

MTX *User-Club Deutschland*

Info 29
20.10.1988

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

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

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

Verpflichtungen: Einsendung unseres Anmeldeformulars.

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

Club-Info, unser Blatt, verschicken wir ca. 8-wöchentlich. Inhalt ist alles was uns über den MTX/FDX (ohne Copyright) in die Hände fällt. Es kostet nicht über DM 12.- (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, Grundwehrdiensleistende, 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 Grevenbleck 24 3005 Hemmingen 1 (0511) 41 78 77	Thomas Wulf Roritzer Str. 8 8500 Nürnberg 90 (0911) 33 52 52	Hans Gras Statenhoek 49 NL 1506 VM Zaandam (0031-75) 17 49 91
---	---	---	--

Telefon-Sprechzeiten

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

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

Inhaltsverzeichnis**C L U B**

Korrektur und Nachtrag	Seite 2
Besprechung / Softwarequelle	Seite 3
Der ideale Programmierer ?	Seite 23
Anmerkung von Holger Göbel	Seite 43

H a r d w a r e

Boot-EPROM	Seite 4
------------	---------

S o f t w a r e

CONVERT	Seite 4
FORMSTAR	Seite 5
R A M 4 . 3	Seite 19
BRADFORD-Dokumentation	Seite 48
BRADFORD-Zeichensätze	Seite 56
VIRUS	Seite 59

d S ?

d.h. dunkle Sunden von Wem-Auch-Immer und Lösungen	Seite 6
---	---------

S u p e r C a l c

Tips & Tricks	Seite 9
---------------	---------

d B A S E

Steuerzeichen	Seite 11
Probleme	Seite 13
Patch für Umlaute	Seite 14

L e s e r b r i e f

Kurt-Bernd Rohloff, 8000	Seite 15
Hartmut Traber, 5720	Seite 18
Hans Gras, NL 1506	Seite 20

H a r d w a r e

Drucker	Seite 24
Z80PIO und Interrupts	Seite 25
Datenblatt Z80-PIO	Seite 28
SDX - 80Zeichen-Karte	Seite 30
Diskettenformate	Seite 31
SDX-Sidisc	Seite 32
Ports	Seite 33
Volkseprommer	Seite 35
WAIT-Generator	Seite 36
c't-IO-Karte und Interrupts	Seite 38
8-MHz-Platine	Seite 44
8 MHz - Kochbuch	Seite 46

G r a f i k

Chaos	Seite 57
-------	----------

Preis für dieses Info: DM 12,-

Liebe Leserin, lieber Leser,

wir gehen langsam aber sicher auf die 30 zu - mit der Nummer des Infos! Es ist wieder mal eine längere Pause zwischen zwei Infos verstrichen, weil ich mitte September einen Mangel an Informationen hatte, und erst einmal in Urlaub fuhr. Sag einmal, hast Du nicht vielleicht etwas für's Info? Ich bin für alles zu haben von einfachen Fragen/Antworten bis hin zu fertigen Beiträgen! Falls Du meinst, Dein Beitrag sei für unser Info zu niveaulos, d.h. schlicht und ergreifend zu simpel, dann irrst Du! Denn gerade diese 'simplen' Beiträge sind meines Erachtens die für die meisten interessantesten. Selbst kleine Tips, mit denen Du Dir Dein Memotech-Leben hie und da vereinfachst können anderen helfen, die evtl. einfach anders an die Sache herangegangen sind! Gerade auch die Erfahrungen, die ein Anfänger gemacht hat, um ein Problem zu lösen, sind die wertvollsten. Wenn nämlich ein Experte versucht, Dinge einfach aufzuschreiben, begeht er sehr leicht viele Unterlassungssünden, da er gar nicht merkt, daß er Kenntnisse vorsetzt, die nur für ihn selbstverständlich sind!

Das letzte nordische Clubtreffen fand Anfang September in Hemmingen statt. Zu meiner Freude war der Einzugsbereich von Neumünster (Schleswig-Holstein) bis Puchheim/Fürstenfeldbruck (bei München) recht groß, und gab mir die Möglichkeit, einige der 'Südlichter' kennenzulernen. Als Höhepunkt außer dem eigentlichen Kennenlernen und Klönen empfand ich die Hardware-Basteleien von H. Göbel & G. Witzel (MTX/FXDX in Aluturm umgebaut, FDX-ECB-Einschub als Prototyp, 8MHz-Platine) sowie RAM 4.3 (siehe weiter hinten). Das nächste nordische Clubtreffen soll 1 1/4-Tägig sein, d.h. incl. Übernachtung und geselligem Beisammensein abends und beim Frühstück, und nach der Schneeperiode stattfinden. Von einigen, die aus Bayern kamen, habe ich mir sagen lassen, daß sie dieses stark befürworteten. Ich glaube, daß gerade so ein gemütlicher Plausch nach dem eigentlichen Treffen ... - den 'Wuppertalern' und mir hat es auf jeden Fall definitiv gefehlt!! Außerdem wollen wir die Dinge, die Ihr mitbringen und vorstellen wollt, in einem Zeitplan so zusammenfassen, daß jeder weiß, was wann gezeigt wird, damit jeder alles was ihn interessiert sehen kann, und nichts unter den Tisch fällt.

Hast Du Lust, daran teilzunehmen, dann schick mir doch bitte einmal die Antwort auf folgende Fragen: Wann paßt es Dir? Willst Du übernachten, und wenn ja, was darf Übernachtung kosten? Lieber einfache/billige Übernachtung, Doppelzimmer, oder wie sonst? Was möchtest Du vorführen, was sehen? Was möchtest Du sonst zum Treffen sagen? (Dies ist noch keine Anmeldung)

Thema TEAC FD 55 GFR für 1.76 MB umändern: Vor einiger Zeit veröffentlichte Herbert Oppmann einen Artikel, in dem er beschrieb, wie die älteren TEAC FD55GFV zu belöten/beschalten sind, damit sie im HD-Modus mit 300 statt 360 Umdrehungen rotieren, um so das Format 1C mit 1.76 MB statt 'nur' 1B mit 1.44MB zu verkräften. Um die TEAC FD55GFR auch so zu tunen, brauchen wir anscheinend das technische Handbuch, welches DM 150.- (evtl. zzgl. MwSt und P&V) kostet - Versuche da ohne, jedoch mit gutem Menschenverstand, schlugen leider fehl. Bisläng sind nur H. Traber, H. Ahrensfeld und E. Gottspenn daran zu beteiligen. WARUM nur so wenige? Schließlich sind noch wesentlich mehr Laufwerke dieser Art im Einsatz! Wartest Du mit Deinem GFR etwa auf die Veröffentlichung im Info? Wenn ja, beschleunige doch den Informationsfluß etwas mit einem Betrag zu den Kosten! Ob ich diese Informationen dann ins Info nehmen kann, hängt von denen ab, die das Handbuch bezahlt haben!

Speicherbausteine haben in der letzten Zeit eine recht spektakuläre Preisentwicklung durchgemacht. Hätte ich Anfang dieses Jahres DM 50.000,- übrig gehabt, hätte ich heute leicht DM 150.000,- mein Eigen nennen können. Speziell die RAMs HM 50257-12, die ich für die 768k-Karten benötige, sind von ca. 10.-/St. bis hin zu 'nicht lieferbar' gekommen. Mittlerweile sind sie wieder erhältlich, und stehen sogar mit echten Preisangaben in einigen Preislisten: ca. DM 25.-/St. Ab Januar 1989 dürfen sich die Preise wieder einigermaßen normalisiert haben. Wer jetzt 512 kB haben möchte, muß ca. DM 650,- bis 700,- anlegen; Tendenz fallend!

Es gibt leider auch schlechte Nachrichten: Die Kosten für Papier und daher für den Druck der Infos sind um ca. 15 % gestiegen, was sich leider im Preis der Infos niederschlagen muß!

Herbst zur Nedden

C L U B: Besprechung / Softwarequelle**Kommentar und Besprechung von Software**

(Herbert zur Nedden, 2000)

Ich bin gerne bereit, meine Beurteilung von Software, die im Club entstanden ist, und die der Autor an den Rechner (und damit an den Mann/ die Frau) bringen will, ins Info zu packen. Doch dank der Vielfalt der Anwendungen und der Möglichkeiten des Systems ist es mittlerweile so, daß es einiges an Software gibt, die eigentlich richtig toll ist - nur ich sie nicht einsetzen kann/will. Was nutzen mir die besten Programme, die etwas erledigen, was ich gar nicht brauche ? Klar, ich kann mich auch über solche Produkte auslassen, doch leider übersehe ich vermutlich einige der Vorzüge. Daher möchte ich diejenigen unter Euch, die von einem Programm oder einer Hardware überzeugt oder gar begeistert sind bitten, mir mal Euren Kommentar für's Info zu geben.

Beispiel hierfür ist das Boot-EPROM von Michael Kessler, welches ich für mich eigentlich mit 'klingt toll, brauch ich nicht' abhakte, da ich gar nicht erkannte, was das eigentlich Gute an diesem EPROM noch so ist, und es zumindest für viele andere interessant macht! Und wenn ich als angeblicher Freak einige Dinge nicht sehe, geht das vielleicht anderen ebenso, was oft auch daran liegt, von welcher Seite aus man Dinge betrachtet. Also Ihr Begeisterten: Bringt Licht in unser Dunkel!

Quelle für M80/L80 (Jan Brederke, 2000)

Bei dem M80 und L80 handelt es sich um einen komfortablen Makro-Assembler samt dazugehörigem Linker nebst einiger weiterer Utilities. Die meisten Assemblerprogramme des Clubs, insbesondere die Klix-Programme, sind damit geschrieben, und so existieren auch schon entsprechend viele fertige Makro- und Programmbibliotheken für den MTX allgemein, CP/M samt P2DOS-Erweiterung und zur Klix-Programmierung.

Laut c't 9/88, Seite 25 gibt es das MS-Entwicklungspaket, bestehend aus MBASIC-Interpreter und -Compiler sowie M80-Assembler, L80-Linker, Cross-Reference und Library-Manager bei der Kemptener Graf Elektronik Systeme GmbH (GES) zum Preis von DM 248. Die genaue Adresse stand in der c't leider nicht.

Den dazu kompatiblen Z80ASM von SLR gibt es bei dem "Elektronikladen", Detmold (Tel. 052 32/81 71) für DM 199 (ohne Cross-Reference, ... ??).

c't empfiehlt, sich vor einer Bestellung über die lieferbaren Diskettenformate zu informieren. Dank RAM4 sollten wir da aber kaum Probleme haben, solange nur die Diskettengröße stimmt.

Anm.d.HzN.: Anruf bei Elektronik-Laden Detmold kurz nach Erscheinen von c't:
'HÄ? was ist das, da muß ich mal den Chef fragen - aber der ist verreist.'
Naja, vielleicht gibt's den Assembler mittlerweile.
Billiger ist eine Bestellung bei SLR-Systems direkt in den USA: ca. US\$ 49.-

H a r d w a r e: Boot-EPROM / S o f t w a r e: CONVERT

Boot-EPROM von Michael Kessler

(Herbert zur Nedden, 2000)

Michael Kessler hat ja vor einiger Zeit ein neues Boot-EPROM herausgebracht, mit dem es möglich ist unter anderem von Format 09 zu booten. Da ich nun bekanntlich von einer CMOS-Floppy boote, war dieses EPROM für mich eigentlich nicht interessant - ABER da habe ich doch garnicht gepeilt, daß das Boot-EPROM eine gewisse Feinheit bietet:

Mit dem Boot-EPROM von MK stehen die wichtigen RAM4-Formate 09, 0A, 1A und diverse andere auch ohne RAM 4.x zur Verfügung!!

Fällt nun mal die CMOS-Floppy oder gar die Speichererweiterung aus, kann ich immerhin noch von RAM4-Formaten booten, und diese auch ohne RAM 4.x verarbeiten!!

Also ist dieses EPROM die Chance für Nicht-RAM4-ler, auch die Vorzüge des Formates 09 nutzen, oder mit geeignetem Laufwerk gar des Formates 1B, die da 390kB (bzw. 1400kB) und deutlich schnellerer Zugriff sind.

Ich habe also dieses Boot-EPROM mal getestet: Es hält, was es verspricht!

Fazit: Das neue Boot-EPROM ist auch für Besitzer von CMOS-Floppies durchaus, für Nicht-RAM4-ler sogar sehr interessant.

Neuerungen: Zum neuen FDX-Boot-EPROM, Version 8/1988 ist übrigens zu vermerken, daß das Format 6F ersatzlos gestrichen wurde (schade), dafür jetzt aber bei B:03 auch unser gutes FDXB läuft (juhu). Auch CMOS-Floppys (Formate 56, 53) sind jetzt drin!

Update: Updates gegen Portoübernahme.

CONVERT.COM: Aus 1A alt mach 1A neu

(Herbert zur Nedden, 2000)

Michael Kessler hat vorgeschlagen und realisiert, daß das Format 1A statt mit 4kB je Block auf der Diskette genauso wie 1B mit 2kB je Block implementiert wird. Was ist das, und wozu ist es gut, werdet Ihr fragen. Nun:

Auf der Diskette werden Dateien immer in Sektoren à 128 Bytes abgespeichert, und diese Sektoren zu Blöcken (1 Block = 1kB, 2kB, ... je nach Format) zusammengefaßt. Je belegtem Block wird die Nummer dieses Blockes im Direktory eingetragen. Da evtl. der letzte Block nicht ganz von der Datei eingenommen wird, steht im Direktory-Eintrag auch noch, wieviele Sektoren des letzten Blocks zu der Datei gehören. Speichere ich nun z.B. eine Text-Datei ab, die lediglich meinen Vornamen enthält, enthält die Datei eigentlich nur 7 Bytes an Information und 1 Byte mit Inhalt 1AH als Ende-des-Textes-Kennung. Schau ich jedoch nach, wie groß diese Datei ist (z.B. mit STAT), stelle ich fest, daß diese auf einer 03-er Diskette immerhin ganze 2kB frißt! Tja, auf einer 03-er Diskette werden die Dateien in 2kB-Schnipseln (den Blöcken) angelegt. Meine 8 Bytes belegen einen Block auf der Diskette, wobei im Direktory vermerkt ist, daß nur im ersten Sektor dieses Blocks wirklich Daten stehen, d.h. der Rest ist ungenutzt.

Die Original-Bernd-Preusing-1A-Disketten arbeiten mit 4kB-Blöcken, was bedeutet, daß es hier meine Datei mit Inhalt 'Herbert' immerhin 4kB belegt. Da Michael nun festgestellt hat, daß der Inhalt seiner System-Diskette im 1A-Format schlappe 350kB mehr Platz benötigt, als auf einer 1B-Scheibe, da 1A mit 4kB- und 1B mit 2kB-Blöcken arbeitet, hat er sich vorgenommen, das 1A-Format auf 2kB-Blöcke umzustellen.

S o f t w a r e: FORMSTAR

Für diese Umstellung ist folgendes zu tun:

1. Das Format 1A neu mit FORMATE4 definieren. Dabei sind folgende Änderungen erforderlich: (In Klammern alter Wert)
Blocksize 2 (4), Blöcke 711 (355), EXM 0 (1), ALO #F0 (#C0).
2. Alte 1A-Disketten mit Michael's Programm CONVERT auf das Neue Format konvertieren.

Michael schreibt über CONVERT:

CONVERT.COM funktioniert tadellos (bis jetzt konnte ich jedenfalls keine Fehler mehr feststellen), dennoch sollten alle diejenigen, die nur ganz wenige 1A(alt) Disketten haben, diese per Umkopieren ins neue 1A übertragen. Der Grund: Durch die verkleinerte Blockgröße brauchen die Files auf der Disk weniger Platz (insbesondere kleine Files), dadurch entstehen auf der Diskette zwischen zwei Files sehr kleine Freiräume, wird nun ein neues File auf Disk geschrieben, so werden diese Freiräume aufgefüllt, das File also über die ganze Disk verteilt. Dadurch muß der arme Lesekopf ordentlich hin-und hersteppen, um das File zu lesen. Dies ist sicherlich nachteilig, aber derzeit noch nicht zu vermeiden. Allerdings bin ich bereits in der Konzeptphase für einen Disketten-Aufräumer, der die Files auf der Disk ordentlich anordnet und somit die Zugriffszeit verkürzt.

FORMSTAR

(Herbert zur Nedden, 2000)

Früher kursierte mal so ein Merkwürd-Programm COMPAT, mit dem man angeblich auf alle möglichen Formate unter gewissen Umständen zugreifen kann, wenn Laufwerk B: und C: als echte Laufwerke vorhanden waren - das tun nun RAM 4.x und MS wesentlich besser.

Und dann gab es dazu noch so ein FMAT mit dem man alle (un)möglichen Formate formatieren konnte - das tun nun FORMSTAR und MSFORM4 wesentlich besser.

Spaß beiseite: Immer wieder kam die Anfrage, warum Bernd Preusing seinem FORM4 nicht all die Formate beigebracht hat, die RAM 4.x auch verarbeiten kann - immerhin stehen in FORMATE.DAT ganze 178 Formate (wobei die MSDOS-Formate nur einen Platz belegen). Die Antwort ist verblüffend banal: Weil das richtig Arbeit macht. Lesen und Schreiben von fremden Formaten ist vergleichsweise einfach, falls diese mit einem Floppy-Controller-Chip beschrieben wurden. Es genügt eigentlich zu wissen, wie der DiscParameterBlock und SkewTable aussieht. Aber das Formatieren bedeutet, daß man wissen muß, wieviele Bytes welchen Inhaltes in die Lücken (Gaps) zwischen den Sektoren gehören, und und und ??

Michael Kessler, der sich mit unserem Controller-Chip eingehend beschäftigt, hat diesem Manko nun genial abgeholfen:

FORMSTAR ist in der Lage knapp 100 (d.h. über 90) Formate zu formatieren.
Dabei überprüft FORMSTAR auch noch gleich die Formatierung,
was RCHECK überflüssig macht!

(Wem die MSDOS-Formate fehlen, dem empfehle ich MSFORM4.)

Und zu allem Überfluß gibt es FORMSTAR auch für das KCLICK, d.h. Speicherresident und natürlich für die SDX! Der Format-ID wird stets gesetzt (natürlich nur bei Formaten mit Systemspuren).

Format-ID: Siehe Info 27 Seite 26 und verwandte Artikel.

d S ? : d.h. dunkle Sunden von Wem-Auch-Immer und Lösungen

COPYD

(Herbert zur Nedden, 2000)

Da schreibt man so für SICH ein Progrämmle zum Duplizieren von Disketten, und stellt fest, daß man eigentlich weder das Programm noch das KnowHow den Clubmitgliedern vorenthalten möchte. Da das Programm mir eigentlich die Erstellung von PublicDomain-Disketten (insbesondere KLICK.xxx) vereinfachen sollte, baute ich auch gleich Default Quell- und Ziellaufwerk ein. Der simple Befehl COPYD tat schon alles gewollte: B: wurde radikal mit dem Senf von D: überschrieben. Nun machte ich mich an das Kommentieren, damit sie Chose auch für's Info geeignet ist, und packe den Quelltext auch auf eine CLUB-PD.

Was ist der Erfolg:

Alle möglichen Leute stört die bequeme Bedienung, aber auf die einfache Idee, entweder die Default-Verarbeitung aus dem Programm zu entfernen, oder noch einfacher mit DDT, MONI oder ThePatch Default Quell- und Ziel-Laufwerk auf den selben Wert zu setzen (da COPYD das nicht mag) kommt niemand - statt dessen hagelt es Kritik.

Was sagt HzN dazu:

Ich finde es toll, daß eine so rege Reaktion erfolgte, mit der ich nie im Leben gerechnet hatte! Aber zu dem Inhalt möchte ich folgendes Anmerken:

PD-Programme sollten nicht den Anspruch vollständiger Perfektion für sich beanspruchen müssen, zumindest falls sie im dokumentierten Quelltext herauskommen. Es ist besser, ein jeder gibt seine für SICH SELBST geschriebenen Programme, die einen sinnvollen Einsatzbereich FÜR DEN AUTOR SELBST darstellen, in Public Domain und schert sich 'n feuchten darum, daß die Ausführung einzelner Teile nicht jedem paßt, als, daß jeder seine FÜR SICH GUT GENÜGENDEN Programme den anderen nur deswegen meint vorenthalten zu müssen, weil er meint, sie seien nicht gut genug! Man füge den Quelltext mit anbei. Wem dann Einzelheiten nicht passen, der kann den Quelltext ja ändern, und so vielleicht dem Autor noch gute Ideen zukommen lassen! Es gibt zwar einige Perfektionisten im Club (z.B. Olaf Krumnow), die dann im Laufe der Zeit die eigenen PD-Programme trotzdem immer wieder überarbeiten. Was wäre, wenn Olaf z.B. sein KLICK-HARDCOPY für sich behalten hätte. Wie kämen wir dann an die nächste erweiterte und installierbare Version, die auf KLICK.004 sein wird ?

CP/M und RAM 4.x

(Herbert zur Nedden, 2000)

Der BIOS-Befehl Block-Read, mit dem auf einen Schlag ab dem aktuell Sektor mehrere Sektoren eingelesen werden können ist MURKsig! Dieser Befehl sollte wie folgt funktionieren:

```

Loop:  Lies Sektor an DMA-Adresse
        Zähler = Zähler - 1
        Wenn Zähler = 0, dann Fertig
        DMA-Adresse erhöhen (um Sektorlänge = 128)
        Wenn Sektor = Spurlänge
            dann Sketor = 1
            Spur = Spur +1
        sonst Sektor = Sektor +1
        Goto Loop

```

Memotech hat leider die Spurlänge fest auf 26 gesetzt, was unter RAM 4.x so geblieben ist, was BP bewußt tat, und in der RAM 4-Dokumentation erwähnt hat. Wenn ich daher Block-Read auf eine Diskette mit mehr als 26 Sektoren je Spur an, überspringt Block-Read einfach die Sektoren ab 27 bis zum Ende der Spur.

d S ? : d.h. dunkle Sunden von Wem-Auch-Immer und Lösungen**Turbo-Pascal-Loop**

(Herbert zur Nedden, 2000)

Folgendes Programm loopt, und loopt, und loopt, ...:

```
var s: string[255];
begin
  s:= '   ';
  while s[1]=' ' do delete(s,1,1);
end.
```

Klar, was es tun soll ? Die führenden Leerzeichen entfernen. Was aber, wenn der String nur Leerzeichen enthält ? Irgendwann ist durch dieses Programm der String s leer. Delete(s,1,1) ist daher wirkungslos, da es nur bei nicht-leeren Strings das erste Zeichen entfernt, d.h. die Zeichen alle nach vorne schiebt, und die Länge um eins verringert. Nach dem dritten WHILE-Durchlauf ist die Länge von s null, d.h. Delete(s,1,1) tut nix.

Nichtsdestotrotz steht an s[1] ein ' ' ... also LOOP!

So geht's:

```
var s: string[255];
begin
  s:= '   ';
  while pos(' ',s)=1 do delete(s,1,1);
end.
```

BRADFORD

(Herbert Oppmann, 8520)

Wenn die Größe der mit BRADFORD zu druckenden Datei ein Vielfaches von 128 ist, hängen einige Editoren (z.B. WordStar) kein 1A als Textende-Kennung dahinter, da der Text am physikalischen Ende der Datei zu Ende ist. Turbo Pascal, und damit BRADFORD, mag das nicht:

IO-Error 99

und die letzte Zeile wird nicht ausgedruckt.

