind ständig verfügbar, um den Text in Standard-Alternate- oder Grafikmode ('PIP' liest sich wie: 'Herz-Pfeil-Herz' !!!) anzuzeigen. Zum Erstellen von Glotzen-Masken ist per '^K^T' eine spezielle 'Type-Funktion' anwählbar, die den spezifizierten Text mit allen eingegebenen Ctrl-Codes und Attributen richtig anzeigt (Rahmen, Blockgrafik, Hell/Dunkel etc.).

Software: E45

---

Peter Lang, Banderbacher Str.24, 8502 Zirndorf

---

E 45

Des Dramas 2. Seite

---

Fast alles was ein/ausgeschaltet oder alternativ verändert werden kann, wird in der Statuszeile angezeigt, z.B. Datei, Zeile, Spalte, freier sowie belegter Speicherplatz, Wildcards ein/aus etc. etc. ...

Aber ...

--> Drei Wochen später ... <--

An dieser Stelle hatte ich die Höflichkeit besessen, wegen der Ausgewogenheit einige Kleinigkeiten aufzuführen, die mir an "E45" nicht ganz 'schmeckten', bzw. die ich nach einigen Stunden Arbeit mit "E45" irgendwie vermisste. Also wiegesagt hier stand Negativ-Kritik ! (Nu - so wild wars ooch wieder nich !)

Zu meckern gibts nix

Mittlerweile habe ich eine Update-Version des Werkes erhalten! Und das ca. 3 Wochen nach meinem Gemotze - das nenne ich wirklich Leistung! Vor allem, weil restlos Alles was ich irgendwie bekritelte (sorry Herbert, vielleicht war ich wohl ein bisserl ekelig !?) behoben, verbessert, erweitert und was weiß ich alles wurde. Selbst Phantastereien (meinerseits) wurden verwirklicht. Wie nennt man denn sowas? Kundendienst? Ehrgeiz? Oder wie? (jetzt kommt garantiert wieder ein: Anm.d.HzN !!) Aber warum so bescheiden? Hier ein kurzer (!) Auszug aus den Sachen die neu/behoben sind:

MTX-Edit schaltet zur Arbeit auf F-Tabelle 2 (Newword) und nach dem Verlassen wieder (hurra!!) auf die vorher aktive. "^KD" ändert nicht nur das Bezugslaufwerk, sondern jetzt auch noch den Userbereich (!!!) im Dateinamen. Und jetzt kommts: Alles was ein-, aus- oder umgeschaltet werden kann ist jetzt installierbar. Per Non-Doc-Patch-Datei (ThePatch). Das heißt: Ich kann mir meine Lieblings-defaults vorher fest installieren! Ändern geht ja dann wieder per Ctrl-Kommando. Auch die Dokumentation ist erweitert. Nicht nur quantitativ sondern hauptsächlich qualitativ! Genau richtig für so Hirnis wie mich. Jetzt kapieren selbst Leuchten wie meine Vielseitigkeit, worum es geht!

Usw. usw. ...

Na Bravo! Dieser HzN hat es wirklich fertiggebracht, all meine schönen Kniefieseligkeiten zu entkräften.

Kurz und Gut:

Wir haben endlich einen wirklich voll zufriedenstellenden, robusten und bedienungsfreundlichen Non-Doc-Editor für's Klick unseres Silikon-Trottels! Da gibt's nix zu meckern!

Na denn: 'Bess demnäx ...'

Leserbrief: Hans Gras, NL 1506

**Tip für MBASIC user**

Es gibt beim Compiler (BASCOM) die Möglichkeit um ein RUN-Time Library zu benutzen (BRUN.COM). Dies ist ein schöne Lösung für Disk Platz Problemen, oder "out of space errors" beim linken (mit L80).

Aber benützer von ZCPR 2 haben dann Schwierigkeiten um diese Programmen über ein Suchpfad ans laufen zu kriegen. Das Compilietes Basic Programm such dann BRUN.COM aufs eingelogte Laufwerk. Und meistens steht BRUN.COM auf eine andere Diskette.

Die Lösung ist Einfach. Das Programm met DDT, MONI, PATCH o.ä. Patchen. Ins letzte Sector (ist 100hex byte) des programms steht mehrere malen die Tekst "BRUN" usw. (siehe SchirmDump "ETIKET.DMP" auf diese Seite).

FileName: E:ETIKET.COM (Eof=134) Record: 133/134 Base= 0100 18:14:50

Address	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	0123456789ABCDEF
4380	8F	9A	DA	3A	90	21	80	FF	19	C3	0F	90	11	45	90	C3	.....!.....E..
4390	3D	90	11	5A	90	0E	09	CD	05	00	C3	00	00	42	52	55	=..Z.....BRU
43A0	4E	2E	43	4F	4D	20	6E	6F	74	20	66	6F	75	6E	64	0D	N.COM not found.
43B0	0A	24	42	52	55	4E	2E	43	4F	4D	20	74	6F	6F	20	6C	.\$BRUN.COM too l
43C0	61	72	67	65	0D	0A	24	**	42	52	55	4E	20	20	20	20	arge..\$_BRUN
43D0	43	4F	4D	00	00	00	00	00	00	00	00	00	00	00	00	00	COM.....
43E0	00	00	00	00	00	00	00	00	00	00	00	00	3A	D1	02	FE	1E
43F0	C2	C9	90	21	28	4E	22	D2	02	2A	E6	8F	22	D4	02	2A	....!(N"...*..."*!
4400	94	8F	22	D6	02	E5	EB	CD	EA	04	0A	19	22	D8	02	D1	..."....."
4410	EB	CD	54	03	21	FC	FF	39	22	DA	02	22	DC	02	C3	83	..T.!...9"..."....
4420	19	CD	44	1A	56	65	72	73	69	6F	6E	20	6D	69	73	6D	..D.Version mism
4430	61	74	63	68	0D	8A	C3	15	01	4D	20	74	6F	6F	20	6C	atch.....M too l
4440	61	72	67	65	0D	0A	24	00	42	52	55	4E	20	20	20	20	arge..\$.BRUN
4450	43	4F	4D	00	00	00	00	00	00	00	00	00	00	00	00	00	COM.....
4460	00	00	00	00	00	00	00	00	00	00	00	00	3A	D1	02	FE	1E
4470	C2	C9	90	21	28	4E	22	D2	02	2A	E6	8F	22	D4	02	2A	....!(N"...*..."*!

Display Record Base OverLay Ucase Search Cont Address Edit +/- Quit

Wir brauchen nur das Byte aufs angegebene Stelle (\*\*) zu ändern: Für Disk A: eine 01, Disk E: eine 05 und fur I: eine 09 usw. Das alte Byte (00hex) bedeutet aufs eingelogte Laufwerk suchen. Ans Ende dieses File steht es noch ein mal. Dies ist ein Unschönheit des Linkers (L80), und hat kein Bedeutung.

Um es zu erleichteren habe ich auch ein einfachere Weise gedacht. Mit Hilfe von BRUNZOEK.COM (CLUB.???) kan es auch gepatch werden. Einfach BRUNZOEK.COM starten, Name des zu Patchen File und das Laufwerk fur BRUN angeben. Als Versuch File ist BRUNTEST.COM dabei, diese sucht BRUN.COM.

Viel Erfolg!

Ihr MBASIC Bastler Hans Gras

Es gibt auch eine neue SuperCalc. Dies heißt Supercalc2. Ich habe Version 1.00. Dies war ein sonder Ausführung für eine Osborne Computer. Ich habe mir viel mühe gemacht om es zu Patchen. Es läuft auch mit RAM 4.3. Mit unterstützung verschiedene Pactches aus (alte) Infos. Eine Version ohne RAM 4.3 patches ist auch möglich. Nur bei ausreichende Fragen. Selbstverständlich auch mit alle HilfsProgrammen, wie beim alte SuperCalc (1), Version 1.12.

Leserbrief: Claudio Romanazzi, 3070

### Z C P R 3 - Manuals

Ich habe vor um alle Manuals aus zu Drücken auf einer Canon LBP 8A1 oder einer NEC P6+ printer, nach einer Bearbeitung mit WS 4.0, um Papier zu sparen (original etwas von 400 Seiten?).

Aber wann du es bereits gemacht habe is es für mich nicht intressant. Copieren oder ins Info ist einfacher!

**Anm.d.HzN.:** Habe ich noch nicht!

Bitte beim nächstes mal angeben.

Ihre Holländische Bastler  
Hans Gras

**Anm.d.HzN.:** Hans Gras versucht gerade, eine Festplatte zum laufen zu bekommen.

**Anm.d.HzN.:** Hannes Mehrstens hat es schon geschafft.

Leserbrief: Claudio Romanazzi, 3070

Dear Herbert,

anbei auch einmal ein Leserbrief von mir:

Auch ich möchte mal einen Kommentar abgeben. Alle Infos sind bisher sehr gut gelungen. Es ist (für mich!) eine gute Mischung zwischen Sachen, die mir bekannt und anderen, die mir eben neu sind. Dazwischen eben immer mal ein Leckerbissen, von dem die Konkurrenz (MSDOS, etc.) nur träumt, oder nur mal nebenbei ein paar Märker mehr hinblättern muß. Auch bei den Sachen, die mir schon bekannt sind, ist immer mal wieder etwas dabei, das einer Auffrischung bedarf und dessen Kenntnis mir gerade jetzt (seltsam!) von Nutzen ist.

Jetzt zum perfekten Programmierer. Lieber Olaf, du kannst's. Doch wer sich in z. B. in OK-Menu einarbeiten soll, der guckt doch lieber in die vorhandenen Esc-Sequenzen und schreibt seinen Window-Bedarf selbst. Wer in Assembler programmiert, erstellt sich seine eigene Syslib und schaut, soweit ein Source vorhanden, eben nur mal in deine rein, um evtl. was abzukupfern. So ist es jedenfalls mir ergangen. Vielleicht bin ich eine Ausnahme oder besonders eigensinnig, doch du wolltest ja wissen, wie es anderen mit deinen Schöpfungen ergeht.

Zum Thema Dokumentation. Olaf's gehen ja. Aber, was sonst noch so kreucht und fleucht hat mich schon viel Papier gekostet. Vor allem Herbert tut sich mit Zeilenüberlängen so sehr hervor, daß mein Drucker schon eine neue Zeile anfängt, der Zeilenzähler von Newword aber das gar nicht mitbekommt. Das Ergebnis: Eine Seitennummer am Kopf der nächsten Seite und ein Seitenvorschub. Bis auf besagte Nummer ist das Blatt dann leer. Gratulation. Bitte, schreibt doch wenigstens als Punkt-Punkt-Kommando an den Anfang einer Dokumentation **euer** Format hin, so daß man sich einrichten kann und nicht voller Vertrauen mehr Papier verbraucht als nötig!

Die I's in den Adressenlisten lassen vermuten, daß der Club sich weiterhin gesundschrumpft. Nur noch die wirklich interessierten werden weiterhin **ihrer** schwarze Kiste treu bleiben und nicht in den Club der Konsumenten eintreten, der durch ein paar Scheinchen das Programmieren ersetzt. Ich hoffe nur, daß die 'neuen' dann mehr zu bieten haben werden.

Leserbrief: Claudio Romanazzi, 3070

Ich biete euch jetzt meinen Wortmanager an. Das ist ein Programm zur Rechtschreibung. Näheres entnehmt bitte der weiter hinten abgedruckten Einleitung der Dokumentation dazu. Erst wollte ich das Programm in den PD-Pool geben, doch dann habe ich beschlossen, die Kontrolle zu behalten, um die Bibliothek zu pflegen und um das unten angesprochene Textprogramm zu verwirklichen. Das geht nicht, wenn einjeder im Source herumpfuscht. Bitte habt Verständnis. Der Preis ist jedoch fast der gleiche, wie der einer PD-Diskette: DM 10.- incl. Porto/Verpackung, incl. Update der Bibliothek im Oktober bei Einsendung der eigenen. Ich war schon immer gegen die hohen Preise für Software, die sowieso nur im Club läuft und hier will ich ein Zeichen setzen. Auch Programme, die wirklich viel Arbeit gemacht haben, können für jedermann erschwinglich sein, wenn nur ein einigermaßen guter Wille dahinter steckt! Geliefert werden immerhin ca. 240Kb. Davon nimmt allein die Bibliothek knapp 190 Kb ein.

Zu Manfreds Leserbrief. Das war konstruktive Kritik! Ich bin genau deiner Meinung. Nur eins hast du vergessen. Ich gebe ein Beispiel: Bezogen auf Info 30 möchte ich nämlich wissen, was Wahrheitstafeln/KV-Diagramme sind. Also manchmal wirds ein bischen hoch für unsereinen und dann besteht die Gefahr, daß der vielleicht gute Artikel überlesen wird, wie hier geschehen, weil 0 Durchblick. Es existiert eine Zeile Dokumentation, die nichts erhellt.

Jetzt zum Thema Festplatte. Immer wieder tauchte mal im Info die Frage nach einer Festplatte auf. Da ich auch interessiert war buchte ich in c't 4/89 eine Anzeige, in der ich gebrauchte ECB-Adapter für einen OMTI-Controller, bzw. die reine c't-ECB-Lösung suchte. Von je einer Lösung bekam ich daraufhin ein Exemplar. Inzwischen läuft bei mir die OMTI-Lösung am ECB-Bus. Die Kapazität beträgt knapp 21 MB (**Megabyte**). Die Blockgröße beträgt z.Zt. 4 KB, die Directory-Größe ist 1024 Einträge. Die Software dazu besteht aus einem Klick-Overlay, welches die Read-Write-Routinen des G-Laufwerks verzweigt. Im Klick-Directory ist der Titel HDTreiber eingetragen. Bei Aufruf durch Funktionstaste parkt die Festplatte und man kann reinen Gewissens ausschalten. Für die Zukunft ist eine Partitionierung in Blöcken von ca. 6MB geplant, damit der TOAM nicht mit dem Allocationvector zugestellt wird (TOAM ist der Platz zwischen BIOS und Systemtabellen, Allocationvector ist eine Tabelle, die dem BDOS anzeigt, welche Blöcke auf der gerade eingeloggten Diskette frei/belegt sind). Auf einer Festplatte wären dann mehrere logische Laufwerke, die durch CFG anzusprechen sind. Dieses ist aber Zukunftsmusik. Ich hoffe, daß ich bis zu meinem Urlaub ab Mitte Juli alles fertig habe. Die Software wird (incl. Source) auf der Klick-PD 007 zu finden sein. Alles jubelt und fragt 'quanta costa?'. Zur Zeit kostet ein Set, bestehend aus Controller unbekannter Genese (Hersteller) & Seagate ST225 ca. DM 550.- fertig anschließbar an einen Kompatiblen. Für meine Lösung benötigt man einen OMTI-Controller (5510 oder 5520, beide so um DM 200.-). Dazu käme dann eine Festplatte eigener Wahl. Dazu kommt dann noch ein Adapter, der zwischen ECB-Bus und Bus des OMTI-Controllers gebastelt werden muß. Z. Zt. habe ich zwei Controller. Meine lauffähige Version würde ich bei Interess im Club für DM 700.- incl. alles verkaufen. Alles klar? Für nähere Information kann man mich gewöhnlich immer anrufen, oder wer sich für Einzelheiten interessiert in c't 4 + 9 1987 nachlesen.

**Anm.d.HzN.:** Claudios Platte läuft gut, wie auf dem Clubtreffen zu sehen war!

Was mich noch interesieren würde:

Lieber Thomas Wulf, bringe doch noch ein paar Erfahrungen und **Bilder** deiner Forschungen mit Apfelmännchen und dergleichen.

Wer hat sich schon mal intensiever mit Forth beschäftigt? Bitte bei Gelegenheit bei mir melden!

Es grüßt euch alle Claudio Romanazzi



Software: Wortmanager**Wortmanager**

(Claudio Romanazzi, 3070)

Jetzt folgt der Vorspann der Dokumentation des Wortmanagers:

Einleitung:

Im Februar '88 erzählte mir Horst Kupka von einem Programm, welches die Rechtschreibung von Texten überprüft. Dazu lieferte er mir ein paar Gedanken und Algorithmen. Für unseren MTX existierte ein solches Programm noch nicht. Mich interessierte die Problematik und ich machte mich an die Arbeit. Zu erst mußte ein geeigneter Algorithmus zur Erstellung einer Bibliothek gefunden werden, danach einer, der das Einsortieren neuer Worte ermöglicht. Weiterhin brauchte ich einen Wortfinder, der Worte für den Syntaxcheck aus der Bibliothek extrahiert. Nach und nach (man hat ja noch eine Beruf!) löste ich die Probleme und heute, knapp ein Jahr später liegt ein lauffähiges Programm vor, daß in akzeptabler Zeit Texte jeder Art auf ihre Rechtschreibung untersucht und bei Fehlern in geeigneter Weise mit dem Benutzer zusammenarbeitet.

