

MTX *User-Club Deutschland*

Info 40
25.11.1990

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.- je Exemplar. Jeder kann dazu Beiträge liefern und hier gratis Kleinanzeigen veröffentlichen.

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

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

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

Kontaktadressen:

Herbert zur Nedden
Sonnenau 2
2000 Hamburg 76
(040) 200 87 04

Ab Januar 1991:
Herbert zur Nedden
Alte Landstraße 21
2071 Siek

Hans Gras
Statenhoek 49
NL 1506 VM Zaandam
(0031-75) 17 49 91

Telefon-Sprechzeiten

Herbert zur Nedden: Do 18 - 22 Uhr, Sa 9 - 14 Uhr

Inhaltsverzeichnis**C l u b**

Korrektur & Nachtrag	Seite 2
Fragen & Antworten	Seite 3

R A M 6.x

Extended Command Processoren (ECPs)	Seite 5
Archiv-Flag oder Archiv-Bit	Seite 7

B r a d f o r d

Watt'n datt'n	Seite 8
Befehlsübersicht	Seite 12

H a r d w a r e

Tips	Seite 4
768k-Karte	Seite 15
512k-Hauptplatine	Seite 15
768k-Karte Schaltplan	Seite 16
768k-Karte Trennungen	Seite 17
768k-Karte Legungen Oberseite	Seite 18
768k-Karte Legungen Unterseite	Seite 19
512k-Hauptplatine Schaltplan	Seite 20
512k-Hauptplatine Layout	Seite 21
512k-Hauptplatine Lötanweisungen	Seite 22
8"-Laufwerke	Seite 23
12 Volt-Bräter	Seite 23

C P / M

BDOS-Dateifunktionen	Seite 24
----------------------	----------

S o f t w a r e

Quick Copy 'n Compare	Seite 24
Besprechung von QCC	Seite 26
StopCB4	Seite 26

A s s e m b l e r

Tastatur	Seite 26
----------	----------

Preis für dieses Info: DM 8,70

Redaktionsschluß für's nächste Info: 15.01.91

Wettbewerb

(Herbert zur Nedden, 2000)

Auf KLIICK.012 ist ein Brettspiel. Jeder kann eine eigene Bewertungsfunktion schreiben. Die beste gewinnt, dem Ersteller winken Lorbeeren und PD's im Werte von DM 99.-

Clubtreffen

(Herbert zur Nedden, 2000)

Geplant im April 1991 oder in dessen Nähe; Sa/So mit Übernachtung vor Ort. Der Ort ist wieder südl. von Hannover. Bitte teile mir Termineinschränkungen, Wünsche etc. mögl. bald mit. Das Anmeldeformular erscheint mit Info 41.

Kontostand

(Herbert zur Nedden, 2000)

Bitte prüfe den Deinen!

Anzeigetexte samt Absender bitte schriftlich an Herbert zur Nedden!

V E R K A U F (Preise sind i.a. ohne Porto & Verpackung)

Herbert zur Nedden, Sonnenau 2, 2000 Hamburg 76, 040 - 2008704:

--> Interessiert Dich einer der von mir angebotenen Posten: Mach ein Angebot!
Meine hier genannten Preise sind nicht unbedingt unumstößlich!
(Neue/geänderte Posten haben einen * statt des - vorne weg)

- * Drucker Star Gemini 15x, breit, d.g. A4 Quer/A3-Format: DM 200.-
- * Evtl. Grünmonitor: DM 100.- (Kommt erst darnächst, muß ich testen)
- Rikadenki Plotter RY21, VB DM 1500,-
Flachbettplotter, DIN A4, Aufnahme für Rotringstifte, incl. Handbuch und Schaltplan, 8085 CPU (Z80-aufwärtskompatibel), Centronics-Schnittstelle. Positioniergenauigkeit: 0,1 mm, Zeichengeschwindigkeit 200 mm/s, 19 Kommandos wie u.a. Kreise, Kreisbögen, Rechtecke, verscheiden gestrichelte Linien, Textausgabe in 4 Richtungen und verschiedenen Größen, absolute und relative Koordinaten, Markierungen auf Linien und Graphen. Kann in Textmodus gesetzt werden, um als Drucker zu arbeiten.
- Ich habe RAMs für 768kB-Erweiterung!
- Disketten-Laufwerke: (gebraucht und getestet)
 - EPSON SD 560: 5 1/4", 80-Spur, Nur-HD-Lw., d.h. Formate 1A, 1B, aber nicht 03, 07, 09, 0A usw. DM 80.-
- * dBASE Version 2.41: DM 100.-
- * NewWord Version 2.16, im Tausch gegen Original-NW-Diskette: DM 40.-
- * NewWord Version 2.02 DEUTSCH, im Tausch gegen Original-NW-Diskette: DM 40.-
- * Original Microsoft-BASIC-Lizenz incl., M80/L80 Assembler/Linker, Handbuch, auf dem MTX lauffähig: DM 180.- (Microsoft liefert dieses Teil nicht mehr aus!)
- * NonDoc-Editor für CP/M incl. Source, zeilen- und bildschirmorientiert, Macro-Fähig, mit Handbuch im Ringordner: VB DM 40.-
- Z80ASM (Club-PD Z80-Ass.): Listing und Handbuch, ca. 1.4 cm DIN A4: VB DM 5.-
- * Infos 11-36, gebraucht VB DM 60.- frei Haus
- Bücher
 - Programmierung des Z80, Rodney Zaks, 606 Seiten: DM 45.-
 - Mikroprozessor Interface-Techniken, A. Lesea/R. Zaks, 425 Seiten: DM 30.-
 - Operationsverstärker Anwendung, 164 Seiten, DM 10.-
 - ECA-Tabelle ttl-IC's, endet im Bereich der 74xx400-er: DM 20.-
 - ECA-Tabelle dat 1: Transistoren A..BUY: DM 5.-
 - ECA-Tabelle tht: Thyristoren, Triacs,: DM 5.-
- Einbau-Drehspulmeßgerät 0-50uA: DM 10,-
- Solange der Vorrat reicht:
 - MTX-Tasten je DM 1.-, Tastenkappen je DM -.50
 - EPROMS 2564 für je DM 15.-
 - Dynamische RAMs 4116 (VRAMs) 8 Stück: DM 25.-
 - Dynamische RAMs 3732 (32k x 1Bit) 8 Stück: DM 1.50
 - Statische RAM's 2k x 8 Bit (6116): je DM 2.-
 - TTL-IC's: 74LS175, 74LS368, 74LS173, 74LS158, 74LS258 je DM 0.50;
74LS10, 74LS11, 74LS21 je DM 0.30
 - Original-Memotech-Spielecassetten: Toado, Kilopede, Knuckles, Draughts, Reversi, Snappo, Blobbo, Utilities, Demo, StarCommand je DM 4.-;
 - 10 Disketten FUJI HD 5 1/4", gebraucht & o.k. DM 35.-, Originalverpackt: DM 60.-