Die MENU.CPR-16k-Grenze überlistet!

(Peter Würfel, 7262)

Wer sich mit MENU.COM bzw. MENU.CPR eine Benutzeroberfläche geschaffen hat, hat sicher schon festgestellt, daß MENU.CPR nicht größer als 16 k sein darf. Das ist nicht gerade viel, besonders dann, wenn man verschiedene Auswahlmenüs optisch übersichtlich gestaltet und dazu fast den gesamten Bildschirm ausnutzt.

Wie sich diese 16k-Grenze umgehen läßt, möchte ich an folgendem Beispiel zeigen:

Auf meiner SRAM (LW I:) befindet sich die Datei MENU.CPR. (hier nur in minimalen Auszügen, ohne Bildschirmsteuerzeichen. In Wirklichkeit ist die Datei so groß, daß ich bei der Option 'C' = dBASE keinen Platz mehr für ein Untermenü habe) Das Untermenü für verschiedene dBASE-Programme befindet sich in der Datei MENU.002. Wenn ich nun die Option C (menu 0) in MENU.CPR anwähle, wird die bisherige Datei menu.cpr in menu.001 umbenannt und sofort anschließend die Datei menu.002 in menu.cpr. Es stehen mir somit weitere 16 k zur Verfügung. Wie das Beispiel zeigt, kann ich sofort in einem bestimmten Untermenü weitermachen (im Beispiel menu 1).

d S ? : d.h. dunkle Sunden von Wem-Auch-Immer und Lösungen

MENU.CPR :

```
-dpx
#
(menu 0)
H a u p t m e n ü

1: Inhaltsverzeichnisse
A: New Word
B: Super Calc
C: dBASE
#
Aa;;nw;I:
Ba;;i:sc;i:
Cren i:menu.001=i:menu.cpr;ren i:menu.cpr=i:menu.002;menu 1
1:3
#
menu 1 (Bildschirm)
#
menu 1 (Befehle)
#
menu 2 (Bildschirm)
#
menu 2 (Befehle)
#p
(menu 3)
I n h a l t s v e r z e i c h n i s s e

A: XDIR A:
B: XDIR B:
C: XDIR C:
#
A!XD A?:
B!cf4;type cls.typ;XD B?:
C!cf4;type cls.typ;XD C?:

##
```

MENU.002:

```
-dpx
#
menu 002/0 (Bildschirm)
#
menu 002/0 (Befehle)
#
(menu 002/1)
dBASE-Menü

A: dBASE
B: Telefonliste

0: Hauptmenü
#
Oren i:menu.002=i:menu.cpr;ren i:menu.cpr=i:menu.001;menu
Aa;;i:db i:dbs;i:
Bi:db c:telef;ren i:menu.002=i:menu.cpr;ren i:menu.cpr=i:menu.001;menu

##
```

SuperCalc: Tips & Tricks**SuperCalc: Tips und Tricks**

(Kurt-Bernd Rohloff, 8000)

1. Wie rum rechnet er denn nu?

SuperCalc kann die Tabelle bekanntlich zeilenweise (ist voreingestellt) oder spaltenweise berechnen. Leider ermangelt es unserer uralten Version an einer Statuszeile, so daß man einer Tabelle später nicht mehr ansehen kann, welcher Modus gerade eingestellt ist. Man kann dies aber leicht durch zwei Formeln herausfinden, wie unten angegeben. Ihre absolute Lage in der Tabelle ist egal, lediglich ihre relative Anordnung muß eingehalten werden.

```

1   A   11   B   1
11          A2+1
21 B1+1

```

Ist die Zahl in der oberen rechten Zelle kleiner als die andere, rechnet er zeilenweise, anderenfalls spaltenweise. Wenn man das raus hat, kann man die beiden Formeln ja wieder rauslöschen (/B).

2. Update nur einmal!

Einige Male stand ich vor folgendem Problem: Ich habe eine Zelle mit einem Wert. Dieser Wert wird im Laufe der Zeit durch Zu-/Abgänge verändert. Als Beispiel betrachten wir ein Bankkonto. Den Zugang (z. B. Zahlung von Zinsen) kenne ich, aber nicht den neuen Saldo. Man könnte natürlich den neuen Saldo im Kopf berechnen und gleich den neuen Saldo in die Tabelle eingeben. Aber wer kann heute schon noch Kopfrechnen? Und wer will es denn, wenn er doch SuperCalc hat? Also zu primitiv.

Nächste Lösung: Ich editiere die Zelle mit dem Saldo. Hinter den aktuellen Wert schreibe ich "+ZUGANG", wobei ZUGANG durch den aktuellen Zahlenwert zu ersetzen ist. Wenn ich auf diese Art mehrere Zugänge verarbeitet habe, kann ich ja mal wieder den entstandenen Saldo in die Zelle eintragen, damit die Formel nicht zu lang wird. Das funktioniert recht gut. Man hat nur den Nachteil, daß man den letzten Zugang nicht immer sieht, sondern nur, wenn der Cursor auf der Saldo-Zelle steht. Wenn ich dann mal die Formel durch den neuen Wert ersetze, ist diese Information sogar ganz weg. Man weiß dann manchmal nicht mehr, ob man den letzten Zugang schon verarbeitet hat oder nicht. Kurz und gut: ich möchte den Zugang in einer eigenen Zelle sehen.

Nächste Lösung: Man nimmt zwei Zellen, eine für den Zugang und eine für den Saldo. Die Saldo-Zelle enthält dann die Formel "ZUGANG+SALDO", wobei hier ZUGANG und SALDO für Zellen stehen. Diese Lösung hat den Haken, daß der Zugang bei jeder Neuberechnung wieder addiert wird, und daß ist natürlich falsch! Der Zugang soll nur ein einziges Mal addiert werden. Um das zu erreichen, habe ich mir einen Trick ausgedacht, der allerdings noch eine weitere Zelle für den alten Saldo (d. h. den Wert vor dem Zugang) braucht. Mit Zahlenwerten sieht das dann so aus:

```

1           A           11           B           11           C           1
11          Zugang          Stand          Stand
21          alt
31-----
41          247.00          1000.00          1247.00

```

SuperCalc: Tips & Tricks

Interessanter sind natürlich die dahinterstehenden Formeln:

1	A	11	B	11	C	1
11		Zugang		Stand		Stand
21				alt		neu
31	-----					
41		247		IF(A4+B4=C4,B4,C4)		IF(B4=C4,C4+A4,C4)

Nur dann, wenn sich der Zugang **ändert**, wird der Saldo neu berechnet. Wie der Trick funktioniert, werde ich nicht erklären, sondern es euch zum Knobeln überlassen. Wichtig ist nur, daß die Berechnung in der Reihenfolge "Zugang" >> "alter Stand" >> "neuer Stand" voranschreitet. Eine vertikale Anordnung wäre also ebenfalls möglich. Aus dem eben gesagten geht aber auch schon der Nachteil dieser Methode hervor: man kann nicht zweimal hintereinander **denselben** Zugang verarbeiten. Um dies zu erreichen, muß man erst den Zugang auf Null setzen und danach den vorigen Wert nochmal eingeben. Außerdem kann man den Zugang nicht mit N/A vorbelegen.

Frage an alle SuperverCalcten: Wer löst das gleiche Problem mit nur zwei Zellen?

3. Zufallszahlen

Wer mal in SC Zufallszahlen braucht, kann den folgenden Generator verwenden:

1	A	11	B	11	C	11	D	11	E	11	F	11	G	1
11	Zufallszahlen													
21	Start/Stop(1/0) 1													
31	Anfangswert .608													
41	Zufallszahl IF(B2=0,B3,B4*9821+.211327-INT(B4*9821+.211327))													
51														
61	Grenzen													
71	von bis													
81	1 49													
91	INT(B4*(B8+1-A8)+A8)													

In B4 wird bei jeder Neuberechnung eine andere Zufallszahl zwischen 0 (einschl.) und 1 (ausschl.) erzeugt. Eigentlich bräuchte man dafür eine FRACTION Funktion. Da SC diese nicht bietet, ist die Formel etwas länglich geraten. Alternativ dazu könnte man den gemeinsamen Ausdruck $B4*9821+0.211327$ auch in einer Hilfszelle berechnen. In B3 wird ein Startwert, ebenfalls zwischen 0 und 1, vorgegeben. Mit B2=0 kann man den Generator anhalten, z. B. um einen anderen Startwert vorzugeben. Der Generator stammt aus einem Programm der HP-65 Anwenderbibliothek von Don Malm. Er liefert unabhängig vom Anfangswert 1 Mio. verschiedene Zufallszahlen.

Wer ganzzahlige Zufallszahlen im Bereich von beispielsweise 1 bis 49, beide einschließlich, braucht (wozu wohl???), kann die erzeugten Zahlen durch die Formel in B9 weiterverarbeiten. Bei anderen Grenzen (4 aus 36 oder was auch immer) braucht man nur die Grenzen in Zeile 8 entsprechend anzupassen.

d B A S E: Steuerzeichen

dBASE: Bildschirm- und Druckersteuerzeichen

(Kurt-Bernd Rohloff, 8000)

Ich beziehe mich hier auf den sehr informativen Artikel von Peter Würfel in Info 27 auf S. 45. Als Alternative zu seiner Lösung möchte ich hier meine Methode vorstellen. Sie besteht aus einem dBASE Programm ATTR, das im wesentlichen die Steuervariablen durch STORE Befehle vorbelegt bzw. sie wieder freigibt. Das Programm stellt nur eine kleine Auswahl der Möglichkeiten zur Verfügung, die unser Bildschirm bzw. Drucker bieten. Dies hat sich jedoch in der Praxis als durchaus ausreichend herausgestellt. Das Programm wird durch eine Variable ATTRSTR gesteuert. Diese kann ein oder zwei Zeichen enthalten. Das erste Zeichen ist für die Bildschirmsteuervariablen zuständig, das zweite für die Druckersteuervariablen. In der folgenden Tabelle sind die möglichen Werte mit ihrer Wirkung dargestellt ("*" bedeutet "irgendwas"):

	<u>setzen</u>	<u>freigeben</u>
Bildschirm	! B	b
Drucker	! *D	*d

Als Default wird "BD" angenommen, wenn ATTRSTR undefiniert ist. Wenn ich beispielsweise nur die Druckervariablen setzen will, gehe ich folgendermaßen vor:
 STORE "xD" TO ATTRSTR
 DO ATTR

Das Programm sieht nun folgendermaßen aus (meine Kommentare dazu in *kursiv*):

```

IF TYPE(ATTRSTR) <> "C"
    STORE "BD" TO ATTRSTR
ENDIF
DO CASE
    CASE $(ATTRSTR,1,1) = "B"
        * Bildschirmattribute
        STORE CHR(6)+CHR(2) TO NORM           halbhell
        STORE CHR(6)+CHR(16) TO INV          invers
        STORE CHR(6)+CHR(48) TO INVINT       intensiv invers
        STORE CHR(20) TO BRI                 hell
        STORE CHR(19) TO UL                  unterstreichen
        STORE CHR(21) TO BRIUL               hell+unterstreichen
        STORE CHR(14) TO BLION               blinken an
        STORE CHR(15) TO BLIOFF             blinken aus
        STORE CHR(14)+"**"+CHR(15) TO STARS  ** blinkend
    CASE $(ATTRSTR,1,1) = "b"
        RELEASE NORM,INV,INVINT,BRI,UL,BRIUL,BLION,BLIOFF,STARS
ENDCASE
IF LEN(ATTRSTR) > 1
    DO CASE
        CASE $(ATTRSTR,2,1) = "D"
            * Druckersteuerzeichen (DMX80 Drucker):
            STORE CHR(27)+"=" TO MSB0        Bit 7 auf 0 ziehen
            STORE CHR(27)+"#" TO MSBOFF     Bit 7 so lassen
            STORE CHR(27)+"P"+CHR(1) TO PICA
            STORE CHR(27)+"P"+CHR(128) TO ELITE
            STORE CHR(15) TO COMPRON         Schmalschrift ein
            STORE CHR(18) TO COMPROFF       Schmalschrift aus
            STORE CHR(14) TO DBLON          Breitschrift ein
            STORE CHR(20) TO DBLOFF         Breitschrift aus
            STORE CHR(27)+"E" TO BOLDON     Fettdruck ein
            STORE CHR(27)+"F" TO BOLDOFF   Fettdruck aus
            STORE CHR(27)+"-"+CHR(1) TO ULON Unterstreichen ein
            STORE CHR(27)+"-"+CHR(128) TO ULOFF Unterstreichen aus
    
```

d B A S E: Steuerzeichen

```

CASE $(ATTRSTR,2,1) = "d"
    RELEASE MSBO,MSBOFF,PICA,ELITE,COMPRON,COMPROFF,DBLON,DBLOFF,;
        BOLDON,BOLDOFF,ULON,ULOFF
ENDCASE
ENDIF
RELEASE ATTRSTR
RETURN

```

Wenn ich z. B. ein Menü mit intensiv hervorgehobenen Auswahlnummern ausgeben will, gehe ich folgendermaßen vor:

- ? BRI+"1"+NORM,"- neuen Satz aufnehmen"
- ? BRI+"2"+NORM,"- Satz ändern"
- ? BRI+"3"+NORM,"- Satz löschen"

Auf dem Bildschirm erscheint dann:

- 1 - neuen Satz aufnehmen
- 2 - Satz ändern
- 3 - Satz löschen

Noch einige Bemerkungen dazu: Für einige Bildschirmsteuerzeichen habe ich einfach die Farbcodes genommen. Das kann (sollte) man natürlich auch anders machen. Die ULOFF Variable für den Drucker funktioniert nur, wenn man vorher MSBO an den Drucker geschickt hat (vergl. dazu Info 18-7 und 27-47)! Die DBLON Variable schaltet den Drucker nur für die aktuelle Zeile in den Breitschrift Modus. Er wird durch den nächsten LINE FEED oder durch DBLOFF wieder ausgeschaltet, je nachdem, was zuerst eintritt. Als Alternative dazu gibt es einen ESC+'w' Steuerbefehl. Daß ich im ersten CASE nicht einfach ATTRSTR="B" verwende, hat den Grund, daß die Abfrage so unabhängig vom EXACT Modus funktioniert.

Zum Schluß noch einige Anmerkungen zu Peters Artikel. Die von ihm vorgeschlagene Möglichkeit zur **Speicherung von logischen Variablen** kam mir natürlich auch zuerst in den Sinn, ich fand es aber reichlich umständlich. Inzwischen hab ich eingesehen, daß es mit dem DBFLAG Programm für einfache Bedingungen auch nicht bequemer ist. Für zusammengesetzte Bedingungen (etwa "Wenn A und B und nicht C") scheint mir meine Methode aber einen kleinen Vorteil zu bieten. Im übrigen ließe sich das von Peter angegebene Beispiel auch in einer Zeile formulieren.

Zu PACK wäre zu sagen, daß dBASE einfach die gelöschten Sätze ans Ende der Datei verfrachtet und davor ein ^Z setzt. Wenn man also Sätze löscht, um Diskettenkapazität freizumachen, muß man mit COPY eine Zwischendatei anlegen, die alte Datei löschen und der Zwischendatei wieder den alten Namen geben.

Die von Peter angegebenen Startadressen für **Assemblerprogramme** habe ich zum Anlaß genommen, mein DBFLAG für jede dieser Adressen zu assemblieren. Die drei HEX Dateien habe ich in der Datei DBFLAG.TXT zusammengefaßt und an Herbert geschickt (ist auf CLUB.033). Jeder Benutzer kann sich dann den für ihn passenden Teil daraus kopieren. Ich bleibe aber dabei, daß sich das Assemblerprogramm mit RET beenden muß. Der RETI Befehl dient nur für einen Return von einer Interrupt-routine.

In Info 28-5 sprach Peter das Problem **Zeile voll** an. Seine Interpretation ist nicht ganz korrekt. Denn dBASE weiß ja nicht, daß der Drucker im Modus "Pica compressed" betrieben wird, insofern schickt es auch keinen LF hinterher. Man muß jedoch wissen, daß dBASE im RAW OFF Modus an die Ausgabe noch ein Blank anhängt, es könnte ja noch mit "??" weitergehen (s. Info 26-5). Also hat Peter in seinen Beispielen 133 Zeichen an den Drucker ausgegeben, und dieser hat den Zeilenumbruch bewirkt. Peter sollte sein Beispiel also nochmal im RAW ON Modus wiederholen, dann müßte es klappen (ich hab's nicht probiert).

d B A S E: Probleme**RESET oder Schampus oder was??**

(Peter Würfel, 7262)

(die Fortsetzungsgeschichte von Info 27-49 und Info 28-06)

Auf meiner RAM-Disk A: befindet sich dBASE, in C: eine Diskette mit der Datei 'test1.dbf', eine zweite Diskette mit 'test2.dbf' liegt bereit. Also müßte nach Wolfgang Kühn (Info 28-06) folgendes klappen:

Laufwerk C: anmelden, dann

```
C>a:dbase
```

```
set defa to a
```

```
do test
```

```
...
```

```
* ---- test.cmd
```

```
? 'testlauf'
```

```
use c:test1
```

```
appe blank
```

```
? 'Leersatz angehängt'
```

```
use
```

```
? 'Diskette in C: wechseln'
```

```
wait
```

```
reset C:
```

```
use c:test2
```

```
appe blank
```

```
? 'Leersatz angehängt'
```

```
use
```

```
? 'Ende Testlauf'
```

doch nach dem Diskettenwechsel der Frust:

```
P2DOS error on C: R/O
```

```
Function =34 File =TEST2 .DBF
```

Also wieder nix! (übrigens auch dann nicht, wenn ich mir das 'set defa to a' spare und von C: aus die Befehle im Direktmodus eingebe). Liegt es vielleicht an meiner (gepatchten) dBASE-Version? Ein Anruf bei Wolfgang und zwei Tage später zeigen mir das Programm DIFF.COM, daß wir grundsätzlich mit der gleichen M&T-dBASE-Version arbeiten, und mit seinem dBASE auf meiner Maschine klappts auch nicht.

...vier Stunden später, die Lösung:

nicht: 'reset C:' sondern nur: 'reset', das aber zweimal(!!) hintereinander (natürlich jeweils in einer neuen Zeile)!

Ob das nur bei meiner Maschine (sie läuft mit 5MHz) notwendig ist, konnte ich bisher nicht feststellen. Nun klappt das Programm auch, wenn ich dbase auf meiner SRAM-Floppy I: habe (C>i:dbase) und die Overlays auf der EPROM-Floppy H:, das Wichtige ist, daß von einem mechanischen Laufwerk aus aufgerufen wird (in diesem Beispiel C:), nicht von einem RAM-Laufwerk; welches Laufwerk mit 'set defa to ...' angemeldet wird, ist nebensächlich.

Und dann noch eine Lösung, bei der ein 'reset' ausreicht: Ich rufe dBASE von D: aus auf (ein 3''-LW), habe mein(e) Programm(e) auf A: (set defa to a) und suche die Datenbank-files auf C:; das LW D: wird nur beim Reset gebraucht.

d B A S E: Patch für Umlaute

dBASE-Patch

(Wolfgang Dexheimer, 6719)

Patch für Umlaut-Behandlung von dBASE:

- a) ! wandelt Umlaute auch in Großbuchstaben um
- b) Erkennt Umlaute (ohne ß) bei PICTURE "A" als Buchstaben

dBASE-Version 2.40

- a) Adresse 12F5h: alt 7B neu 7E
- b) Adresse 3688h: alt 5B neu 5E
368Fh: alt 7B neu 7E

dBASE-Version 2.41

- a) Adresse 1317h: alt 7B neu 7E
- b) Adresse 373Dh: alt 5B neu 5E
3744h: alt 7B neu 7E

Hier nun die Test-Programme dazu:

- a) .? !("test für dbase mit äöüß")<RET>
Resultat: TEST FÜR DBASE MIT ÄÖÜß
- b) .STORE " " TO Testwort<RET>
.ERASE<RET>
.§ 10,10 SAY "Buchstabentest";<RET>
.GET Testwort PICTURE "AAAAAA"<RET>
es erscheint die Eingabemaske bei 10,10
.READ<RET>
.? Testwort<RET>
.? !(Testwort)<RET>

Leserbrief: Kurt-Bernd Rohloff, 8000

Kurt-Bernd Rohloff, 8000
München, 08.10.88

Lieber Herbert,

nachdem das RESET Problem in dBASE wohl nun endlich gelöst ist, kann ich so gleich von einem anderen berichten. In dem Programm MS.COM werden für eine Suchabfrage zwei sehr lange Variablen MSSUCHA und MSSUCHB aufgebaut, in denen eine Bedingung formuliert wird. Die Variablen sollen dann in dem Unterprogramm MS1 wie folgt verwendet werden:

```
IF &MSSUCHA
  IF &MSSUCHB
```

```
  ...
ENDIF
```

```
ENDIF
```

Beide Variablen sind kürzer als 254 Zeichen. Trotzdem bricht das Programm u. U. bei der vorletzten STORE Anweisung für MSSUCHB einfach ab! Die Umstände sind genau gesagt folgende:

1. In der Eingabemaske werden alle Felder ausgefüllt, und
2. für den Datumsvergleich wird nicht "=" oder "<>" gewählt, und
3. der Suchmodus ist mit "3" ("irgendwo im Feld") gewählt worden.

Noch schlimmer: dBASE hat mir in der zugehörigen Datenbank MUSIK den ersten Satz versaut. Wenn Du ihn Dir mal anschaust, erkennst Du darin Teile von einer der beiden Variablen! Beachte dabei bitte auch das Löschkennzeichen (!)! Zum Glück enthielt der Satz keine Nutzdaten, sondern diente nur als Wachhund für den Dateianfang. Auch ein Heraufsetzen der Startadresse von DBFLAG auf C000, wie von Peter Würfel angegeben, brachte keine Besserung. Ich habe zwar schon eine Umgehungs-lösung im Kopf, aber trotzdem die

Frage: Ist das Problem bekannt? Weiß jemand Abhilfe (Patch)?

Auf den folgenden Seiten findest Du Teile (kommentierten) Protokolls, einmal für einen erfolgreichen Durchlauf (Suchmodus 1) und einmal für den Fehlerfall (Suchmodus 3).

Anm.d.HzN.: Wer Kurt-Bernd helfen möchte, kann von Ihm sicherlich eine Diskette mit den Routinen und Daten dieses Fehlerfalles erhalten.

Übrigens, 80 Zeichen/Zeile finde ich reichlich blöd, weil NewWord dann immer horizontal scrollt. Ich selbst benutze 77 cpl und ELITE Druck (wohl cp 10) und Page offset 10. Dann paßt es gerade auf den Bildschirm und der Ausdruck sieht auch g'scheit (man muß sich ja anpassen) aus.

Anm.d.HzN.: RAM 4.x + Aufgerüstete 80Zeichen-Karte beheben dieses Manko!!

Wenn Du mir Deine Artikel mit 77 oder gar weniger Zeichen je Zeile schickst, macht das nichts. In diesem Fall bitte KEINE SILBENTRENNUNG (Hyphen Help) verwenden. Ich reformatiere sie dann gerne.