Systemvoraussetzungen:

Eine Bibliothek mit ca. 60.000 Worten belegt ca. 200 Kilobyte auf einer Diskette. Wenn dazu noch Newword mit seinen ca. 100 Kilobyte kommt und ein Text dazu, dann wird klar, daß ein System mit 'nur' 03-Formaten ungeeignet ist. Um ein Wort eines Textes in der Bibliothek zu finden, liest der Wortmanager mindestens 2 Kilobyte ein. Das bedeutet, daß eine Textprüfung ziemlich langsam wird. Deshalb ist eine Ramdisk unbedingt erforderlich. Von allen Bildschirmtreibern wird nur der von RAM ((c) BP) unterstützt. Beim Einsortieren neuer Worte wird vor der Fertigstellung der neuen Bibliothek eine Sicherheitskopie angelegt, die natürlich größer ist als die alte Bibliothek. Das erfordert natürlich entsprechenden Platz auf der Diskette oder Ramdisk. Ich halte deshalb auch ein größeres Laufwerk für erforderlich. Prinzipiell ist der Wortmanager jedoch auf allen Ramdisk-Systemen lauffähig. Am allerbesten und befriedigendsten ist natürlich ein System mit großem Laufwerk und erhöhter Systemfrequenz (optimal 8 Megahertz). Hier purzeln die Worte nur so und es ist eine Freude zuzusehen.

Fenster:

Wortmanager ist in reiner Maschinensprache geschrieben. Es wird die Fenster-technik von Ram4.x (kleinere Versionen nicht getestet, aber sicher lauffähig oder leicht anpaßbar) ausgenutzt. Meistens zu sehen ist die zur Verfügung stehende Wortanzahl und das Auswahlmeneue. Die Uhr (sofern vorhanden) hat ein eigenes Fenster. Ebenso in Fenstern befinden sich alle Fehlermeldungen und Mitteilungen, die den Benutzer zu spezifischen Reaktionen auffordern.

Was kann der Wortmanager nicht?:

Das Programm prüft die Rechtschreibung gespeicherter Worte. Das bedeutet, findet es ein zu prüfendes Wort, so ist das Kriterium der Beurteilung erfüllt. Es werden also keinerlei grammatikalische Funktionen ausgeführt (es gibt eine Ausnahme, siehe unten).

S o f t w a r e: WortmanagerWas kann der Wortmanager?:

Er testet Worte in Texten auf richtige Schreibweise (Syntaxcheck).  
Er testet die Großschreibung von Worten, die **ausschließlich** groß geschrieben werden.  
Er testet die Großschreibung am Satzanfang.  
Er korrigiert den Text während des Testens.  
Er findet Worte und Wortteile in der Bibliothek (Joker zugelassen).  
Er speichert Worte, die er nicht kennt.  
Er sortiert die gespeicherten Worte in die Bibliothek ein.  
Er sortiert als falsch erkannte Wort wieder aus.  
Schaltbar ist die Funktion der automatischen Großschreibung, sowohl am Satzbeginn, als auch beim Einzelwort.

Was wird geliefert?:

Auf der Systemdiskette befinden sich fünf Files:

- |                          |              |
|--------------------------|--------------|
| 1) der Wortmanager       | Worte.Com    |
| 2) die Bibliothek        | Biblioth.Ek  |
| 3) die Dokumentation     | Worte.Doc    |
| 4) eine leere Bibliothek | Keinewor.Te  |
| 5) die Installation      | Wortinst.Com |

Die Installation:

Sie erklärt sich selbst. Worte.Com muß sich auf der selben Diskette befinden. Danach sollte Worte.Com gesichert werden.

Nachwort:

Der Aufbau der Bibliothek hat viel Arbeit gemacht (ungefähr zwei Monate!). **Jeder** Benutzer verpflichtet sich, mir **seine** Bibliothek im September 1989 zu schicken. Im Oktober werde ich daraus eine große gemeinsame Bibliothek machen, die dann jedem Einsender zugesendet wird. Das wird die Quote der unbekanntenen Worte drastisch senken. Außerdem werden alle neuen Worte nochmal auf ihre Richtigkeit überprüft. Sicher sind in meiner Bibliothek auch noch Fehler, die dann mit ausgemerzt werden können.

Im Laufe der Programmierung von Wortmanager ärgerte ich mich über verschiedene Kleinigkeiten von Newword. Ich wollte nämlich den Syntaxcheck einbinden. Da Newword sehr komplex aufgebaut ist, bekam ich jedoch keine Chance. Zudem ärgerte ich mich über die Länge (ca. 100 Kilobyte). Herbert's Edi ist auch nur 9 Kilobyte lang. Setzt man die Komplexität von Newword dagegen, müßte man mit höchstens 30 Kilobyte auskommen. Dazu käme dann ein Druckertreiber, den jeder in einem Installationsprogramm selbst erstellen könnte. Alles in allem nicht mehr als 50 Kilobyte. Lange Rede, kurzer Sinn, im Laufe der Zeit soll ein neues Textsystem entstehen. Dieses Textsystem soll dann auch Graphik einbinden können (hallo Manfred). Es soll eine automatische Trennhilfe haben. Dank der Aufmerksamkeit im Club konnte schon ein geeigneter Algorithmus gefunden werden. Auch für Ratschläge bin ich jederzeit dankbar. Willkommen sind auch Co-Autoren nach Absprache.

Nun noch etwas Statistik:

Die Suchzeit pro Wort beträgt 0.27 Sekunden. Der Datendurchsatz der Bibliothek beträgt ca. 3.46, d.h., pro Wort werden 3.46 Byte gebraucht.

H a r d w a r e: Umbau auf 8 MHz

## Verantwortlichkeit:

Jeder ist für seine Bibliothek selbst verantwortlich. Tut man Mist hinein, kommt Mist heraus. Man sollte immer die Möglichkeiten und Grenzen von Wortmanager beachten und nicht etwas verlangen, für das er (noch) nicht vorgesehen ist. Ebenso wenig sollte man ihn aber unterschätzen.

## Copyright:

Das Programm und diese Dokumentation unterliegen meinem Copyright, die Struktur der Bibliothek ebenfalls. Nicht meinem Copyright unterliegen natürlich die Worte, die in einer Bibliothek gespeichert sind:

Wortmanager 1.0 (c) Claudio Romanazzi 1.1989

**8 Mhz - Umbau**

(Akira Fukumoto, 1000)  
Berlin, den 17.03.89

Herbert hat mich gebeten meine Erfahrungen beim Umbau meines MTX auf 8 Mhz zu schildern und das soll an dieser Stelle geschehen.

Zunächst möchte ich die Ausgangssituation beschreiben, die die Basis meines Umbaus darstellt. Die Hauptplatine des MTX mit der Speichererweiterung und die RS-232 Schnittstellenkarte ist auf dem Deckelblech des FDX-Gehäuses untergebracht. Dabei habe ich mir den mechanischen Luxus geleistet, aus zwei eingefrästen Messing-Rechteckprofile einen Halterahmen zu bauen. Dabei ergaben sich allerdings, wie nicht anders zu erwarten, einige Schwierigkeiten, da Memotech einige Leiterbahnen mit Versorgungsspannungspotential derart nahe an den Rand der Hauptplatine angebracht hat, daß ich relativ präzise im Bereich der Hauptplatine fertigen mußte. Ihr werdet fragen, warum ich diesen Aufwand getrieben habe - das werde ich im folgenden erklären.

Erstens habe ich damit dieses Umbauteil sehr starr und damit schwingungsfrei aufgebaut. Zweitens habe ich durch Anbringen zweier Scharniere die komplette Hauptplatine auf die Lötseite umdrehbar machen können, was zukünftige Modifikationen erleichtert hat, als da wären der Umbau auf 8 Mhz, das Zusammenlöten der Einzelplatinen und das Ansetzen eines Abschirmbleches. Ferner wurde damit gleichzeitig eine gute Masseverbindung an der Speichererweiterungskarte ermöglicht. An dieser Stelle sollte erwähnt sein, daß die Speichererweiterungskarte mit 512 kBytes aufgerüstet ist. Als nächstes muß ich erwähnen, daß ich den Lüfter und das Netzteil gegen ein leiseres bzw. ein stärkeres getauscht habe. Bezüglich des Lüfters mußte ich bei der geforderten Qualität die Fa. Papst befragen, bezüglich des Netzteils ist mir ein Artikel der Zeitschrift C't in die Hände gefallen, in dem ein stärkeres Netzteil für einen Atari St beschrieben wurde und ausgerechnet dieses Netzteil wurde bei einem hier ansässigen Elektronikmarkt für DM 50.- angeboten. Auch hiermit hatte ich zunächst so meine Schwierigkeiten, da es Euro-Karten-Format hat. Dies habe ich im wesentlichen damit gelöst, daß ich das Netzteil quer eingebaut habe und über eine Registerschiene alle Verbraucher angeschlossen habe. Erstaunlicherweise konnte ich bei diesem Netzteil den Heizkörper am FDX-Bus entsorgen. Die meisten Probleme bereiteten mir die zusätzlichen Steckverbinder an der Rückseite des FDX-Gehäuses.

Nennenswert ist allerdings nur ein 23-poliger Sub-D Ausgang an dem über ein entsprechendpoliges, 1.5 m langes, Rundkabel die Tastatur angeschlossen ist. Hier wurde schon frühzeitig eine Verbindungslinie für den 8 Mhz-Umschalter eingeplant. Weniger von Bedeutung ist, daß ich das FDX mit Moosgummi ausgeschlagen habe und alle wesentliche Verschraubungen durch Linsenkopfförmige M4-Innensechskantschrauben ausgewechselt habe, was eher ästhetische Gründe hat. Dies ist der Ausgangspunkt für den 8 Mhz Umbau - genannt sei noch eine Grundbedingung die ich an dieses Projekt geknüpft hatte:

Es sollte zunächst mit einfachsten elektronischen Mittel ein Maximum an Performance erzielt werden, damit fiel zunächst eine Wait-State-Schaltung aus.

Hardware: Umbau auf 8 MHz

Zunächst bestellte ich mir von Holger Göbel die 8 Mhz-Karte und ging bei der Schaltung so vor, wie es das 8 Mhz-Kochbuch vorsah:

Z80 H CPU...

Einige Bustreiber in HC- oder ALS- Qualität

und Rams mit 100 ns Zugriffszeit (Hier sei erwähnt, das sich im nachhinein die kürzere Zugriffszeit nicht einen schelleren Erfolg nach sich zog)

und eine Hand voll keramische Kondensatoren im pF-Bereich

Nachdem alle Verbindungen gelegt waren und ich auch die zusätzlichen Bauteile untergebracht habe, bis auf die Logik in gedrehten Präz.-Sockeln, habe ich zunächst den Grundrechner aktivieren wollen. Dies misslang kläglich, was mich allerdings nicht weiter irritierte, da ich sowas bereits gewohnt bin. Nach einer Kontrolle der Verbindungen mit dem OHM-Meter stellte ich fest, daß ich beim Auslöten des CPU-Sockels zwei Adressleitungen abgerissen hatte. die Reparatur gestaltete sich glücklicherweise relativ leicht. Trotzdem hatte der Grundrechner auf meinen Genesungsversuch noch keine positive Antwort zu bieten, als ich einen weiteren Startversuch probte. An dieser Stelle des Projekts mußte ich meine Meßgeräte scharf machen und auf den Rechner loslassen. Mehrere Versuche die RAS bzw. CAS-Timings zu beeinflussen waren zwar auf dem Oszilloskop erkennbar brachten jedoch nicht den gewünschten Erfolg beim Booten, sondern nur im Grundgerätemodus. Dies äußerte sich der Gestalt, daß das CP/M-System zwar bootete jedoch nach kurzer Zeit abstürzte. Auch die im letzten Info veröffentlichten Timings von RAS, CAS und MPX Signalen konnte ich zwar aus einigen Datenblättern zu den Speichern nachvollziehen, brachten jedoch in der Praxis keinen Erfolg, d.h. der Einsatz von Treibern für doppelte Inverter wirke nicht.

Erstaunlich war auch, daß der Rechner bei angeschlossenem Oszilloskop länger funktionierte als ohne. Dies brachte mich auf eine neue Idee: Es bestand nämlich die Möglichkeit, daß das Meßgerät nicht nur auf das Timing des Signals Einfluß nahm, sondern auch auf dessen Form. Abgesehen davon war bei angeschlossenem Ossi, das Signal auch eher mit einem Trümmerhaufen zu vergleichen. Kurz entschlossen lötete ich auf alle mit dem Speicherzugriff beschäftigten Bauteile einen Blockkondensator in Keramikqualität mit 100 nF Kapazität. Nach dieser Behandlung erschien auf dem Oszilloskop endlich eine erkennbare Rechteckfunktion. Jetzt konnte ich auch die Verzögerungsglieder vollständig weglassen. Man beachte stets, daß nie allein ein Kondensator bei kurzgeschlossenem Widerstand geschaltet werden sollte, da sonst das TTL-Gatter wenigstens kurzzeitig einen Kurzschlußladestrom aufbringen muß. In dieser Phase konnte ich mit 8 Mhz komplett booten und Programme aufrufen. Dabei beschwerte sich nicht einmal der Controller über die Mehrbelastung. Meine Story ist allerdings noch nicht beendet, da bei den Testläufen doch noch ein erstaunlicher Fehler auftrat. Beim schnellen scrolen mit Newword oder Turbo-Pascal verschluckte sich anscheinend der Rechner und stürzte ab. Dieser Zustand war nur noch mit Hard-Reset zu beheben, an sonsten war die Tastatur tot. Ein weiterer Punkt war, daß dieser Fehler erst nach einigen Stunden Betriebszeit auftrat, was mich zur Annahme trieb, daß es sich um ein thermisches Problem handeln mußte. Also schraubte ich das Gerät wiedereinander auf und sieh da, der Fehler tauchte nicht wieder auf auch nicht nach 5 Stunden Betriebszeit. So allerdings konnte ich das Gerät nicht auf meinem Tisch deponieren, so daß ich mich mit Heizlüfer und Pappblende nach dem Corpus Delikti auf die suche machte. Relativ schnell fand ich heraus, das die CTC Ärger bereitete. Da ich wie oben beschrieben ohne Wait-State auskommen wollte probierte ich es zunächst mit einem Kühlkörper. Ein U-Profil in passender Länge und Breite mit ausreichend Wärmeleitpaste wurde der CTC auf den Rücken geschnallt und bei der Gelegenheit der CPU auch. Des weiteren sägte ich einen Schlitz von der Länge des User-Ports an die Seite des FDX, so daß der Luftstrom des Lüfters über die Kühlkörper hinwegstreichen konnte.

Danach baute ich alles wieder zusammen und nahm einen weiteren Dauertest vor. Dieser konnte mit Erfolg abgeschlossen werden.

H a r d w a r e: Umbau auf 8 MHz

Es sei noch erwähnt, daß ich die Drahtverbindungen der 8 Mhz-Karte nicht mit dem Prozessor-Chip oder der CTC verbunden habe, sondern die meisten Verbindungen auf den Durchkontaktierungen des User-Ports wiedergefunden habe.

An dieser Stelle möchte ich an die Konstrukteure der 8 Mhz-Karte und der Hardware-Uhr noch einige Kritik loswerden.

1. Die 8 Mhz-Karte ist so aufgebaut, daß sie bei gesockelten Rams, speziell beim dem links unten gefährlich nahe an stromführende Teile kommt.
2. Von der Hardware-Uhr mußte ich ein gehöriges Stück abkneifen, was mir hinreichend Schweiß auf die Stirn trieb, da ich selbst diesen verstrickten Aufbau bestimmt nicht wieder zusammengekriegt hätte.

**Anm.d.HzN.:** Sorry Akira, Du hast ein aus Versehen sehr breit geratenes Exemplar der Uhr erhalten. Die Designer der 8 MHz-Karte trifft daher hier keine Schuld - sie hatten die Norm-Uhr als Muster.

Trotzdem möchte ich nicht die Gelegenheit verpassen vor allem Herbert zur Nedden meinen Dank für die selbstlose Unterstützung bei der Realisation des Projektes auszusprechen, da ich ohne ihn heute bestimmt noch mit 4 Mhz arbeiten würde.

Bis hierhin habe ich den Bericht unter 8 Mhz und Newword geschrieben und da ich nicht über ein Vierfinger-Suchsystem hinauskomme kann man sich so ungefähr vorstellen wie lange das gedauert hat.

Noch ein Letztes möchte ich los werden:

Warum kriegt man den VS4 unter Turbo-Pascal nicht zum Laufen?

Kann mir darauf irgend jemand eine Antwort geben.

Mit vielen Grüßen und der Hoffnung, daß sich noch mehr MTX Besitzer dazu hinreißen lassen ihren Rechner zu tunen verbleibe ich bis zum nächsten Mal

Eurer

Akira Fukumoto

p.s: Ich bitte für den an mancherlei Stelle fatalen Satzbau und der mangelhaften Zeichensetzung um Verständnis, denn es ist jetzt 02:23 morgens.

p.s.s:Mal etwas ganz anderes: Kann mir irgend jemand aus dem Club das sogenannte Speculator-Rom zur Verfügung stellen, ich möchte doch mal sehen ob soetwas wirklich funktioniert.