Philemon Lopatta, Konrad-Adenauer-Str. 121, 7560 Gaggenau, 07225-76225

* Infos 1-39 für DM 50.- frei Haus.

Werner Steiner, Burghofweg 30, 7318 Lenningen 2, 07026-5413

* RS232-Karte und 128k-Karte billig abzugeben.

Liebe Leserin, lieber Leser,

Mein Umzug in unser Häusle wird irgendwann Mitte Januar 1991 stattfinden. Selbsturmelnd werde ich einen Nachsendeantrag bei der Post stellen, so daß ich schriftlich unter meiner alten Adresse auch noch etwas länger erreichbar bin. Meine neue Telefonnummer kenne ich noch nicht.

Zum Thema Info auf Diskette habe ich bislang sechs Nein's, drei Ja's und ein Egal erhalten. Der Grund für die Ablehnungen war, daß dann Zeichnungen und Komik schlecht möglich wären und viele das Info eh selbst ausdrucken würden. Zum einen könnte ich Zeichnungen ggf. auf Papier beilegen und das Ausdrucken ist sicherlich etwas preiswerter als die Druckkosten. Ich hatte vor, das Info in drei Versionen auf die Diskette zu packen: 1. mein Wordstar-Original mit Michael-Keßler Grafik (d.h. in ^Ä ^Ö eingerahmt), 2. dto. konvertiert auf Erik d'Hondt's New-Word-Grafik (in ^X eingerahmt), die auf DMX80 ausgedruckt werden kann und 3. mit Simpel-Graik (d.h. + - ! als Rahmen). Alle Versionen würden selbsturmelnd die erforderlichen Punkt-Kommandos (.pl, .mt usw.) enthalten, um einen problemlosen Ausdruck zu gewährleisten. Da ich mich jedoch aufgrund von zehn Meinungen schlecht mit gutem Gewissen entscheiden kann, bitte ich um weitere Stellungnahmen. In diesem Zusammenhang wurde gefragt, ob das Info nicht auf Umweltpapier erscheinen könne. Die Druckerei war von dieser Anfrage weniger begeistert, da das Umweltpapier relativ staubig ist, und damit die Druckmaschine verdreckt. Außerdem ist das Papier leider noch teurer als normales. Daher wird vorerst nichts aus Infos auf grauem Papier. Ich werde das jedoch auch weiterhin im Auge behalten.

Anton Reiser erwähnte, daß für den Überblick 'grundsätzliche' Artikel oder auch Aufstellungen wie die der KLIX-Overlays gut sind. Davon möchte er gerne mehr lesen. In diesem Zusammenhang möchte ich Euch alle um Anregungen bitten. Schlummert da noch eine Idee oder gar der Anfang eines Artikels bei Dir auf Diskette oder in der Schublade? Ich bin dafür ein dankbarer Abnehmer!

Es begab sich vor einiger Zeit, daß Olaf und ich statt am Computer sitzen zu wollen, Lust auf ein Spiel hatten. Also kramte ich Uisge hervor. Das ist ein strategisches Brettspiel mit sehr einfachen Regeln. Es ist jedoch verblüffend interessant im Spiel an sich. Es ist ein Spiel der Form 'mit fast nichts viel erreicht'! Da Regeln und Aufbau des Spiels schön einfach sind, kamen wir auf die Idee, Uisge zu programmieren. Das Resultat ist auf KLICK.012 zu finden. Das Herzstück eines jeden 'Denk-Spiels' die Bewertungsfunktion, d.h. die Routine, die berechnet, wie gut eine Stellung ist. Olaf kam auf die Idee einen Wettbewerb zu starten: Jeder, der Lust hat, kann eine eigene Bewertungsfunktion in Turbo-Pascal (oder Assembler via InLine) programmieren und damit am Wettkampff teilnehmen. Dem Gewinner winken insbesondere Lorbeerblätter sowie Public-Domain-Software im Wert von DM 99.- aus dem Club. Die genauen Regeln sind in der Doku zu Uisge vermerkt.

Klarerweise ist im April oder in dessen Nähe 1991 wieder ein Clubtreffen in Kuckenmühle, der Nähe von Hemmingen südlich von Hannover geplant. Es soll wieder auf einem Wochenende Sa/So mit Übernachtung vor Ort liegen. Hast Du Wünsche, Termineinschränkungen, etwas Vorführenswertes oder was auch immer dazu anzumerken, laß es mich bitte wissen. Im kommenden Info wird dann wieder das Anmeldeformular liegen. Wenn ich bis dahin schon Terminprobleme kenne, kann ich evtl. schon die möglichen Termine andeuten.

Frohe Weihnachten und einen guten Rutsch ins Neue Jahr sowie auch weiterhin viel Spaß mit dem MTX wünscht Euch

Euer
Rutsch

C l u b: Korrektur & Nachtrag**Korrektur & Nachtrag**E45.COM: (Hans Gras, NL-1506)

Ich habe ein Problem mit E45.COM: wenn ich eine Datei lade, die größer als der verfügbare Speicher ist, kann ich nicht über RET (Meldung akzeptieren) und dann ^KQ (quit) E45.COM verlassen. Das System hängt und SHIFT-ESC bringt auch nichts mehr (System ist 60K groß). Nur wenn ich auf die Meldung 'Speicher voll' sofort über SHIFT-ESC einen Warmboot auslöse gelange ich wieder auf die CP/M-Ebene.