In DBFLAG.TXT habe ich mein DBFLAG Programm noch einmal für die drei von Peter Würfel angegebenen Startadressen assembliert. Damit ersetzt diese Datei die alte DBFLAG.HEX.

Servus
Kurt-Bernd

Leserbrief: Kurt-Bernd Rohloff, 8000

```

*
* Protokoll von MS.CMD (Lange Zeilen nach Spalte 80 einfach umgebrochen!)
*
* 1. Durchlauf: Suchmodus 1 (EXACT OFF), alle Felder der Eingabemaske
* ausgefüllt, Datumsvergleich auf ">="
*
* Dieser Durchlauf wird ohne Fehler abgearbeitet.
*
*
  IF MSSA <> " "
    STORE MSSUCHA+" .AND. SA=MSSA" TO MSSUCHA
  .AND. SA=MSSA
  ENDIF
  IF MSDM <> " "
    STORE MSSUCHA+" .AND. SPDM=MSDM" TO MSSUCHA
  .AND. SA=MSSA .AND. SPDM=MSDM
  ENDIF

* u.s.w.

  IF TRIM(MSTON) <> " "
    IF PEEK(41987) = 0
      STORE MSSUCHA+" .AND. TONART=TRIM(MSTON)" TO MSSUCHA
    .AND. SA=MSSA .AND. SPDM=MSDM .AND. BANDNR=MSNR .AND. BANDTYP='DP18' .AND. VEL
O=MSVELO .AND. STEREO=MSST .AND. SCHUTZ=MSJN .AND. $(DATUM,7,2)+$(DATUM,4,2)+$(D
ATUM,1,2)>='801212' .AND. KOMPON=TRIM(MSKOMP) .AND. TONART=TRIM(MSTON)
    ELSE
  ENDIF

  IF TRIM(MSSTCK) <> " "
    IF PEEK(41987) = 0
      STORE MSSUCHB+" .AND. STUECK=TRIM(MSSTCK)" TO MSSUCHB
    .AND. STUECK=TRIM(MSSTCK)
    ELSE
  ENDIF

* u.s.w.

  IF MSART <> " "
    STORE MSSUCHB+" .AND. ART=MSART" TO MSSUCHB
  .AND. STUECK=TRIM(MSSTCK) .AND. NUMMER=MSNUM .AND. OPUS='33' .AND. SPIELER=TRI
M(MSPLAY) .AND. GERAET=MSGER .AND. BEM=TRIM(MSBEM) .AND. LTG=TRIM(MSLTG) .AND. S
OLIST=TRIM(MSSOLIST) .AND. SPUR=MSSPUR .AND. ART=MSART
  ENDIF
  SET EXACT OFF
  SET ECHO OFF
+++Länge von MSSUCHA: 230
+++Länge von MSSUCHB: 214
*
***** ENDE DES ERSTEN DURCHLAUFS

```

Leserbrief: Kurt-Bernd Rohloff, 8000

- *
 * 2. Durchlauf: Suchmodus 3 (Prüfung auf Enthaltensein), sonst alles
 * wie im 1. Durchlauf.

```
IF MSSA <> " "
  STORE MSSUCHA+" .AND. SA=MSSA" TO MSSUCHA
.AND. SA=MSSA
ENDIF
```

- * u.s.w.

```
IF TRIM(MSTON) <> " "
  IF PEEK(41987) = 0
    STORE MSSUCHA+" .AND. (TRIM(MSTON) $ TONART)" TO MSSUCHA
.AND. SA=MSSA .AND. SPDM=MSDM .AND. BANDNR=MSNR .AND. ('DP18' $ BANDTYP) .AND.
VELO=MSVELO .AND. STEREO=MSST .AND. SCHUTZ=MSJN .AND. $(DATUM,7,2)+$(DATUM,4,2)
+$(DATUM,1,2)>='801212' .AND. (TRIM(MSKOMP) $ KOMPON) .AND. (TRIM(MSTON) $ TONAR
T)
  ENDIF
ENDIF
```

```
IF TRIM(MSSTCK) <> " "
  IF PEEK(41987) = 0
    STORE MSSUCHB+" .AND. (TRIM(MSSTCK) $ STUECK)" TO MSSUCHB
.AND. (TRIM(MSSTCK) $ STUECK)
  ENDIF
ENDIF
```

- * u.s.w.

```
IF MSSPUR <> " "
  STORE MSSUCHB+" .AND. SPUR=MSSPUR" TO MSSUCHB
.AND. (TRIM(MSSTCK) $ STUECK) .AND. NUMMER=MSNUM .AND. ('20' $ OPUS) .AND. (TR
IM(MSPLAY) $ SPIELER) .AND. GERAET=MSGER .AND. (TRIM(MSBEM) $ BEM) .AND. (TRIM(M
SLTG) $ LTG) .AND. (TRIM(MSSOLIST) $ SOLIST) .AND. SPUR=MSSPUR
  ENDIF
IF MSART <> " "
  STORE MSSUCHB+" .AND. ART=MSART" TO MSSUCHB
***** HIER BRICHT DAS SCHWEIN EINFACH AB *****
* Es erfolgte kein Echo des neuen Wertes von MSSUCHB, wie es bei TALK ON
* sein müßte. Das nächste, was auf den Bildschirm geflattert kam, war der
* folgende Punkt!
. use
```

Leserbrief: Hartmut Traber, 5720

Hartmut Traber

Hohbeulstr. 8, 02.10.88
5270 Gummersbach, 02261/65399

Lieber Herbert,

Besprechung:

Formstar

Formstar ist prima, es formatiert sogar Formate, für die das Laufwerk nicht geeignet ist (gilt glaube ich nur für 1C und 1D)!

Scherz beiseite: Es ist prima, aber die MS-DOS-Formate fehlen ihm.

Alle Klicker sind nicht nur brauchbar, sondern Spitze!

DDT benutze ich auch nicht mehr, MONI2 ist besser!

Etikett.com benutze ich häufig ...

Ansonsten: Auf jeder Club-Disc war Brauchbares drauf, manchmal auch nur zum Lernen, was kein Nachteil ist, billiger kann man nämlich nicht lernen!

Zur Hardware:

Alle Hardware aus dem Club ist nützlich und sagenhaft preiswert.
Auf die Uhr und den ECB möchte ich nicht mehr verzichten.

Frage:

1. Spricht was dagegen, die ECB-Platinen in die FDX zu packen, ohne Konverter-Platine?
2. Brauchen wir dann noch eine Terminierung oder kann man drauf verzichten?
3. Was hätte das für Auswirkungen auf die ECB-Option bzw. ECB-PAL?
(Bevor ich die Interface-Platine behandle ...)

Antw.d.HzN.:

1. Nein, wenn Du Lust hast das zu Lötten !
2. Das kann nur der Versuch zeigen. Besser ist JA!
3. Weg mit beiden, falls ECB-Karten an FDX-Bus angeschlossen
Ent- oder weder, falls ECB-Karten in FDX und an Flachbandkabel (ECB-Adapter). Was und wie hängt von Port-Belegung ab.

Anm.d.HzN.: Gerhard Witzel ist gerade dabei, eine FDX-Karte zu entwerfen, die eine ECB-Karte aufnehmen kann. Vielleicht kommt da auch gleich noch die Hardware-Uhr oder eine Terminierung oder ... drauf.

copyd.com ist gefährlich!

copyd.com ist in meiner vorliegenden Fassung voreingestellt auf b:=d:.

(Ich habe beide Laufwerke!)

Rufe ich auf: copyd^M, ist sofort das Directory von b: zerstört.

Abhilfe ? Antw.d.HzN: Patch Ziel=Quelle (s.o.)

Ich denke, analog sonstiger Progr's, sollte man die Slashes einbauen und damit eine kurze Bedien-Anweisung ausgeben.

Nichts für ungut!

Viele Grüße nach Hamburg
Hartmut

S o f t w a r e: R A M 4 . 3

RAM 4.3 on Stage

(Herbert zur Nedden, 2000)

Was ist neu?

1. TOAM ist nun bei 0E780H, d.h. es können sogar 4 Laufwerke mit großen Formaten (1A, 1B, 1C, ...) CONFIGuriert werden.
2. Tastaturpuffer ist länger und piepst, wenn voll.
3. Tastaturtreiber: AlphaLock+Shift+Buchstabe = KleinBuchstabe.
4. AlphaLock-LED (d.h. Lämpchen) wird unterstützt.
5. Interrupt-Tabelle bei 0F080H um 12 Einträge verlängert.
6. P2DOS-Zeileneingabe: Mit ^E, ^X in beide Richtungen durch Zeilenstack
7. Zusätzlicher Zeileneditor mit NewWord-artiger Eingabe als RAM4-Sprung verfügbar. (Ctrl-Codes nur über ^P)
8. Spooler kann auch auf selbstdefinierte Ausgaberroutine gelegt werden.
9. Bildschirmschoner.
10. Bildschirm-Sequenz Esc Ä E Zeichen füllt bis EoL mit Zeichen.
11. KLIX-Blöcke können so gekennzeichnet werden, daß ein KLICK-Warmboot sie löscht. (Gedacht für CP/M-Programme)
12. CFIG43 kann die Formate 03, 07, 09, 0A und 1B (unter geeigneten Bedingungen) automatisch erkennen.
13. RCHECK43 macht beim Format 1C keinen Bildschirm-Müll.
14. MS kann auch das PS/2-1.44MB-Format verarbeiten
15. Aktualisiertes FORMATE4.DAT ist dabei.

Was trotz gegenteiliger Dokumentation nicht funktioniert:

1. Zusätzlicher Zeileneditor (7.) kopiert Zeilenpuffer nach der Eingabe auf Bank 0. Das gilt auch für den P2DOS-Zeileneditor (6.) Soll dieser in KLICK-Overlays genutzt werden, kann der Zeilenpuffer zwischen 0F980H und 0F9FF liegen.

Was bei RAM43-Versionen vom Clubtreffen getan werden muß:

1. Damit die Software-Uhr nicht so schnell ist, sollte aber auch bei Hardware-Uhr erfolgen: Adresse 15E8: alter Inhalt: 0CAH, neu: 0C2H.

Was der Spaß kostet:

Als Update zu RAM 4.2 popelige DM 5.- incl. D&P&V, sonst siehe Angebotsliste.

Bestellung:

Mit beiliegendem Bestellzettel bei BP oder HzN.

RAM4.3-Bestellzettel

Ich bestelle hiermit das RAM4-System und verpflichte mich, die Urheberrechte des Autors in jeder Hinsicht zu wahren.

Name: _____ Vorname: _____

Straße: _____ Wohnort: _____

- Ich bin RAM4.2-Besitzer und zahle DM 5.- (Update)
- Ich bin RAM4.1-Besitzer und zahle DM 10.- (Update)
- Ich bin RAM3-Besitzer und zahle DM 45.-
- Ich bin kein RAM3-Besitzer und zahle DM 55.-

Ich möchte das RAM4-System für die FDX SDX
 ich habe EDICTA- und/oder 1MB-DISI-Ramfloppy und die umseitig genannten.

Den oben angekreuzten Betrag habe ich

- auf das Konto Bernd Preusing, Gerhardstr. 16, 2200 Elmshorn
Volksbank Uetersen, BLZ 221 916 17, KtoNr. 62812 überwiesen.
- als Verrechnungs-Scheck beigelegt.
- Ich wünsche Abbuchung von meinem Clubkonto.

Datum, Unterschrift

Einsenden an Bernd Preusing, Eichenkamp 6, 2200 Elmshorn
oder an Herbert zur Nedden, Sonnenau 2, 2000 Hamburg 76

↑ hier durchtrennen

L e s e r b r i e f: Hans Gras, NL 1506

Hallo, hier etwas über meine Experimenten mit Library Utilities.

Das LayOut von das Programm **LDIRZ.COM** gefällt mir nicht so gut. Folgendes eine beschreibung von meine Änderung von **LDIRZ.COM**.

```
DDT LDIRZ.COM [ret]
S0195 [ret]
31 [ret]           Versions nummer 2.1
. [ret]           Ende Teil 1
S0543 [ret]       Kopfzeile aendern
20 [ret]
20 [ret]
00 [ret]
00 [ret]
2A [ret]
9A [ret]
06 [ret]
EB [ret]
7A [ret]
B3 [ret]
CA [ret]
78 [ret]
05 [ret]
1B [ret]           Ende Teil 2
. [ret]

S0576 [ret]
4B [ret]
. [ret]           Ende Teil 3
S0580 [ret]
00 [ret]
00 [ret]
00 [ret]
. [ret]           Ende Teil 4
S05D8 [ret]
20 [ret]
6B [ret]
42 [ret]           Ende Teil 5
[ret]
[ret]
[ret]
2E [ret]
. [ret]           Ende Teil 6
S060B [ret]
. [ret]
[brk]
SAVE 23 LDIRZN.COM [ret]
```

Das gibt bei mir folgendes Schirm:

E>LDIRZN -COMMAND.LBR *.*

LDIRZ Version 2.1

Filename	Typ	Size									
ASM	.COM	8k	INITDIR	.COM	2k	MKDIR	.COM	8k	STBACKUP	.COM	14k
BASICODE	.COM	6k	IOLOADER	.COM	4k	MODEMB	.COM	17k	STLZ	.COM	4k
BAUD	.COM	1k	KILL42	.COM	4k	MS	.COM	24k	STSZ	.COM	2k
BSAVE	.COM	2k	L80	.COM	11k	MSFORM4	.COM	3k	STSAUTO	.COM	2k
COMPARE	.COM	5k	LABEL	.COM	1k	OLDCAT	.COM	5k	SYSCOPY4	.COM	1k
CRC	.COM	3k	LD	.COM	5k	PUTCAT	.COM	13k	TOUCH	.COM	1k
DATE4	.COM	2k	LOAD	.COM	2k	RCHECK4	.COM	1k	UNARC	.COM	5k
DD	.COM	16k	M80	.COM	20k	RELDUMP	.COM	2k	UNERA	.COM	1k
DDT	.COM	5k	MASTSTAT	.COM	2k	SAP	.COM	2k	UNLOAD	.COM	1k
DEVICE	.COM	3k	MCHECK	.COM	5k	SAVSPOOL	.COM	12k	VIEW	.COM	4k
DZCONV	.COM	7k	MDC-CAT	.COM	3k	SETSPOOL	.COM	13k	WHEEL	.COM	2k
ECHO	.COM	1k	MDC-DZ	.COM	3k	SIZE	.COM	14k	XDIR3	.COM	11k
FORM4	.COM	2k	MDC-MB	.COM	3k	ST	.COM	4k	XLATE2	.COM	6k
GENINS	.COM	13k	MDC-NW	.COM	3k	STARTUP	.COM	1k	Z8OASM	.COM	24k
GETREL41	.COM	12k	MDC-RD	.COM	3k	STAT	.COM	6k	ZEX	.COM	6k
IDC	.COM	11k									

61 Files Selected occupying a Total of 377 kByte.

Dies ist ein wenig schöner, wass?

L e s e r b r i e f: Hans Gras, NL 1506

Weiter habe ich eine Library mit verschiedene Info-Dateien. Um diese Info-Dateien anzusehen habe ich ein speciele LTYPEZ.com geschrieben. Arbeitet auf gleiche Weise als LRUNZ.com. Als Beispiel:

LTYPEZ IOBYTE ---> Typed das File IOBYTE.INF aus das Default Library INFO.LBR.

LTYPEZ MBASIC.HLP ---> Das gleiche mit MBASIC.HLP.

LTYPEZ -HELP BASCOM.HLP ---> Das gleiche mit BASCOM.HLP aus Library HELP.LBR.

Als Default gilt: Library ist INFO.LBR und Extensions sind bzw. .INF und .LBR.

Wie man sehen kan in Inhalt meines Libraries, habe ich das Programm UNARC.COM. Dies ist ein Programm um ARCHivierte Files wieder zurück zu holen aus eine Library des Types .ARC (für MS-DOS Geräten). Wer hat ARC.COM für CP/M?? Es combiniert ungefähr SQ/USQ (Squeezes) mit LU (Libraries). Sehr vernünftig für BackUp Zwecken!! Es gibt normalerweise ca. 40% "mehr" Raum auf eine Diskette.

So. Nächstes Subject: Formattieren des Memotech Scheiben Format #03.

Ich habe für viele Infos zurück etwas gefragt über langsame Disketten des types #03. Leider kam kein Lösung für meine Probleme. Dafür habe ich selbst mal herausgefunden warum es ein Problem war.

Bernd Preusung hat in seine FORM4x.com eine Sektor Skew von 2. Das geht normalerweise (ohne RAM z.B.) schneller, aber das Verify einer Programm dauert zu lange. Einer Skew von 2 geht nicht. Ich habe meine Original Memotech System Disc mal untersucht mit das schöne Programm READTRK.COM von Herbert Oppmann und entdeckt: Memotech hat eine Sektor Skew von 3. Dan bleibt die Frage: wie formattiere ich die Sektor Skew von 3. Ganz einfach: FORM4x.com ein wenig patchen. Aber ich habe FORM4x.com bereits gepatcht für Highest Density 1C Format, (Fehlerchen ins Info 26: 8" einsetzen!) en 2x patchen und Format 10 daraus smeißten ist mir zu viele. Dafür ein neues FORM45.com.

Dann haben wir immer noch RCHECK4.com mit Problemen mit Format 1C. Dafür auch etwas schönes: RCHECK45.com. Versuch es mal. Programmen sind auf CLUB.03x. Herbert Oppmann, ich danke dich für ihre Einsatz. Format 1C (und wahrscheinlich Format 1D auch) läuft bei mir (auf eine Teac FD 55 GFV) einwandfrei.

Eine **Warnung** für COPYD (Version HzN): sofort das teil "Defaults" entfernen. Bei Anruf ohne Parametern kan eine "Unschuldiges" Disc zerstört werden. Einfach eben unter das Label Start: jp nz,Parse --> jr Parse (ohne nz,)!

Anm.d.HzN.: Du kannst die Defaults ja ändern (z.B. B:=B:), aber warum Dir Disketten dabei kaputt gehen, verstehe ich nicht. Gerade diese Defaults machen das Programm doch erst schön nutzbar. Dann muß ich weniger eingeben!

Noch eine **Warnung**. Das Programm NSWEEP verarbeitet eine Rename nicht immer richtig, wann eine File größer ist als 512kB!! Die überlaufende Extents werden gelöscht!

Hans Gras.

Leserbrief: Hans Gras, NL 1506

F: (Hans Gras, NL-1506)

Wer hat auf seine GraficSchirm (VS4) ein höhero auflösung als 256x192 erreicht (zB. 256x256 pixel in weniger Farben, nur BitMap, kein Sprites usw.)? Ist so etwas überhaupt möglich?

A: (Herbert zur Nedden, 2000)

Ich glaube, daß Du bei unserem Video-Controller im MTX da keine Chancen hast!!! Der ist nur für diese eine Auflösung gemacht.

F: (Hans Gras, NL-1506)

Wer hat sich bemüht mit "FELIX", eines (PD) "Grafic Animation System"? Bitte bei mir melden. Wer nicht weißt was es ist, auch bei mir melden.

F: (Hans Gras, NL-1506)

Wer hat seine Memotech Newword 2.02 auf eine andere Rechner ans laufen? Ich habe versucht es auf eine Osborne 1 (Z80, 5.25") und eine Ampex (8085, 8") ans laufen zu kriegen. Ich wollte der zweiter Rechner als PrintRechner einsetzen. Man kan alles machen, ausser abspeichern und printen!

* Es ist möglich um NewWord mit alle Overlays aufs eine Osborne 1 Diskette (90KB netto) zu packen! Man muß nur auf die Hilfsschirmen versichten. Wer intressiert ist, bei mir melden.

F: (Hans Gras, NL-1506)

Wer hat eine Auto-Repeat ins RAM 4.x Tastatur Treiber eingebaut?

A: (Hans Gras, NL-1506)

Marco Bakker und ich habe ein "Overlay" gebastelt welche es möglich macht um eine Auto Repeating Keyboard Treiber (USA/Deutsch) zu installieren. Verschiedene Sachen sind einstelbar. Klick fähig! Leucht Diode Steuerung fürs Alpha-Lock und mehr. Mehr Info in einer der nachsten Infos.

Anm.d.HzN.: siehe RAM 4.3-Ankündigung in diesem Info!

Ob das o.g. Programm von HG und MB mit RAM 4.3 noch harmoniert wage ich stark zu bezweifeln!

F: (Herbert zur Nedden, 2000)

Ich verstehe nicht, wozu dieses beknackte AutoRepeat gut sein soll!!! Entweder kann die Tastatur mit AutoRepeat leicht mal prellen, oder die Zeit, bis es los geht ist uninteressant lang. Ich kann mit der RAM 4.x Tastatur, d.h. mit LineFeed als RepeatTaste, wesentlich besser arbeiten als mit einigen professionellen Tastaturen. Diese prellen zwar nicht, aber es dauert Jahrhunderte bis das AutoRepeat einsetzt. Dann ist das Repeat meist noch zu träge. Eigentlich braucht man das AutoRepeat doch nur bei wenigen Tasten, i.a. doch wohl den Cursorstasten! Da ist die Steuerung über AutoRepeat nicht so leicht mit der Präzision wie mit LineFeed möglich!

C L U B: Der ideale Programmierer ?**Bin ich der ideale Programmierer?**

(Olaf Krumnow, 2050)

Manchmal habe ich das Gefühl, ein solcher zu sein. Eingebildet, meinst Du? Klingt natürlich so, ist aber nicht so gemeint. Ich will hiermit vielmehr andeuten, daß von Euch so wenig Reaktionen zu Programmen kommen, daß ich manchmal meinen könnte, meine Programme würden es allen Recht machen. Und das wäre doch ideal, oder?

Es sind nunmehr 3 KLIICK-PD's erschienen, die eine Vielzahl von Routinen und Programmen enthalten. Diese Programme (soweit sie von mir sind) gehen immer von meinen eigenen Bedürfnissen und Hardwarebeständen aus. Logisch, ich kann ja nicht hellsehen. Daß aber immer wieder Probleme mit anderen Einstellungen und Geräten auftauchen, macht mir im Allgemeinen nur ein einziger klar. Das ist Herbert, der mich deshalb mehr oder weniger ständig am Telefon beschäftigt hält. Wenn aber schon mit Herberts System und Bedürfnissen solche Differenzen auftreten, wie muß es dann erst bei Euch sein?

In diesem Zusammenhang interessiert es natürlich auch, welche Programme Ihr überhaupt benutzt und welche am verstauben sind. Ich kann mir vorstellen, daß z.B. Hardcopy doch häufiger benutzt wird, aber das ist nur eine Vermutung. Ohne Eure Reaktionen weiß ich nicht, ob es sich lohnt, ein Programm weiterhin zu pflegen, oder ob es auf einer Stufe stehenbleibt, auf der ich damit leben kann.

Vermißt Ihr irgendwelche Programme, traut Euch aber nicht zu, sie selbst zu schreiben? Vielleicht hilft eine Anfrage im Info, ob bereits jemand etwas in diese Richtung unternommen hat. Vielleicht unternimmt ja auch jemand etwas, der nur bisher nicht die Idee hatte oder den Aufwand für sich alleine gescheut hat.