Hardware: Tips**Tips zum Timung und den Bustreibern**

(Herbert zur Nedden, 2000)

Hier kurz einige Erfahrungen und Ideen aus meinen letzten 8 MHz-Aktionen und atv-erwandten Basteleien:

1. Warum die 74LS245 besser als die 74ALS245 auf den Treiberkarten laufen, die es mit dem Memotech-Flachbandkabel zu tun haben. Und warum die 74HC245 gute Resultate liefern - wohingegen die 74HCT245 weniger gute.

Die schwarzen TTL-Taschenbücher von IWT sind eine sehr interessante Lektüre, da sie die Funktion der IC's 74xx00 - 74xx??? vernünftig und relativ vollständig im Klartext beschreiben (wobei xx = nix, ALS, AS, F, H, L, LS oder S).

Und bei dem 74xx245 steht zu lesen, daß es diesen zwar in ALS, AS, F, L und LS gibt, aber nur bei der LS-Version die Eingänge eine Schmitt-Trigger-Funktion mit einer Hysteresis von 0.4 Volt besitzen, sich also auch sehr gut als Empfänger auf verrauschten Leitungen eignen.

Bei den HC und HCT (sind nicht in den IWT-TTL-Taschenbüchern drin, da sie CMOS und nicht TTL sind) bedeutet das T, daß die Signale TTL-Kompatibel sind. Doch die Vermutung, daß 74HCT245 als Bustreiber geeignet sind, ist falsch, da die HCT i.a. nur max. 10 cm Leitung an den Ausgängen sicher abkönnen (habe ich gelesen). Grundsätzlich liefern allerdings HC/HCT sauberere Flanken und Signale als die TTL-Typen, dafür sind insbesondere die HC meist langsamer.

2. Häufig sind es die RAMs auf der MTX-Hauptplatine, die Ärger machen, wenn der Takt beschleunigt wird. Mit den HM50256-12 auf den 768k-Karten hatte ich bislang nie Probleme. Daher ging ich an Olaf's Rechner pragmatisch ran, deaktivierte erst einmal die RAMs auf der Hauptplatine und spendierte ein neues PROM für die 768k-Karte, welches den Speicher ab Bank 0, 0000h an aufwärts einblendete. So lief der Rechner problemlos mit 8 MHz/1 Wait (wegen CTC). Anschließend habe ich das MTX-PAL erneuert, damit die letzten 16k adressierbares RAM auf Bank 15, von 8000h-0C000h durch die langsamen 150ns-RAM's der Hauptplatine gestellt werden. Da hier eh nur der Spooler hin kommt, tun es diese RAMs hier vollauf.
3. Wer hat ausklamüsert, warum einige 80-Zeichen-Karten gut gekühlt gut laufen, jedoch bei etwas mehr Wärme Zeichen verschlucken ??  
Je nach Karte ist das Verschlucken durchaus sehr hartnäckig!

Hardware: Sprachausgabe

**Sprachausgabe mit dem MTX**

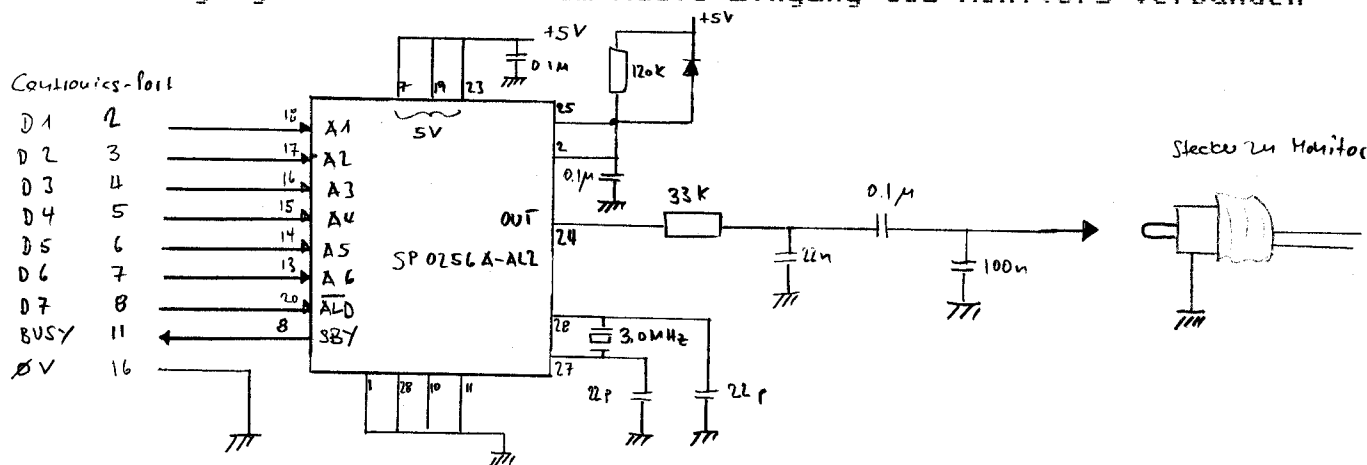
(Andreas Fischer, CH-4303)

die Sprachausgabe auf meinem MTX basiert auf der Basis des Sprachsynthesizer-IC "SP 0256A-AL2". Dieser IC kostet nur ca. DM 15.- und eignet sich bestens fuer diese Aufgabe. Mit nur 11 Bauteilen kann diese Schaltung sehr einfach aufgebaut werden.

Da der OUT-Port 7 schon durch diverse Funktionen belegt ist (z.B. die Datums- und Zeitanzeige) habe ich den Centronics - Ausgang benutzt. Damit weiterhin mit dem Drucker gearbeitet werden kann, habe ich einen elektronischen Schalter eingebaut (C-MOS 4066). Man kann aber auch einen handelsueblichen Umschalter anschliessen. Entsprechende Data-Switch-Centronics kosten bei uns ca Fr.50.-.

Nun zur Schaltung:

- Aufbau auf einer Lochraster-Printplatte
- externe Stromversorgung mit 5 V
- die Schaltung aus der Info 17-45 habe ich noch etwas abgespeckt - es laeuft auch so..
- den Ausgang habe ich mit dem Audio-Eingang des Monitors verbunden



das Programm:

- das Turbo-Pascal-Programm sieht wie folgt aus

```

program sprache; { $U+ }
var a,b,c,code:byte;
    phon:char;
    wort:string[80];
procedure schlaufe; kontrolliert, ob Buchstabe fertig gesprochen ist
var a:byte;
    test:boolean;
begin
    test:=false;
    write(phon);
    repeat
        a:=port[4];
        write(a,' ');
        if (a=127) or (a=255) then test:=true;
        if (a=127) then writeln;
    until test=true;
end;
    
```

Hardware: Sprachausgabe

```

procedure ruhe;
begin
  port[4]:=0+64;
  port[4]:=0;
end;

procedure cod_def;
begin
  case phon of
    '0':code:=0;(pause 10 msek)
    '1':code:=1;(pause 30 msek)
    '2':code:=2;(pause 50 msek)
    '3':code:=3;(pause 100 msek)
    '4':code:=4;(pause 200 msek)
    '5':code:=62;(kehlkopf el)
    'o':code:=30;
    'a':code:=23;
    'e':code:=20;
    '(':code:=7;(ae)
    '8':code:=8;(ch)
    'p':code:=9;
    'j':code:=10;
    'n':code:=11;
    't':code:=13;
    'm':code:=16;
    'd':code:=21;
    'i':code:=19;
    '9':code:=20;(ei)
    'h':code:=27;
    'u':code:=31;
    'g':code:=34;
    'w':code:=46;
    '7':code:=37;(sch)
    'z':code:=38;
    'r':code:=39;
    'f':code:=40;
    'k':code:=42;
    '6':code:=44;(ng)
    'l':code:=62;
    'v':code:=35;
    's':code:=55;
    'b':code:=63;
    'c':code:=42;
    'z':code:=43;
    'x':code:=47;
    'y':code:=49;
    '!'':code:=28;(oe)
  end;
end;

```

```

begin
  (sprechen aller phoneme)
  for a:=63 downto 0 do begin
    writeln(a);
    port[4]:=a+64;
    port[4]:=a;
    schlaufe;
    readln;
  end;

  repeat (sprechen ganzer woerter)
  write('Eingabe Wort :');
  readln(wort);
  for a:=1 to length(wort) do begin
    phon:=copy(wort,a,1);
    cod_def;
    port[4]:=code+64;
    port[4]:=code;
    schlaufe;
  end;
  ruhe;
  until wort='ende';

end.

```



Hardware: 80-Zeichenkarte**80-Zeichen-Karte: Schaltpläne**

(Herbert zur Nedden, 2000)

In diesem Info findet Ihr zum einen den Schaltplan der originalen 80-Zeichen-Karte der FDX sowie gleich die Modifikationen für die Aufrüstung des Speichers der 80-Z-Karte auf die größere Textauflösung, d.h. 16k statt 4k RAM. Dieser ist in Holland entstanden und 5 Seiten lang, wobei die Seiten 1 und 2 eigentlich den selben Ausschnitt zeigen - je nach Größe des RAMs.

Zusätzlich ist der Schaltplan von Horst Kupka noch einmal anbei, da sich doch der eine oder andere Fehler in dem im letzten Info gedruckten Schaltplan eingeschlichen hatte, und nun nimmer drin ist.

Hinweis: Der 'Holland-Plan' benutzt für die zusätzlichen Adreßleitungen durch IC 3B' zu den RAMs die einfache Version mit einem 74LS367, während der 'Kupka-Plan' auf einem LS257 basiert.

**80-Zeichen-Grafik**

(Horst Kupka, 4019 und Holger Göbel, 8630)

**Anm.d.HzN.:** Die Lösung und diese Anleitung ist von Horst entwickelt, und von Holger getestet und in Details zum Teil korrigiert worden.

Wie versprochen, die Umbauanleitung.

Etwas so zu beschreiben, daß auch jeder alles versteht, ist unheimlich schwer. Hier ein Versuch von mir. Bitte seid nicht ärgerlich, wenn es nicht sofort hin-  
haut.

Bevor die Beschreibung startet, folgendes zum Schaltplan:

Alle neu hinzugekommenen IC's sind mit 'Nx', also N1 bis N7, bezeichnet. Die anderen Nummern entsprechen den Platinenkoordinaten. Zur Verdrahtung benutze ich Kupferlackdraht mit einem Durchmesser von 0,3 bis 0,4 mm. Er läßt sich gut mit einer Pinzette verlegen und die Isolierung schmilzt durch die LötKolbenwärme. Und noch ein kleiner Tip: Benutzt LötKolben mit einer Leistung von max. 50 Watt und feiner Spitze, sonst verlötet Ihr alles andere, nur nicht was gewollt ist. Die Umbauanleitung sieht zwar recht lang aus, aber der Umbau ist in ca. 4-5 Stunden geschafft.

Als erstes nenne ich die IC's, die auf der 80-Zeichenkarte eingebaut und vom Umbau betroffen sind.

- |                      |    |                                  |
|----------------------|----|----------------------------------|
| 1.) 1E (74LS393) Pin | 2  | / Takterzeugung Blink u. Density |
| 8E (74LS374) Pin     | 1  | / Attributregister               |
| 7A ((E)PROM) Pin     | 18 | / ASCII-ROM                      |
| 9A ((E)PROM) Pin     | 18 | / ATTRIBUT-ROM                   |

Die Ausgänge dieser IC's müssen im Grafikbetrieb ausgeschaltet werden. Die genannten Pins werden kurz über der Leiterplatte abgeschnitten, hochgebogen, miteinander verbunden und später zum Signal HGR (N1 Pin 2) geführt. Zu den ROM's folgendes: Rom raus, P18 hochbiegen und wieder in den Sockel.

- 2.) 6A (74LS166) / Schieberegister Pixelerzeugung

Pin 1 und Pin 6 über Platine abkneifen, hochbiegen und Pin 6 mit Pin 8 verbinden. Pin 1 wird später mit dem Ausgang (Pin 13) von IC N4 verbunden.

Hardware: 80-Zeichenkarte

## 3.) 7F (74LS161) / Takterzeugung Systemtakte

Pin 2 Clockeingang abkneifen und hochbiegen.  
Später mit Ausgang IC N2 (Pin 4) verbinden.

## 4.) 1B (74LS30) / Port-Dekodierung 38-3B

Pin 8 abkneifen und hochbiegen. Er wird zur genaueren Dekodierung gebraucht.

## 5.) 5F (74LS20) / Erzeugung SRLD Laden Schieberegister

Pin 10 abkneifen und hochbiegen. Hier muß noch eine UND-Verknüpfung vor, damit im schnellen Betrieb nur einmal geladen wird.

So, das war es erstmal. Oh je, so viel, aber das war das Einfachste. Jetzt gehts ans Basteln.

Ich gehe davon aus, daß jeder schon den Umbau auf 8Mhz und auf die 16k-RAM-Erweiterung (INFO 21/30 bzw. 22/15) gemacht hat; wenn nicht, dann sollte man sich den Hinweis bei IC 1C und 3B' anschauen. Jetzt ist eine Hälfte von IC 1C frei. Die benötigten Beinchen müssen leider von der Leiterplatte abgeknipst und hochgebogen werden. (es sind die Pins 1, 2, 3, 4, 5 und 6). Pin 2 wird mit Pin 6 und Pin 1 mit Pin 13 verbunden. Zu den restlichen Pins später bei der Verdrahtung.

Nun IC 3B': Es gibt zwei Versionen der bisherigen 16k-RAM-Erweiterung, die eine ist von Jan Bredereke (INFO 22/15), auf welche sich der Schaltplan bezieht. Die ältere Version von Hagen Wenzek (INFO 17/69) hat ja bekanntlich keinen Umschalter. Es muß also, wenn 3B' schon vorhanden, folgendes geändert werden:

Die Leitung, die zu MA12 (Pin 16) des CRT-Controller (MC 6845) geht, wird entfernt, und Pin 5 (bei H. Wenzek: Pin 6) wird von der Masse gelöst. Diese beiden Leitungen kommen später an den Adress-Zeilen-Multiplexer IC N2.

IC 7E (Attribut-Zeichen-Multiplexer): Da IC 8 im Grafikbetrieb ausgeschaltet ist, müssen noch 3 Widerstände zur Voreinstellung der Attributwerte eingebaut werden. Es sind 3 \* 1kOhm (mit neu bezeichnet). Background = Pin 14, Bright = Pin 13 und Revers = Pin 5. Die andere Seite der Widerstände wird an Pin 8 von IC 7E (Masse) angelötet.

Und jetzt zu dem Teuersten des Umbaus. Es handelt sich, wie erraten, um die RAM's. Mein Vorschlag: 61256. Ich weiß sie sind teuer, aber bieten auch Vorteile. Dazu später mehr.

**Anm.d.HzN.:** kleinere SRAM's kosten kaum weniger Geld, wohl aber beim Retten der Bildschirme im KLICK dort einiges an Speicherplatz! Bei Verwendung der 61256 wird kein Platz auf dem KLIX-Heap benötigt, da das Retten völlig entfallen kann - es wird einfach im RAM der 80Zeichenkarte gepaged! Holger Göbel kennt eine Quelle, wo diese RAMs kürzlich unter DM 30.- das Stück kosteten.

Die Pins 23 und 26 werden hochgebogen, bis die Beinchen auf dem IC aufliegen. Dann wird das RAM so in den Sockel gesteckt, daß die Beinchen 1, 2, 27 und 28 links überstehen. Im Sockel sind nun 2 Buchsen frei, sie werden nun mit dem RAM verbunden: Sockelpin 24 mit RAMpin 28 und Sockel 21 mit RAM 27. An den RAMs sind nun noch folgende Beinchen unbelegt:

Pin 1 kommt später nach HGR

Pin 2 kommt nach IC 3B' Pin 9 = RA 12 (auch bei H. Wenzek)

Pin 26 später zu Zusatzplatine IC N2 Pin 9 = NA13

Pin23 wird mit IC 3B' Pin 12 verbunden = RA11 (bei H. Wenzeks Version mit Pin 3 von IC 3B).

H a r d w a r e: 80-Zeichenkarte

Damit hätten wir auch die RAMs vorbereitet.

Jetzt ist alles klar auf der 80-Zeichen-Karte oder...?

Nein, N6 (74LS374) und N5 (74LS138) müssen noch vorbereitet und huckepack auf IC 4C bzw IC 2C gelötet werden. Von IC N6 wird nur ein Ausgang benötigt (Q1 = Pin 5). Die restlichen Ausgänge kneift man ab. (Pin 2, 6, 9, 12, 15, 16 u. 19). Pin 5 = RA13 und Pin 11 = OUT3A hochbiegen und dann auf IC 4C löten. Mit IC N5 verfährt man genau so: Pin 10 u. 11 abkneifen, die Pins 3, 4, 7, 9, 12, 13, 14 u. 15 hochbiegen und auf IC 2C löten.