Anm.d.HzN: Das stimmt und wird mit E60 behoben: KLICK.012.

RAM 6.0-Handbuch: (Herbert zur Nedden, 2000; gefunden von Jan Brederke, 2000)

Auf Seiten 115/116 steht zu lesen, daß die Uhrendaten HINTER den von KLICK geretteten Registern gerettet werden um beim Verlassen des KLICK wieder restauriert zu werden. Dem ist nicht ganz so. Die sechs Bytes Uhrendaten stehen VOR den Registern. Die Adresse der geretteten Uhrendaten kannst Du wie folgt ermitteln:

```
ld iy,(VarPtr)      ; iy = Zeiger auf RAM-Variablen
ld l,(iy+26h)
ld h,(iy+27h)      ; hl = K1xRegs = Adresse geretteter Register
ld de,-6
add hl,de          ; hl = K1xRegs-6 = Adresse geretteter Uhrendaten
```

MONI: (Jan Brederke, 2000)

"MONI file.ext" funktioniert nicht, es wird eine zu lange Datei gemeldet, und im Speicher steht der erste Sektor von Moni vielfach. Meine Vermutung: Einer der BDOS- oder BIOS-Einsprünge verändert (erlaubterweise) neuerdings Register. Grund für die Vermutung: schon Dijey reagierte sehr allergisch, als mein MountLib die Register beim BIOS-Read nicht rettete.

Anm.d.HzN: Einige der alten MONI-Versionen laufen in der Tat nicht richtig unter RAM 6.x. MONI 2.11 vom 4.8.88 läuft. Ggf. bei Olaf bestellen.

Boot-EPROM: (Hans Gras, NL-1506)

Michael Keßler's BOOT-Eproms haben nicht unbedingt Rahmen und Fenster! Mittlerweile gibt es auch Rahmenlose. Du erkennst Michael's Boot-EPROMs an einer Versionsnummer 9.x.

CTR40: (Hans Gras, NL-1506)

Eine weitere Möglichkeit für diesen Treiber wäre ein Grafik-Treiber. Ich habe eine 80x25 Karte ohne HRG (HiRes Grafik). Für Grafik ist der VS 4 mit seinen 192x256 Punkten und 16 Farben sicher nutzbar - auf jeden Fall besser als die 100x160 Punkte der 80Zeichen-Blockgrafik. Evtl. gar unter Verwendung der im BASIC-ROM enthaltenen Routinen, allerdings als Kopie derselben im RAM, da die ROM-Routinen die Speicherstellen oberhalb von FA00h benutzen.

Datei-Extensions: (Hans Gras, NL-1506)

.PIC Dateien, die Bilder enthalten. Je nachdem mal zum TYPEn auf den Bildschirm, und mal zum Drucken.

.Z3T ZCPR 3 Terminal Capabilities Definition Pakete.

Fragen & Antworten**Fragen & Antworten**

F: (Anton Reiser, 8977)

Seit längerer Zeit ist viel Wirbel um DTP (Schreibtisch-Verleger-Spiel). Das Standard-Ausgabeformat ist PostScript. Gibt es CP/M-Software mit PostScript-Druckertreiber oder -Ausgabe? Die Schwierigkeit dürfte der fehlende Grafik-Standard unter CP/M sein - denn ohne WYSIWIG (What you see is what you get - also mal sehen, wie es aussieht) kein DTP.

A: (Jan Brederke, 2000)

Zu Info 39-4: Hans Gras sagt, daß das Diskettenformat des "Kaypro 10" 128 Dir-Einträge hätte, im Format C5 aber nur 64 Dir-Einträge eingetragen seien. Das Format C5 heißt "Kaypro 4", und dies hat offensichtlich auch seinen Grund: Ein Freund von mir hat einen Kaypro 4. Gelegentlich tauschen wir Disketten, und er nimmt zur Zeit sein BIOS mit einem Disassembler auseinander. Daher habe ich mich mit dem Kaypro 4-Format bereits näher beschäftigt und kann sagen, daß dieses Format auf der Diskette Platz für 128 Dir-Einträge bietet, aber definitiv nur 64 davon nutzt. (Herbert hat extra dafür den Menüpunkt "Schweineformat" in FMT6.CHN eingebaut.) Falls also der Kaypro 10 wirklich alle 128 Dir-Einträge nutzt, müßten wir ihm ein eigenes Format spendieren (oder ihn ignorieren).

F: (Jan Brederke, 2000)

H LPC.COM und HLPQ.COM aus NEWHLP*.LBR von CLUB.902 sollen ja angeblich gesqueeze-te bzw. gecrunchte Help-Dateien lesen können. Bei mir gibt es nur Schrott, sobald ich einen Unterpunkt aufrufe, durch den eine weitere Help-Datei eingeladen wird.

F: (Hans Gras, NL-1506)

Ich habe eine Frage zur Format-Verwaltung mit FMT6. Wie basteln wir folgende Skew-Tabelle:

SKEW:	DB	11,12,17,18,23,24,29,30,3,4,9,10,15,16,21,22
	DB	27,28,1,2,7,8,13,14,19,20,25,26,31,32,5,6

Diese Tabelle brauche ich, um eine modifiziertes ITT3030-Format zu lesen. Ich habe das bislang durch manuelles Patchen des "Drive X:" auf dem KLIX-Heap geschafft. Automatisch ist mir das noch nicht einfach gelungen.

A: (Herbert zur Nedden, 2000)

Einfacher geht es, wenn Du das Laufwerk erst selektierst und dann den Skew-Table auf Bank 0 (Adresse steht im DiskParameterHeader) patchst. Das geht dann auch leicht per Programm. FMT6 kann Dir da nicht helfen.