Das gilt natürlich nicht nur für mich, sondern für alle Leute, die Programme im Club veröffentlichen. Und soweit ich weiß, sind auch alle Programmierer bereit, soweit möglich auf Eure Bedürfnisse, Ideen und Anforderungen einzugehen. Daß nicht immer Konsens zu erreichen ist, versteht sich von selbst. Aber viele Leute sorgen für viele Ideen und damit für vielseitige und gute Programme. Und dazu sollte ein Club doch eigentlich da sein, oder?

Also Leute, schreibt, telefoniert, was das Zeug hält. Nur so können wir für Euch alle dasein!

H a r d w a r e: Drucker**DMX80-Tip ... Out of Paper**

(Herbert zur Nedden, 2000)

Der DMX80 hat einen etwas ungünstig platzierten Papier-Ende-Sensor, der bei DIN A4-Papier etwa drei Zeilen zu früh meldet, daß das Papier alle ist: OUT OF PAPER

Wenn dem Drucker mittels Esc '8' (DMX 80) gesagt wurde, er soll das Papier-Ende ignorieren, so liefert der Drucker zwar an seinem PAPER-END-Signal den Hinweis ab, daß das Papier zu Ende ist, aber sein BUSY-Signal sagt weiterhin 'ich bin bereit', d.h. der Drucker bleibt On Line - es kann also weitergedruckt werden, auch auf die Walze.

Damit dieses funktioniert, ist die Abfrage des Centronics-Status' unter RAM 3.x und 4.x so geändert worden, daß nur noch das BUSY-Signal ausgewertet wird. Wer ohne RAM 3.x/4.x arbeitet: Siehe Info 14, Seite 13.

So weit so gut.

Was aber nun, wenn zwar die BUSY-Abfrage für die Centronics-Schnittstelle richtig ist, aber leider vergessen wurde, dem Drucker mitzuteilen, daß er das Papierende ignorieren soll.

Da ist nun die Seite fast fertig - es fehlen nur noch ein paar Zeilen, und die Gurke meldet 'Out of Paper' und geht Off Line ?

Wenn ich jetzt eine neue Seite von hinten einlege, damit der Drucker merkt, daß doch Papier da ist, versucht er gleich diese neue Seite einzuziehen, d.h. er schiebt die fast fertige Seite raus. MIST! Da sollten doch noch ein paar Zeilen draufgedruckt werden.

Habe ich die Gefahr frühzeitig erkannt, dann kann ich natürlich das zweite Blatt Papier schon früher dazulegen, und alles klappt. Aber wann bin ich schon so plietsch (= plattdeutsch für schlau) ?

Erfreulicherweise bietet der DMX80 eine raffi-tückische Lösung für solche Fälle:
Die Online Taste!

Wenn eine DIN A4-Seite fast fertig bedruckt ist, der DMX 80 meldet, daß er kein Papier mehr hat, der Drucker daher Off Line ist, genügt ein zarter Druck auf die On Line Taste, und er druckt eine weitere Zeile.

Für jede noch fehlende Taste muß also einmal On Line gedrückt werden! Aber nicht zu oft, sonst wird die Walze immer schwärzer!

Hardware: Z80PIO und InterruptsZ80-PIO und Centronics/Carrera

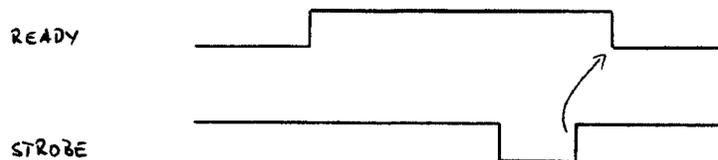
(Herbert zur Nedden, 2000)

Um einen Z80-PIO als Schnittstelle (Freaks sprechen auch von einem Interface) für einen Centronics-Drucker oder die Olympia-Carrera zu betreiben, muß etwas gelötet werden.

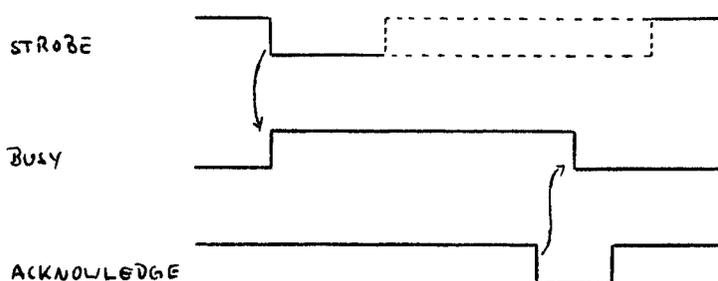
Da diese drei Kandidaten verschiedene Steuersignale alle nach eigenem Gusto verwenden, hier erst einmal die Bedeutung und das Timing (d.h. Zeitverhalten) der verwendeten Steuersignale.

Z80-PIO Timing im Ausgabe-Modus

1. Der Z80-PIO signalisiert mit einem High-Pegel an seinem READY-Ausgang, daß er Daten zur Übernahme hat.
2. Er erwartet an seinem STROBE-Eingang eine Empfangsbestätigung (Acknowledge), wobei der Wechsel dieses Pegels von Low auf High ausschlaggebend ist. Im Ausgabe-Modus (um den es hier geht) ist der STROBE-Eingang des Z80-PIO eigentlich ein ACKNOWLEDGE-Eingang.
3. Hat er die Empfangsbestätigung erhalten, geht sein READY-Ausgang wieder auf Low (d.h. 'keine Daten bereit'), und er löst, falls dafür programmiert, einen Interrupt aus.

Centronics Timing

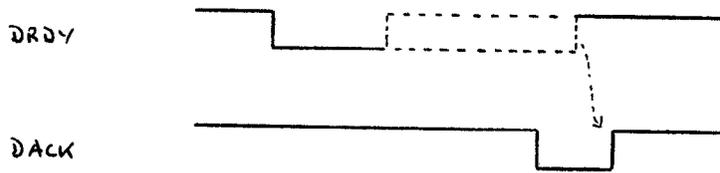
1. Erhält der Centronics-Empfänger (Drucker) an seinem STROBE-Eingang einen Low-Pegel, so übernimmt er die anstehenden Daten, sowie er dazu bereit ist.
2. Wenn er die Daten übernimmt, signalisiert er an seinem BUSY-Ausgang durch einen High-Pegel, daß er beschäftigt ist.
3. Hat er die Daten entgegengenommen, übermittelt er an seinem ACKNOWLEDGE-Ausgang einen Low-Impuls (d.h. High-Low-High) als Empfangsbestätigung.
4. Ist er dann bereit, weitere Daten zu übernehmen, so legt er seinen BUSY-Ausgang wieder auf High.



H a r d w a r e: Z80PIO und Interrupts

Carrera Timing (ohne doe Centronics-Interface-Box)

1. Erhält die Olympia Carrera an ihrem DRDY-Eingang einen Low-Pegel, so übernimmt sie die anstehenden Daten, sowie sie dazu bereit ist.
2. Wenn die Daten entgegengenommen wurden, legt sie ihren DACK-Ausgang auf Low.
3. Falls nun der DRDY-Eingang wieder auf High ist, geht auch der DACK-Ausgang wieder auf High.

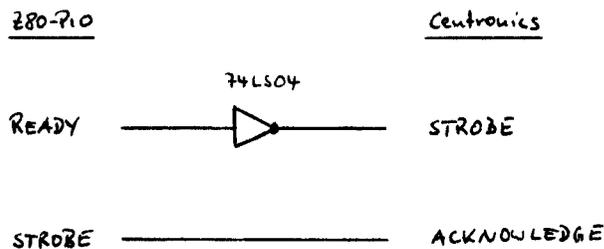


Z80-PIO und Centronics

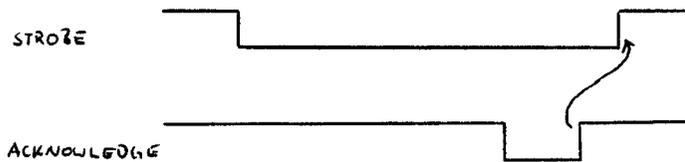
Wenn man mal das BUSY-Signal unter den Tisch fallen läßt, so verhält sich ein Centronics-Drucker genau so, wie es der Z80-PIO tut:

1. Der Z80-PIO signalisiert an seinem READY-Ausgang, daß Daten bereit sind, was der Centronics-Drucker an seinem STROBE-Eingang erfahren möchte.
2. Hat der Drucker die Daten verarbeitet, teilt er dies an seinem ACKNOWLEDGE-Ausgang mit, was der Z80-PIO an seinem STROBE-Eingang erfahren möchte.

Und genau so simpel funktioniert die Sache auch:



Timing:



Das BUSY-Signal ist nur für solche Centronics-Schnittstellen erforderlich, wo die CPU direkt die Statussignale des Druckers abfragt. Schließlich soll die CPU nach dem Absenden eines Zeichens an den Drucker nicht so lange warten müssen, bis sie den ACKNOWLEDGE-Impuls des Druckers erwisch hat - der Z80-PIO hingegen, hat sowieso nichts besseres zu tun, und informiert die CPU über den Erhalt des ACKNOWLEDGE mittels Interrupt.

H a r d w a r e: Z80PIO und Interrupts

Z80-PIO und Olympia Carrera

Wenn man sich die Signale der Carrera ansieht, macht alles eigentlich einen fantastischen Eindruck:

Die Carrera erwartet an ihrem DRDY-Eingang die Information, daß Daten bereit sind, und übermittelt via DACK die Empfangsbestätigung.

Leider ist da eine Tücke:

Der Z80-PIO will die steigende Flanke der Empfangsbestätigung haben, die Carrera liefert jedoch erst einmal nur die fallende Flanke, und erst wenn ihr DRDY-Eingang wieder 'keine Daten bereit' signalisiert, kommt die steigende Flanke der Empfangsbestätigung.

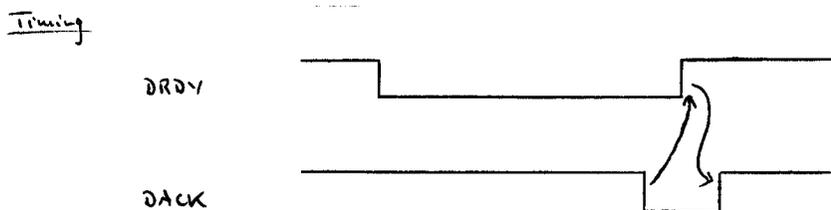
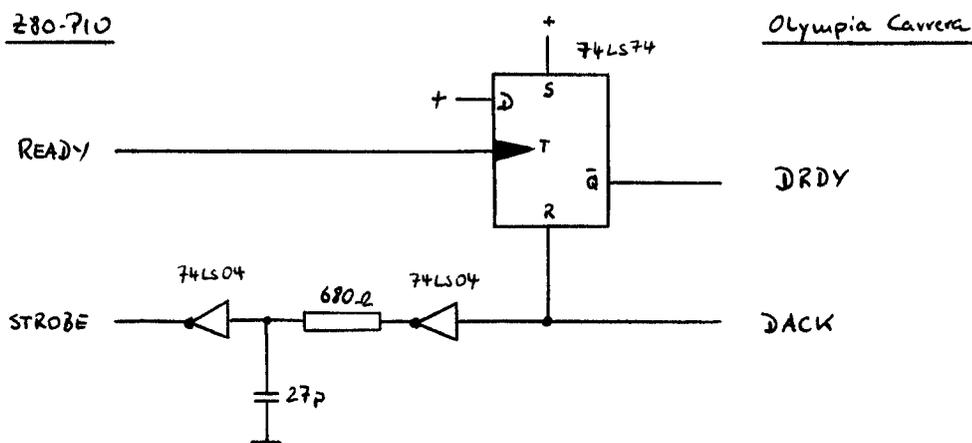
Die Lösung dieses Problems gestaltet sich erfreulicherweise auch nicht allzu kompliziert:

1. Die steigende Flanke des Z80-PIO-READY legt den DRDY-Eingang der Carrera auf Low (d.h. wenn der Z80-PIO sagt 'ich habe etwas' wird das der Carrera mitgeteilt).
2. Sowie der DACK-Eingang der Carrera auf Low geht, d.h. die erste Flanke der Empfangsbestätigung der Carrera kommt, wird der DRDY-Eingang der Carrera wieder auf High gelegt.

Damit ist sichergestellt, daß die Carrera ihren DACK-Ausgang wieder auf High legt, und der Z80-PIO kann diese steigende Flanke, d.h. seine Empfangsbestätigung erhalten:

1. Z80-PIO-READY setzt ein FlipFlop, dessen Ausgang das DRDY-Signal für die Carrera liefert.
2. DACK von der Carrera setzt dieses FlipFlop wieder zurück.

Die Tücke liegt im Detail, mußte ich feststellen. Damit sich die langsame Carrera nicht verschluckt, mußte ich ihr DACK-Signal noch etwas verzögern, bevor es an den Z80-PIO geht - sonst gehen einige Zeichen verschütt.



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.

Hardware: SDX - 80Zeichen-Karte

SDX-80Zeichenkarte mit mehr Kontrast

(Herbert zur Nedden, 2000)

Unten findet Ihr einen Schaltplan, wie Ihr den Kontrast der SDX-80Z-Karte deutlich verbessern könnt:

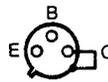
Ein einfacher Transistorverstärker.

Benötigte Teile

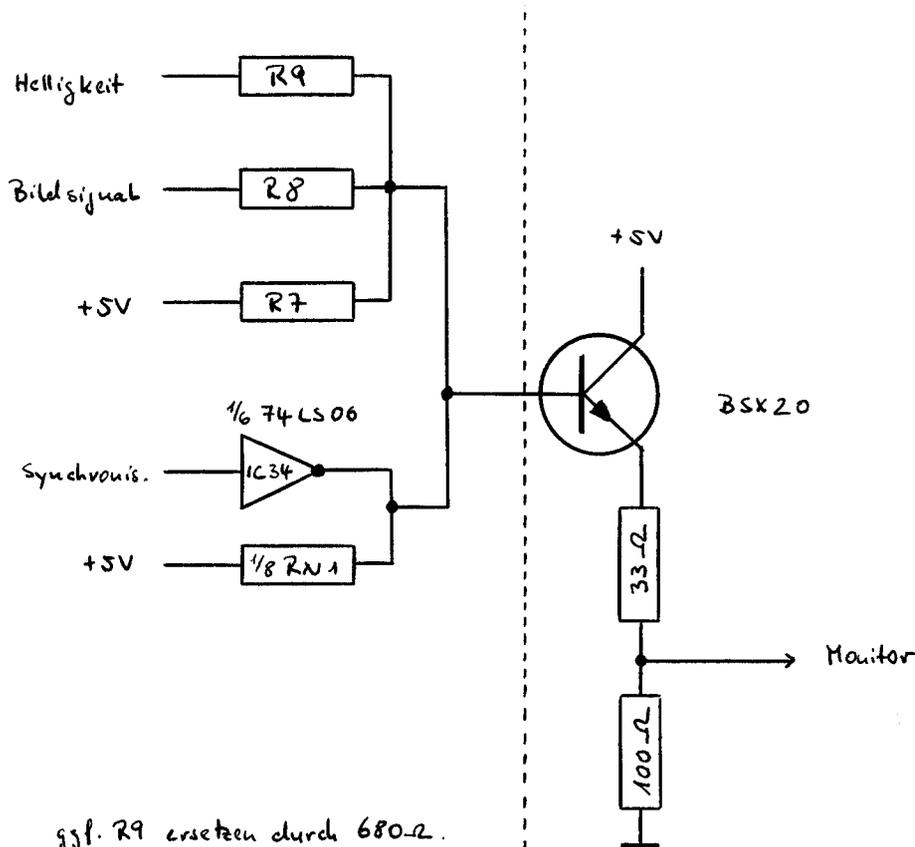
- Transistor BSX 20
- Widerstand 33 Ohm
- Widerstand 100 Ohm
- Kabel, Lötzinn, ...

Die Abtrennung des Monitorausganges von der 15-pol-Stiftleiste ist tückisch, da sie von zwei Seiten Zuleitungen hat, d.h. die Leitung von R7-R9 und vom LS06 treffen sich erst am Monitor-Pin.

Ansicht des BSX20 von unten:



Schaltung:



ggf. R9 ersetzen durch 680Ω.

Original SDX 80Zeichen-Ausgabe

Video-Verstärker

H a r d w a r e: Diskettenformate**Memotech's Diskettenformate: dSM**

(Herbert zur Nedden, 2000)

Es ist ja bekannt, daß Memotech jegliche Formate derart vergewaltigt, daß aus der Sicht von CP/M stets 26 Sektoren je Spur herauskommen. D.h. die sog. logische Spur ist 26 Sektoren lang.

Das ist ja auch der Grund, warum unser 40-Spur-Format (Format 03) kaum auf anderen Rechnern realisierbar ist.

Bei 'normalen' Diskettenformaten entspricht einer logischen Spur, d.h. einer Spur aus Sicht des CP/M immer

- eine physikalische Spur der Diskette auf einer Seite,
- eine physikalische Spur der Diskette auf Seite 1 plus eine auf Seite 2,
- oder ein vielfaches davon.

Auf jeden Fall gilt daher: (bei 'normalen' Diskettenformaten)

$$(\text{physikalische Spurlänge}) = (\text{ganze Zahl}) * (\text{logische Spurlänge}).$$

Betrachten wir nun Memotech's Format 03, bei dem eine physikalische Spur aus 16 Sektoren besteht (die logische Spur hat ja die bekannte Länge von 26 Sektoren), so ist der Quotient logische/physikalische Spur weit davon entfernt, eine ganze Zahl zu sein - und jedes 'normale' Betriebssystem hat damit Probleme.

Memotech-BIOS muß daher auch immer folgende Rechnung durchführen:

$$1. \text{ relativer Sektor} = (26 * \text{logische Spur}) + \text{logischer Sektor}$$

Diese relative Sektornummer wird dann umgerechnet in die physikalischen Werte Sektor und Spur:

$$2. \text{ physikalische Spur} = \text{rel. Sektor} / \text{physikalische Spurlänge}$$

$$3. \text{ physikalischer Sektor} = \text{rel. Sektor modulo physikalische Spurlänge}$$

Ein 'normales' BIOS berechnet diesen Murks gar nicht!

Um dem ganzen den letzten Schliff zu geben, hat Memotech auch noch seine Silicon-Disc-Formate ebenso vergewaltigt! WÜRG!!!! Das kostet Zeit für Umrechnungen und bringt gar nichts.

Wer also einen Disketten-Treiber von Memotech disassembliert, wird immer den Code wiederfinden, um eine Zahl mit 26 zu multiplizieren!

Übrigens stammt diese 26 als Spurlänge von den 8" CP/M-Standarddiskettenformaten. Ja, die Memotech-Formate 10, 11, 12 und 13 sind die Standard-Formate für CP/M-Scheiben auf 8" - so etwas gibt es (Standard meine ich). Memotech war dann der dringenden Auffassung, dieses Format sei der Weisheit letzter Schluß (machte also Schluß mit derselben) und verbog das physikalische Format der anderen Disketten und der SiliconDiscs auf dieses 26-Sektoren-je-Spur-Format.

H a r d w a r e: SDX-Sidisc

SDX-RamDisc im 3 1/2"-Laufwerk: SIDSIC.COM

(Herbert zur Nedden, 2000)

Ich habe mich drangesetzt und den SDX-SiDisc-Treiber disassembliert, um auch diese RAM-Floppy in RAM 4.x aufnehmen zu können. Das Ergebnis war recht interessant. (SIDISC.COM samt der REZILOG-Dateien findet Ihr auf der CLUB.032. REZILOG ist ein Disassembler-Tool von CLUB.001.)

Dieser Silicon-Disc-Treiber unterstützt immerhin zwei Arten von RAM-Floppies (ja, so schreibt sich der Plural von Floppy im Englischen!):

Formate 5x: SiDisc's, die den 'überschüssigen' Hauptspeicher, d.h. alles ab Bank 1 als RAM-Floppy nutzen, d.h. genau das, was RAM 4.x auch tut.

Formate 4x: SiDisc's, die keinen Z80-Hauptspeicher verwenden, sondern über drei Ports (Spur, Sektor, Daten) angesprochen werden:

- Schreibe Spur/Sektor auf die entsprechenden Ports
- Schreibe oder Lies 128 Bytes über den Daten-Port.

Sie werden übrigens genauso angesprochen, wie die Memotech-Silicon-Discs, die es für die FDX gibt.

Je nachdem, welcher Laufwerksbuchstabe (F, G, H oder I) genommen wird, ist die unterste der drei Port-Adressen (die sog. Basis-Adresse) 50h (F:), 54h (G:), 58h (H:) bzw. 5ch (I:).

Die Port-Adressen verteilen sich wie folgt:

```
Sektor equ Basis-Adresse +0
Spur   equ Basis-Adresse +1
Daten  equ Basis-Adresse +3
```

Natürlich schlägt hier Memotech auch mit der beliebten Umrechnung in Spuren mit 26 Sektoren voll zu - was für ein genialer MURKS!!

Stellt sich die Frage, wie diese Floppies in RAM 4.x eingebaut werden können:

Formate 5x: Tunlichst nicht verwenden, da schließlich unter RAM 4.x dieser Speicher schon für das Betriebssystem, die RAM-Floppy F:, den Drucker-Spooler und den KLIX-Heap verwendet wird.

Formate 4x: Werden von Haus aus von RAM 4.x unterstützt.

Allerdings verlangt RAM 4.x, daß die Port-Adressen ab 54h liegen:

```
Sektor equ 54h
Spur   equ 55h
Daten  equ 57h,
```

d.h. die Silicon-Disc für Laufwerk G: verdrahtet wurde: Im Original SDX-System muß CONFIG G:4x (für geeignetes x) laufen. Funktioniert nur CONFIG F:4x, oder H:4x oder I:4x, so stimmt die Port-Adresse nicht, und die Platine muß umgejumpert werden. Wie das geht, weiß ich nicht, da ich noch keine solche Platine gesehen habe.

Läuft mit dem Original-Memotech-System CONFIG G:4x, verweigert jedoch RAM 4.x den Zugriff, so:

- Falls unter RAM4 CONFIG G:4x nicht klappt, mit FORMATE4 oder INST4 prüfen, ob der Config-Code 4x in RAM4 installiert ist - ggf. einrichten.
- Klappt es immer noch nicht, RAM4.COM auf eine Diskette kopieren und an mich schicken. Evtl. hast Du eine Version, die diesen Treiber nicht beinhaltet, was recht unwahrscheinlich ist.

H a r d w a r e: SDX-Sidisc / Ports

Ich bin gerne bereit, eine RAM 4-taugliche SDX (d.h. eine mit mindestens 96 kB RAM auf der Hauptplatine) zu untersuchen, und diese dann ggf. entsprechend anzupassen, damit alles läuft!

Dafür benötige ich allerdings sowohl das Grundgerät, den Floppy-Teil als auch eine Original-Memotech-Systemdiskette (mit SIDISC.COM), die RAM4-Scheiben und zwei Leerdisketten, die ich dann fertig installiert mit RAM4 vorbereiten kann.

Falls Du keine RAM 4.3-Lizenz hast, leg bitte den ausgefüllten Bestellzettel anbei. Dann kommt die Kiste gleich mit RAM 4.3 zurück.