Die restlichen neuen IC's, ausgenommen N3, habe ich auf einer kleinen Zusatzplatine untergebracht. Die Platine ist so aufgebaut, daß die neuen IC's in die Zwischenräume zwischen IC 8D und 3D passen. Zweckmäßigerweise beginnt man mit N4, das dann zwischen 8D und 7D liegt, und hört mit N7 auf, das zwischen 3D und 4D zu liegen kommt. So kann man sofort von 8D bzw. 3D auf die Zusatzplatine springen, und die Platine wird durch diese Verbindungen gehalten. Die Reihenfolge von oben nach unten lautet bei mir N4, N2, N1 u. N7.

Achtung: Wenn der Status der Platine abgefragt werden darf (was ja sinnvoll ist, damit sich irgendeine Software auf den Modus der Karte einstellen kann), sollte der dazu benötigte 74LS244 (s. großer Schaltplan links unten) auch auf diese Platine, und zwar als fünftes IC nach unten. Die Belegung ist ganz am Ende der Umbauanleitung beschrieben. Die Verdrahtung muß jetzt geschehen, später kommt man nicht mehr an IC 4D ran. (Es bliebe dann nur noch übrig, die Datenleitungen direkt vom 6845 abzugreifen.)

Legt man die Zusatzplatine auf die 80-Zeichenkarte (Bauteilseite unten), so sollten die IC's genau in die Lücken passen, und Pin 16 von IC N4 liegt an der Ramseite. Alle Verbindungen, die auf der Zusatzplatine erforderlich sind, werden nun hergestellt: Spannungsversorgung, HGR-Signal (N4 P9, N2 P1, N1 P2), TXTPG (N2 P11 --> N1 P7) und folgende Pins auf Masse legen: N4 P1, N4 P6 und N2 P15. Jetzt wird die Zusatzplatine auf die 80-Zeichenkarte gelegt und auf zur letzten Etappe.

Die Verdrahtung:

Die erste dickere Leiterbahn rechts neben der Zusatzplatine ist Masse, links neben der Platte ist Plus 5V.

Von IC 8D Pins 6, 5, 2, 15, 16, 19 nach N4 Pin 11, 10, 5, 4, 3, 2 (also 8D Pin 6 nach N4 Pin 11, 8D Pin 5 nach N4 Pin 10 usw.).

Weiter Von IC 3D P. 7 --> N4 Pin 14 und IC 3D PIN 10 --> N4 Pin 12. Das waren die 8 Attributbits.

IC N3 habe ich auf eine kleine Platine zwischen 9B und 7B gelegt, die Leitungen S0 bis S7 zwischen N3 und 9B verdrahtet und P0 bis P7 sofort nach IC 6A gezogen, also:

N3 Pin 13, 15, 17, 2, 4, 6, 8, 11 nach

9B Pin 2, 5, 6, 9, 12, 15, 19, 16.

N3 Pin 7, 5, 3, 18, 16, 14, 12, 9 nach

6A Pin 2, 3, 4, 5, 10, 11, 12, 14.

Pin 1+19 werden mit HGR-QUER auf der Zusatzplatine N1 Pin 3 verbunden.

Hardware: 80-Zeichenkarte

## Verbindungen 80-Zeichen-Platine --&gt;Zusatzplatine

von 1B Pin 8 --> N5 Pin 4 / 38-3B Portdekod.  
 von 2C Pin 3 --> N7 Pin 4 / RDQuer zur Portdek.  
 von 2C Pin 2 --> N7 Pin 5 / A1 Portdek.  
 von N5 Pin 3 --> N7 Pin 6 / Umschaltung Read-Write  
 von N5 Pin 15 --> N7 Pin 1  
 von N5 Pin 14 --> N7 Pin 2  
 von 1C Pin 12 --> N7 Pin 3 / neues CRTCCS  
 von 1C Pin 5 --> N7 Pin 9 / SRLD-Freigabe  
 von 1C Pin 4 --> N1 Pin 2 / HGR  
 von 7F Pin 2 --> N2 Pin 4 / neuer Takt Systemtakterzeugung  
 von 8F Pin 8 --> N2 Pin 2 / 15 Mhz - Takt und weiter nach 1C Pin 3  
 von 1C Pin 6 --> N2 Pin 3 / 7,5 Mhz. Kontrolle: 6+2 1C verbunden ?  
 freier Lötunkt unter 5F Pin 10 --> N7 Pin 10 / SRLD Takt  
 von 5F Pin 10 --> N7 Pin 8 / SRLD-Erzeugung  
 von 6A Pin 15 --> N4 Pin 15 / Seriell Laden  
 von 6A Pin 7 --> N4 Pin 7 / SRCK  
 von 6A Pin 1 --> N4 Pin 13 / Pixels von Attribut  
 von 4A Pin 2 --> N1 Pin 1 / RESET  
 von 4A Pin 38 --> N2 Pin 6 / Z0 = Zeile 0  
 von 4A Pin 37 --> N2 Pin 13 / Z1 = Zeile 1  
 von 4A Pin 16 --> N2 Pin 5 / MA 12  
 von 4A Pin 17 --> N2 Pin 14 / MA 13  
 von 3B'Pin 11 --> N2 Pin 7 / Multiplexsignal MA 12 und Z0  
 Achtung: bei H. Wenzek: 3B' Pin 10 --> N2 Pin 7  
 von 3B'Pin 5 --> N2 Pin 12 / MUX MA 13 und Z1  
 Achtung: bei H. Wenzek: 3B' Pin 6 --> N2 Pin 12  
 von 3B'Pin 7 --> N2 Pin 10 / RA 13  
 (bei H. Wenzek ebenfalls 3B' Pin 7 --> N2 Pin 10)  
 und weiter nach N6 Pin 5 (huckepack 4C)  
 von 4C Pin 3 --> N1 Pin 13 / D07 Romseite1  
 von 4C Pin 18 --> N1 Pin 12 / D06 Romseite2  
 von 4C Pin 4 --> N1 Pin 5 / D05 TXTPG Textseite u. Adr.13  
 von 4C Pin 17 --> N1 Pin 4 / D04 Umschaltung Grafik  
 von N5 Pin 7 --> N1 Pin 9 / OUT 3B  
 von N6 (huckepack 4C) Pin 11  
 --> N5 (huckepack 2C) Pin 9 / OUT 3A

## Und zum Schluß die RAMs:

Pin 1 IC 9C mit Pin 1 IC 7C und weiter nach HGR Pin 18 9A  
 bzw. 7A oder N1 Pin 2  
 Pin 2 IC 9C mit Pin 2 IC 7C und weiter n. IC 3B' Pin 9 = RA 12  
 (bei H. Wenzek ebenfalls 3B' Pin 9)  
 Pin 26 IC 9C mit Pin 26 IC 7C und weiter n. Zusatzpl. IC N2 Pin 9  
 Pin 23 IC 9C mit Pin 23 IC 7C und IC 3B' Pin 12 verbinden = RA11  
 (bei H. Wenzek 3B Pin 3, Achtung: 3B ohne ')

Das wars. Kontrolle: alle HGR-Leitungen angeschlossen? Spannungsversorgungen auch ?

Hardware: 80-Zeichenkarte

Nochmals alle HGR-Anschlüsse:

N1 P 2

N2 P 1

N4 P 9

7C P 1

9C P 1

7A P18

9A P18

8E P 1

1E P 2

74LS244 Pin 17, wenn ihr den Status abfragen wollt (was zu empfehlen ist), s. dazu den Schaltplan und weiter unten.

Ich hoffe, Ihr wie ich habt keine Verbindung vergessen, also dann ready zum Countdown. Doch wo kommt die Software her? Keine Sorge, etwas Testsoftware gibt es bei mir.

Beschreibung zu den Hinweisen 'siehe Text'.

Die Grafikkarte wurde von mir für die Auflösung von 720 \* 348 Pixels (Hercules-auflösung) umgerüstet. Hierzu wird ein Speicher von 32KByte benötigt. (16K Attribut u. 16K ASCII) Damit bei späteren Anwendungen, z.B. <SHIFT ESC> die Grafik nicht im Nirwana landet, wird mit HGR auf die zweiten 32k-Byte umgeschaltet. So stehen zusätzlich noch 2 ganze Textseiten zur Verfügung. Diese sind umschaltbar mit TXTPG (D5 Out 3B). Ist aber mangels Money der Speicherplatz nicht vorhanden, so kann man nur 640 \* 200 Pixels auflösen, weiterhin ist nach Umschaltung auf den Text-bildschirm die Grafik weg !! Ist man sich dieser Sache bewußt, so wird RA13 und NA13 nicht benötigt und der beschriebene Ramumbau kann außer Acht gelassen werden. Ebenso fällt HGR an Pin 1 flach. (ACHTUNG: die Pinbelegung auf meiner Zeichnung stellt einen 61256 dar.) Will man aber eine höhere Auflösung, so ! muß ! der Speicher erweitert werden. Hierzu biete ich noch eine Alternative:

Zwei weitere 8K-RAMs werden huckepack auf die vorhandenen montiert. Die Pins 20 = CS werden aus dem Sockel entfernt und hochgebogen. Nun muß noch IC 9E verdrahtet werden. Die Signale CS1 gehen auf die unteren RAMS Pin 20, und CS2 auf die oberen. Aber auch hier: Nach Umschaltung = Grafik futsch.

Und nun der letzte Clou. Hat einer genug Geld übrig, so kann er durch größere EPROMs auch noch verschiedene Schriftarten auswählen. Hierzu dienen die Signale ROM1 und ROM2 (vielen Dank, Hans).

Und will einer zu allerletzt auch noch wissen, wie er seine 80-Zeichenkarte vergewaltigt hat, so muß er einen LS244 opfern. Mit IN3B kann er dann die Konfiguration seiner Karte abfragen. Dazu muß der 74LS244 so verdrahtet werden:

Spannungsversorgung wie üblich

Pin 19 -->	N5 Pin 12	/ IN 3B
Pin 9 -->	4D Pin 11	/ D7
Pin 5 -->	4D Pin 8	/ D6
Pin 7 -->	4D Pin 13	/ D5
Pin 3 -->	4D Pin 6	/ D4
Pin 17 -->	N1 Pin 2	/ HGR
Pin 13 -->	N1 Pin 7	/ TXTPG
Pin 11 -->		/ ROM1 für evtl. ROM-Erweiterung
Pin 15 -->		/ ROM2 für evtl. ROM-Erweiterung

Mehr kann ich Euch leider nicht bieten, und durch diesen Umbau bekommt man keine Hypesupergrafikcadmaschine, aber Grafik ist endlich machbar, und mit den geeigneten Routinen gar nicht so schlecht.

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

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

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



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

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

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

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

H a r d w a r e: Diskettenwechsel erkennen**Disc-Change-Signal**

(Herbert zur Nedden, 2000)

Warum und wieso mich dieses Signal interessiert, ist leicht gesagt:

Damit dieses Signal von Olaf's Cache ausgewertet werden kann, und so meine Disketten(-Laufwerke) und Nerven geschont werden. Klar ?

Also noch einmal (denn das o.g. hätte ich auch nicht verstanden!):

Olaf Krumnow hat einen Directory-Cache geschrieben, d.h. eine Systemerweiterung, die das Direktory von Diskettenlaufwerken im KCLICK-Heap zwischenspeichert, und so die Zugriffe auf das Direktory von Disketten erheblich beschleunigt. Das funktioniert so, daß beim ersten Zugriff auf das Direktory dieses komplett in einen Puffer gelesen wird, und bei weiteren Direktory-Lese-Zugriffen die Scheibe nicht mehr bemüht wird. Das spart u.a. reichlich viele Bewegungen des Schreib-Lese-Kopfes (also weniger Schrab-Schrab). Schreibzugriffe werden natürlich sowohl in den Puffer als auch auf die Diskette gepackt. Als Beispiel sei an eine dBASE-Anwendung gedacht, die mit diversen .CMD-Dateien (also Kommandodateien) arbeitet. Jedesmal, wenn eine solche gebraucht wird, geht der Laufwerks-Kopf auf das Direktory, sucht die Datei, wandert dort hin, wo die Datei ist, und liest sie. Mit Cache wandert der Kopf direkt zur gewünschten Datei. Auch die Arbeit mit großen Overlays (z.B. NW.OVR) ist so deutlich schneller, da diese Datei i.a. in einem Stück auf der Scheibe ist, der Kopf daher kaum noch bewegt wird. Lausche einfach mal Deinem Laufwerk. Meistens folgt einem Schrab bei dem der Kopf auf die äußeren Spuren zwecks Direktory-Zugriff geht, ein zweiter Schrab mehr oder weniger dorthin zurück, wo er her kam, um die Daten zu lesen. Und dieser Schrab-Schrab entfällt bei Cache-Betrieb bei Lesezugriffen.

Stellt sich natürlich die Frage, woran der Cache merkt, daß die Diskette gewechselt wurde, d.h. das Direktory neu einzulesen ist. Das BDOS (bei uns heißt es P2DOS) erfährt von diesem Umstand meist durch einen WarmBoot, der i.a. mittels ^C oder einem JP 0 am Ende von Programmen ausgelöst wird. Das BDOS teilt dem BIOS beim 'Select Drive' (Funktion 9) im E-Register mit, wenn es die Diskette für gewechselt hält (d.h. dem BDOS neu angemeldet wurde), und der Cache kann entsprechend reagieren.

Aber genau hierin ist die Krux: Habe ich drei Laufwerke, und wechsele eine Diskette, was ich dem BDOS mittels ^C mitteile, so erklärt das BDOS gleich alle drei Laufwerke für gewechselt, und verleitet folglich den Cache dazu, bei einem erneuten Zugriff auf irgendeine dieser drei Disketten deren Direktory neu einzulesen, auch, wenn die Diskette garnicht gewechselt wurde. Eigentlich ist das nicht allzu tragisch, da die Zugriffe trotzdem deutlich schneller mit als ohne Cache sind. Dumm ist nur, daß die meisten Programme mit einem JP 0 enden, also einen WarmBoot auslösen (und danach also das BDOS via BIOS dem Cache klar macht, daß ein erneutes Einlesen der Direktories angesagt sei). Der WarmBoot hat den Vorteil (und gelegentlich auch den Grund), daß dadurch BDOS und CCP (bei uns P2DOS und ZCPR2) nachgelesen werden.

Einige Laufwerke (nämlich die 'großen') liefern ein Signal, mit dem sie mitteilen, daß die Diskette gewechselt wurde, und daran könnte sich der Cache orientieren, d.h. müßte dem BDOS nicht mehr glauben.

Diese Laufwerke liefern am Pin 34 des Daten-Flachbandkabels je nach Stellung eines Jumpers wahlweise das Signal

DRIVE READY d.h. Laufwerk ist bereit  
DISC CHANGE d.h. Diskette wurde gewechselt

Da unser Controller sein eigenes 5"-Ready-Signal erzeugt, ist es beim 5"-Bus, an dem die 5"-, 3 1/2"- und 3"-Laufwerke hängen, für den Betrieb des Controllers völlig wurscht, was die Laufwerke an diesem Pin so erzählen.

H a r d w a r e: Diskettenwechsel erkennen

Und das brachte mich auf den Gedanken, die Laufwerke, die es können, so zu jumpern, daß sie das 'DISC CHANGE' melden. Dabei liegt dieses Signale auf 0 Volt, wenn die Scheibe aus dem Laufwerke genommen wurde, oder der Rechner eingeschaltet wurde. Nachdem eine Diskette wieder eingelegt und mindestens ein Spurwechsel ausgeführt wurde, geht das Signal wieder auf 5 Volt.

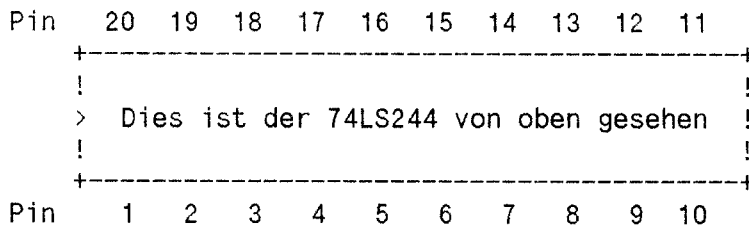
Und Olaf's Cache kann just dieses Signal auswerten. D.h. solange die Diskette nicht gewechselt wurde wird das Direktory nicht unnötig nachgelesen. Klar, Diskette raus und gleiche wieder rein liefert ebenfalls die Information, daß die Diskette gewechselt wurde.

Was ist zu tun ?

Leider muß wieder einmal der LötKolben angeschmissen werden, und ein 74LS244 auf die Floppy-Controller-Platine gelötet werden. Außerdem müssen die Laufwerke, die das DISC CHANGE liefern können entsprechend gejumpt werden, d.h. die Steckbrücken müssen z.B. bei TEAC-Laufwerken auf DC statt RY gesteckt werden. Wie diese bei anderen Herstellern heißen ist mir unbekannt - vermutlich DC. Allerdings ist mir kein 40-SPUR-Laufwerk bekannt, welches das DISC CHANGE liefern kann!

Um nun wirklich in den Genuß der Neuerungen zu kommen muß Olaf's Cache geladen werden, der - wie zu erwarten - nur unter RAM 4.5 läuft.

Ach, eh ich's vergesse, die Lötarbeiten:



Dieser 74LS244 muß auf den Floppy-Controller gelötet werden.

Er stellt dann den Input-Port 45h dar.