F: (Hans Gras, NL-1506)

In dem o.g. Zusammenhang habe ich eine weitere Frage:

Utility-Funktion 64 (Search Block Name) hängt mein System auf, aber Funktion 65 meldet richtig, das Program "Drive X:" nicht da ist. Warum hängt sich RAM auf?

A: (Herbert zur Nedden, 2000)

Grund ist ein Fehler, der mit RAM 6.1 behoben ist.

F: (Hans Gras, NL-1506)

Und noch eines in Sachen Format-Verwaltung:

Es passierte mir einige Male, daß ein gewünschtes Format nicht in RAM 6.x eingebaut war. Also FMT6 starten, C(onfig), H(inzu), C(onfig)x:Format usw. Geht das nicht einfacher?

A: (Herbert zur Nedden, 2000)

Ja, und zwar mit dem neuen XCFIG6 (Extended CFIG6) von KCLICK.012. Es kann zwar nur ein Laufwerk zur Zeit bearbeiten, dafür aber ggf. das Format erst aus FORMATE.R6 laden.

Hardware: Tips

F: (Hans Gras, NL-1506)

Ich habe noch eine/n Unschönheit(e)/Fehler. Dieses Mal mit der Uhr auf dem FDX-OMTI Adapter von Gerhard Witzel. Die Hardware-Uhr läuft nicht! (Die Anzeige stimmt nicht, steht immer). So weit ich das Schaltbild zur Bildung des IO-Port 3Eh verstehe, müßte es klappen. Wer hat dafür eine Lösung um die Uhr wieder zu benutzen?

A: (Herbert zur Nedden, 2000)

Anscheinend ist das gelegentlich ein Problem. Häufig liegt es daran, daß vom 74LS373 oder 74LS374 zwei Pins vertauscht sind. Das einzige, was ich da empfehlen kann, ist, ein Testprogramm zu schreiben, welches a) auf Port 3Eh Daten ausgibt und mit einem Meßgerät zu prüfen, ob die Daten ankommen und b) von dort einliest und anzeigt und die einzelnen Eingänge mal an +5 Volt und mal an Masse anzuschließen. Also: Den Port testen!

Hardware-Tips

Info 37-3: (Uwe Sander's Problem)

(Anton Reiser, 6977)

Bei Kälte fand meine Floppy die Spuren auf der Disk auch nicht. Bis ich mal ein paar Tropfen Haushaltsöl auf die Führungsschienen des Schreib/Lesekopf-Schlittens träufelte. Ich habe säurefreies Öl genommen, in der Hoffnung, daß dieses das Platsik des Schlittens nicht angreift. Es hat geklappt! Seitdem bootet die schwarze Kiste auch bei 10 Grad (plus).

Bootprobleme:

(Anton Reiser, 6977)

Die RS232-Karte fest anlöten bringt's. Zwischendurch hatte ich immer wieder Schwierigkeiten. Bootprobleme und Abstürze. Alle Kabelverbindungen wieder und wieder überprüft. Kein Erfolg. Bis ich mich (endlich) dazu überwunden habe, die Karte anzulöten. Im Info stands ja schon lange.

3 1/2"-Laufwerke:

(Anton Reiser, 6977)

Manche Informationen im Info kann man nicht gleich richtig einordnen und geraten dann in Vergessenheit. Zum Beispiel war mir nicht klar, wie einfach es ist, ein 3 1/2 Zoll-Laufwerk anzuschließen. Für DM 158 ein solches im 5 1/4" Einbaurahmen erworben, stehen im Format 1C preiswerte 1,7 MB zur Verfügung. Da habe ich meinen Wunsch nach einer Festplatte wieder zurückgestellt. Nun bin ich dabei, mir eine Organisationsform für die Files zurechtzulegen. Der Möglichkeiten gibt es ja verschiedene, nur fehlt mir noch der Überblick.

Anm.d.HzN: Olaf Krumnow hat mit seinen Projekt-Aliasen aus Info 39 eine recht automatische Idee geliefert - sie läuft jedoch nur unter RAM 6.x.

Disc Change für Dumme: (Info 37-24)

(Hartmut Traber, 5270)

Ich überfliege immer zuerst mal das Info, dann folgt das Studium. So war ich zunächst mal geplättet angesichts dieser Überschrift.

Nach dem Studium (ich habe mittlerweile wieder ein SD 521 wegen 40-Spur-Formatieren usw.) war ich von der Idee angetan und fing an den LötKolben zu schwingen. Ich hatte ein 74, ein 376 war nicht schnell zu beschaffen, aber ein anderer Treiber tat es auch: Funktioniert einwandfrei auf Anhieb!

Und in diesem Zusammenhang, das Laufwerk war ja probeweise angeschlossen und in RAM 6 mit der Steprate 3 ms installiert, es läuft auch mit dieser Steprate einwandfrei.

Meine Bedenken, daß die Diskette ganz herausgezogen werden müßte, waren unbegründet. Drückt man "Push" und schiebt die Diskette dann wieder hinein, ist sie als neu angemeldet. Wie das allerdings bei schreibgeschützten Disketten aussieht, werde ich ganz bestimmt heute abend noch herausbekommen, bin allerdings schon jetzt sicher, daß dann nur ein echter Wechsel (ganz herausziehen) wirkt, und so soll es ja auch sein.