Ports

(Herbert zur Nedden, 2000)

Mittlerweile ist auf dem ECB-Bus ein starker Wildwuchs zu verzeichnen, der dazu geführt hat, daß kaum noch 8 oder gar 16 zusammenhängende Port-Adressen frei sind, geschweige denn 64 davon für die HD64180-CPU, so diese die Z80 ersetzen sollte

Daher habe ich mir mal angesehen, ob da nicht etwas zu machen ist, und festgestellt, daß JA! Betroffen von diesen Änderungen sind die Besitzer von

- Conitec DISI SRAM-Floppy
(0B8h-0BBh und 0BCh-0BFh ---> 09Ch-09Fh)
ACHTUNG: Jetzt wird mit Format 53 und 54 auf die selbe Adresse zugegriffen! Daher sollte bei der Wahl des Formates für diese Floppy Vorsicht walten, d.h. nur das verwendete in RAM x.x installiert werden!
- c't 1 MB EPROM-Floppy
(0E4h-0E7h ---> 0ACh-0AFh)
- I/O-Karte wie die von c't, die ab Port 0D0h liegt
(0D0h-0DFh ---> 0B0h-0BFh)
- HD64180-Subprozessorkarte an Ports 0F8h-0FBh
(0F8h-0FBh ---> 084h-087h)
- ich mit meinem JK-Programmer
(0CCh-0CFh ---> 034h-037h, d.h. in FDX-Bereich!)

Für diese Karten habe ich neue Port-Adressen vorgesehen.

NEIN, Ihr müßt jetzt nicht alle Eure Karten umlöten oder umjumpern! Die Adressen der Floppy-Karten dürfen auch noch gar nicht verändert werden, da auch RAM 4.3 mit den alten Adressen arbeitet.

ABER mit der nächsten RAM-Version (RAM x.x mit x.x>4.3) müssen die Conitec-SRAM- und c't-1MB-EPROM-Floppies auf die neuen Adressen umgesetzt werden!! Wie das in Sachen Jumper genau aussieht wird dann im Handbuch zum neuen RAM x.x beschrieben. Die anderen Port-Adressen müssen nicht geändert werden. Ob, bleibt jedem überlassen.

Wenn ich richtig orientiert bin, betrifft diese Geschichte nur wenige (B. Preusing, U. Hönisch, U. Grass, Dr. H. Göbel, C. Romanazzi und mich). Wer Einwände hat, möge bitte nicht aufschreien, da ich das kaum hören werde, sondern mir sagen/schreiben warum.

WER BOOTET von der CONITEC-DISI ???? Bitte dringend melden zwecks Austausch des Boot-EPROM's. Tausch kostet lediglich das Porto!

H a r d w a r e: SDX-Sidisc / Ports

ÜBRIGENS gibt es die CONITEC-DISI nicht mehr als Leerplatine, sondern nur noch fertig gelötet, was diese im Vergleich zur c't-Karte, die immerhin 512 kB ab kann, unvergleichlich teuer. Zum anderen ist die c't-Karte betriebssicherer!

Für diejenigen unter Euch, die sich mein PAL für die Bussteuerung der FDX (Info 26) eingesetzt haben/einsetzen werden, d.h. ihre ECB-Option eliminier(t)en, werden übrigens auf Wunsch auch die Adressen 060h-07Fh für den ECB-Bus freigegeben, da ich kaum glaube, daß diese in der FDX vonnöten sind, zumal dort noch genug frei ist (z.B. 020h-02Fh).

Port-Adressen Zukunft (d.h. ab der nächsten RAM-Version)

(---> kennzeichnet die Änderungen.)

MTX	00 - 07	Diverses auf der Hauptplatine
	08 - 0B	Z80CTC
	0C - 0F	Z80DART
	10 - 14	SDX-Floppy-Controller
	18 - 1B	8255-PIO-Box (wer hat die ???)
	1C - 1E	frei: 3
	1F	Cassettenmotorsteuerung
FDX	20 - 2F	frei: 16
	30 - 33	80Zeichen-Karte
--->	34 - 37	JK-Programmer
	38 - 39	80Zeichen CRTC 6845
	3A - 3F	frei: 6
	40 - 47	FDX-Floppy-Controller
	48 - 4F	frei: 8
	50 - 53	Memotech SiDisc #1 (auch unter RAM 4.x)
	55 - 5F	Memotech SiDisc #2 - #4 (nicht RAM 4.x)
	60 - 6F	frei: 16
	70 - 73	Aachen-Floppy (wer hat die eigentlich ???)
	74 - 7F	frei: 12
ECB	80 - 83	Edicta Grafik-Karte
	84 - 87	HD 64180 Sub-Prozessorkarte
	88 - 8F	frei: 8
	90 - 97	I/O-Karte: nach Belieben (z.B. 2x 8255-PIO)
	98 - 9B	c't RAM-Floppy, Formate 58-5B
--->	9C - 9F	Conitec SRAM-Floppy, Formate 53-54
	A0 - A3	Edicta RAM-Floppy, Formate 5C-5F
	A4 - A7	c't 256k EPROM-Floppy, Format 57
	A8 - AB	c't SRAM-Floppy, Format 56
	AC - AF	c't 1MB EPROM-Floppy, Format ??
--->	B0 - B7	I/O-Karte: nach Belieben (z.B. Z80CTC/DART/PIO)
--->	B8 - BF	I/O-Karte: mögl. 2x Z80PIO
--->	C0 - FF	64 I/O-Adressen der HD64180-CPU

WICHTIG

Wer gegen diese Aktion Einwände erheben möchte, teile mir dieses bitte baldmöglichst mit! Danke!

H a r d w a r e: Volkseprommer**VOLKSEPROMMER ZUM SPOTTPREIS AN CENTRONICS-SCHNITTSTELLE !!!!**

(Michael Keßler, 5600)

In der Zeitschrift mc, Heft 8/84 wird ein Eprommer vorgestellt, der einerseits konkurrenzlos billig ist (max. DM 50,--) und andererseits professionellen Geräten kaum nachsteht. Der Eprommer ist hier bei mir seit 3 Monaten im Einsatz, ich bin sehr zufrieden damit. Ein großer Vorteil des Geräts ist sein Anschluß an der ganz gewöhnlichen Centronics-(Drucker)-Schnittstelle, so daß wirklich jeder ohne Hardware-Umbau damit arbeiten kann. Trotz des geringen Preises ist der Eprommer recht leistungsfähig. So sind die folgenden Eprom-Typen zu brennen:

2716	2732	2732-A	2758-A	2758-B	2764	27128	27256
2508	2516	2532	2564				

Da die Einstellung der Eprom-Pinbelegung SOFTWAREMÄSSIG vonstatten geht, sind auch weitere (ausgefallene) Eprom-Typen nachträglich installierbar (max. 28 Pins). Ätzworlagen (Folie) für die einseitige Platine liefert die Firma Neucom, Hangweg 4, 8893 Hilgertshausen-Tandern zum Preis von ca. 6 DM/Stück (Positiv oder Negativ). Die Software, die von Manfred Flume an unseren Rechner angepaßt und verbessert wurde, ist bei mir für DM 5,-- zu erhalten. Da der Abdruck von Schaltplänen aus Zeitschriften hier im Info aus urheberrechtlichen Gründen bedenklich ist, muß sich jeder Interessierte an die mc 8/84 halten, die allerdings oft in öffentlichen - bzw. Unibibliotheken einsehbar ist, ggf. kann man sich auch an mich wenden. Will man Eproms mit 12,5 Volt Programmierspannung brennen, so muß die Schaltung aus mc 8/84 wie folgt geändert werden: Man bringt parallel zur 4,7 Volt Zenerdiode eine 12 V Zenerdiode so an, daß mittels eines Umschalters entweder die 4,7 V oder die 12 V Zenerdiode zum Einsatz kommt. In Stellung 4,7 V hat man eine Programmierspannung von 21 V, in Stellung 12 V beträgt die Programmierspannung nur noch ca. 12,5 V. Der Anschluß des Eprommers an die Centronics-Schnittstelle geschieht wie folgt:

Clock (Ck)	an D0	(Pin 2)
Strobe (S)	an D1	(Pin 3)
Input (In)	an D2	(Pin 4)
Output (Out)	an PE	(Pin 12)
Masse (0V)	an 0V	(Pin 16)
+5 V (+5)	vom Netzteil	

Ich habe die entsprechenden Anschlüsse separat an eine 9-polige Joystickbuchse gelegt und betreibe den Eprommer über ein normales 6-poliges Joystickkabel. Vielleicht weiß ja wer, ob man direkt +5 Volt an einen noch freien Pin der Centronics legen darf, um so direkt das Druckerkabel benutzen zu können.

Anm.d.HzN.: Wenn ich dem Handbuch meines EPSON LQ850 glauben darf, ist kein Pin mehr frei. Der einzige nicht mit einer Funktion benannte Pin ist Nummer 35, an dem allerdings vom Drucker her +5 V über einen 3,3 kOhm-Widerstand liegen. Allerdings ist zu beachten, daß das Memotech-Druckerkabel die Pin's 35 und 36 gar nicht mit verbindet (vermutlich, weil 34-poliges Flachbandkabel leichter zu erhalten ist - schließlich ist es das, was für Floppy-Anschlüsse verwendet wird).

Ich schlage daher vor, einen eigenen Port für diesen Zweck zu verwenden. Sei es nun ein 8255 oder ein Z80PIO im Bit-Modus. Der Anschluß eines solchen IC links am MTX ist nicht weiter problematisch - auch ohne ECB-Bus! Dann bliebe sogar der Drucker unbehindert!

Hardware: WAIT-Generator

WAIT-Generator für 8 MHz

(Herbert zur Nedden, 2000)

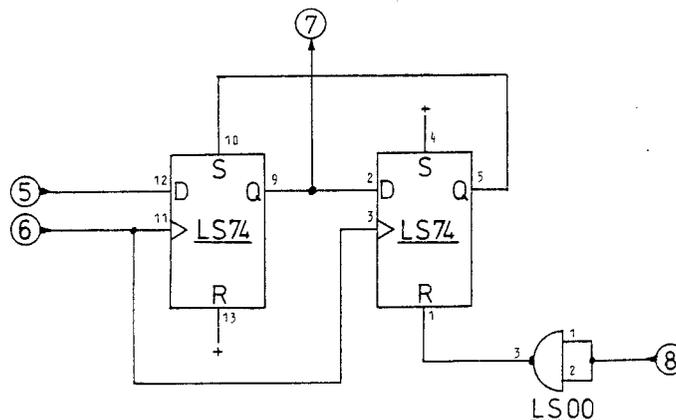
Im Info 21, Seite 31 hatte ich eine Schaltung veröffentlicht, die einen WAIT auf den M1-Zyklus der Z80CPU erzeugt, d.h. den Befehlshol-Zyklus um einen Takt verlängert, was langsameren RAM's ein gewisses Mitspracherecht einräumt. Seinerzeit habe ich es jedoch unterlassen, die Funktionsweise dieser Schaltung zu beschreiben - ich hatte sie einem Z80-Buch entnommen und darauf vertraut, daß sie auch tut. Holger Göbel hat es sich zum Ziel gemacht, das was ich so mittels Experiment mehr recht als schlecht schaffe sauber zu durchleuchten und mit allem möglichen Datebüchern bewaffnet so lange zu analysieren, bis er herausgefunden hat, warum ich es tatsächlich geschafft habe - oder auch nicht geschafft habe. Anschließend wird das ganze verfeinert, getestet und zu Diskette für's Info gebracht. Nun verwendet er bei seinem Interrupt-Erfolg eben diesen WAIT-Generator - daher habe ich ihm zugesagt, zu erläutern wie dieser funktioniert:

Erst einmal zur Funktion eines FlipFlop:

1. Sind S und R auf High arbeitet das FlipFlop als solches, d.h. mit steigender Taktflanke wird die Information (Low oder High) vom Eingang D auf den Ausgang Q übernommen.
2. Ist S auf High und R auf Low, d.h. R = Reset aktiv, wird das FlipFlop zurückgesetzt, d.h. Q geht auf Low.
3. Ist R auf High und S auf Low, d.h. S = Set aktiv, wird das FlipFlop gesetzt, d.h. Q geht auf High.
4. Sind R und S auf Low ist der Zustand des FlipFlop nicht definiert. Dieser Zustand ist nicht zulässig.

Der Ausgang Q-quer ist immer invers zu Q.

Hier erst noch einmal die Schaltung:



Damit die Beschreibung nicht zu einem Roman ausartet, hier meine Bezeichnungen:

- die in Kreisen genannten Anschlüsse 5-8 schreibe ich als (5) bis (8).
- die FlipFlops bezeichne ich als rFF (rechtes FlipFlop) und lFF (linkes).
- die Pins der FlipFlops: Q des rechten FF: Q(rFF), u.s.w.
(zum Glück brauche ich die Q-quer's nicht)

Die Bedeutung der Anschlüsse (5) - (8):

- (5) Steuereingang, auf Grund dessen Low-Pegel WAIT's erzeugt werden können.
- (6) Takteingang für die FF's, der der CPU-Takt sein muß.
- (7) WAIT-Ausgang: Low = WAIT
- (8) Sperr-Eingang: High = keine WAIT's werden erzeugt.

H a r d w a r e: WAIT-Generator

Nun zur Schaltung:

1. Gesperrter Zustand (8) = High

- > R(rFF) = Low, d.h. rFF zurückgesetzt, also Q(rFF) = Low
- > S(lFF) = Low, d.h. lFF gesetzt, also Q(lFF) = High, d.h. kein WAIT

Da beide FF's mittels Reset bzw. Set festgehalten werden, ändert sich am Zustand der Ausgänge Q(rFF) und Q(lFF) nichts --> Kein WAIT

2. Aktiver Zustand (8) = Low

- > R(rFF) = High, d.h. rFF arbeitet als FlipFlop, d.h. übernimmt mit steigender Taktflanke die Daten von D(rFF) nach Q(rFF).

2.1 Ruhezustand, d.h. kein WAIT (5) = High

- > Q(lFF) ist/geht auf High, entweder
weil S(lFF) = Low, d.h. lFF gesetzt, oder
weil s(lFF) = High und mit steigender Taktflanke die Daten von D(lFF) = High nach Q(lFF) übernommen werden
- > Q(lFF) = High, d.h. kein WAIT und D(rFF) = High
- > mit nächstem Takt: Q(rFF) = High,
d.h. S(lFF) = High (also insbesondere arbeitet nun lFF auch als FF).

Nun arbeiten beide FF's als FlipFlops, d.h. die Reset- und Set-Eingänge liegen auf High.

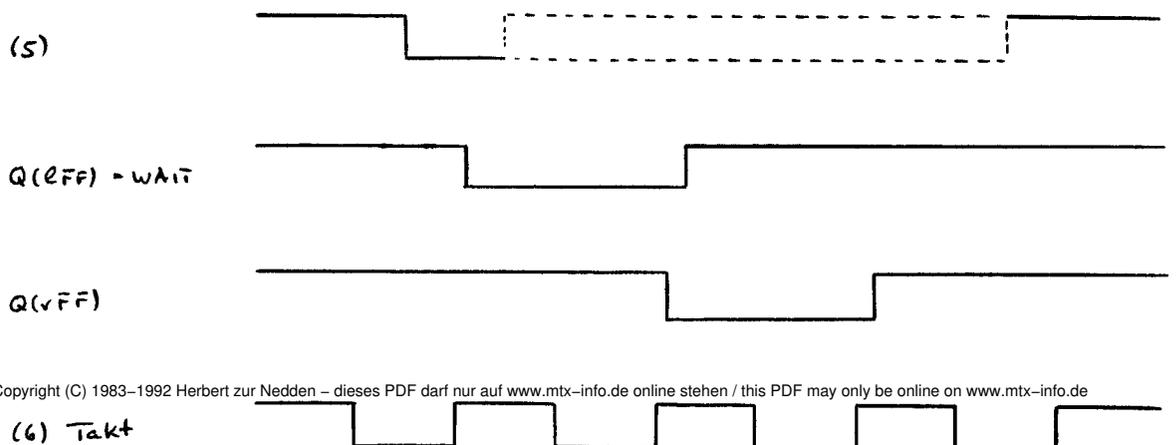
2.2 WAIT-Erzeugung (5) = Low

Ich gehe davon aus, daß vor dem absinken des Signals am Anschluß (5) beide FF's als solche arbeiten, sonst ist auf jeden Fall das linke FF gesperrt (S(lFF) = Low). Dann ist der Pegel von (5) sowieso uninteressant, und die Schaltung geht erst einmal in den Ruhezustand oder ist gesperrt.

- > mit der nächsten steigenden Taktflanke: Q(lFF) = Low,
d.h. WAIT = Low (also aktiv) und D(rFF) = Low
- > einen Takt später: Q(rFF) = Low,
d.h. S(lFF) = Low, d.h. lFF gesetzt, also Q(lFF) = High,
d.h. WAIT = High und D(rFF) = High
- > einen Takt später: Q(rFF) = High,
d.h. die Schaltung ist wieder im gleichen Zustand wie zu Beginn dieses Zyklus - Es wird insbesondere immer ein WAIT-Impuls erzeugt.

Ist nun (5) auf High, bleibt die Schaltung im Ruhezustand; ist (5) = Low, so geht dieser Durchlauf von vorne los, d.h. es folgt der nächste WAIT.

Timing für einen Durchlauf:



Hardware: c't-IO-Karte und Interrupts**c't-IO-Karte und Interrupts**

(Dr. Holger Göbel, 8630; 09561/15131)

Im INFO 27/56 hatte ich kurz vorgeschlagen, die c't-IO-Karte zu testen und meine Erfahrungen mitzuteilen.

Das hat deswegen etwas länger gedauert, weil die Handhabung der Interrupts Schwierigkeiten bereitet hat. Doch der Reihe nach:

1. Warum eine IO-Karte und warum die von c't?

Die erste Frage ist natürlich auch eine Angelegenheit der Computeranwendungen jedes einzelnen. Ohne Zweifel stellt eine IO-Karte die Schnittstelle des Computers zur Außenwelt her. Ich selber verwende die PIO-Bausteine (PIO = Parallel Input/Output) zur Ansteuerung elektronischer Schaltungen, HzN hat sich mit Hilfe derselben eine einfache zweite Centronics-Schnittstelle für seinen Matrixdrucker aufgebaut. Da sind der Phantasie keine Grenzen gesetzt (hallo, Modelleisenbahn-Fans!).

Ein zusätzlicher SIO (Seriell Input/Output) ist z.B. erforderlich, wenn HzN in seiner nächsten RAM4xx-Version, wie versprochen, eine Maus unterstützen will und am SIO der RS232-Platine schon etwas Anderes hängt (z.B. ein Modem). Auch eine verbesserte Tastatur könnte man anschließen oder einen Drucker/Plotter mit serieller Schnittstelle ...

Der CTC (Counter-Timer-Chip) in unserem Grundgerät ist ja schon voll ausgelastet, ein zusätzlicher also gar nicht so schlecht (z.B. für periodische Interrupt-Generierung). Die Digitalisierer-Software von Otmar Rucker ist sicherlich auf diese Weise einfacher zu realisieren.

Die IO-Karte der Zeitschrift c't (4/85, beachte Errata 7/85) besitzt 1 CTC, 1 SIO und 2 PIOs, alles Z80-Bausteine. Die Treiber für eine V24-Schnittstelle sind realisiert, ebenso eine (allerdings unvollständige) Look-Ahead-Logik, dazu später mehr. Ein kleines Lochrasterfeld kann für eigene Erweiterungen genutzt werden. Sie ist voll interruptfähig, nach einigen kleineren Basteleien auch unter 8 MHz (trotz der Z80B-Versionen, die eigentlich nur bis 6 MHz gehen dürften). Allerdings müssen für den Interrupt-Betrieb auch kleinere Änderungen im Grundgerät/FDX vorgenommen werden. Dazu später mehr.

Die Platine ist unter der Bestellnummer 850484dB bei eMedia GmbH, Bissendorfer Str. 8, Pf 610106, 3000 Hannover 61, für DM 79,-- zu erhalten. Als Z80-Bausteine sollten die B-Versionen genommen werden, falls die Karte auch unter 8 MHz laufen soll. Der Aufbau ist relativ unproblematisch, insgesamt belaufen sich die Kosten auf etwa DM 200,--.

Falls Ihr die genannten c't-Zeitschriften nicht zur Verfügung habt, kann ich Euch die entsprechenden Artikel zukommen lassen.

2. Interrupts - Eleganz mit Hindernissen**a) Der INTACK-Zyklus**

HzN hat im letzten Info (28/10 und 28/14) ausführliche Berichte über Interrupts zu Papier gebracht, ich darf mich dankenswerterweise auf diese berufen. Folgende wichtige Eigenart des Interruptmodus 2 soll noch einmal kurz beleuchtet werden:

Löst z.B. der PIO einen Interrupt aus, weil an seinem einen PIN das Ereignis eingetreten ist, auf welches er aufpassen sollte, dann muß die Z80-CPU ihm diesen Interrupt bestätigen (man nennt das "INTACK" = Interrupt-Acknowledge oder "ich habe verstanden"). Dies tut sie, indem sie ihre beiden Signale /M1 und /IOREQ auf LOW legt, was sonst nie vorkommt. Jetzt ist die große Stunde (besser: Nanosekunde) des PIO gekommen - er legt nämlich ein Byte auf den Datenbus, und zwar das, das er sich von der Initialisierung her tapfer gemerkt hat.

H a r d w a r e: c't-I0-Karte und Interrupts

Dieses Byte ist die untere Hälfte einer Speicheradresse, die obere Hälfte steht im I-(Interrupt)-Register der CPU, so sie der Programmierer da hineingeschrieben hat. Jetzt haben wir also eine ganze 2-Byte-Adresse, in die die CPU hineinschaut. Was sieht sie dort? - Eine zweite Adresse, und zu dieser springt sie jetzt hin, denn das ist die Interruptroutine, die extra geschrieben wurde, um auf das Ereignis an besagtem PIN zu reagieren.

Das eben Ausgeführte war sicherlich nicht unbedingt etwas Neues, man muß es sich nur immer wieder vor Augen führen, denn mit dieser Adresse in einer Adresse ist es doch etwas kompliziert. Wo aber liegt das Problem?