Pins 1 2 4 6 8 11 13 15 17 19 hochbiegen.

Nun den 72LS244 huckepack auf das IC B5 (ebefalls 74LS244) setzen, wobei die Pins 3 5 7 9 10 12 14 16 18 20 mit den gleichen des unteren verlötet werden.

Nun zu den hochgebogenen Pins:

- Pins 1 und 19 verbinden, und an Pin 14 des IC D1 (74LS138).
- Pins 2 4 6 8 13 15 17 allesamt an Masse (= Pin 10)
- Pin 11 mit Pin 1 des Steckkontaktes J3 (d.h. dem Anschluß für 5"-Laufwerke) verbinden. Das ist der Pin, der direkt neben dem 'J' von 'J3' ist, und dank Memotechs ungedrehter Numerierung dem Pin 34 der Laufwerke entspricht. Außerdem muß noch ein Widerstand von 1 kiloOhm zwischen diesen Pin 11 des 74LS244 und +5 Volt (Pin 20 des 74LS244) gelötet werden, da das uns hier interessierende Signal der Laufwerke i.a. mit einem sog. Open-Collector-Treiber erzeugt wird.

Folgendes hast Du gelötet: Port IN 45h (das ist der neue 74LS244), dessen Bit 0 das DC-Signal der Laufwerke ist, und dessen Bits 1-7 auf Masse liegen.

**WANN ?** Olaf's Cache ist leider noch nicht 100%-ig betriebssicher. Er versucht es, ihn mögl. bald fertigzustellen, und auf KLICK.007 zu packen. Zumindest kannst Du, wenn Deine Kiste schon mal auf ist, vorbereitend löten.

Hardware: c't SRAM-Floppy

### Datenverlust auf der c't SRAM-Floppy

(Herbert zur Nedden, 2000)

Immer, wenn ich meine Boot-Floppy aus ihrem Steckplatz herausgenommen habe mußte ich anschließend von einer echten Diskette booten, da diese Platine die Angelegenheit hat, Daten zu verlieren. Nein, ich vergaß nicht, den Rechner abzuschalten! Der Datenverlust kommt vor allem bei abgeschaltetem Rechner. Dahingegen habe ich mit meiner SRAM-Floppy von CONITEC keinerlei Datenverluste - egal wann und wie schnell ich diese Karte vom ECB-Bus ziehe und wieder draufstecke - auch bei laufendem Rechner.

#### Zur Technik:

Die statischen RAMs und der 74HC138 (c't) bzw. 74HCT137 (CONITEC), der zur Freigabe der RAMs dient, werden bei laufendem Rechner aus dessen +5 Volt gespeist, bei abgeschaltetem Rechner oder natürlich ausgestöpselter Platine aus einem Akku bzw. einer Batterie. Wenn ich daher im folgenden von 'Recherspannung' schreibe, meine ich immer die Spannung, die das Netzteil des Computers liefert!

Der Grund für den Einsatz von 74HC138 bzw. 74HCT137 ist, daß dieses CMOS-IC's sind, die sehr wenig Strom verbrauchen, so sie nicht aktiv am geschehen teilnehmen, d.h. sich ihre Eingangssignale nicht ändern. Diese beiden IC's stehen immer unter Strom, damit deren Ausgänge insbesondere im nicht-aktiven Zustand auf high liegen, und so die RAM's sperren.

Warum c't einen 74HC und CONITEC einen 74HCT verwendet: .... ???

#### Schauen wir uns mal die Schaltung der c't-SRAM-Floppy an:

Die Chip-Enables (Freigabe-Signale) der RAMs wird von einem Binärdecoder 74HC138 erzeugt, der ebenso wie die RAMs durch den Akku mit Strom versorgt wird. Dabei wird der 74HC138 nur dann aktiviert, wenn wirklich auf die Floppy zugegriffen wird - naja fast nur ... (sonst verlöre die Karte ja keine Daten!)

Interessieren tut mich der Pin 6 dieses 74HC138: Liegt hier Low an, ist der 74HC138 gesperrt, d.h. seine Ausgänge liegen alle auf High und damit sind die RAMs deaktiviert - egal was sich so an den anderen Pins des 74HC138 tut, solange er unter Strom steht. Liegt hier High an, ist der 74HC138 nicht gesperrt.

1. Das Original-Design der c't-Platine sieht vor, daß dieser Pin mit +5 Volt vom Rechner verbunden wird, d.h. eigentlich nur dann auf High liegt, wenn der Rechner an ist. Nun, wenn der Rechner aus ist, stellt das getaktete Computer-Netzteil relativ sicher, daß dieser Pin auf Low liegt - da es einen niederohmigen Ausgang besitzt. Ziehe ich die Karte jedoch aus dem Gerät, hat es sich mit dem Zustand dieser Leitung - sie hängt völlig frei - und ggf. Daten ade!
2. Die neuere Version dieser Platine hatte hier einige kleinere Änderungen - nein ich sagte nicht Verbesserungen: Pin 6 war über einen Widerstand von 4,7 kiloOhm an +5 Volt vom Rechner angeschlossen (Lötbrücke J9) und außerdem mit dem Reset-Signal des Rechners (Lötbrücke J8). Damit wird der 74HC138 zusätzlich gesperrt solange Reset anliegt - mehr haben wir nicht davon.
3. Uwe Grass empfahl, einen Widerstand von 1 kiloOhm zwischen diesen Pin 6 und Masse zu löten, damit dieser Freigabe-Eingang bei ausgestöpselter Karte an Masse hängt. Das hat einige Datenverluste erspart, die einigen bei noch eingesteckter Karte Ärger bereiteten, aber das Ausstöpseln der Karte war für mich immer noch das schnellste Mittel, den Rechner zum Booten von Diskette zu bewegen.

Fazit: Das Design der c't-Karte ist einfach unzulänglich.

Hardware: c't SRAM-FloppySchauen wir uns mal die Schaltung der CONITEC-SRAM-Floppy an:

Im Prinzip ist die Lösung hier die gleiche, nur daß statt des 74HC138 der 74HCT137 verwendet wird. Der '138 ist ein Binärdecoder, der '137 ist ein Binärdecoder mit Zwischenspeicher (Latch) - ansonsten sind diese beiden IC's Funktionsgleich. Den Pin 6, um den es hier geht, haben beide (74HC138 und 74HCT137) gemeinsam.

CONITEC mußte immerhin 16 statt acht große RAM-Sockel unterbringen, hatte also keinen Platz für ein Latch plus Binärdecoder in zwei ICs.

Nun zu CONITEC's Lösung bei der Beschaltung des Pins 6:

1. An diesem Pin ist ein Widerstand von 220 Ohm gegen Masse, und eine Leuchtdiode gegen die +5 Volt der Rechnerspannung.  
Die Leuchtdiode zeichnet sich insbesondere durch einen Spannungsabfall von mindestens 0,7 Volt und durch einen konstanten Strom aus. Sobald die Rechnerspannung sinkt, sinkt die Eingangsspannung am Pin 6 des 74HCT137 um mindestens 0,7 Volt schneller auf Low als die Rechnerspannung, d.h. er macht die Schotten zügig dicht, und die statischen RAMs sind sicher.  
Der kleine Widerstand von 220 Ohm ist eine deutlich kräftigere Masse-Verbindung als bei der c't-SRAM-Floppy, und stellt sicher, daß dieser Eingang auch bei ausgestöpselter Platine auf Low bleibt, d.h. die Daten sind sicher.

Nun zur Lösung des Problems:

1. Alles was so am Pin 6 des 74HC138 ankommt dort abklemmen, d.h. vermutlich einfach die Lötbrücken J8 und J9 öffnen, und ggf. den von Uwe Grass spendierten Widerstand von 1 kOhm (oder was auch immer), der von hier gegen Masse sitzt entfernen.
2. Einen Widerstand von 220 Ohm zwischen Masse und diesen Pin 6.
3. Eine LEUCHT-Diode (keine andere Diode!!!) zwischen diesen Pin 6 und + 5 Volt der Rechnerspannung. Dabei muß diese Diode bei eingeschaltetem Rechner leuchten. Eine 3mm-LED ist gut geeignet.  
Achtung: die +5 Volt dürfen nicht vom Pin 16 des 74HC138 genommen werden, da dort die Akku/Batteriespannung anliegt.

Hast Du eine auf 512 kB aufgerüstete c't-SRAM-Floppy, dann ist auf dem einen 74HC138 gleich ein zweiter huckepack drauf. Da jedoch bei beiden die Pins 6 miteinander verbunden sind, ist nichts weiter zu tun als das o.g.

Das bringt's NACHWEISLICH!!!

Im Rahmen meiner Basteleien habe ich diese Platine mehrfach beiseitegelegt, wieder auf den Bus gesteckt und direkt davon gebootet.

FRAGE - PROBLEM: HILFE!!

(Herbert zur Nedden, 2000)

Ich habe es mehrfach erlebt, daß eine c't-SRAM-Floppy 100%-ig im Lesezugriff lief. Nachdem ich lediglich mit ERA TEST.COM dieses Teil mal eben von dieser Floppy gelöscht habe, war es zwar fehlerfrei gelöscht, aber trotzdem erntete ich einen BDOS-Err, d.h. der Kontroll-Lesezugriff von RAM 4.x nörgelte. Nach dieser Aktion war jeglicher Zugriff auf die SRAM-Floppy unmöglich. Selbst nach einem Reset ging kein Booten von diesem Teil. **ABER** ein einfaches Aus/Anschalten des Rechners, und alles lief wieder 100%-ig - bis zum nächsten Schreibzugriff auf diese Floppy. Wenn ich RAM 4.5 so installiere, daß es kein Schreib/Lese-Verify für die RAM-Floppies mache, kann ich sogar mehrmals löschen, muß aber vor dem nächsten Booten die Kiste Aus/Anschalten! **Wer weiß Rat ??**



Hardware: Maus**Anschluß einer seriellen Maus**

(Herbert zur Nedden, 2000)

Nach einigen Frustrationen, gefolgt von einer Keine-Lust-Pause, einem Test an einem echten MsDos-Recher (den ich für meinen Bruder gekauft und installiert habe), und einer daraufhin mit viel mehr Informationen gefolgten Untersuchung am Memotech ergab: 'Hurra, es klappt'.

Der Grund, warum die Maus bei mir zuerst nicht wollte, liegt in einer Unfeinheit der MsDos-Welt. Bei beiden Geräten (MTX + MsDos) gehen die Eingangs-Signale der seriellen Schnittstellen durch einen Konverter, der den  $\pm 12$ -Volt-Pegel der seriellen Schnittstelle auf den 0-5-Volt-Pegel für den DART, oder wie der Baustein auch immer heißen mag, umsetzt. Nur war Memotech (zu) gründlich beim Design, und hat die Eingangs-Signale vor diesem Konverter (d.h. an dem Ende, wo das Signal ankommt) einem Widerstand von 1 kOhm gegen +5 Volt versehen, damit bei offener Schnittstelle hier keine undefinierten Pegel auftreten. In der MsDos-Welt ist dieser Widerstand nicht vorhanden. Ich kann zwei Memotechs via RS232 und 100 Meter Kabel sicher verbinden, bei MsDos soll das kritisch sein. Und meine Maus (ich meine nicht meine Frau damit!!!) sendet ein derart schwaches Signal, daß dieser 1 kOhm-Widerstand genügt, es tot zu machen. Widerstand raus: Alles tut.

Wie ich das und weiteres herausgefunden habe:

1. Maus an MsDos angeschlossen und mal geschaut, was sich so auf den Leitungen tut. Zu untersuchen waren RxD, TxD, RTS und DTR (aus Sicht der RS-232, d.h. RxD ist das einzige Signal von der Maus an den Rechner). Erfreulicherweise lagen die drei vom Rechner gelieferten Signale konstant auf -12 Volt. Die Maus lief am MsDos auch fehlerfrei, nachdem ich TxD, RTS und DTR alle zusammen an einen dieser Pins gelötet hatte.  
Fazit: Die Maus erhält vom Rechner keine Initialisierungs-Befehle.
2. Was die Maus von sich gibt, war mittels Oszillo gut zu sehen. Also mein Testprogramm, welches Daten mit einer RS-232 sendet, und über die andere des gleichen Rechners empfängt, gestartet, und so lange an der Übertragungsfrequenz gedreht, bis meine Signale denen der Maus entsprachen. Allerdings habe ich getippt, daß die Maus keine Parity und 2 Stop-Bits sendet - daß sie mit 8 Bit Daten arbeitet habe ich einfach angenommen.
3. Soweit so gut. Maus an Memotech angstöpselt, RS232 initialisiert und auf Empfang: NICHTS! Auf der RxD-Leitung kam zwar etwas, was von der Form her richtig aussah an, nur daß statt der  $\pm 12$  Volt nur von  $5 \pm 0.5$  Volt die Rede war. Die Kontrolle der Einstellung des Spannungsbereiches meines Oszis änderte nichts an diesem Ergebnis. Nachdem ich RxD von Memotech getrennt hatte, war dieses Signal wieder o.k. Ein Anruf bei Horst Kupka, der mir über die Leitungs-Längen-Probleme in der MsDos-Welt berichtete, bestätigte meine sehr wage Vermutung, das da ein kiloOhm zu viel im Spiel ist.
4. Diesen dummen 1 kOhm-Widerstand zwischen RxD und +5 Volt ausgelötet. So, nun kamen die Daten an. Und nach einigem Geteste lief die Maus, d.h. ich erhielt ihre Daten und zeigte sie an. Dabei ist zu bemerken, daß das Einlesen sehr schnell gehen muß, da die Maus jeweils fünf Bytes auf einen Schlag, d.h. in sehr rascher Folge sendet. Mit sehr schnell meine ich, daß unter 4 MHz und Turbo-Pascal ein Polling-Betrieb (d.h. ständiges Abfragen des DART-Statusregisters) keine Chance hat, die fünf Bytes zu erwischen. Selbst meine Assembler-Routine 'Read:' auf der übernächsten Seite schafft nur Bytes 1-3 sicher unter 4 MHz. Da mein Maustreiber per Interrupt läuft: null Problemo.
5. Meine Maus hat einen Schalter, mit dem ich festlegen kann, ob ihre drei Tasten oder nur zweie wirken sollen. In der Schalterstellung zwei läuft die Maus nimmer am Memotech - vermutlich will sie dann initialisiert werden.

Hardware: Maus

Eh ich's vergesse, die Maus, die ich untersucht und angeschlossen habe ist die

GENIUS Dyna-Maus GM-6000

Warum ich gerade diese Maus untersucht habe, hat zwei Gründe:

1. sollte es eine serielle Maus sein, damit ich nicht irgendwelche Hardware zum Analysieren von merkwürdigsten Zustandsmeldungen oder mit etwas Glück lediglich einen zusätzlichen Eingabeport basteln mußte. Der JoyStick-Port ist tabu, weil er zur Tastatur parallel liegt.
2. diese Genius GM-6000 ist erfreulich preiswert und trotzdem gut: Sie kostet unter DM 100.-, rollt recht gut, hat drei Tasten, einen 9-poligen weiblichen d-SUB-Stecker, einen Adapter auf einen 25-poligen weiblichen d-SUB-Stecker, Maus-Pad (= unnötige Rollunterlage), Halte-Tasche zum ankleben/schrauben. Außerdem ist die Konkurrenz bei seriellen Mäusen teurer oder schlechter.

Als BonBon kann die serielle Maus über die RS 232 und den DART ihre Daten per Interrupt abliefern, d.h. ich vergeude keine Rechenzeit damit diese Teil zu befragen, ob ich mich ihm widmen soll!

Ich habe also für DIESE Maus einen Treiber geschrieben, der sich in den RAM 4.5-Tastaturtreiber einschleift. Mein Maustreiber unterstützt nicht unbedingt andere Mäuse - es sei denn, diese senden die selben Daten und haben ein gleiches Interface. Ich glaube, daß das in Anbetracht der Qualität und des Preises meiner Maus nicht so schlimm ist. Da ich die Sources des Maustreibers auf der KLICK.006 freigegeben und mit dem einen oder anderen Kommentar versehen habe, steht einem Versuch Deinerseits, eine andere Maus oder einen Trackball anzuschließen nichts im Wege, solange dieses Teil serielle Daten sendet, da mein Treiber auf DART-Interrupts basiert, um das System mögl. wenig zu belasten!