R A M 6.x: Extended Command Processoren (ECPs)**Extended Command Processoren (ECPs)**

(Herbert zur Nedden, 2000)

Der Name klingt toll, aber was sich dahinter verbirgt ist sicherlich nicht allen ganz klar. Zu deutsch heißt ECP Erweiterter Kommando-Prozessor. Toll, aber auch nicht besser in Bezug auf das Verständnis. Hier also mal ein Versuch, das Ganze etwas zu durchleuchten:

Unter CP/M sind Programme in Dateien mit der Extension .COM auf Diskette (oder RAM-Floppy oder Festplatte) abgelegt. Wenn CP/M (also kein laufendes Programm) auf die Eingabe eines Befehls wartet bin ich als Anwender gefordert dieses zu tun. Der Teil von CP/M, der diese Arbeit erledigt heißt übrigens CCP = Command Console Processor, also Konsol-Kommandoausführer. Einige Befehle, wie z.B. DIR oder ERA sind im CCP enthalten, d.h. werden ausgeführt, ohne, daß ein entsprechendes Programm (also DIR.COM oder ERA.COM) auf Diskette stehen muß. Gebe ich jedoch einen Befehl ein, der nicht in den CCP eingebaut wurde (z.B. OTTO), so sucht der CCP auf Diskette nach OTTO.COM. Dabei sucht der Original CP/M 2.2-CCP nur auf dem aktuellen Laufwerk, der neue unter RAM 6.x laufende CCP mit Namen ZCPR 3.3 sogar auf mehreren Laufwerken (nämlich auf denen, die im sog. Suchpfad stehen). Wird OTTO.COM gefunden, so wird es in den Speicher geladen und gestartet. Wenn nicht, erhalte ich i.a. die Meldung 'OTTO?', d.h. CP/M fragt mich, was OTTO sein soll. Gebe ich einen Befehl zusammen mit einer Laufwerksangabe (z.B. A:OTTO) an, so wird nur auf A: nach OTTO.COM gesucht.

Soweit so gut: Der CCP sucht nach dem entsprechenden .COM, wenn der angegebene Befehl nicht in ihm selbst enthalten ist.

Jetzt kommt eine geniale Idee von ZCPR ins Spiel. Statt bei einem vergeblichen Versuch, das .COM zum eingegebenen Befehl zu suchen, einfach klein beizugeben, stellt der ZCPR den Befehl 'CMDRUN OTTO' in seine Kommandozeile, und versucht diesen auszuführen. D.h. nun wird CMDRUN.COM gesucht und dieses Programm soll nun OTTO irgendwie ans Laufen bringen. Soetwas kennst Du schon: Die .SUB-Dateien. Diese kannst Du ja nur dadurch ausführen lassen, daß Du sie mit Hilfe von SUB.COM (oder XSUB.COM usw.) aufrufst: 'SUB OTTO' führt OTTO.SUB aus. Benenne ich also SUB.COM einfach in CMDRUN.COM um, so kann ich durch Eingabe von OTTO die Datei OTTO.SUB ausführen lassen, ohne das SUB vorne weg einzugeben - das tut dann der ZCPR für mich.

Also: Der ZCPR stellt vor einen nicht ausführbaren Befehl (d.h. ist nicht eingebaut und das .COM ist nicht zu finden) CMDRUN. Dieses Programm kann dann versuchen, den eingegebenen Befehl auszuführen.

Da CMDRUN (also in unserem o.g. Beispiel SUB) sozusagen eine Erweiterung von ZCPR ist, wird es Erweiterter Kommandoprozessor (ECP) genannt.

Für ZCPR 3.3 gibt es einige interessante ECPs:

1. **ARUNZ** sucht den angegebenen Befehl in seiner Datei ALIAS.CMD. Findet es ihn, so wird die zugehörige Kommandozeile gestartet. Diese Kommandozeile kann mehrere Befehle enthalten.
2. **XRUN** sucht entlang des Suchpfades nach einer Datei, deren Name ohne Extension der des angegebenen Befehls ist. Wird eine gefunden, wird die Extension der Datei angeschaut und, falls XRUN damit etwas anfangen kann, die zugehörige Kommandozeile gestartet. Anderenfalls sucht XRUN weiter.
3. **CHAINDU** kann Turbo-Pascal .CHNs ausführen. .CHNs sind Turbo-Pascal-Programme, die nicht alleine ausführbar sind, jedoch von einem anderen Turbo-Pascal-Programm aus gestartet werden können. (Siehe Info 37-9).

R A M 6.x: Extended Command Processoren (ECPs)

Üblicherweise sind alle drei der o.g. ECPs unter RAM 6.x (genauer unter ZCPR 3.3) aktiv! ARUNZ.COM wird in CMDRUN.COM umbenannt und folglich als erster ECP bei Erfolgloser Suche nach dem .COM gestartet. Findet ARUNZ den Befehl nicht in ALIAS.COM, so startet es XRUN. Findet XRUN den Befehl auch nicht, gibt es klein bei; findet es das passende .CHN, so startet es CHAINDU.

Beispiel: Ich gebe OTTO ein.

1. Ist OTTO im CCP (also bei uns ZCPR 3.3) fest eingebaut, wird OTTO ausgeführt.

Sonst:

2. Wird OTTO.COM entlang des Suchpfades gefunden, wird OTTO.COM in den Speicher geladen und gestartet.

Sonst:

3. Wird CMDRUN OTTO in die Kommandozeile gestellt. Folglich wird CMDRUN (also ARUNZ) gestartet. Findet es den Befehl OTTO in ALIAS.COM, wird die entsprechende Kommandozeile gestartet.

Sonst:

4. Wird XRUN OTTO in die Kommandozeile gestellt. Folglich wird XRUN gestartet. Findet es OTTO.??? mit einer geeigneten Extension entlang des Pfades, wird die entsprechende Kommandozeile gestartet. Falls so OTTO.CHN gestartet wird, lautet die Kommandozeile CHAINDU du:OTTO wobei du: das Laufwerk/User ist, auf dem OTTO.CHN gefunden wurde.

Sonst:

5. Gibt XRUN klein bei, meldet also z.B. Watt'n dat.

Übrigens werden weitere Parameter, die ich zusammen mit meinem Befehl eingebe, natürlich auch weitergegeben!

Damit das Ganze so funktioniert, müssen ein paar Kleinigkeiten stimmen:

Zuvor eine Anmerkung zum Suchpfad. Er besteht aus mehreren Directory-Angaben, nämlich denen, die abgesucht werden sollen. Das letzte Directory im Pfad heißt Wurzelverzeichnis. Es spielt eine besondere Rolle: CMDRUN (also unser ARUNZ) und ALIAS.COM sowie sinnvollerweise auch XRUN und CHAINDU müssen dort stehen! CMDRUN, ALIAS.COM werden nämlich nicht entlang des Pfades, sondern nur im Wurzelverzeichnis gesucht. Das hat den einfachen Grund, daß es schneller geht!