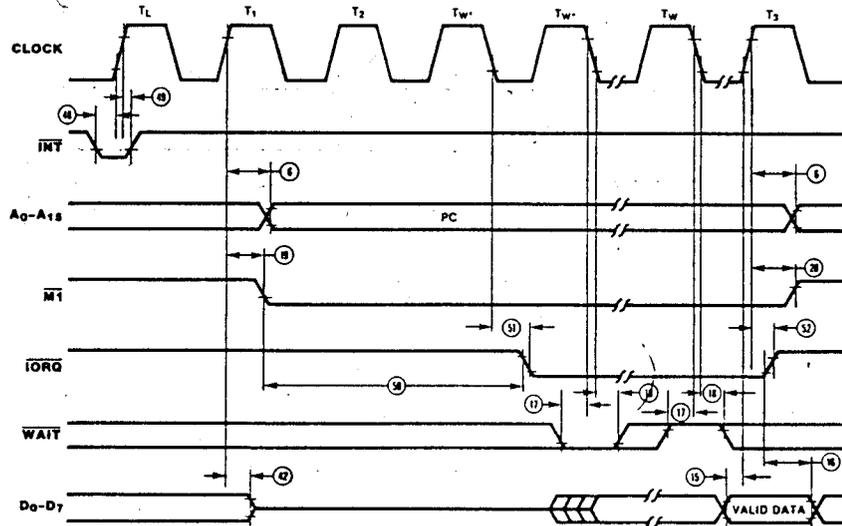
Wie immer - im Timing. Wenn die CPU mit irgendwelchen IO-Bausteinen kommuniziert, muß sie zeitlich etwas Rücksicht nehmen, erstens sind diese von Haus aus nicht so schnell, und zweitens sind oft noch ein paar Treiber und Gatter zwischengeschaltet. Deshalb generiert die CPU von sich aus bei IO-Zugriffen immer 1 oder 2 WAITs (sie wartet also 1 oder 2 Takte). Bei einem INTACK spendiert sie 2 WAITs, was im Allgemeinen auch ausreicht, nicht aber unter 8 MHz. Hier muß künstlich ein zusätzliches WAIT eingefügt werden, damit den IO-Bausteinen beim INTACK nicht schwindlig wird.

Jetzt müssen wir, falls Ihr noch mitdenken wollt, doch etwas ins Eingemachte gehen: WAITs werden der CPU dann aufgezwungen, wenn ihr /WAIT-Eingang auf LOW gezogen wird. Sie erkennt diesen Zustand aber nur

a) während der abfallenden Flanke des Taktzyklus T2,

b) während der abfallenden Flanke eines WAIT-Taktes.

Die CPU läßt sich also nur nach T2 (da hat sie ihren Programmzähler auf den Adreßbus gelegt und wartet auf die Daten) unterbrechen, dann aber (beliebig) lange, denn am Ende eines jeden WAITs ist sie empfänglich für ein neues WAIT.



NOTE: 1) T_L = Last state of previous instruction.

2) Two Wait cycles automatically inserted by CPU(*).

Figure 8. Interrupt Request/Acknowledge Cycle

Wenn Ihr Euch jetzt einmal das Timing-Diagramm eines INTACK oben anschaut dann seht Ihr, daß das Charakteristikum des INTACK sehr spät aktiv ist: erst während des 2. "freiwilligen" WAITs T* der CPU folgt /IORQ dem /M1 und geht auf LOW. Liest man jetzt im Datenblatt nach, so garantiert ZILOG als Z80-Hersteller selbst bei seiner schnellsten H-Version minimal lediglich 15 ns, in denen der CPU jetzt ein weiterer WAIT aufgezwungen werden kann.

Wenn man weiß, daß die Treiberbausteine jeder etwa 10 ns die Signale verzögert, wird klar, daß der WAIT-Generator, den wir aufbauen wollen, direkt neben der CPU sitzen und außerdem sehr schnell sein muß (ein 74LS123 z.B. scheidet schon mal aus, er braucht ca. 30 ns!).

H a r d w a r e: c't-IO-Karte und Interrupts

Wie gut, daß Herbert immer so hervorragende Vorarbeit leistet:

Also, die Schaltung aus INFO 21/31 genommen und leicht modifiziert:

1. Sicherer ist es, einen 74ALS zu nehmen, weil der erheblich schneller ist.
2. An PIN 12 kommt /IORQ (PIN 20 am Z80).
3. An die beiden Eingänge des LS00 (da, wo die eingekreiste 8 steht), kommt /M1 (PIN 27 am Z80).
4. Besser ist es (das hätte sowieso ins Schaltbild gehört), das /WAIT-Signal (an der eingekreisten 7) über eine Diode (z.B. 1N4148) an den WAIT-Eingang der CPU zu führen (PIN 24). Dabei muß die Kathode (also der schwarze Balken) zum 74ALS74 zeigen, an die Anode gehört noch ein Pull-up-Widerstand (z.B. 4,7 kOhm) an die + 5 V.

Übrigens kann ich mir sehr gut vorstellen, daß mancher Rechner, der unter 8 MHz zwar rechnete, aber die Tastatur nicht beachtete, und deswegen ein M1-WAIT aufgezwungen bekam, jetzt auch voll mit 8 MHz läuft: Für die Tastaturabfrage ist ja der CTC im Grundgerät verantwortlich, auch eine B-Version, dem ein WAIT im INTACK-Zyklus zum Verschnaufen gut tun würde.

b) Die Daisy-Chain und Look-Ahead

Auch zur Daisy-Chain hat Herbert schon Grundsätzliches gesagt (28/11 ff.). Ganz kurz (Insider, bitte überspringen):

Alle Bausteine, die einen Interrupt auslösen können, sind in einer Kette angeordnet, an erster Stelle steht der wichtigste, an letzter einer, na ja, sagen wir, wie Du und ich. Wenn der wichtigste gerade die CPU unterbrochen hat, teilt er das mit, indem sein IEO (Interrupt-Enable-Output) auf Low geht. Da dieser mit dem IEI (richtig geraten: Interrupt-Enable-Input) des niedriger priorisierten Bausteins verbunden ist, muß dieser und alle folgenden schön ruhig bleiben. Die folgenden Bausteine geben deswegen klein bei, weil der IEO des zweiten auf LOW geht, sobald sein IEI das tut (typisches Frustrationsverhalten!). Allerdings braucht er ein bißchen Zeit, bis die Information von seinem IEI an seinen IEO gelangt. Das ist der Grund, warum bei maximaler Taktfrequenz eine Daisy-Chain nur bis zu höchstens 4 Gliedern funktioniert. Bei 8 MHz liegen wir aber schon oberhalb der garantierten maximalen 6 MHz der B-Versionen.

Ein Vorausschauer muß also her, eine Look-Ahead-Logik. Diese spannt sofort, wenn irgendein IEO auf LOW geht, und gibt diese Information an alle niedriger priorisierten Bausteine blitzschnell weiter.

Die c't-Karte besitzt nur eine rudimentäre Look-Ahead-Logik: Ein IEO irgendeines der vier IO-Bausteine wird zwar schnell an den IEO-PIN der ganzen Karte gegeben, d.h., eine weiter hinten liegende Karte bekommt die Information rechtzeitig genug, aber die vier IO-Bausteine der Karte selbst kommunizieren auf ganz normalem Wege miteinander.

Also sollte man auf dem Lochrasterfeld (für irgendwas ist es doch gut) eine eigene Look-Ahead-Logik aufbauen. Man benötigt einen 74LS08 und einen 74LS21 (besser noch ALS). Mag jemand beim Anblick des Schaltbildes (siehe unten) denken, daß ein 74LS08 auch genügt - aber dann muß die IEO-Information im schlimmsten Fall durch drei Gatter laufen, was zu Komplikationen führen kann (Achtung, jetzt wird's wieder etwas speziell):

H a r d w a r e: c't-IO-Karte und Interrupts

'RETI' ist ein Zwei-Byte-Befehl, er besteht aus den beiden Bytes 'ED' und '4D' (HEX). Der König erkennt nun, und zwar nur dann, wenn er wie in unserem Beispiel einen Interrupt angefordert, aber noch nicht bestätigt bekommen hat, das erste 'ED' und setzt dann seinen IEO gnädigerweise für ca. 4 Taktzyklen auf high, damit der Bettelmann also 'sein RETI' erkennen kann.

Im eben beschriebenen Beispiel ist es also notwendig, daß eine logische '1' möglichst schnell die Daisy Chain durchläuft. Das leistet die Look-Ahead-Logik allerdings nicht, wenn man nicht einen unvertretbar hohen Schaltungsaufwand betreiben will.

Ein nicht oder falsch erkanntes 'RETI' bewirkt in jedem Fall einen Fehler. Man kann das vermeiden, indem man softwaremäßig den RETI-Befehl nur bei erlaubten Interrupts (also nach 'EI') durchführt, also: EI - NOP - RETI.

Nachteilig kann nun sein, daß der Bettelmann sein 'RETI' sehr viel später erhält und deswegen einen erneuten Interrupt nicht erkennen kann, weil ja der König während des NOPs unterbrechen durfte.

Aber keine Angst: auch unter 8 MHz habe ich ohne das NOP mit meiner Look-Ahead-Schaltung keine Probleme gehabt, und das, obwohl ich im Grundgerät auch meinen SIO mit einer Look-Ahead-Logik versehen habe (was wohl nicht nötig sein dürfte).

Wenn Ihr die Look-Ahead-Schaltung aufbauen wollt, dann müßt ihr ja die IEO-IEI-Verbindungen der einzelnen IO-Bausteine untereinander trennen. Tut das bitte, bevor Ihr die Sockel eingelötet habt, damit Ihr die Leiterbahnen besser verfolgen könnt. Beachtet dabei, daß die IEO ihren Pull-up-Widerstand brauchen, weil sie Open-Kollektor-Ausgänge sind, und daß die Verbindungen zu dem 74LS09 der Karte gewährleistet sind. Ich habe auf der Bestückungsseite 2 und auf der Lötseite 3 Leiterbahnen durchtrennt und auf der Lötseite 2 neu gelegt.

3. Weitere Verbesserungen der Karte

Die IO-Bausteine sollten einen RESET erfahren auch dann, wenn man die beiden RESET-Tasten drückt, nicht nur bei einem Neuanschalten des Rechners. Deswegen muß die Leitung von PIN 1 des 74LS08 zum ECB-Anschluß 26c (/PWRL) getrennt und zum Anschluß 31c (/RESET) geführt werden.

4. Veränderungen am ECB-Bus und am MTX/FDX

Zunächst einmal müßt Ihr prüfen, wie Herbert in 28/11 ausgeführt hat, ob auf der ECB-Busplatine wirklich 16c (IEO) mit 11c (IEI) des folgenden Steckplatzes verbunden ist. Sind die Leitungen 11c und 16c einfach durchgeschleift (wie bei mir), so müßt Ihr sie trennen und entsprechend neu verdrahten. Wenn ein Steckplatz dann unbenutzt bleibt, müßt Ihr dafür Sorge tragen, daß an diesem Stecker 11c und 16c überbrückt sind.

S. INFO 28/10 und 25/32: Auf der kleinen Interface-Platine in der FDX müssen die Leitungen INT, WAIT (und BUSRQ) über Dioden angeschlossen werden.

S. INFO 25/32: Wenn ein IO-Baustein einen Interrupt auslöst, dann darf er ja, wie oben beschrieben, ein Byte auf den Datenbus legen. Dieses Byte muß natürlich zur CPU gelangen. Das tut es nur sauber, wenn die FDX nicht dazwischenfunkelt. Die FDX kann schachmatt gelegt werden, wenn das Signal DIR245 entsprechend geschaltet wird. Vielleicht erinnert Ihr Euch noch an die ECB-Option aus INFO 13/3, da hat dieses Signal schon eine Rolle gespielt. Kurz und gut: Der ECB-Bus wird nur dann interruptfähig, wenn dieses Signal ausreichend berücksichtigt wird. Wolfgang Zierenberg hat in INFO 25/32 eine Schaltung dazu entworfen. Diese hat - Welch ein Glück - Gerhard Witzel zur Grundlage einer kleinen Platine genommen, so daß Ihr nur folgendes zu tun braucht: Platine bestellen, PAL aus der RS232-Platine raus, da die Platine rein, PAL in die Platine, und dann die 245DIR-Leitung neu verlegen, und zwar genau so, wie in INFO 13/4 im letzten Absatz geschildert und bebildert.

H a r d w a r e: c't-IO-Karte und Interrupts

Zweite Möglichkeit: Ihr baut Euren ECB-Bus gleich zukunftssicher auf und unterzieht Euch einer größeren Bastelei auf der kleinen Interface-Karte in der FDX, siehe dazu INFO 27/29 von HzN.

5. Resumee

Für den Interruptbetrieb müssen kleine Veränderungen an der FDX-Interface-, der RS232- und evtl. der ECB-Bus-Platine vorgenommen werden. Sollen Interrupts auch unter 8 MHz laufen, müssen eine kleine WAIT-Schaltung für den INTACK-Zyklus und eine Look-Ahead-Logik für die Daisy-Chain das regelrechte Timing sicherstellen (das 8-MHz-Kochbuch habe ich diesbezüglich aktualisiert).

Wer sich die IO-Karte aufbauen will, dem stehe ich gern zur Seite; für uns beide (die Karte und mich) hat sich der Aufbau gelohnt: sie tut ihre Dienste, und ich kenne mich jetzt ein wenig mit Interrupts aus.

Für die Programmierung der IO-Bausteine bedarf es einiger Kenntnisse, da kann ich das Buch "Z80-Anwendungen" von James W. Coffron aus dem SYBEX-Verlag empfehlen; interessante Hinweise erhält man auch aus dem Artikel "Computus interruptus" der c't 9/84 (evtl. über mich zu bekommen).

Das abgedruckte Listing in der c't für den Test dieser Karte habe ich stark modifizieren müssen, es läuft jetzt unter RAM43 tadellos (RAM43 stellt schon Interrupt-Vektor-Adressen für die IO-Bausteine zur Verfügung). Das Programm könnt Ihr auch von mir lauffähig haben (dann mit weiteren Informationen).

Die Modifizierungen des Testprogrammes waren v.a. wegen folgenden Phänomens erforderlich:

Wenn eine Interruptroutine dazu dient, irgendetwas auf den Schirm auszugeben, dann wird in den Bildschirmroutinen, die RAM4xx bereitstellt, ein 'EI' durchlaufen. Wenn dann ein zweiter IO-Baustein interruptet, kommt alles durcheinander, und der Rechner stürzt ab, also Vorsicht!

Wie man Interruptroutinen schreibt, daß sie z.B. in der Common-Bank liegen sollten, das wäre ein eigener Artikel, Ihr findet schon etwas (allerdings etwas fehlerbehaftet) im INFO 25/34.

Wer die IO-Karte nicht aufbauen will, dem war jetzt zuviel Information geboten. Trotzdem hoffe ich, daß er etwas aus dem Artikel herausziehen konnte.

Dazu sei es mir erlaubt, eine Anmerkung zu machen (am Ende eines so langen Artikels wird man immer so moralisch, morgens um halb Drei): Der Club ist für MTXler die einzige Lebensquelle, und ich erachte meine 'Maschine' für so gut (v.a. in der jetzigen Ausbaustufe), daß ich sie so schnell gegen keine andere eintauschen will (und kann). Ich habe nun allerdings den Eindruck, als hätten die letzten INFO-Artikel hauptsächlich die bestritten, die die Entwicklung ihres MTX weit vorangetrieben haben, hard- und softwaremäßig. Diese dürften allerdings in der Minderheit sein. Warum traut sich aber nicht einer von der anderen "größeren Hälfte", auch mal z.B. ein BASIC-Listing abzudrucken, denn BASIC ist doch keine Schande, und viele könnten davon profitieren. Oder macht Ihr nichts mehr mit dem MTX? Kann ich mir einfach nicht vorstellen. Auch die Kritik eines Spieles (und wer lädt nicht auch mal ein solches ein) kann interessant sein... Wir alle machen doch das INFO.

Hardware: 8-MHz-Platine**8-MHz-Platine**

(Dr. Holger Göbel, 8630; 09561/15131)

Beim Clubtreffen hatte ich den Eindruck, daß breiteres Interesse an einer Beschleunigung des Rechners auf 8 MHz besteht, viele jedoch den nicht unerheblichen Schaltungsaufwand scheuen. Aus diesem Grund haben Gerhard Witzel und ich eine 8-MHz-Platine geätzt, die Ihr bestückt und getestet für insgesamt DM 35,-- haben könnt (siehe auch INFO 27/55).

Was ist dann noch zu tun?

Das IC 9D muß ausgelötet und an seine Stelle ein 14-poliger gedrehter Sockel eingelötet werden. In diesen Sockel wird die Platine einfach eingesteckt. Die Platine hat dann einen Stecker mit Flachbandkabel, dessen 14 Adern an 10 Punkten auf der Hauptplatine angelötet werden müssen. Zwei Adern gehen zu einem Umschalter (4/8 MHz), eine Ader muß zur EPROM-Floppy, falls vorhanden.

Das ist der ganze Aufwand für die Umschaltlogik. Einige ICs, Widerstände und Kondensatoren müssen dann noch ausgetauscht werden (schnellere Typen). Alles das ist aber mehrfach beschrieben und in tabellarischer Form in meinem (in diesem INFO aktualisierten) 8-MHz-Kochbuch aufgelistet.

Wer sich also aufgrund genügender eigener Löterfahrung selbst an den Umbau wagt, der kann die Platine bei mir beziehen. Wer sich das nicht traut, für den nimmt HzN den Umbau vor (übrigens verwendet er unsere 8-MHz-Platine).

Die Platine hat den angenehmen Vorteil, daß sie aufgrund ihrer Abmessungen nicht mit der Uhr in PORT 7 kollidiert, und daß der Deckel des Grundgerätes auch wieder zu schließen ist. Anbei habe ich einen Schaltplan der 8-MHz-Platine gelegt, der sich eng an die Vorarbeiten von HzN (21/29) und Horst Kupka (25/19) hält.

Hinzugekommen sind: ein Eingang /ECB (der nicht unbedingt beschaltet werden muß), damit bei einem Zugriff auf die EPROM-Floppy 1 WAIT erzeugt wird. Auch der Eingang /BUSRQ muß nur dann mit der CPU verbunden werden, wenn der DMA-Transfer nicht klappt (wenn also beim Lesen/Schreiben der Disketten Fehler auftreten). Bei mir ist das nur dann der Fall, wenn der ECB-Bus angeschlossen ist.

Der Ausgang SPEED gibt an, wann der Rechner mit welcher Frequenz getaktet wird (z.B. zur Ansteuerung eines kleinen Displays). Wozu ich über einen Pull-up-Widerstand 5 V herausgeführt habe, kann man in INFO 27/55 nachlesen.

Die eingekreisten Ziffern geben die Nummer der Adern im Flachbandkabel an, wenn man die Platine eingesteckt hat und die Adern von links nach rechts zählt. Wo die Adern angeschlossen werden müssen, steht genau daneben.

Die Buchstaben 'o' und 'u' in den NOR-Gattern unterscheiden die beiden 74LS02-Chips. 'o' steht für 'oben' (also Lötseite), 'u' für - na?

Viel Erfolg!

Umbau auf 8 MHz

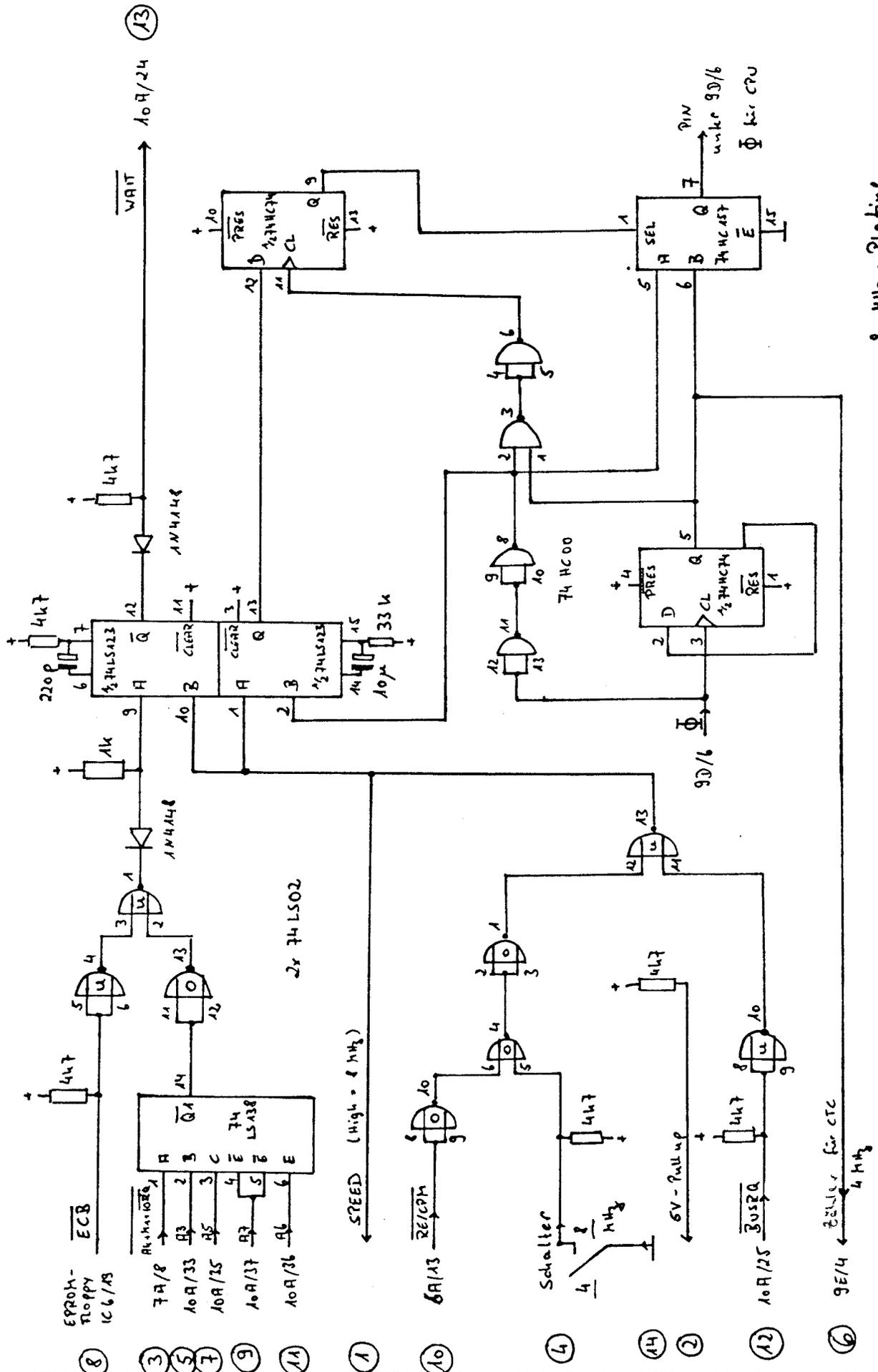
(Herbert zur Nedden, 2000)

Ich verwende zum Umbauen eines MTX/FDX auf 8 MHz grundsätzlich diese 8 MHz-Platine von Holger und Gerhard. In Anbetracht der Arbeitserleichterung bleibt mein Angebot unverändert bestehen - obwohl die Platine DM 35.- kostet.

Den ersten Rechner habe ich übrigens mit einer solchen (etwas älteren Version dieser, aber im großen und ganzen gleichen) Platine auf 8 MHz getrimmt. Es war richtig erfreulich, was so an Arbeit plötzlich wegfiel. Noch angenehmer als die Arbeitersparnis finde ich jedoch, daß diese Platinen immer getestet sind, ich also weiß, daß wenn's nicht klappt ist der Fehler dort nicht zu suchen. Wer kennt das nicht, daß man beim Verdrahten von 5 IC's auf einer kleinen Lochrasterplatine den einen oder anderen Pin vertauscht.

Bleibt nur noch 80Zeichen-Karte anpassen, 8 RAM's der Hauptplatine CPU, CTC und DART austauschen, am Timing basteln ... FERTIG! Dabei ist es i.a. das Timing, was die meiste Arbeit macht, insbesondere beim Versuch, ohne WAIT auszukommen.