Wie Ihr sicherlich schon ahnt, müßt Ihr etwas löten:

1. Der Widerstand von 1 kOhm vom RXD-Eingang der RS232 muß weg. Dümmlischerweise ist dies nicht der einzige solche Widerstand, und Memotech hat daher ein Widerstands-Array eingesetzt. Welcher der Widerstände weg muß, müßt Ihr mit einem Ohm-Meter leicht feststellen können. Bei der RS232-A ist das der mit Pin 2, bei der RS232-B der mit Pin 3 der 25-poligen Buchse verbundene Widerstand. (Das Widerstands-Array ist einfach eine Reihe von Widerständen, deren eines Ende auf je einen Pin des Arrays geführt wird, während alle anderen Enden der Widerstände zusammengeschlossen an einem der End-Pins des Arrays herausgeführt sind. Damit hat das Widerstands-Array eine gewisse Ähnlichkeit mit einem Puppen-Kamm. Auf der RS232/Interface-Platine habt Ihr es mit dem Array RN1 zu tun.)
2. Der schöne 9-ploge Stecker der Maus:

Pin-Nr.	Farbe	Bedeutung der Drähte meiner Maus:
2	Weiß	RXD (Daten von Maus zum Rechner)
3	Rot	TxD (Daten vom Rechner zur Maus)
4	Braun	DTR (vom Rechner zur Maus)
5	Schwarz	Masse
7	Blau	RTS (vom Rechner zur Maus)
Gehäuse	Grün	Masse

Falls Ihr den Stecker tatsächlich (d.h. so wie ich) abkneift, bitte unbedingt die Farb-Pin-Zurodnung nachmessen (Ohm-Meter), da man nie weiß, ob die Farben bleiben! Wenn Ihr die Maus aufschraubt, könnt Ihr zuvor prüfen, ob die Drähte überhaupt noch farbig sind - aber das wird sicherlich der Fall sein.

H a r d w a r e: Maus

3. Dieser 9-polige Stecker muß abgekniffen und durch einen Memotech-freundlicheren 25-poligen ersetzt, oder über einen Konverter sonst irgendwie an unsere RS-232 angepaßt werden. Der mit der Maus gelieferte Adapter ist ungeeignet, da bei MsDos die RS-232-Stecker eigentlich Buchsen sind.

Unter Verwendung meiner o.g. Farben, je nach der gewünschten RS-232, müssen die Drähte an die folgenden Pins der 25-poligen RS-232-Buchse des Memotech:

RS-A	RS-B	(beachte: eigentlich genügt ein 3-poliger Stecker!!)
2	3	RxD: Weiß
5	20	DTR: Rot, Braun und Blau (ja!!!!, drei auf einen Streich)
7	7	Masse: Schwarz und Grün

**Einlesen der Maus ohne Interrupts**

(Herbert zur Nedden, 2000)

1. CTC und RS-232 initialisieren:

Ports: RS-A: CTC = 09h, DartCtl = 0Eh, DartData = 0Ch  
 RS-B: CTC = 0Ah, DartCtl = 0Fh, DartData = 0Dh

Baudrate von 1200 Baud einstellen:

folgende Daten auf den Port CTC schicken: (dezimal)

3 3 69 16: Zähler, Teilen durch 16

DART fertigmachen, d.h. initialisieren:

folgende Daten auf den Port DartCtl schicken: (dezimal)

1 0 2 0 3 225 4 76 5 104: 8 Bit, 2 Stop, keine Parity, DTR=RTS=0

2. RS232-Lesen der Maus:

```

Read:  ld      c,DartCtl      ; Port
       ld      hl,Tasten     ; Daten-Puffer
       call   WaitSync      ; 1. Byte = Tasten
       inc    hl
       call   GetMaus       ; 2. Byte = X
       inc    hl
       call   GetMaus       ; 3. Byte = Y
       ret
  
```

```

WaitSync: call   GetMaus      ; Warten auf 1. Datenbyte einer Sequenz
          and   0f8h         ; d.h. Lesen, bis ein Byte der Form
          cp   080h         ; 10000??? Binär angekommen ist
          jp   nz,WaitSync  ; (dieses Byte liefert die Tasten)
          ret
  
```

```

GetMaus:  in     a,(c)        ; Warten auf Daten
          bit   0,a
          jp   z,GetMaus
          in   a,(DartData)  ; einlesen
          ld   (hl),a        ; speichern
          ret
  
```

```

Tasten:  db      0           ; Taste gedrückt: Bit = 0
          ; Bit 0 = Rechts
          ; Bit 1 = Mitte
          ; Bit 2 = Links
  
```

```

X:       db      0           ; X-Änderung ) Binärzahl mit Vor-
Y:       db      0           ; X-Änderung ) zeichen in Bit 7
  
```

H a r d w a r e: Maus

**Der Maustreiber** <--- benötigt RAM 4.5 (Herbert zur Nedden, 2000)

Um die GENIUS Dyna-Maus GM 6000 Maus zu aktivieren, genügt es (nachdem die Hardware wie oben angegeben vorbereitet wurde), mein KLICK-Overlay **Mouse 1** zu laden. Dadurch wird mein Maustreiber in den RAM 4.5-Tastaturtreiber eingeschleift. Dabei werden die Maus-Daten per DART-Interrupt eingelesen, damit zum einen keine Daten verloren gehen, und sich zum andern die CPU nur dann um dieses Teil kümmert, wenn es auch etwas von sich gibt. Hat die Maus einen Satz Daten (5 Bytes) abgeliefert, wird diese Information ausgewertet und die entsprechenden Tastencodes in den RAM 4.x-Tastaturpuffer gestellt.

**WICHTIG:** Bevor das KLICK-Overlay eingesetzt wird, muß es installiert werden, was insbesondere die Port-Adressen des CTC und des DART betrifft.

Klarerweise liegt die Mausbewegung auf den Pfeiltasten. Da die Maus unheimlich schnell in ihren Bewegungsmeldungen ist, kann eine Ansprechschwelle installiert und jederzeit im KLICK verändert werden. (Die Maus sendet ihre Bewegung über zwei Bytes, eines für die X- und eines für die Y-Richtung. Dabei sind diese Bytes als Binärzahlen mit Vorzeichen in Bit 7 zu interpretieren, was die Verarbeitung in Assembler sehr einfach gestaltet.)

Die mittlere Maustaste liegt natürlich auf Home, die linke auf Maus-Shift. Die rechte Maustaste ist frei installierbar, und zwar auf eine beliebige Taste außer Shift, AlphaLock oder LineFeed. Dazu muß lediglich deren Tastaturcode installiert werden, wobei ein Code von Null (00h) diese Taste als Maus-Ctrl wirken läßt. Über das KLICK-Overlay kann diese Taste jederzeit auf einen beliebigen über die Tastatur erreichbaren Code umgelegt werden, wobei Maus-Ctrl als Ctrl-Leerzeichen einzugeben ist, da eben diese Tastenkombination 00h liefert.

Nun zu Maus-Shift und Maus-Ctrl: Für Tasten, die die Maus liefert, und die zum 10-er-Block gehören, d.h. insbesondere die Funktionstasten im Sinne von RAM 4.x sind, wirken Maus-Shift und Maus-Ctrl eben so, wie erwartet. D.h. in Verbindung mit der mittleren Maustaste (Home) und der Mausbewegung (Pfeile) ist alles bestens. Liegt die rechte Maustaste jedoch auf einer F- oder Nicht-Funktionstaste, so liefert diese zusammen mit Maus-Shift i.a. nicht die geschiftete Taste, da Maus-Shift lediglich eine Addition von 10h auf den Tastencode bewirkt. Liegt die rechte Maus-Taste z.B. auf Esc (1bh), so liefert sie zusammen mit Maus-Shift 2bh als Code, und nicht das KLICK! Der Grund für diese Lösung ist zum einen, daß ich damit einen deutlich einfacheren und daher schnelleren Treiber schreiben konnte und es außerdem in meinen Augen eh nicht sinnvoll ist, die rechte Maustaste nicht auf eine Funktionstaste zu legen, da so nur Flexibilität verloren ginge. Du kannst ja die rechte Maustaste ja z.B. auf Ctrl-PAGE legen, also Deinen eigentlichen Funktionstasten-Bereich vermutlich unberührt lassen.

**HINWEIS:** Maus-Shift und -Ctrl wirken NUR für die Maus. Genauso wirken die normalen Shift, Ctrl, AlphaLock und LineFeed nur für die Tastatur!

Natürlich werden die Maustasten entprellt. Dabei habe ich mich entschlossen, die Entprellung wie RAM 4.x auf Tasten-Ebene zu machen. Wird also die mittlere Maustaste (Home) festgehalten, und die linke Maustaste (Maus-Shift) gedrückt und losgelassen, so liefert der Maus-Treiber nur Home. Die Mausbewegung wird dadurch entprellt, daß die Bewegung erst die o.g. Ansprechschwelle überschreiten muß, eh sie in die entsprechende Pfeiltaste umgesetzt wird.

In Anbetracht dessen, daß die Pfeile und Home mit und ohne Shift/Ctrl alles RAM 4.x-Funktionstasten sind, kann die Maus immerhin 20 Tasten liefern, hinter denen sich Funktionstasten verbergen.

Hardware: DART per Interrupt für die Maus

Über das KLICK-Overlay kann die Maus jederzeit an- und abgeschaltet werden, damit der Cursor nicht unnötig durch die Gegend flitzt, nur weil man aus Versehen die Maus vom Tisch fällt - äh - ich meine, wenn die Maus vorübergehend nicht benötigt wird. Hierbei wird der Maustreiber nicht aus dem System entfernt, sondern lediglich die Auswertung und Übergabe der Mausinformation in den Tastaturpuffer abgeschaltet. Der Maustreiber wird hierbei bewußt nicht vollständig deaktiviert, sondern die Maus-Interrupts auch weiter laufen gelassen, da das Abschalten der Interrupts eine Umprogrammierung des DART bedeutet, was ich vermeiden wollte. Willst Du den Maus-Interrupt vollständig abschalten, mußt Du lediglich das Klick-Overlay mit Olaf Krumnow's KILL5 aus dem Heap entfernen.

**Wie der DART warum initialisiert werden muß**

(Herbert zur Nedden, 2000)

Einen Z80-DART muß man unter Interrupt-Betrieb immer als eine Einheit betrachten, d.h. man kann die beiden Kanäle nicht völlig unabhängig für Interrupts initialisieren. Was ein Wunder auch, ist es schließlich auch nur EIN Baustein.

Ich werde im folgenden NICHT detailliert auf die verschiedenen Möglichkeiten, wann und wofür der DART wieso und weshalb einen Interrupt beim empfangen welcher Dinge auch immer auslösen kann, da ich das selbst (noch ?) nicht so ganz verstanden habe. Wenn ich den DART per Interrupt betreibe, dann so, daß jedes ankommende Zeichen einen Interrupt auslöst!

Der DART kann, wie es sich für die Z80-Welt gehört, beim Auslösen eines Interrupts die untere Hälfte der Adresse des Interrupt-Vektors liefern. Die obere hat die Z80-CPU schon in ihrem I-Register (zu haben). Und an dieser zusammengesetzten Adresse findet die CPU dann die Adresse des Interrupt-Handlers. Da nun ein DART immerhin aus verschiedenen Gründen interrupten kann (Sende-Puffer leer, Zeichen angekommen, externer Interrupt (?) sowie besondere Empfangsbedingung), und das für zwei Kanäle, ist er in der Lage acht verschiedene Interrupt-Handler zu adressieren. Dafür modifiziert er einfach seine Hälfte der Adresse des Interrupt-Vektors. Die Interrupt-Vektortabelle, in der die Adressen der Interrupt-Handler stehen, hat folgenden Aufbau:

1. Kanal B: Sendepuffer ist leer
2. Kanal B: Externer- oder Status-Interrupt
3. Kanal B: Zeichen ist angekommen
4. Kanal B: besondere Empfangsbedingung
5. Kanal A: Sendepuffer ist leer
6. Kanal A: Externer- oder Status-Interrupt
7. Kanal A: Zeichen ist angekommen
8. Kanal A: besondere Empfangsbedingung

Der DART erwartet daher, daß diese Tabelle von acht Adressen auf einer Adresse beginnt, deren unteren vier Bits Null sind. Da das Ganze auch noch unter RAM 4.5 laufen soll, ist wegen der Interruptnutzung durch RAM 4.5 für die Tastatur und die Uhr die Wahl der Lage dieser Interrupt-Tabelle weiter eingeschränkt. Ich habe einfach die von RAM 4.5 vorgesehene Tabelle an der Adresse 0F080h gewählt. Die zweite an 0FOA0h ist natürlich auch möglich.

Um den DART zu initialisieren müssen die Informationen zur Lage und Behandlung dieser Tabelle dem DART übermittelt werden, nämlich die untere Adreßhälfte (80h bzw. 0A0h in den o.g. Beispielen), sowie daß er auch die gesamte Tabelle, und nicht immer den ersten Tabelleneintrag nutzen soll. Und just diese Übergabe dieser beiden Anweisungen an den DART machen den eigentlichen Ärger aus, da sie nur im Kanal B für den gesamten DART gesetzt werden. (Man kann nämlich den DART dazu verleiten, nur einen Interrupt-Handler anzusprechen, was bedeutet, daß dieser erst das DART-Statusregister lesen müßte, um zu erfahren was Tango ist).

H a r d w a r e: DART per Interrupt für die Maus

Damit ergeben sich folgende Einschränkungen, die allerdings eigentlich nicht weiter schädlich sind:

1. Wird DART-Kanal B für die Maus gewählt, bleibt Kanal A von seiner Initialisierung unverändert, d.h. so wie er war.  
Läuft Kanal A unter Interrupts, muß die Basis-Adresse der Interrupt-Vektortabelle, die dort im Einsatz ist, in diesem Programm installiert werden, falls ungleich 0F080h, da diese neu gesetzt wird!!!
2. Wird DART-Kanal A für die Maus gewählt, werden Interrupts von Kanal B abgeschaltet, da dessen Register 1 neu geschrieben werden muß.  
Soll jedoch auch Kanal B mit Interrupts laufen, so muß dieser nach dem diese Mausroutine eingeschleift wurde initialisiert werden. Hierbei ist allerdings wieder zu beachten, daß beide Kanäle eine gemeinsamen Interrupt-Vektortabelle benutzen müssen.
3. Soll der freie DART-Kanal eh nicht via Interrupts laufen, werden nur die Parametern der benötigte Schnittstelle umprogrammiert. Die andere bleibt so wie sie vorher war initialisiert.
4. Soll der freie DART-Kanal allerdings via Interrupts laufen, muß er ebenfalls die vier möglichen Interrupt-Vektoren nutzen. D.h. die Variante, daß der DART immer nur den ersten Tabellen-Eintrag nutzt, ist nicht.
5. Falls Du mein KLIICK-Overlay RsInit benutzt, muß die für die Maus genutzte RS232 unbedingt dort herausgenommen werden. Das gilt übrigens für deren BEIDE Kanäle, falls die Maus am Kanal A hängt, oder der freie Kanal über Interrupts läuft.

Hier zur Information die Initialisierungswerte für die DART-Register. Register 2 (gibt es nur für Kanal B) wird mit der unteren Hälfte der Adresse der Interrupt-Vektortabelle gefüllt, die anderen vier wie folgt:

```

DartReg1:    db      00010100b
              ;          ^----- Keine externen Interrupts
              ;          ^----- Keinen 'Sendepuffer leer' Interrupt
              ;          ^----- Alle acht interrupt Vektoren nutzen
              ;          ^^----- Je empfangenen Zeichen ein Interrupt,
              ;                   Parity-Fehler -> Fehler-Interrupt
              ;                   Wait/Ready unterbunden
DartReg3:    db      11100001b
              ;          ^----- Empfangsteil eingeschaltet
              ;          ^----- Hardware-Handshake eingeschaltet (egal!)
              ;          ^^----- 8 Bit je empfangenen Zeichen
DartReg4:    db      01001100b
              ;          ^^----- Keine Parity
              ;          ^^----- Zwei Stop-Bits
              ;          ^^----- Datenrate = 16x Taktrate
DartReg5:    db      01101000B
              ;          ^----- RTS = 0
              ;          ^----- Sendeteil eingeschaltet
              ;          ^----- Keine Breaks senden (egal)
              ;          ^^----- 8 Bit je gesendetem Zeichen
              ;          ^----- DTR = 0

```

Falls Interesse besteht, bin ich gerne bereit, im nächsten Info einen Artikel über den Aufbau der erforderlichen Interrupt-Routinen zu beschreiben.

Hardware: Hard-DiscHarddisk Anschluß an die schwarze Kiste

(Hannes Mehrrens, 7900)

Grundlagen:

Wie bei normalen Floppy-Laufwerken benötigt man für Harddisks einen speziellen Controller. Hierfür sind Karten für den ECB-Bus verfügbar (siehe c't 8/86 ff.). Ich favorisiere die Verwendung von fertigen Harddisk-Controllern, die eigentlich für IBM PC/XT's angeboten werden. Beide Lösungen haben Vor- und Nachteile.

**Harddisk-Controller für den ECB-Bus:**

- Passen direkt an den ECB-Bus
- Sind meist Bausätze und müssen abgeglichen werden (heikel!!!)
- Haben keine eigenen Prozessor zur Steuerung
- Ihre Programmierung ist daher recht aufwendig

**Harddisk-Controller für IBM-PC/XT's:**

- Können nur über ein Interface an den ECB-Bus oder MTX/FDX-Bus angeschlossen werden
- Sind fertig aufgebaut, müssen nicht abgeglichen werden
- Besitzen meist einen eigenen Prozessor zur Steuerung
- Ihre Programmierung ist daher relativ einfach
- Sind recht billig zu bekommen

Leider sind die PC/XT-Harddisk-Controller in der Programmierung nicht zueinander kompatibel. Unter der Flut der angebotenen Controller haben die Typen OMTI 5510, OMTI 5520 und OMTI 5527 weite Verbreitung, speziell an nicht IBM-PC/XT-Computern gefunden. Die Zeitschrift c't hat mehrere Artikel über den Anschluß dieser OMTI-Controller an den ECB-Bus (siehe c't 4/87, c't 9/87, c't 10/87), an den Amiga und den Atari veröffentlicht. Die OMTI-Controller sollen übrigens zu den Schnellsten gehören.