Abgesehen davon, daß also die o.g. Dinge im Wurzelverzeichnis zu finden sein müssen, müssen noch weitere Bedingungen erfüllt sein:

1. Die letzte Zeile in ALIAS.COM muß wie folgt lauten:
: xrun \$!
Damit wird nämlich CMDURN (also ARUNZ) dazu gebracht, XRUN aufzurufen, wenn es den angegebenen Befehl nicht findet.
2. In XRUN muß für die Extension .CHN folgende Kommandozeile definiert sein:
CHAINDU \$B\$!
damit es auch CHAINDU richtig startet (XRUN wird so ausgeliefert).

R A M 6.x: Archiv-Flag oder Archiv-Bit**Archiv-Flag oder Archiv-Bit**

(Herbert zur Nedden, 2000)

Dieses nützliche kleine Ding hat leider einen irgendwie doch verwirrenden Namen. Ich möchte versuchen, Euch zu erläutern, was es damit auf sich hat:

Unter CP/M besteht ein Dateiname aus 11 Zeichen: Dem eigentlichen achtstelligen Namen und der dreistelligen Extension. Da für den Dateinamen selbst nur 7 Bit je Zeichen erforderlich sind, kann das 7. Bit (man zählt die Bits ja von 0 an) für Systemzwecke verwendet werden. Diese 7. Bits werden Flags genannt. Einige haben einen Namen und werden dann auch oft statt Flag einfach Bit genannt.

Unter Original-CP/M 2.2 werden nur die Flags in den ersten beiden Zeichen der Extension verwendet: Das erste als R/O- und das zweite als SYS-Flag. Ist das R/O-Flag gesetzt, gilt die Datei als ReadOnly, d.h. NurLesen - sie ist schreibgeschützt. Ein gesetztes SYS-Flag kennzeichnet die Datei als System-Datei, was nichts anderes bedeutet, als das sie vom Befehl DIR nicht mit angezeigt werden soll. Es macht also die Directory-Anzeige übersichtlicher. Von unserem P2DOS, welches unter RAM 6.x läuft werden zwei weitere Flags verwendet: Das zweite des Dateinamens ist das Public-Flag. Ist es gesetzt, gilt die Datei als Public, d.h. existiert auf allen Usern. Das andere ist das, um das es hier geht: Das Archiv-Flag, welches im dritten Zeichen der Extension zu finden ist.

Das Archiv-Flag (oft auch Archiv-Bit genannt) kann von Programmen gesetzt werden. Das P2DOS löscht dieses Flag automatisch, wenn die Datei verändert oder neu angelegt wird. Und genau das ist der Witz dieses Flags. Wurde es gesetzt, kann man an ihm erkennen, ob die Datei seit dem geändert wurde. Daher wäre der Name Changed-Flag (also Geändert-Flag) etwas sprechender.

Wenn ich irgendwelche Dateien bearbeiten will, kopiere ich sie mir i.a. von meiner Sicherungs- (oder WasAuchImmer-) Diskette auf mein Arbeitslaufwerk, sprich die RAM-Floppy. Habe ich das, was ich tun wollte getan oder will ich sicherheitshalber mal zwischendurch sichern, kann ich natürlich einfach alles oder mit DiJey selektiv einige Dateien auf die Sicherungs-Diskette kopieren.

Holger Hansen hat zur Vereinfachung dieser Arbeit seinerzeit das Programm Back-ItUp geschrieben, welches die Zeiteinträge verglich, um zu entscheiden, welche Dateien auf dem Arbeitslaufwerk neuer als auf der Sicherungs-Diskette sind, um nur diese zu kopieren. Der Nachteil war, daß dabei die Gefahr besteht, Dateien, die garnicht auf die Sicherungs-Diskette gehören dorthin zu kopieren, da eine Datei neuer als keine Datei ist - schließlich kann die Datei ja neu angelegt worden sein.

Mit Hilfe des Archiv-Flags geht das etwas besser. Ich hole alles, was ich brauche auf das Arbeitslaufwerk und setze bei allen Dateien das Archiv-Flag, d.h. kennzeichne sie als 'unverändert'. Ändere ich nun das eine oder andere, so löscht das P2DOS für mich das Archiv-Flag der betroffenen Dateien, kennzeichnet sie also als 'verändert'. Zum Sichern kopiere ich nur die Dateien mit gelöschtem Archiv-Flag auf die Sicherungs-Diskette. Damit kopiere ich nur die veränderten und die neu angelegten Dateien, aber nicht die, die ich von einer anderen Diskette geholt habe - es sei denn ich habe auch die verändert.

Zum Kopieren kannst Du z.B. ACPY oder mein neues QCC verwenden. Entscheidend sind die Optionen: Nach dem Kopieren soll das Archiv-Flag der Dateien auf dem Arbeitslaufwerk gesetzt werden.

```

auf's Arbeitslaufwerk: /d   Setze Archiv-Flag der Ziel-Dateien
auf Sicherungs-Disk:  /nos  Kopiere nicht-Archivierte Dateien, überschreib
                        evtl. existente Dateien und setze anschließend
                        das Archiv-Flag der Quell-Dateien

```

BradfordBradford - Watndatn ?

(Peter Würfel,7262)

Wer, so wie ich, nur den 'alten' DMX80 besitzt, wird ab und zu den Wunsch verspüren, auch mal was 'schöner' auszudrucken. Wenn dieser Wunsch nur 'an und ab' auftaucht, dann bietet sich dazu das Druckprogramm Bradford an (dauernd damit zu arbeiten ist nicht empfehlenswert, denn der Ausdruck eines Textes mit Bradford dauert schon seine Zeit.)

Wo gibt's was ?

Und wer also Bradford sucht, der wird in unserer PD-Bibliothek gleich fünfmal fündig: auf CLUB.027, CLUB.035, CLUB.808, CLUB.053 und auf CLUB.054.