Hardware: 8-MHz-Platine



8-MHz-Platine
HG 11.10.88

Hardware: 8 MHz - Kochbuch

Bauteil	Maßnahme	Begründung	INFO
A) Allgemeine Maßnahmen			
a) Netzteil			
1. FDX-Netz R24	Widerstand verändern (Poti) oder stärkeres Netzteil	Netzteil hochregeln, weil 9 MHz mehr Strom brauchen evtl. Gleichspannungen auf die MTX-Platine führen, um der Hitzeentwicklung im MTX Herr zu werden (dicke Verbindungskabel nehmen!)	18/37,21/34 25/29-30

B) Hauptplatine

* 1. XTAL1	8 MHz-Quarz	klar	21/30
* 2. R32, R33	3,3 kOhm	damit der Quarz den Schwingkreis optimal anregt	21/30
* 3. IC 10A	Z80 <u>H</u> CPU	ist für diese Taktfrequenz ausgelegt	21/30
* 4. IC 8A	Z80 <u>B</u> CTC	eigentlich bis 6 MHz, funktioniert aber (?)	21/30
* 5. IC 2C-5C,2D-5D	RAMs höchstens 150 ns	schneller, beim MTX 500 genügen 3732 (gibt's die?), sonst wie beim MTX 512: 4164	21/30
* 6. IC 9D	raus, gedrehter Sockel rein muß 74 <u>HC</u> 04 sein	zur Not evtl. statische RAMs verwenden da wird die 8 MHz-Platine reingesteckt; falls man nicht gleich die 8MHz-Platine nimmt	25/19 27/55 21/30
* 7. IC 9E	P4 abkneifen, hochbiegen	damit die Zeitkonstanten des CTC gleich bleiben können	18/19,21/30
8. R35, R36	besser drinlassen	laut ZILOG macht sonst die CPU evtl. Schwierigkeiten	21/30
* 9. C5, C6	auslöten	damit das CAS-Signal für die RAMs eher kommt	21/30,24/9
10. R14, R15	evtl. durch andere ersetzen	gerade diese RAS-/CAS-Signale sind sehr zeitempfindlich	24/9
11. 8 MHz-Platine	evtl. überbrücken	CAS noch eher (nur wenn 9. nicht genügt)	25/21
	reinstecken, Verb. löten	ist von mir getestet, Anschlußplan anbei	25/19,27/55

C) 768/512k-Karte

1. Platinenstecker	evtl. fest verlöten	keine Wackelkontakte, Kapazitäten vermindert	22/19
2. C1, C2	evtl. durch andere ersetzen	RAS/MPX/CAS-Signalfolge für die RAMs sind manchmal auch hier kritisch (RAM von BP läuft dann nicht). Bei mir: C1 raus, C=47 pF	24/10,25/20
3. R1, R2	evtl. andere Werte	s. "B) Hauptplatine: 10."	24/10,25/20

D) RS232-Platine

1. Platinenstecker	evtl. fest verlöten	s. C) 1.	22/19
* 2. IC 1	Z80 <u>B</u> DART (oder SIO)	für die schnellere Frequenz nötig	21/30
3. IC 5	evtl. 74HC244	"sauberere" Steuersignale (z.B. RFSH, M1, PHI)	25/20
4. IC 6	evtl. 74HC241 oder 74ALS241	dto.	25/26

E) 80-Zeichen-Karte

* 1. IC 1C	P5 abkneifen, unter der Platine P5 mit P8 verbinden	der Video-Kontroller bekommt dadurch die Daten schneller	21/30
------------	---	--	-------

F) Floppy-Kontroller-Karte

1.	läuft gut (dank 8MHz-Platine und Horst Kupkas Ideen)
----	--

Hardware: 8 MHz - Kochbuch

Bauteil	Maßnahme	Begründung	INFO
G) Interface-Karte (kleine Karte in der FDX mit dem CP/M-EPROM)			
1. Treiber	evtl. <u>HC</u> oder <u>ALS</u>	schnellere Signale (hat bei mir keinen Vorteil gebracht)	24/13
2. CP/M-EPROM	kann bleiben	gebootet wird mit 4 MHz	
H) ECB-Bus			
* 1. Anordnung	MTX zwischen FDX und ECB	Abschluß nach beiden Seiten	
2. Terminierung	<u>evtl.</u> bestimmte Signale oder ganze Terminierung raus	bestimmte Steuersignale werden zeitlich verschoben nur nötig, wenn Disk-MTX-Transfer wirklich mit 8 Mhz laufen soll, evtl. hier auf 4 MHz umschalten (Anschluß auf der 8-MHz-Platine vorgesehen), da nur geringer Zeitverlust	25/31,26/27 27/55
I) ECB-Karten			
a) c't-CMOS-RAM-Floppy			
1. IC1, IC11 (74LS174, 74HC138)	je einen Blockkondensator (100 nF) von +5V auf Masse, d.h. P8->P16, direkt einlöten	diese ICs machen beim <u>Schreiben</u> mit 8 MHz evtl. Probleme, der Blockkondensator stabilisiert die Schaltströme	24/11
b) Conitec-DISI-CMOS-RAM-Floppy			
1.		läuft gut	
c) c't-EPROM-Floppy			
* 1. IC6 (74LS688)	von P19 Kabel zur 8MHz-P1.	an WAIT-Logik anschließen, da die EPROM-Floppy mit reinen 8 MHz nicht mitkommt (warum?). Dieses Signal geht auf low, wenn ein IORQ in Verbindung mit M1 an die EPROM-Karte geht.	24/11
d) EDICTA-Graphik-Karte			
1.		läuft gut	
e) c't-IO-Karte			
* 1. IC 5-8	Z80 B CTC/SIO/PIO	eigentlich bis 6 MHz, funktioniert aber	29/?
<u>Für den Interruptbetrieb:</u>			
* 2. Look Ahead	auf dem Lochraster aufbauen	Daisy Chain schneller machen	29/?
* 3. WAIT	direkt neben der CPU aufbauen	Damit der INTACK-Zyklus klappt	29/? , 21/31
* 4. 74LS244 auf Interfacekarte	Leiterbahnen von P 12/14/16 zum Bus-Anschluß 50/51/53	Damit nicht TTL-Ausgänge gegeneinander geschaltet werden	25/32, 28/10
	der FDX rechts durch Dioden ersetzen		
* 5. IC8 der RS232- Karte (PAL)	IM2-ECB-Platine einsetzen, 245DIR-Signal neu verdrahten	Das 245DIR-Signal muß bei einem INTACK umgeschaltet werden	25/32, 29/?
<u>oder:</u>	PAL von HzN auf die Interfacekarte		13/3 27/29

S o f t w a r e: BRADFORD-Dokumentation

(Herbert zur Nedden, 2000)

ALLGEMEINES

BRADFORD ist ein Druckprogramm, welches auf CLUB.027 steht.

BRADFORD Version 1.20 ist Public Domain, das Handbuch jedoch ist durch Copyright geschützt (schließlich will der Autor von BRADFORD, Aaron M. Contorer, zumindest hiermit etwas Geld verdienen). Wer das Handbuch haben möchte, kann es in den USA bestellen. Wo und Preis steht in einem der DOC's, die bei BRADFORD dabei sind. Damit Ihr jedoch nicht allzusehr im Dunkeln tappt, hier eine Einführung in BRADFORD.

BRADFORD ist ein NLQ-Druckprogramm für Matrixdrucker. Der Autor sagt:

'Prints near letter quality, which means that the printout is not quite able to make you think it came from a typewriter, but good enough that you don't care.'

Eingedeutscht:

'Druckt Fast-Brief-Qualität, was bedeutet, daß der Ausdruck nicht ganz so gut ist, daß Du glaubst er wäre von einer Schreibmaschine, aber gut genug, daß es Dir nichts ausmacht.'

Zusätzlich bietet BRADFORD mehrere Schriften, Proportionaldruck, Blocksatz, ...

ANMERKUNGEN ZU NEWWORD/WORDSTAR

WordStar ist ein bekannter Texteditor, zu dem NewWord kompatibel ist. Der Vorteil von NewWord ist nicht nur, daß Du als Memotechler(in) dafür eine Lizenz hast, sondern, daß NewWord in einigen Kleinigkeiten besser ausgereift ist als WordStar. Schließlich ist NewWord dadurch entstanden, daß Mitarbeiter vom WordStar-Hersteller MicroPro ihre eigene Firma NewStar aufgemacht haben, um NewWord auf den Markt zu bringen, und daher besser sein mußten - und sind!

BRADFORD unterscheidet zwei Arten von Texten, nämlich WordStar (also NewWord) Dokumente und sonstige ASCII-Texte. In NewWord-Dokumenten ist es ja bekanntlich so, daß man einfach einen Absatz munter eingeben kann, und nur am Ende des Absatzes RET eingeben muß, damit NewWord weiß, wo dieser zu Ende ist. Den Zeilenumbruch und Randausgleich innerhalb des Absatzes macht NewWord für uns.

Am Rechten Rand des Bildschirms erkennt man unter NewWord das Ende des Absatzes (d.h. ein erzwungenes Zeilenende) am '<', während innerhalb eines Absatzes dort nichts steht. (Steht dorten ein '+', so geht die Zeile über den rechten Bildschirmrand hinaus. Verschiebst Du die Anzeige nach rechts, erscheint irgendwann dort das obengenannte.) Logischerweise ist diese Information, ob das Zeilenende ein verschiebbares Zeilenende innerhalb eines Absatzes (wird ggf. beim Einfügen/Löschen/Reformatieren verschoben) oder ein erzwungenes Zeilenende, ist natürlich im Dokument abgespeichert.

Sagt man BRADFORD, daß alle Zeilen mit Blocksatz ausgegeben werden sollen, macht BRADFORD auch genau das. Soll eine Zeile als Absatz-Ende nicht im Blocksatz, sondern, so wie es sich gehört, einfach linksbündig ausgegeben werden, muß BRADFORD davon irgendwie Kenntnis erhalten. Dazu kann man an das Ende dieser Zeilen einen 'drucke linksbündig'-Code hängen. (Siehe Kommando .DC)

S o f t w a r e: BRADFORD-Dokumentation

Weiß nun BRADFORD, daß es ein NewWord (oder WordStar) Dokument ausdrückt, so tut es von sich aus das, was man erwartet: Im Blocksatz-Modus hängt BRADFORD an eine Zeile, die ein erzwungenes Zeilenende enthält automatisch und selbstständig den 'drucke linksbündig'-Code (auch, falls hierfür kein besonderes Zeichen mittels .DC definiert wurde).

Außerdem werden die Steuer-Codes für Fettdruck und Unterstreichen auf die von NewWord her bekannten ^B und ^S vordefiniert, die unter NewWord mittels vorangestelltem ^P in den Text eingegeben werden können. (Siehe auch hierfür Kommando .DC)

WAS BRADFORD NICHT KANN

Kopf- und Fußzeilen (Footer/Header) werden leider von BRADFORD Version 1.20 nicht unterstützt. Soll ein NewWord-Dokument mit Footer/Header mittels BRADFORD ausgedruckt werden, so muß das Dokument zuvor mit NewWord in eine Datei gedruckt werden (Drucker PRVIEW, liefert die Datei PREVIEW.NW), die dann mit BRADFORD ausgedruckt wird. Hierbei ist es ggf. erforderlich die Datei PREVIEW.NW zuvor etwas zu überarbeiten. Ebenso dürfte es Probleme mit dem Übereinanderdrucken von Zeilen geben (NewWord-Befehl ^P^M oder ^P RET), was durch ein '-' am rechten Bildschirmrand kenntlich gemacht wird.

Der Wechsel des Zeichensatzes oder der Zeichenbreite innerhalb einer Zeile ist in BRADFORD Version 1.20 auch nicht vorgesehen.

BRADFORD ignoriert alle Control-Codes außer dem Tab (^I), dem Zeilenende (^M^J) und den mit dem .DC-Kommando (siehe unten) definierten. Wird BRADFORD beim Aufruf mitgeteilt, daß es ein NewWord (WordStar) Dokument bearbeitet, werden automatisch ^B und ^S für Fettdruck/Unterstreichen vordefiniert (s.o.). Solche Schicki-Micki-Dinge wie ^Q, ^W, ^E, ^R oder ^A, ^N, ^Y oder ^T, ^V u.s.w. sind wirkungslos. (Eingegeben werden diese unter NewWord natürlich mit ^P gefolgt vom o.g. Ctrl-Code!)

BRADFORD unterstützt keine Footer/Header (.h1-.h3/.f1-.f3), kein Merge-Print oder ähnliche Feinheiten. Derartige Dokumente müssen erst mit dem PRVIEW-Drucktreiber in die Datei PREVIEW.NW gedruckt und dann evtl. manuell bearbeitet werden, bevor sie BRADFORD überlassen werden können.

AUFRUF

Vor dem Aufruf von BRADFORD muß sichergestellt werden, daß die gewünschten Zeichensätze, die in den Dateien FONT?.BIN stehen, auf dem aktuellen Laufwerk, User 0 sind. (Das ? steht für eine Ziffer 0-9 oder einen Buchstaben A-Z).

BRADFORD wird nicht mit einer Kommandozeile aufgerufen. Es muß nach dem Aufruf ein Quiz (wie es der Autor nennt) durchlaufen werden.

In der folgenden Beschreibung des Quiz ist links die Frage, die BRADFORD stellt genannt, und rechts daneben sind die gültigen Eingabewerte angegeben. Darunter stehen dann Erläuterungen zu den einzelnen Fragen.

Der Hinweis 'Dieser Wert kann mit ... verändert werden' weist auf BRADFORD-Kommandos hin, mit denen diese Voreinstellung im Text verändert werden kann.

In diesem Quiz sind folgende Angaben zu machen:

S o f t w a r e: BRADFORD-Dokumentation

- File [d:filename.ext]:** Laufwerk:Dateiname
Name der zu druckenden Datei mit Laufwerksangabe, falls die Datei nicht auf dem aktuellen Lw. ist.
- User number (0-15):** Zahl von 0 bis 15 oder RET (= 0)
User-Bereich, wo die zu druckende Datei ist (i.a. 0). Findet BRADFORD die genannte Datei nicht, so verlangt es Dateinamen/User noch einmal.
- Copies (0-99):** Zahl von 0 bis 99 oder RET (= 1)
Anzahl der Ausdrücke (0 = nichts)
- Printer Type (A-D):** Buchstabe von A bis D
Art des Druckers. Was BRADFORD zu bieten hat wird weiter oben auf dem Bildschirm angezeigt. Um herauszufinden, welcher Druckertreiber am besten klappt, sollten einfach mal die vier Möglichkeiten ausprobiert werden.
- Width, in 20th inch (1-9):** Ziffer von 1 bis 9
Zeichenbreite in 1/20 Zoll. Standardbreite (Pica) ist 2, d.h. 1/10 Zoll
Wird bei Verwendung des Druckertreibers A ein ungerader Wert angegeben, nimmt BRADFORD automatisch und kommentarlos die nächst höhere gerade Zahl. Dieser Wert kann mit .CW verändert werden.
- Strikes (1-9):** Ziffer von 1 - 9
Legt fest, wie oft der Text übereinandergedruckt werden soll. Mit 3 erhält man ein recht gutes kontrastreiches Bild, insbesondere bei alten Farbbändern. Mehr als 4 ist unsinnig. Für einen Probedruck empfiehlt sich 1, um das Farbband zu schonen. Dieser Wert kann mit .ST/.DK verändert werden.
- Font:** Zeichen von 1 bis 9 oder von A bis Z
Zeichensatz, mit dem gedruckt werden soll.
Sinnvoll nur, wenn zugehörige Datei FONT?.BIN auf dem aktuellen Laufwerk, User 0 ist. Wird der geforderte Zeichensatz nicht gefunden versucht BRADFORD Zeichensatz 1 zu finden. Wurde jedoch Zeichensatz 1 angefordert probiert BRAD/ORD Zeichensatz 3. Schlägt dies fehl, bricht BRADFORD ab.
Dieser Wert kann mit .FN verändert werden. (Siehe auch .SF/.OK)
- Left/Right/Center/Justify:** Einer der Buchstaben L, R, C oder J
Art der Druckaufbereitung: Linksbündig, Rechtsbündig, Zentriert, Blocksatz.
Dieser Wert kann mit .FL/.LF, .FR/.RF, .CE/.CN und .JU/.UJ verändert werden
- Proportional spacing?** Y (yes = ja) oder N (no = nein)
Proportionalschrift ja/nein
Aaron M. Contorer empfiehlt die Verwendung von Proportionalschrift für Texte, die mit Blocksatz ausgedruckt werden, weil das besser aussieht, und es keinen Grund gibt, es nicht zu tun!
Proportionalschrift beeinträchtigt natürlich weder Tabulatoren (Ctrl-I) noch die Leerschritte am Anfang einer Zeile. Daher können sogar Pascal-Programme vernünftig mit Proportionalschrift ausgedruckt werden.
Dieser Wert kann mit .P verändert werden.
- Number Pages?** Y (yes = ja) oder N (no = nein)
Seitennumerierung ja/nein
Sind unten auf der Seite nicht mindestens 4 Zeilen frei (.BM/.MB), wird keine Seitennumerierung gedruckt. Dieser Wert kann mit .N verändert werden.
- Pause between Pages?** Y (yes = ja) oder N (no = nein)
Soll beim Seitenwechsel pausiert werden?
Der Zweck dieses Schalters ist es, die Verarbeitung von Einzelblättern zu vereinfachen.

S o f t w a r e: BRADFORD-Dokumentation

Top and Bottom margins? Y (yes = ja) oder N (no = nein)
Setzt den Rand oben/unten: Y: oben: 3, unten 8 Zeilen.
N: oben: 0, unten 0 Zeilen.

Left and Right margins? Y (yes = ja) oder N (no = nein)
Setzt den Rand rechts/links: Y: Rand links: 8, Zeilenlänge: 65 Zeichen
N: Rand links: 0, Zeilenlänge: 80 Zeichen
BRADFORD verwendet für diese Voreinstellungen die Standard-Zeichenbreite von 2/20 (= 1/10) Zoll, d.h. Pica.

WordStar document file? Y (yes = ja) oder N (no = nein)
Legt fest, ob es sich um ein WordStar/NewWord-Dokument handelt.
Wird diese Frage mit Y beantwortet, so gelten die oben und im folgenden gemachten Bemerkungen zur Verarbeitung von WordStar/NewWord-Dokumenten:
Letzte Zeile eines Absatzes bei Blocksatz linksbündig, Vordefinition von ^B und ^S für Fettdruck bzw. Unterstreichen an/aus.
Dieser Wert kann mit .WS verändert werden.

Command character or RET: Irgend ein Zeichen oder RET (= nichts)
Legt Kommando-Prefix fest, d.h. das Zeichen, mit dem Kommandos in Spalte 1 kenntlich gemacht werden. Sinnvoll: Punkt ('.')

In dieser Kommando-Beschreibung wird der Punkt als Kommando-Prefix angenommen. Wird nur RET gedrückt: keine Kommandos

Is all this correct (Yes/No/Abort) ?
Y (ja), N (nein), A (Abbruch)
Y: BRADFORD fängt an zu Drucken
N: Das Quiz geht von vorne los
A: Dann eben nicht, zurück zu CP/M

KOMMANDOS

Bei der nun folgenden Beschreibung der Kommandos verwende ich den Punkt ('.')

als Kommando-Prefix (Command Character), da dieser auch tunlichst verwendet werden sollte, damit die Chose mit NewWord harmoniert.

Einige verschiedene BRADFORD-Kommandos bewirken genau das selbe! Ob ich .LM oder .ML verwende bleibt sich gleich. Aaron M. Contorer wollte dem Anwender die Wahl des Kommandos überlassen.

Im folgenden beschreibe ich die Kommandos, wobei ich die Alias-Namen eines Kommandos stets zusammen angebe.

Der Text 'wird ein NewWord-Dokument ausgedruckt, so ...' hat nur dann Gültigkeit, wenn BRADFORD beim Aufruf oder mit dem Kommando .WS1 gesagt wurde, daß es sich um einen solchen Text handelt.

S o f t w a r e: BRADFORD-Dokumentation

.DC	Define Character	Definiere Zeichen
.CD	Character Definition	Zeichen-Definition

BRADFORD kennt 4 verschiedene Funktions-Zeichen:

B: Bold	Fettdruck an/ausschalten
U: Underscore	Unterstreichen an/ausschalten
L: Left	Zeile linksbündig
C: Center	Zeile zentrieren

Diese Zeichen werden mit .DC (oder .CD) definiert:

.DCB! bewirkt, daß BRADFORD alle in ! eingerahmten Texte fett ausdrückt. Das ist die NewWord-Funktion ^P^B. Wird ein NewWord-Dokument ausgedruckt, so setzt BRADFORD automatisch .DCB^B

.DCU\$ bewirkt, daß BRADFORD in \$ eingerahmten Texte unterstreicht. Das ist die NewWord-Funktion ^P^S. Wird ein NewWord-Dokument ausgedruckt, so setzt BRADFORD automatisch .DCU^S

.DCL\$ bewirkt, daß BRADFORD alle Zeilen, die in als letztes Zeichen ein \$ enthalten linksbündig ausdrückt. Wird ein NewWord-Dokument ausgedruckt, so druckt BRADFORD automatisch alle Zeilen, die mit RET beendet wurden (d.h. unter NewWord ein '<' rechts am Rand haben) linksbündig.

.DCC# bewirkt, daß BRADFORD alle Zeilen, die in als letztes Zeichen ein # enthalten zentriert ausdrückt.

Soll ein Ctrl-Code als Zeichen mit .DC definiert werden, wie Z.B. ^B für Fettdruck, so muß diese Zeile unter NewWord wie folgt eingegeben werden:

Punkt D C Ctrl-P Ctrl-B (ohne Leerzeichen!)

D.h. das ^B muß mittels Ctrl-P Ctrl-B in den Text eingefügt werden, d.h. die Angabe des gewünschten Zeichens erfolgt genauso, wie man dieses Zeichen in den Text eingibt. (Ctrl-P = Ctrl-Code in Text einfügen, und Ctrl-B ist der einzufügende Ctrl-Code.)

Um eine mit .DC festgelegte Definition wieder aufzuheben, wird als Zeichen ein Blank (Leerzeichen) eingegeben:

.DCB bewirkt, daß das zuvor mit .DCB als Fettdruck-Kennung definierte Zeichen diese Funktion nicht mehr hat.

.PA	Page	Seitenvorschub
	Beginnt eine neue Seite (wie NewWord).	

.DM	Display Message	Gib Meldung aus
	.DM Dies ist eine Meldung	
	gibt den Text 'Dies ist eine Meldung' auf dem Bildschirm aus.	
	(wie NewWord)	

.WK	Wait for KeyPress	Warte auf Tastendruck
	BRADFORD hält mit der Verarbeitung an, bis eine beliebige Taste gedrückt wurde. (Ähnlich wie NewWord ^P^C, wobei NewWord jedoch unbedingt ein ^C zum Weitermachen verlangt, BRADFORD jedoch jedes Zeichen akzeptiert).	