Die 3 genannten HD-Controller sind in der Programmierung fast identisch.

**Eigenschaften und Unterschiede der 3 HD-Controller:****OMTI 5510:**

- Die Größe und Anzahl der Sektoren/Spur läßt sich über Jumper einstellen. Dies erlaubt eine bessere Ausnutzung der Harddisk (größere Kapazität)
- Ansonsten ist der OMTI 5510 identisch mit dem OMTI 5520.

**OMTI 5520:**

- Erlaubt nur den PC-Standard von 17 Sektoren mit 512 Bytes pro Spur. Gegenüber dem beim OMTI 5510 einstellbarem Format von 18 Sektoren mit 512 Bytes pro Spur, ergibt sich hier eine etwas geringere Kapazität.
- OMTI 5510 und OMTI 5520 sind für den Einsatz von Standard-Harddisks (MFM-Laufwerke)

Hardware: Hard-Disc**OMTI 5527:**

- Ist nur für den Einsatz der neueren RLL-Laufwerke.  
Diese RLL-Laufwerke ermöglichen eine höhere Übertragungsgeschwindigkeit, und haben wegen des hier verwendeten Formats von 26 Sektoren mit 512 Bytes pro Spur eine um ca. 50% höhere Kapazität als MFM-Laufwerke.  
Der Preis für 1Mbyte Kapazität bei RLL-Controller/RLL-Harddisk ist geringer als für die MFM-Controller/MFM-Harddisk Kombination. Sollten mehr als die üblichen 20MByte der MFM-Harddisks benötigt werden ist eine RLL-Controller/RLL-Harddisk Kombination sicherlich preiswerter.  
Nach Angaben eines Verkäufers sollen die RLL-Laufwerke einwenig empfindlicher gegenüber Hitze und Änderungen der Versorgungsspannung zu sein als MFM-Laufwerke. Man sollte hier also für ausreichende Kühlung (Lüfter) und eine stabile Versorgungsspannung (notfalls seperates Netzteil) sorgen.

Da ich nur OMTI 5520 besitze kann ich leider keine Geschwindigkeitsunterschiede zwischen den 3 Controllern angeben. Die obigen Informationen sollen nur einen Überblick geben, bei Bedarf ist das jeweilige Manual zu Rate zu ziehen!

Die augenblickliche Treiber-Software ist für den OMTI 5520 Controller ausgelegt. Ich werde diese aber in Kürze so anpassen, daß alle 3 OMTI Controller sowie ein weiterer Spezial-Controller unterstützt werden. (Siehe weiter hinten)

**Wissenswertes über Harddisks:**

Die am weitesten verbreitete und wohl billigste 5.25" MFM-Harddisk ist momentan die ST225 von Seagate. Es handelt sich hier um ein MFM-Laufwerk mit 4 Köpfen und 615 Cylindern (=Tracks). Die formatierte Kapazität beträgt laut Hersteller 21.4 MBytes. Am OMTI 5520 mit 17 mal 512 Bytes pro Spur ergibt sich eine Kapazität von  $17 \cdot 512 \cdot 4 \cdot 615 = 20.4$  MByte.

Die bei Harddisks angegebenen Zeiten (bei der ST225: 65ms) geben meines Wissens die mittlere Zeit an, die gebraucht wird um die Köpfe auf eine bestimmte Spur zu positionieren.

Wichtiges Kriterium ist auch, ob die Harddisk automatisch parkt. Während die Harddisk läuft "fliegen" die Köpfe auf einem Luftpolster dicht über den Platten. Beim Abschalten baut sich dieses Luftpolster ab, und die Köpfe landen direkt auf der Plattenoberfläche. Dabei kann die Magnetschicht "abgekratzt" werden und Daten gehen an der entsprechenden Stelle verloren. Die meisten modernen Harddisks parken deshalb nach dem Abschalten automatisch die Köpfe auf einer nicht benutzten Spur. Die ST225 bietet diesen Luxus nicht. Vor jedem Abschalten müssen daher die Köpfe mittels eines Park-Programmes auf eine nicht verwendete Spur geparkt werden.

Die ST225 verbraucht ca. 14.8 Watt (+12V und +5V). Sie gibt beim Laufen ein deutliches gleichmäßiges Brummen von sich, welches aber vom Lüfter der FDX überhört wird.

Bei Platzproblemen könnte auch eine 3.5" Harddisk verwendet werden. Diese kompakteren Laufwerke sind übrigens auch leiser, verbrauchen weniger Strom und erzeugen damit auch weniger Wärme. Sie sind allerdings deutlich teurer.

In der c't 2/89 sind übrigens einige nützliche Artikel über Festplatten enthalten



Hardware: Hard-DiskVoraussetzungen für den Anschluß einer Harddisk**Hardware Voraussetzungen:**

- Harddisk-Controller OMTI 5520  
(OMTI 5510 und OMTI 5527 sind auch verwendbar, die Software wird diese aber erst in Kürze unterstützen)
- Harddisk, z.B.: ST225 (5.25", 20MB)
- Stromversorgung (+12V +5V für Harddisk, ST225 braucht ca. 14.8 Watt, Amperezahl nicht bekannt; +5V ca. 500 mA, +12V <50 mA für den Controller)
- Interface zwischen FDX/SDX und Harddisk-Controller
- 4 freie Portadressen für den Harddisk-Controller

**Software Voraussetzungen:**

- FDX/SDX mit RAM4.x
- Treiber-Software, die bei mir in einem Klick untergebracht ist
- Entsprechende Utility-Programme zum Installieren des Treibers, zum Formatieren der Harddisk, etc.

**Funktionsweise des Harddisk-Anschlusses:**

Das Interface nimmt die nötige Anpassung an den PC-Anschluß des HD-Controllers vor, außerdem wird hier die Port-Dekodierung vorgenommen. Die ganze Dekodierung ist momentan in einem GAL-IC untergebracht. Dadurch konnte der Verdrahtungsaufwand auf ein Minimum reduziert werden. Die Basis-Portadresse läßt sich momentan aber nur durch neues Programmieren des GAL-ICs verändern.

Der HD-Controller wird über ein Flachband-Kabel (momentan 20-polig) an das Interface angeschlossen. Der HD-Controller (Maße: ca. 12x13x1.8cm) müßte sich daher leicht an einer geeigneten Stelle unterbringen lassen. Die Stromversorgung erfolgt entweder über das Interface (FDX-Netzteil) oder über ein separates Netzteil.

Die Harddisk wird über 2 Flachband-Kabel mit dem HD-Controller verbunden. Die Stromversorgung erfolgt genau wie bei normalen Laufwerken über einen speziellen Stecker. Eine 5.25 Zoll Harddisk läßt sich wie ein normales Laufwerk (2. Laufwerk) in die FDX einbauen. Falls das FDX-Netzteil bereits zu stark belastet ist, unbedingt die Harddisk mit einem separaten Netzteil versorgen.

Sollten bereits 2 Laufwerke in der FDX eingebaut sein, dann läßt sich sicher immernoch eine 3.5" Harddisk (z.B. ST125) unterbringen. 3.5" Harddisks sind zwar teurer, verbrauchen aber auch weniger Strom (ST125: 10W), brauchen weniger Platz und sind leiser.

Der Harddisk-Controller wird über die 4 Ports programmiert, Daten werden ebenfalls über die Ports transferiert. Der OMTI 5520 HD-Controller kann 2 Harddisks "verwalten". Das Format ist hier fest eingestellt auf 17 physikalische Sektoren mit 512 Bytes pro Spur.

Selbst unter Verwendung der bis zu 31 User-Bereichen (?) wird ein Directory für solch einen großen Bereich schnell unübersichtlich. Auch werden die Zugriffszeiten bei voller Harddisk größer, da zwischen den Directory-Spuren und den äußeren noch freien Spuren eine immer größere Anzahl von Tracks zurückgelegt werden müssen. Es ist daher sinnvoll, eine physikalische 20MB Harddisk in ca. 4 logische Laufwerke mit je etwa 5MB aufzuteilen. Die augenblickliche Treiber Software unterstützt bis zu 5 logische Laufwerke.

Hardware: Hard-Disc**Vorteile einer Harddisk:**

- Gegenüber Floppy-Laufwerken schnellerer, leiser Zugriff auf viele Programme, Daten, etc.
- Booten mit der Harddisk müßte das Hochfahren der FDX/SDX innerhalb weniger Sekunden ermöglichen.
- Gegenüber RAM-Floppies zwar etwas langsamer, dafür aber erheblich billiger und enorme Speicherkapazität.

**Kosten für den Harddisk-Anschluß:**

Harddisk ST225:	ca. 450.-	(teilweise schon für 428,- zu haben)
OMTI-5520-Controller:	ca. 200.-	(Preisspanne: 150,- bis 230,-)
	=====	
	ca. 650.-	
Interface, Kabel:	?.?	
Programmiertes GAL-IC, (GAL wird nur bei Verwendung meines Interfaces benötigt):	15,-	Bei mir erhältlich
Treiber-Software, Utility-Programme:	18,-	Bei mir erhältlich

Die DM 18,- für meinen Treiber dürften für die meisten von euch recht hoch erscheinen. Dazu möchte ich folgendes bemerken:

1. Für die DM 18,- bekommt man nicht nur den Treiber, sondern auch mehrere Utility-Programme, deren Entwicklung sehr viel Zeit gekostet hat.
2. Ich werde den ersten 5 Bestellern meine Treiber-Software kostenlos zur Verfügung stellen. Dies wird einen präzisen Test ermöglichen. Eventuell noch versteckte Fehler können so beseitigt werden.
3. Nach dieser Phase bekommt man also benutzerfreundliche Qualitäts-Software für sein Geld.
4. Meine Treiber-Software unterstützt in Kürze alle 3 OMTI Controller. Es werden bereits jetzt bis zu 2 physikalische Harddisk-Laufwerke unterstützt.
5. Zusätzlich zu den OMTI-Controllern wird meine Software einen weiteren speziellen Controller unterstützen. Dieser ermöglicht den Einsatz der neuen, durch moderne Technologie sehr preiswerten Harddisks.  
Eine Harddisk/Controller-Kombination (20 MByte) diesen Typs kostet zur Zeit DM 493,- .  
Gegenüber der OMTI 5520/ST225-Kombination (ca. 150,- + 428,- = DM 578 ) ergibt sich also eine Ersparnis von DM 85,- !  
Mit meiner Software kann man also eine Menge Geld sparen.

Informationen über den augenblicklichen Stand können bei mir erfragt werden:  
Hannes Mehrrens, Gutenbergstr.6 G005, 7900 Ulm, Tel.: 0731/28358.

**Ausblick auf zukünftige Entwicklungen/Verbesserungen:**

- Die Uebertragung der Daten zwischen Harddisk-Controller und MTX könnten mit Hilfe eines DMA-Controllers (Z80-DMA) um einiges beschleunigt werden.

Hardware: Hard-Disc

Verdrahtungsplan für das Interface zwischen OMTI-Controller und MTX/FDX

Benötigte Z80-Signale  
(Hier ECB-Bus):

Beschaltung des IBM-PC/XT-Bus  
(Hier am OMTI-Controller)

+5V	=! A1	B1 !=	+5V	=! A1	B1 !=	GND
D5	=! A2	B2 !=	D0	=! A2	B2 !=	RESET
D6	=! A3	B3 !=	D7	=! A3	B3 !=	+5V
D3	=! A4	B4 !=	D2	=! A4	B4 !=	
D4	=! A5	B5 !=	A0	=! A5	B5 !=	
A2	=! A6	B6 !=	A3	=! A6	B6 !=	
A4	=! A7	B7 !=	A1	=! A7	B7 !=	
A5	=! A8	B8 !=		=! A8	B8 !=	
A6	=! A9	B9 !=	A7	=! A9	B9 !=	+12V
	=! A10	B10 !=		=! A10	B10 !=	GND
	=! A11	B11 !=		=! A11	B11 !=	
	=! A12	B12 !=		=! A12	B12 !=	+5V
+12V	=! A13	B13 !=		=! A13	B13 !=	/IOW
	=! A14	B14 !=	D1	=! A14	B14 !=	/IOR
	=! A15	B15 !=		=! A15	B15 !=	+5V
	=! A16	B16 !=		=! A16	B16 !=	
	=! A17	B17 !=		=! A17	B17 !=	
	=! A18	B18 !=		=! A18	B18 !=	
	=! A19	B19 !=		=! A19	B19 !=	
	=! A20	B20 !=		=! A20	B20 !=	
	=! A21	B21 !=		=! A21	B21 !=	
	=! A22	B22 !=	/WR	=! A22	B22 !=	+5V
	=! A23	B23 !=		=! A23	B23 !=	+5V
	=! A24	B24 !=	/RD	=! A24	B24 !=	GND
	=! A25	B25 !=		=! A25	B25 !=	GND
	=! A26	B26 !=		=! A26	B26 !=	+5V
/IORQ	=! A27	B27 !=		=! A27	B27 !=	GND
	=! A28	B28 !=		=! A28	B28 !=	GND
	=! A29	B29 !=		=! A29	B29 !=	/EN_HD
	=! A30	B30 !=		=! A30	B30 !=	A1
	=! A31	B31 !=	/RESET	=! A31	B31 !=	A0
GND	=! A32	B32 !=	GND	=! A32	B32 !=	GND

Folgende Signale müssen für den OMTI-Controller extra erzeugt werden:

- RESET: negiertes /RESET vom Z80-Bus (ein Inverter wird benötigt)
- /IOR: /IORQ oder /RD (ein ODER-Gatter wird benötigt)
- /IOW: /IORQ oder /WR (ein ODER-Gatter wird benötigt)
- /EN\_HD: "Chip-Select" für den Controller  
Dieses Signal wird von der Port-Dekodierlogik geliefert.

Um den Schaltungsaufwand minimal zu halten, habe ich das Ganze in einem GAL-IC untergebracht. GALs sind wie PAL-Bausteine programmierbar, können aber fast beliebig oft elektronisch gelöscht und dann erneut programmiert werden. Sie haben auch einen recht geringen Stromverbrauch von ca. 45 mA !  
Da nicht alle Ein/Ausgänge des GALs für den OMTI-Controller benötigt werden, erledigt das GAL bei mir zusätzlich noch die Port-Dekodierung für eine PIO 8255.

Hardware: Hard-Disc

Für den OMTI-Controller (und die 8255 PIO) kann jede beliebige Port-Adresse mittels eines Programmiergerätes programmiert werden.

Der OMTI-Controller ist mit meiner ECB-Interface-Karte über ein 20 poliges Flachbandkabel verbunden:

**Momentane Anschlußbelegung des programmierten GALs (Typ 16V8):**

**Anschlußbelegung der 20-poligen Pfostenleiste OMTI/ECB:**

-----		-----	
!	Pin1	Pin20	!
/RESET	Pin2	Pin19	RESET
/IORQ	Pin3	Pin18	/IOW
/WR	Pin4	Pin17	/IOR
/RD	Pin5	Pin16	/EN_HD
A7	Pin6	Pin15	/CS_8255
A6	Pin7	Pin14	
A5	Pin8	Pin13	
A4	Pin9	Pin12	A2
GND	Pin10	Pin11	A3
!			!
-----		-----	

-----		-----	
D0	Pin1	Pin20	D1
D2	Pin2	Pin19	D3
D4	Pin3	Pin18	D5
D6	Pin4	Pin17	D7
A0	Pin5	Pin16	A1
RESET	Pin6	Pin15	/EN_HD
/IOW	Pin7	Pin14	/IOR
GND	Pin8	Pin13	GND
+5V	Pin9	Pin12	+5V
+12V	Pin10	Pin11	+12V
-----		-----	

**Bemerkungen:**

Bitte darauf achten, daß der OMTI-Controller mit ausreichend stabilen Spannungen versorgt wird. Falls das FDX-Netzteil nicht stark genug oder schon stark belastet ist, ein separates Netzteil verwenden! Das gleiche gilt für die Festplatte.

Die Treibersoftware für die Harddisk (Stand 25.5.89)

Ich habe folgende Software entwickelt:

**HDFRMED.COM**

Dieses Programm dient zum Verwalten/Ansehen der Daten von verschiedenen Harddisks. Ich habe hier die Daten einiger Seagate Harddisks eingegeben. Die Daten sind/werden in der Datei HDCHARC.DAT gespeichert. Einige der für INSTHD benötigten Daten können hier entnommen werden.