Die Version auf CLUB.027 können wir gleich wieder vergessen, es handelt sich hier um die alte Version 1.20x. (Der Editor für Bradford Zeichensätze auf CLUB.034, BRADEDI?.COM ist für die Zeichensätze dieser alten Version gedacht (und hat außerdem bei mir unter RAM 4.x nie funktioniert)).

Auf CLUB.808 findet man in der Library BFUPDAT1 zwei Versionen von Bradford: BRADFORD.COM und BF2SMALL.COM. Auf meinem System (unter RAM 6.1) läuft nur BF2SMALL (der Aufruf von BRADFORD wird mit der Meldung, daß nicht genügend Speicher vorhanden ist, abgebrochen) und meldet sich als Version 2.02 (140-column-version).

Auf CLUB.035 findet man die Version 2.04 (100-column-version). Was die zwei Hundertstel Versionsunterschied zwischen Vers. 2.02 und 2.04 ausmacht, ist nirgends dokumentiert. Da ich mit einer Seitenbreite von 100 Zeichen bisher immer auskommen bin, arbeite ich mit der Version 2.04 und die weiteren Aussagen beziehen sich immer auf diese Version auf CLUB.035. Das Programm UNFONT.COM auf CLUB.808 ist mit dem gleichnamigen Programm auf CLUB.035 identisch. Das Programm BRADFONT.COM auf CLUB.808 unterscheidet sich an 2 Byte (23FDh und 23FEh) von dem auf CLUB.035. Auch hier arbeitete ich bisher fehlerfrei mit der Version auf CLUB.035. Die Zeichensätze auf 808 und 035 sind mit der Ausnahme von GREEK.BF2 identisch, wobei ein Vergleich zeigt, daß es sich bei dem GREEK-Zeichensatz auf CLUB.035 um die neuere Version handelt.

Auf CLUB.053 findet man zwei Libraries mit insgesamt 16 Zeichensätzen, die von Wolf-Dieter Zimmermann 'eingedeutscht' wurden. Die namensgleichen Zeichensätze in den beiden Libraries unterscheiden sich zwar, wenn man sie mit DIFF.COM vergleicht, wenn man sie jedoch mit UNFONT bearbeitet und dann die *.BFF-Dateien vergleicht, stellt man fest, daß diese identisch sind. Auch im Ausdruck unterscheiden sich diese namensgleichen Zeichensätze nicht. In einem Telefongespräch erklärte mir Wolf-Dieter, er habe halt (quasi als Abschiedsgeschenk an den CLUB) das, was er vor einem Jahr mal gebastelt hat, zur freien Verwendung an Herbert für die PD-Bibliothek gegeben.

Auf CLUB.054 findet man fast alle Zeichensätze, die auf CLUB.035 zu finden sind mit deutschen Umlauten: Von 42 US-Zeichensätzen auf CLUB.035 wurden 38 von mir 'eingedeutscht'. Der 'Klammeraffe' (ASCII-Code 64) wurde nur bei einigen Zeichensätzen durch das Paragraphenzeichen ersetzt. (Da ich das Paragraphenzeichen normalerweise nicht brauche, habe ich mir beim Eindeutschen der Zeichensätze diese Arbeit gespart). Außerdem findet man auf dieser Diskette noch einen neuen Zeichensatz 'FRAKTUR', drei von PIZZA2C abgeleitete Zeichensätze, sowie 5 Hilfsbildschirme für den KLICK.

Der langen Rede kurzer Sinn:

Vergeßt CLUB.027, CLUB.808 und CLUB.053 und arbeitet mit CLUB.035 und CLUB.054

Let's work together

Und nun könnte die Arbeit mit Bradford lustig losgehen, wenn da nicht eine kleine Klippe zu umschiffen wäre: Die Dokumentation auf CLUB.035 ist **HOLLÄNDISCH!**

Bradford

Damit ich nicht mißverstanden werde, ich habe nichts gegen diese Sprache, aber sie zu verstehen, das fällt mir halt sehr, sehr schwer! Nun, im Schweiß meiner Füße hab' ich so nach und nach herausklabüstert, was man mit BRADFORD alles machen kann, und da ich ein vergeßlicher Mensch bin, hab ich mir so einiges in 5 KLICK-Bildschirme geschrieben, die ich gerne weitergeben möchte.

Hier nun noch ein paar Ergänzungen, die nicht so ohne weiteres diesen Bildschirmen zu entnehmen sind:

CONFIG.BF

Zuerst braucht man, will man sich einen Text ausdrucken lassen, BRADFORD.COM und eine Datei CONFIG.BF. Die Datei CONFIG.BF erzeuge man durch Umbenennen aus einer der drei Dateien FX. (für Epson-FX), MX. (für Epson-MX) oder ST. (für Star-Drucker), je nachdem welchen Drucker man ansteuern möchte. Wer 'nen DMX80 hat, wähle die Datei MX.

Diese Datei sieht wie folgt aus: *Das ist Kommentar:*

```

.. epon mx configuration:
.de2                diese drei Zeilen dürfen
.pf2                auf keinen Fall
.pcq                geändert werden !
.. general configuration:
.st1
....                hier folgen nun noch weitere Befehlszeilen.

```

Die 'general-configuration-Zeilen' können nach den eigenen Bedürfnissen geändert werden. Bei mir sehen diese Zeilen wie folgt aus:

```

.st1
.. deutscher Zeichensatz:
.fndpiza
.. Zeichenzwischenraum:
.ai3
.. linker Rand:
.lm8
.. rechter Rand:
.rm76
.. oberer Rand : keiner, Druckbeginn in der ersten Zeile,
.. da Bogen mit Oberkante unter dem Andruckbügel des DMX80:
.tm0
.. unterer Rand:
.bm9
.. Blattlänge 72 Zeilen insgesamt, 63 Zeilen nutzbar:
.pl72
.. In-Line Kommandobuchstabe Ö nach # ändern:
.dcc#
..Ende

```

Wenn nun Bradford (mit den entsprechenden Parametern) aufgerufen wird, sucht es sein CONFIG.BF und übernimmt daraus die Grundinstallation.