.N	Number Pages	Seitennumerierung an/aus
	.NO	schaltet Seitennumerierung aus (Ähnlich NewWord .op)
	.N1	schaltet Seitennumerierung an (Ähnlich NewWord .pn)
	Falls nicht unten auf der Seite mindestens 4 Leerzeilen frei sind (.BM/.MB) wird keine Seitennumerierung gedruckt, da die Seitennummern unten auf der Seite gedruckt werden.	
	Beim Aufruf von BRADFORD wird die Voreinstellung im Quiz erfragt.	

S o f t w a r e: BRADFORD-Dokumentation

- .PN** Set Page Number Setze Seitennummer
 .PN333 veranlaßt BRADFORD, die aktuelle Seitennummer auf 333 zu setzen. Zulässige Werte: 1-32000. (NewWord .pn)
- .PS** Page Numer Style Seitennummerierungs-Art
 .PSR bewirkt, daß die Seitennummern rechts außen stehen.
 .PSC bewirkt, daß die Seitennummern mittig stehen.
 .PSL bewirkt, daß die Seitennummern links außen stehen.
- .WS** WordStar Control WordStar/NewWord
 .WS0 schaltet 'NewWord-Modus' aus, d.h. BRADFORD tut so, als wenn es kein NewWord-Dokument ist.
 .WS1 schaltet 'NewWord-Modus' an, d.h. BRADFORD tut so, als wenn es ein NewWord-Dokument ist, d.h. die o.g./u.g. Hinweise bzgl. NewWord gelten.
 Beim Aufruf von BRADFORD wird die Voreinstellung im Quiz erfragt.
- .PL** Page Length Seitenlänge
 .PL60 setzt die Seitenlänge auf 60 Zeilen.
 Default: 66, zulässiger Bereich: 1-254. (wie NewWord).
- .CE** Center Zentrieren
.CN Center "
 Alle folgenden Zeilen werden zentriert.
 (Siehe auch .JU/.UJ, .FL/.LF, .FR/.RF).
 Beim Aufruf von BRADFORD wird die Voreinstellung im Quiz erfragt.
- .JU** Justify Randausgleich
.UJ Justify "
 Alle folgenden Zeilen werden im Blocksatz gedruckt.
 (Siehe auch .CE, .FL/.LF, .FR/.RF).
 Beim Aufruf von BRADFORD wird die Voreinstellung im Quiz erfragt.
- .FL** Flush Left Linksbündig
.LF Left Flush "
 Alle folgenden Zeilen werden linksbündig gedruckt.
 (Siehe auch .CE, .JU/.UJ, .FR/.RF).
 Beim Aufruf von BRADFORD wird die Voreinstellung im Quiz erfragt.
- .FR** Flush Right Rechtsbündig
.RF Right Flush "
 Alle folgenden Zeilen werden rechtsbündig gedruckt.
 (Siehe auch .CE, .JU/.UJ, .FL/.LF).
 Beim Aufruf von BRADFORD wird die Voreinstellung im Quiz erfragt.
- .ST** Strikes Übereinanderdruck-Häufigkeit
.DK Dakrness Dunkelheit (Druckstärke)
 .ST3 bewirkt, daß die Ausgabe dreifach übereinander gedruckt wird, um ein kräftigeres Bild zu erhalten.
 Sinnvoll ist max. 4, zulässiger Bereich: 1-9.
 Beim Aufruf von BRADFORD wird die Voreinstellung im Quiz erfragt.
- .BN** Boldface No. of Strikes Fettdruck-Stärke
 .BN3 bewirkt, daß die Ausgabe von Fettdruck durch 3-fach Übereinanderdruck erfolgt.
 Sinnvoll ist 3, Default: 3, zulässiger Bereich: 1-9.
 Ist mit .ST3 (oder .DK3) der Normaldruck auf Dreifach gesetzt, und mit .BN2 der Fettdruck als Doppeldruck definiert, liefert Fettdruck Sechsfach-Druck.

S o f t w a r e: BRADFORD-Dokumentation

- .CW** Character Width Zeichenbreite
.CW3 setzt die Zeichenbreite auf 3/20 Zoll.
 Der Druckertyp A (siehe Aufruf von BRADFORD) kann nur mit geraden Werten von .CW arbeiten. Ggf. verwendet BRADFORD kommentarlos die nächst höhere Breite. (Ähnlich NewWord .CW).
 Beim Aufruf von BRADFORD wird die Voreinstellung im Quiz erfragt.
- .FN** Font Number Wähle Zeichensatz
.FNC wählt den Zeichensatz aus der Datei FONTC.BIN aus.
 Zulässige Werte sind 0-9, A-Z, sofern dazu die entsprechende Datei FONT?.BIN auf der Diskette ist. (Siehe auch .SF/.OK)
 Beim Aufruf von BRADFORD wird die Voreinstellung im Quiz erfragt.
- .SF** Single Font Nur ein Zeichensatz
 Mit diesem Kommando können die .FN-Kommandos deaktiviert werden. Diese Funktion ist gedacht, damit man mit dem Zeichensatz 5 (Draft) einen Probeausdruck machen kann, ohne die .FN-Kommandos aus dem Text zu entfernen. Draft ist immerhin deutlich schneller als die NLQ-Zeichensätze
- .OK** O.K. O.K.
 Wenig sprechend: Dieses Kommando hebt die Wirkung von .SF wieder auf, d.h. die .FN-Kommandos sind wieder wirksam.
- .P** Proportional Spacing Proportionalschrift
.P0 schaltet Proportionalschrift aus
.P1 schaltet Proportionalschrift an
 Proportionalschrift beeinträchtigt natürlich weder Tabulatoren (^I) noch die Leerschritte am Anfang einer Zeile. Daher können sogar Pascal-Programme vernünftig mit Proportionalschrift ausgedruckt werden.
 Beim Aufruf von BRADFORD wird die Voreinstellung im Quiz erfragt.
- .LX** Left Margin Exact Linker Rand genau
.XL Left Margin Exact "
.LX200 definiert den linken Rand auf 200/120 Zoll.
 Zulässiger Wertebereich 0-Papierbreite; der DMX80 druckt i.a. auf 8-Zoll Papier, d.h. auf 960/120 Zoll.
- .RX** Right Margin Exact Rechter Rand genau
.XR Right Margin Exact "
.RX900 definiert den rechten Rand auf 900/120 Zoll.
 Zulässiger Wertebereich siehe .LX/.XL
 Der rechte Rand ist keine absolute Druckgrenze. Er dient lediglich als Orientierungswert für Zentrieren und rechtsbündiges Drucken, wird jedoch ignoriert, falls die Textzeile nicht zwischen den linken und rechten Rand paßt.
- .PO** Page Offset Einrück-Bereich
.LM Left Margin Linker Rand
.ML Margin Left Rand links
.LM4 definiert den linken Rand auf 2/10 Zoll, d.h. bei der normalen Zeichenbreite von 1/10 Zoll auf 2 Zeichen.
 Zulässiger Wertebereich 0-245/Papierbreite.
 Wurde beim Aufruf von BRADFORD im Quiz die Frage 'left and right margins' mit N (no = nein) beantwortet, ist die Voreinstellung 0, wurde mit Y (yes = ja) geantwortet ist die Voreinstellung 8.

S o f t w a r e: BRADFORD-Dokumentation

- .RM Right Margin Rechter Rand
- .MR Margin Right Rand rechts
- .RM75 definiert den rechten Rand auf 75/10 Zoll, d.h. bei der normalen Zeichenbreite von 1/10 Zoll auf 75 Zeichen vom linken Rand aus gemessen.
Zulässiger Wertebereich 0-245/Papierbreite.
Wurde beim Aufruf von BRADFORD im Quiz die Frage 'left and right margins' mit N (no = nein) beantwortet, ist die Voreinstellung 80, wurde mit Y (yes = ja) geantwortet ist die Voreinstellung 65.
- .TM Top Margin Oberer Rand
- .MT Margin Top Rand oben
- .TM3 definiert den oberen Rand (d.h. die Zeilen, die oben auf der Seite frei bleiben) auf 3 Zeilen.
Zulässiger Wertebereich 0-253.
Wurde beim Aufruf von BRADFORD im Quiz die Frage 'top and bottom margins' mit N (no = nein) beantwortet, ist die Voreinstellung 0, wurde mit Y (yes = ja) geantwortet ist die Voreinstellung 3.
- .BM Bottom Margin Unterer Rand
- .MB Margin Bottom Rand unten
- .BM8 definiert den unteren Rand (d.h. die Zeilen, die unten auf der Seite frei bleiben) auf 8 Zeilen.
Zulässiger Wertebereich 0-253.
Wurde beim Aufruf von BRADFORD im Quiz die Frage 'top and bottom margins' mit N (no = nein) beantwortet, ist die Voreinstellung 0, wurde mit Y (yes = ja) geantwortet ist die Voreinstellung 8.

AUFBAU DER ZEICHENSATZ-DATEIEN

Die FONT-Dateien FONT?.BIN können mit BRADRES in eine Editierbare Form gebracht werden. BRADRES erstellt aus FONT?.BIN die Datei FONT?B. Diese kann nach Bearbeitung mit BRADCON wieder in eine .BIN gewandelt werden.

In eine FONT?B.-Datei belegt jedes Zeichen genau 16 Zeilen á 9 Spalten.
Ein Zeichen beginnt in der Zeile (ASCII-Code - 32)*16 +1,
und endet in der Zeile (ASCII-Code - 32)*16 +16.
Das große A hat den ASCII-Code 65, geht also von Zeile 529 bis 544.

Der Aufbau der Zeichen am Beispiel von A und a aus FONT1:

		1	2	3	4	5	6	7	8	9	Zeile		1	2	3	4	5	6	7	8	9	Zeile
Groß	1	.	.	.	x	x	x	.	.	.	529	1	1041
	2	.	.	.	x	.	x	.	.	.	530	2	1042
	3	.	.	x	x	.	531	3	1043
	4	.	.	x	x	.	532	4	1044
	5	.	x	x	533	5	1045
Klein	6	.	x	x	534	6	.	.	x	x	x	x	.	.	.	1046
	7	x	x	535	7	.	x	x	.	1047
	8	x	x	536	8	x	1048
	9	x	x	x	x	x	x	x	x	x	537	9	x	1049
	10	x	x	538	10	.	.	x	x	x	x	x	x	.	1050
	11	x	x	539	11	.	x	x	1051
	12	x	x	540	12	.	x	x	1052
	13	x	x	541	13	.	x	x	1053
	-----	14	x	x	542	14	.	.	x	x	x	x	x	x	x
Unter- längen	15	543	15	1055
	16	544	16	1056

Software: BRADFORD-ZeichensätzeTypewriter (1)

THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG 1234567890<>?:"08
 the quick brown fox jumps over the lazy dog !\$#%^&*()_+==ÄÜäüß,./;'

Modern (2)

THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG 1234567890<>?:"08
 the quick brown fox jumps over the lazy dog !\$#%^&*()_+==ÄÜäüß,./;'

Vanilla (3)

THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG 1234567890<>?:"08
 the quick brown fox jumps over the lazy dog !\$#%^&*()_+==ÄÜäüß,./;'

Nostalgia (4)

THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG 1234567890<>?:"08
 the quick brown fox jumps over the lazy dog !\$#%^&*()_+==ÄÜäüß,./;'

Draft (5)

THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG 1234567890<>?:"08
 the quick brown fox jumps over the lazy dog !\$#%^&*()_+==ÄÜäüß,./;'

Italic (6)

THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG 1234567890<>?:"08
the quick brown fox jumps over the lazy dog !\$#%^&()_+==ÄÜäüß,./;'*

Computer (7)

THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG 1234567890<>?:"08
the quick brown fox jumps over the lazy dog !\$#%^&*()_+==ÄÜäüß,./;'

SMALLCAPS (8)

THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG 1234567890<>?:"08
 THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG !\$#%^&*()_+==ÄÜäüß,./;'

ORATOR (9)

THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG 1234567890<>?:"08
 THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG !\$#%^&*()_+==ÄÜäüß,./;'

Pizza (A)

THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG 1234567890<>?:"08
 the quick brown fox jumps over the lazy dog !\$#%^&*()_+==ÄÜäüß,./;'

Jazz (B)

THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG 1234567890<>?:"08
the quick brown fox jumps over the lazy dog !\$#%^&*()_+==ÄÜäüß,./;'

Italic Typewriter (C)

THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG 1234567890<>?:"08
the quick brown fox jumps over the lazy dog !\$#%^&()_+==ÄÜäüß,./;'*

Gotic (D)

ŦŦE QUICK BROWN FOX JUMPS OVER ŦŦE LAZY DOG 1234567890<>?:"08
 the quick brown fox jumps over the lazy dog !\$#%^&*()_+==ÄÜäüß,./;'

Lasce (F)

THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG 1234567890<>?:"08
the quick brown fox jumps over the lazy dog !\$#%^&*()_+==ÄÜäüß,./;'

G r a f i k: Chaos**Einfaches Programm mit viel Wirkung**

(HzN, 2000 & OK, 2050)

Das folgende Programm CHAOS2 aus Spektrum der Wissenschaft 10/87 erzeugt mit ein und dem selben Algorithmus sehr viele verschiedene Bilder. Auf der nächsten Seite sind drei Stück als Vorgeschmack zu weiteren Versuchen. Leider dauert die Erstellung eines Bildes doch schon ein paar Mega-Centi-Sekunden.

```

var i: integer;
    y0,a,x,xx,y: real;
    sqr_x, sin_a, cos_a, y_xx: real;
    Offset: real;
    XAuf1, YAuf1: integer;
    xMax, yMax: real;
    ch: char;
    s: string[10];

label endloop;

begin
randomize;

repeat
    Offset := 0.025; XAuf1 := 250; yAuf1 := 250;
    a := random*2*pi;
    xMax := (750-XAuf1)/XAuf1;
    yMax := (550-YAuf1)/YAuf1;
    InitializeGrafic(768,512);
    ClsGrafic;

    sqr_x := sqrt(9.999999999e+37);
    sin_a := sin(a); cos_a := cos(a); str(a:6:3,s);
    y0 := Offset;

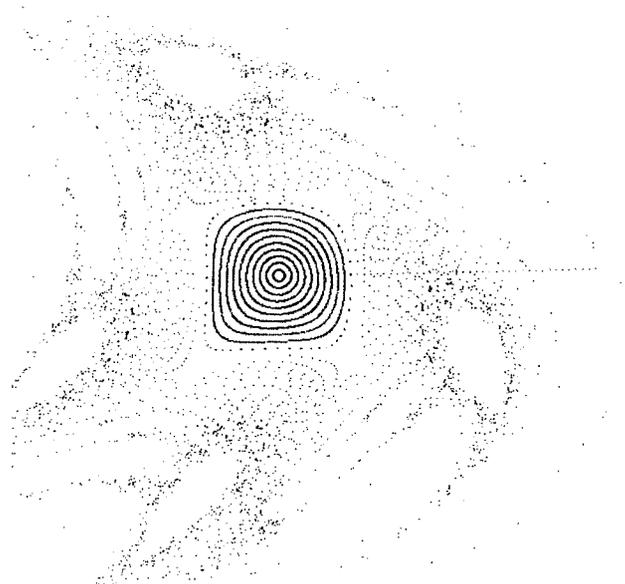
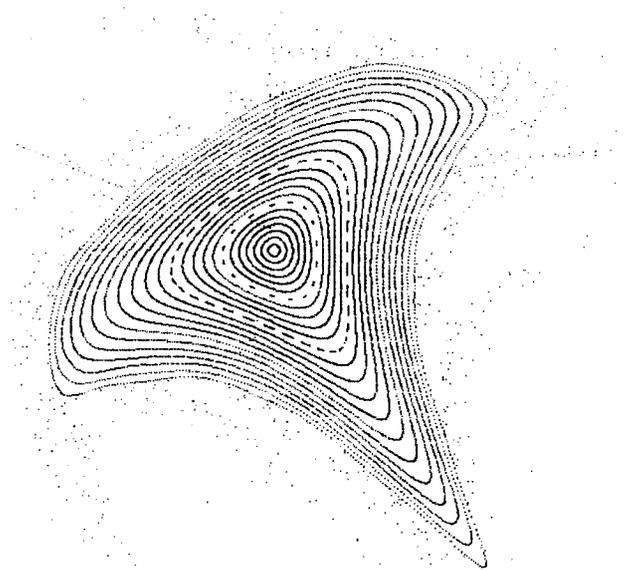
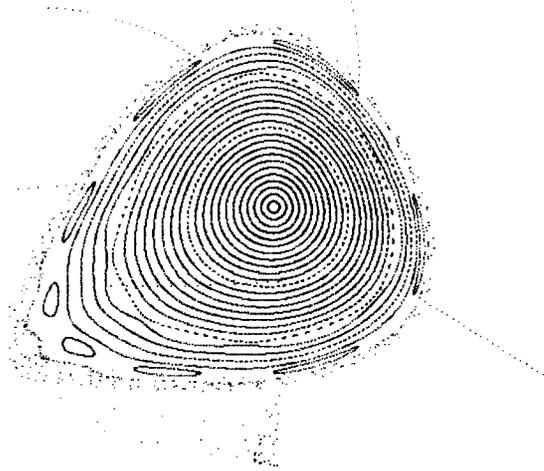
    shile y0 < yMax do begin
        y := y0; x := 0.0;
        for i := 1 to 1000 do begin
            if abs(x) > sqr_x then goto endloop;
            y_xx := y-x*x;
            xx := x*cos_a - y_xx*sin_a;
            y := x*sin_a + y_xx*cos_a;
            x := xx;
            if (abs(x)<xMax) and (abs(y)<yMax) then begin
                x1 := trunc(x*xAuf1)+xAuf1;
                y1 := trunc(y*yAuf1)+yAuf1;
                plot(x1,y1);
            end; { if }
        end; { for i }
    endloop:
        y0 := y0+Offset;
    end; { while }

    writeln('Noch einen ? (J/N') ');
    repeat
        read(kbd,ch);
        ch := upcase(ch)
    until ch in ['J','N'];
until ch='N';
end.

```

G r a f i k: Chaos

Beispiel-Bilder zu CHAOS2 für $a=5.266$, $a=1.958$ und $a=4.669$:



S o f t w a r e: V I R U S**Virus oder Trojanisches Pferd**

(Herbert zur Nedden, 2000)

Ich erhielt folgenden - leider englischen - Artikel über Viren, d.h. über Programme, die andere Programme oder Rechner im wahrsten Sinne des Wortes infizieren. Das geschieht dadurch, daß sich diese Programme in andere Programme oder sogar Teile des Systems einlagern, und je nach dem, wie gemein sie programmiert wurden, nur mal eine dümmliche Meldung ausgeben um sich dann selbst zu löschen, oder gar versuchen sich schnellstmöglichst zu verbreiten, und zusätzlich zerstörerische Tätigkeiten zu entfalten. Zu Eurer Beruhigung möchte ich erwähnen, daß die Boot-Probleme und ähnliche Phänomene des Memotech keine Viren, sondern schlicht und ergreifend Hardwaredefekte zur Ursache haben. Im Bereich von diesen Pseudo-CPU-getriebenen Tastatur-Bildschirm-Laufwerks-Geräten, die den Anwender mit MSDOS bestrafen, sind solche Phänomene (Viren) schon wahrscheinlicher, da diese Umgebung wesentlich standardisierter und die Programme wesentlich größer sind (auch bei gleicher Leistung).

Dear Readers,

If your aren't familiar with PC's and or other more modern 16-bit computers, you've probably never heard of a virus before. Experience has learned that the machines mentioned above can be affected by running very special foul software called VIRUS or TROJAN HORSE. This software was first designed to protect commercial software and their makers (some programs did cost over 1000,- Pounds). Because of the hardware standard of MS-DOS machines the spreading of this type of software was easy. The software did not have to be altered to run on different machines (like CP/M operated computers). Of course, copying of software is a crime and from that point of view the makers and or distributors have all rights to protect their software. No problem at all! These virus programs are mostly effective when the memory is not cleared after a reset and consequently survives a 'soft-reset'. Many computer owners think that if they are working with an 'old' machine, they are immune to this kind of software. A fatal mistake!!!! Making a virus (or trojan horse) is relatively simple and can be made by anyone with moderate assembly programming experience. I don't know if someone in the UK has suffered from a virus on the FDX, but here in Holland, I discovered a virus program. Consequently the question arose; "How to detect and eliminate a virus?". Perhaps the following explanation may clarify things....

VIRUS

A VIRUS is (like its biological brother) a self-propagating routine that alters the system's behaviour and in some cases ruins your discs by erasing or adapting programs on that disc. The more a problem because the virus sneaks in your memory without you being aware of it!! Of course there are many different virus routines, all with another type of disasterous programming. Some suddenly HALT the computer, making you loose a lot of work, whilst others rewrite your directory tracks, making you loose all your programs on the inserted disc. If this happens once, it's a pity, if it goes on for 10 discs it's a calamity! The worst feature of a virus is that it copies itself onto newly inserted discs, thus propagating itself and before you know you're stuck with a box of infected discs.

Trojan Horse

These programs are in fact more dangerous than the average virus, because the program actually does something useful. It normally doesn't propagate itself, but only because it doesn't have to! The user will take care of this! If the programs results are good enough, you are normally most willing to copy that program onto other discs. Finding a trojan horse is quite difficult. You must be able to read the machine code that makes up the program in order to find it.

S o f t w a r e: V I R U S**What can you do to prevent virus attacks?**

First of all; **Don't copy software from others** and if you can't resist the temptation, copy it, but **don't run it just yet!** Most virus routines can be found in the **boot-sectors** of the disc. In fact, it's the only safe place for the virus. You normally can't reach it and the DOS normally never rewrites these sectors. If you suspect a disc from containing a virus, you can eliminate it by **copying certified system tracks onto the believed affected disc** VIRUS RIP! Unfortunately there are some diabolically clever (and foul) virus routines that merge with selected software on the disc! To give an example:

Suppose you are the proud owner of a Bondwell 12/14 portable CP/M computer. Once switched on it expects to find 2 programs on the disc 'AUTORUN.COM' and 'SETUP.COM'. The virus would merge with either of the two, leaving the system-tracks intact. The only way of telling that 'something is wrong' is by checking the length of these two programs (the affected programs are a little larger). This sounds very easy, but is in practice difficult enough. You may know that CP/M filelengths are given in a number of 1K blocks and **not** as in MS-DOS in the exact number of bytes. Since virus routines are normally quite small, the CP/M programs don't show different sizes!

You may find the program COMPARE program listed below very handy.

```
10 REM *** FILE CHECKER ***
20 INPUT "ENTER NAME OF SOURCE PROGRAM:",S$
30 INPUT "ENTER NAME OF COMP. PROGRAM:",C$
40 OPEN "R",#1,S$,128
50 FIELD #1,128 AS S1$
60 OPEN "R",#2,C$,128
70 FIELD #2,128 AS S2$
80 I%=1
90 WHILE NOT EOF(1)
100 GET #1,I%
110 GET #2,I%
120 IF S1$<>S2$ THEN 200
130 I%=I%+1
140 WEND
150 CLOSE #1
160 CLOSE #2
170 END
180 REM *** DIFFERENT STRINGS FOUND ***
200 PRINT "MISMATCH ERROR DETECTED"
210 GOTO 100
```

It's a simple MBASIC program, but it works perfectly!! Assembly language programs of this kind can be found in the vast CP/M Public Domain libraries.