**INSTHD.COM**

Mit diesem Programm kann man die Daten für die verwendete Harddisk, z.B.: die Seagate ST225, eingeben und in der Datei HD.DAT speichern. HD.DAT wird von einigen Dienst-/Testprogrammen benötigt. Ebenfalls installiert werden können der Sektor Skew und die Basis-Portadresse des OMTI-Controllers. In Kürze wird auch der verwendete Controller (OMTI 5510, 5520 oder 5527) in diesem Programm festgelegt. Nachdem alle Daten korrekt eingegeben worden sind, kann das Treiberprogramm installiert werden.

**HDTEST.COM**

Hiermit kann man den Controller und die Harddisk testen. Der Controller ist in der Lage per Befehl sein internes RAM, sich selbst (ROM,etc.) und die Harddisk zu testen.

**HDFORM.COM**

Dient zum Formattieren der Harddisk. Alle notwendigen Daten (Anzahl der Cylinder, Sektor Skew, ...) werden auch hier aus der Datei HD.DAT entnommen. Nach dem Formattieren werden alle Tracks/Sektoren auf Fehlerfreiheit überprüft.

Hardware: Hard-Disc**PARK.COM**

Bevor der MTX ausgeschaltet wird, müssen manche Harddisks "geparkt" werden. Dieses Programm momentan nur für die ST225 zu verwenden. Geparkt wird auf dem letzten Cylinder.

**HDKLICK.COM**

Installiert das Klick mit der Treiber-Software.

**HDBIOS4.COM**

Dieses Programm muß nach HDKLICK aufgerufen werden. Es installiert die Treiber-Software für die gewünschte Harddisk, und bindet diese in das BIOS ein. Es werden dabei ca. 100 Bytes unter TOAM verbraucht. Der Harddisk-Treiber unterstützt bereits bis zu maximal 5 logische Laufwerke. D.h., die physikalische Harddisk ist in bis zu 5 logische Laufwerke aufgeteilt.

Diese Aufteilung ist einmalig mit INSTHD festzulegen.

**HDINIT.COM**

Nach dem Einschalten muß der Harddisk-Controller initialisiert werden. Die obigen Utility-Programme machen dies alles selbständig. HDBIOS4.COM macht dies momentan noch nicht selber.

**CALCFORM.COM**

Berechnet aus: - Anzahl der logischen Tracks für das Format  
 - Der Blockgröße in k  
 - Anzahl der Directory-Einträge  
 - Anzahl der Systemspuren

sämtliche für FORMATE4 benötigte Parameter. Diese können dann ausgedruckt werden und in FORMATE4 eingegeben werden.

### Installieren eines Formats für die Harddisk mittels FORMATE4

Für Harddisks sind die Formate #20..#2F reserviert.

**Beispiel:**

Notwendige Eingaben für ein Format mit

- 153 logischen Tracks (z.B.: für 4 logische Harddisk-Laufwerke)
- Blockgröße 2k
- 1024 Directory-Einträge (mehr sind bei 2k-Blockgröße nicht möglich)
- 0 Systemspuren

CALCFORM aufrufen und die obigen Daten eingeben. Die nun ermittelten Daten in FORMATE4 eingeben.

Nachdem das Format definiert ist, dieses so in RAM4.x und CFG4 installieren wie dies bei einem normalen Laufwerk-Format gemacht wird.

**Anmerkungen:**

Für das Harddisk-Format (volle 20MByte, also 615 log. Tracks) benötigt man bei 2k-Blockgröße ca. 1300 Bytes. Bei einem 58k-System unter RAM4.2 ist hierfür nicht genug Platz unter TOAM. Die System-Größe müßte in diesem Fall entsprechend erniedrigt werden.

Die augenblickliche Version meines Treibers läuft fehlerfrei mit Formaten mit 2k und 4k Blöcken. 8k und 16k Blöcke habe ich bis jetzt noch nicht getestet.

**Anm.d.HzN.:** Mach doch aus dem HDKLICK.COM ein .KLX. Das kann dann eine INIT-Routine enthalten, die Aufgaben von HDBIOS4 und HDINIT gleich mit erledigen! Außerdem sind .COM-KLICKS out, oder ? Trotz TOLL, daß Du es geschafft hast, den Daten-Brummkreislauf Harddisc ans laufen zu bekommen.

Hardware: Hard-Disc und RAM x.x**Zum Thema Hard-Disk-Treiber**

(Herbert zur Nedden, 2000)

Claudio Romanazzi und Hannes Mehrrens haben je einen Hard-Disk-Treiber geschrieben, die sich in folgenden Punkten unterscheiden:

Claudio: + alles in einem .KLX-Klick-Overlay

(Einfügen in BIOS, Treiber und Park-Routine)

+ beliebige Blocksize

- unterstützt nur zwei OMTI-Controller-Typen 5510 und 5520

+ ist Public-Domain --> KCLICK.007 oder jetzt von Claudio gegen D&P&V

Hannes: - ein .COM-KLICK-Overlay, drei CP/M-.COMs

- bislang nur Blocksize von 2k und 4k sicher unterstützt

(das ist ungeeignet, da schon 1,76 MB-Disketten bei 2k-Blocksize 384 DIR-Einträge benötigen und mit 880 Blöcken zwischen FREE und TOAM auf Bank 0 110 Bytes Allocation-Map belegen).

+ unterstützt in Zukunft drei Typen von OMTI-Controllern sowie einen 'Spezial-Controller'. Zur Zeit jedoch nur den 5520.

- Ist nicht PD (verständlich bei der Arbeit !?)

Beide: - Nicht Bestandteil des RAM 4.x-Disketten-Treibers, d.h.:

-> Haben eigene Blocking/Deblocking-Routinen samt des Puffers für physikalische Sektoren (512 - 1024 Bytes)

-> Beim Shift-Esc-Warmboot merkt RAM 4.x, ob noch Daten im Puffer stehen, die eigentlich auf die Diskette gehören. Nicht so für die HardDiscs!

Da Claudios Treiber Public-Domain ist, werden Olaf und ich versuchen, diesen in RAM 5.x fest einzubauen, um den o.g Problemen und Nachteilen abzuhelpfen. Falls Hannes und Claudio einen gemeinsamen Treiber mit den Vorzügen beider Treiber zusammen- und mir zur Verfügung stellen, solls der sein, der in RAM 5.x Einzug findet.

Was ich an gut kommentierten funktionstüchtigen Treibern benötige, die in RAM 5.x fest eingebaut werden sollen:

1. Routine um physikalischen Sektor/Spur/Kopf zu setzen
2. Routine um physikalischen Sektor zu lesen/schreiben
3. Routine zum Parken des Kopfes

Außerdem die Format-Definitionen für FORMATE4. Was sonst noch so an Weichware zusätzlich von Nöten ist sollte auch dabei sein - wobei ich diese er in .COMs vermute.

Warum die Unterstützung von logischen Laufwerken und verschiedenen Block-Größen so ein großes Thema zu sein scheint, ist mir schleierhaft. Das sind doch alles Dinge, die das BDOS für uns übernimmt. Habe ich eine Platte mit 100 Spuren, und will daraus zwei Lw. machen, definiere ich eines mit 0 System-Spuren und 50 Spuren, sowie eines mit 100, jedoch mit 50 Systemspuren. Fertig ist die Lauge. Das mit der Blocksize ist ebenso lediglich das Eintragen der richtigen Werte in den DiskParameterBlock mittels FORMATE4.

Hardware: IBM-TastaturAnschluß einer (kompatiblen) IBM-PC Tastatur an die FDX/SDX:

(Hannes Mehrrens, 7900)

**Hardware Voraussetzungen:**

- Eine Tastatur die an IBM-PC's läuft; (Manche Tastaturen müssen über DIP-Schalter in den PC-Modus umgeschaltet werden.) PC-Tastatur-Layout oder MF-2-Tastatur-Layout (102 Tasten); Tastatur für AT's wäre auch verwendbar, wird aber nicht von der Treiber-Software unterstützt; (Ich verwende eine Cherry MF-2-Tastatur nach DIN-Norm)
- 5 Portleitungen (3 mal 1 Bit Input, 2 mal 1 Bit Output)
- kleines Hardware Interface zwischen Tastatur und Port (1 74LS74, 1 Transistor, 1 Widerstand, 1 5-pol. DIN-Buchse)
- Stromversorgung +5V, ca. 200mA, falls Strom nicht aus MTX-Netzteil entnommen wird.

**Software Voraussetzungen:**

- FDX/SDX mit RAM4.2 und genügend Speicher für "Klicks" (Software müßte auf allen RAM4.x Systemen laufen)
- Angepaßtes SYSIO.IO
- Meine Treiber-Software, die in einem "Klick" untergebracht ist.

**Funktionsweise des neuen SYSIO.IO:**

Zuerst wird das "Klick" installiert. Danach wird das neue SYSIO.IO mit IOLOADER geladen. Dessen KBDRDY-Routine (gibt an ob, eine Taste gedrückt ist) fragt sowohl die alte MTX-Tastatur als auch die neue PC-Tastatur ab. Beim Lesen eines Zeichens (KBDIN) wird das Zeichen von der jeweils bedienten Tastatur geliefert. D.h., beide Tastaturen, die MTX-Tastatur und die IBM-Tastatur, können parallel benutzt werden.

**Funktionsweise des PC-Tastatur-Treibers (im Klick):**

Wenn eine Taste gedrückt wird, sendet die Tastatur einen der gedrückten Taste zugeordneten Code. Dies geschieht auf serielllem Wege (1 Takt- und 1 Datenleitung). Wird die Taste wieder losgelassen, wird ein sogenannter "Break-Code" gesendet, der auch der losgelassenen Taste zugeordnet ist. Die PC-Tastatur kann sich bis zu ca. 8 "Tastendrucke" merken.

Die Treibersoftware ermittelt aus dem empfangenen Code zunächst die Nummer der entsprechenden Taste. In einem Puffer werden alle, im Augenblick gedrückten Tasten gespeichert. Beim Aufruf von KBDIN (lese Zeichen von Tastatur) wird das, der letzten Taste entsprechende Zeichen aus einer Tabelle geholt. Für die Funktionstasten stehen spezielle Tabellen zur Verfügung, die es ermöglichen einer Funktionstaste auch mehrere Zeichen (z.B.: "DIR") zuzuweisen.

**Vorteile des Anschlusses einer solchen IBM-Tastatur:**

- Schnelle, komfortable Eingabe, die Spaß macht.
- Kein Haken und Klemmen, wie bei der recht miesen MTX-Tastatur
- Da die PC-Tastatur intern ca. 8 Zeichen puffern kann, gehen Eingaben auch während von Disketten-Zugriffen (Interrupts gesperrt) nicht verloren!
- Da die MF-2-Tastatur der DIN-Norm entspricht und weit verbreitet ist, entfällt das "Umdenken" bei Bedienung von entsprechenden Tastaturen in Studium und Beruf.

**Vorläufige Nachteile:**

- Da der notwendige Software-Treiber einige kByte an Speicher braucht, wird die PC-Tastatur momentan nur von RAM4.x unterstützt.
- Der Aufruf von Klick mit SHIFT-ESC ist momentan, nur ueber die alte MTX-Tastatur möglich.
- Das Ändern der Belegung von Funktionstasten ist momentan noch nicht "einfach per Tastendruck" möglich. Ein entsprechendes Programm (Klick) ist gerade in Arbeit.

Hardware: IBM-Tastatur**Mögliche Verbesserungen:**

- Denkbar wäre eine Nachbildung von RAM4's Klick (Funktionstasten-Monitor, kleiner Monitor, etc.)
- Wünschenswert wäre das feste Einbinden des Software-Treibers in RAM4.x (in Arbeit)

**Testbericht zu Hannes' IBM-Tastatortreiber**

(Herbert zur Nedden, 2000)

Ich habe von Hannes - leider einen Tag nach dem Clubtreffen - die Software sowie ein Interface für Port 7 erhalten. Da ich von Haus aus misstrauisch bin, und vor allem die geliehene Tastatur nicht ruinieren wollte, habe ich das Interface erst einmal mit dessen Schaltplan verglichen. Oha, ein Lötfehler: OUT0 und OUT1 vertauscht - aber das war schnell behoben. Hattu mir ungetestetes Interface geschickt ?

Gut, dann also los: Interface mit Tastatur an MTX - meine Hardware-Uhr mußte vorübergehend herausgenommen werden. Kiste angeschaltet. Lämpchen auf der Tastatur gingen kurz an - war wohl deren Selbsttest. Klick-Overlay geladen (warum ist das kein .KLX ?), und neues SYSIO eingebunden. Hoppla, war mein Finger auf dem 's'? Nö. Naja, wo das her kam, wollte ich später klären. Jedenfalls kamen alle von mir danach gedrückten Tasten korrekt an - nur leider auch noch viele viele andere - meist ein senkrechter Balken. Eine Original IBM-MF-2-Tastatur, wie sie am AT-03 von eben dieser Firma hängt, läuft nicht so recht. - Das hat Hannes aber auch nicht behauptet.

Immerhin hat sein Programm trotzdem die gedrückten Tasten richtig geliefert, was schon ein gutes Zeichen ist.

Dieser Treiber hat den eklatanten Nachteil, sich nicht so, wie man es gerne hätte in den RAM 4.5-Tastatortreiber einzuschleifen, d.h. er stellt seine Zeichen nicht in dessen Tastatur-Puffer! Auch kann er SHIFT-ESC nicht richtig umsetzen, und verwendet zu allem Überfluß eine eigene Funktionstasten-Verwaltung.

RAM 5.x wird einen RAM-Sprung bereitstellen, über den Zeichen in den Tastaturpuffer eingestellt werden können, was es dann ermöglicht, eigene Tastatur-Treiber sauber einzuschleifen. Dann würde es kein Programm merken, welches nicht direkt auf die Ports, sondern über die System-Einsprünge des BIOS wie z.B. OFFD?h geht, welche Tastatur das Zeichen geliefert hat - und immerhin gehen viele Programme, wie z.B. die KLICK-Overlays diesen Weg. Damit wäre auch nur ein Funktionstasten-Decoder und ein Satz Tabellen im Rechner - dieser wird ab RAM 5.x selbstmurmelnd auch alle 12 F-Tasten der IBM-Tastatur unterstützen.

Hannes war so freundlich, mir auch die Sources zu schicken, und ich habe der Neugier halber mal nachgeschaut, wie er die Tastatur einliest - was eigentlich aus dem Aufbau seines Interfaces schon klar war. Jawohl, Bitweise. Sendet die Tastatur mit 9600 Baud, dauert es eine Millisekunde, um ein Zeichen von der Tastatur einzulesen. Dann muß es noch umkodiert werden, was aber über eine Tabelle leicht möglich ist. Wenn ich mal etwas Luft habe, werde ich die c't-Lösung für den Anschluß einer MF-Tastatur ansehen und testen, da mir diese betriebssicherer erscheint, Port 7 für meine Uhr frei läßt, und vor allem die CPU vom bitweisen Einlesen serieller Informationen befreit. Wie meine Maus wird's dann hoffentlich eine Interrupt-Lösung! Das eine CPU heutzutage dafür herhalten muß ? Hier gehört m.E. unbedingt eine Hardware-Lösung hin, zumal Hannes' Lösung eh ein Interface benötigt, und auch die Uhr nicht einfach vertreiben sollte. Da das c't-Interface eine seriell-parallel und anschließende parallel-seriell Wandlung vornimmt, um mit so ziehmlich jeder Krücke von Übertragungs-Protokoll (9-11 Bit, mal mit, mal ohne Stop-Bit, Start-Bit = high oder low, Ruhepegel = ???), teste ich mal das Einlesen erst mal via PIO und dann mittels DART.



K o m i k: Fremde Sprak is sich schwäre Sprak

The Manager  
Y.M.C.A. Hotel  
London, U.K.

Roma, 24. Aug. 1981

Dear Signore Direttore

Now I am a-tella you a story how I was a-treated at Your hotella.  
I am a-coma from Roma as tourist to London and stay as younga christian man at Your hotella.  
When I coma in my room I see there is no shit in my bed - how can I sleep with no shit in my bed ? So I calla down to the receptione and tella : "I wanna shit". They tella me : "Go to toilette". I say : "No. I wanna shit in my bed". They say : "You better no shit in your bed, you sonnawabitch !".  
I go down for breakfast into ristorante. I order bacon and eggs and two pisses of toast. I getta only one piss of toast. I tella waitress and pointa of toast : "I wanna piss !" She tella me : "Go to toilette". I say : "No.No. I wanna piss on my plate !". Then she say to me : "You bloody hella not piss on the plate, you sonnawabitch !". Second person who do not even know me sonnawabitch ! - What is a sonnawabitch ?  
Later I go for dinner in Your ristorante. Spoon and knife is laid out on the table. But no fock. I tella waitress : "I wanna fock !" and she tella me : "Shure, everyone wanna fock !". I tella her : "No.No. You don't understand me. I wanna fock on the table !". She tella me : "So you sonnawabitch wanna fock on the table ? Get your ass out of here !"  
I go to receptione and ask for a bill. I no wanna stay in this hotella no more. When I have paid the billa, the portier say to me : "Thank you, and peace on you !" I say : "Piss on you too, you sonnawabitch !"  
I go back to Italy. I never more comma stay Your hotella no more, You sonnawabitch.

Sincerely

Enrico Morelli