BRADFORD aufrufen

Im Gegensatz zum alten Bradford 1.20x wird BRADFORD **zusammen** mit der zu druckenden Textdatei aufgerufen, wobei mehrere Textdateien, die hintereinander gedruckt werden sollen, verkettet werden können. Beispiel: ich möchte zuerst 'test1.txt', dann sofort anschließend 'test2.txt' ausdrucken lassen, also rufe ich auf:

Bradford

```
BRADFORD test1.txt test2.txt
```

Außerdem können Steuerdateien im Aufruf genannt werden, die den Druck der folgenden Textdateien steuern. Beispiel:

```
BRADFORD nw.bf test1.txt
```

Es können in der Kommandozeile Optionen übergeben werden. Diese Optionen werden durch das '/'-Zeichen eingeleitet. Wenn mehrere Optionen gewünscht werden, muß zwischen den einzelnen Optionen ein Leerzeichen stehen und der /-Strich wiederholt werden. Ich möchte z.B. Text1 in Draft-Qualität drucken und erwarte eine Pause nach jeder Seite, dann gebe ich ein:

```
BRADFORD test1.txt /D /P
```

(falsch wäre: BRADFORD test1.txt /DP !!)

Den Ausdruck steuern

Die Kommandos, die die Arbeit von Bradford steuern, kann man grundsätzlich auf zweierlei Art und Weise in eine Textdatei schreiben: entweder als 'Punktkommando' wie in NewWord in eine eigene Zeile oder auch innerhalb einer Zeile (sog. In-Line-Kommando).

Punktkommandos:

Wenn eine Zeile in der ersten Spalte mit einem Punkt beginnt (genau wie bei NW), dann wird diese Zeile als Befehlszeile interpretiert. In dieser Zeile kann dann genau ein Befehl stehen. Beispiel:

```
.fnchart
```

Durch diesen Befehl wird der Zeichensatz 'chart' gewählt.

Die Bezeichnung 'Punktkommando' ist übrigens nicht ganz korrekt, denn es ist möglich, ein anderes Zeichen als den Punkt als Steuerzeichen zu definieren (siehe KLICK-Bildschirm 5). Idee.d.HzN: 'Line-Kommando' oder gar 'Out-Line-Kommando'

In-Line-Kommandos:

In der 'Urversion' von BRADFORD2 wird ein 'In-Line-Kommando' durch den Backslash (Ö) eingeleitet. Da ich aber dieses Zeichen im deutschen Zeichensatz als 'normalen' Buchstaben brauche, habe ich in meinem CONFIG.BF als 'In-Line-Kommando'-Zeichen den Gartenzaun (#) definiert (s.o.). Ein In-Line-Kommando ist nun wie folgt aufgebaut: #<Befehl><Leerzeichen>. Das Leerzeichen gehört zwingend zum In-Line-Befehl. Wenn ich z.B. folgende Zeile ausgeben möchte:

Dies ist ein Test!

dann gebe ich in NW diese Zeile wie folgt ein (^ steht für CTRL):

```
Dies ist ein ^S^BT^Best^S!
```

Im Bradford-Dokument sieht diese Zeile wie folgt aus:

```
Dies ist ein #u11 #dk3 T#dk1 est#u10 !
```

Makros (oder Macros ?)

Das obige Beispiel ('Dies ist ein Test!') zeigt, wie unübersichtlich so eine Zeile mit In-Line-Kommandos aussehen kann. In NW hat man ja die Möglichkeit mit <CTRL><O>+<D> die CTRL-Kommandos auszublenden. Wäre es nicht schön, BRADFORD könnte die CTRL-Kommandos von NewWord verstehen? Es kann, man muß es ihm nur beibringen! Dazu werden die Makros verwendet.

Bradford

Bradford kann bis zu 26 Makros verwalten. Diese Zahl kommt dadurch zustande, daß jedem Makro ein Name zugewiesen werden muß und als Name kommen nur die Buchstaben A-Z in Frage.

Was ein Makro ist, und wie man damit umgeht, läßt sich am besten an Beispielen darstellen:

Das erste Beispiel soll zeigen, wie man sich durch ein Makro Tiparbeit ersparen kann. Ich denke da z.B. an Heinrich Bölls 'Dr. Murkes gesammeltes Schweigen' (wer's nicht kennt, unbedingt besorgen und lesen, hat nix mit Computern, sondern mit literarischem Grundwissen für Schmunzler zu tun!). Diese Kurzgeschichte handelt von einer Rede, in der zig-mal die Formulierung 'jenes höhere Wesen, das wir verehren' vorkommt. Diese Formulierung kann ich nun einem Makro zuweisen; dazu muß dieses Makro definiert werden, in unserem Beispiel sieht die Definition wie folgt aus:

```
.mdA "jenes höhere Wesen, das wir verehren"
```

Damit ist dem Makro mit dem Namen 'A' ein bestimmter Text zugewiesen. Die Begrenzer (hier: ") sind beliebig, sie dürfen nur nicht im Makrotext vorkommen, die Zeile könnte somit auch so aussehen:

```
.mdA βjenes höhere Wesen, das wir verehrenβ
```

Wann wird nun dieses Makro ausgeführt? Dazu gibt es zwei Möglichkeiten: Entweder rufe ich dieses Makro mit dem Befehl 'xA' auf (als Punktkommando .xA alleinstehend am Anfang einer Zeile, oder als In-Line-Kommando #xA) oder ich weise einen Buchstaben diesem Makro zu,

Wenn also nun in meinem Text steht:

```
...und nicht nur #xA , sondern auch....
```

dann wird, wenn dieser Text mit Bradford ausgedruckt wird, daraus:

```
...und nicht nur jenes höhere Wesen, das wir verehren, sondern auch...
```

Die zweite Möglichkeit, dieses Makro in einem Text zu verwenden, setzt einen Zuweisungsbefehl zu einem 'Auslöser' voraus. Von diesen Zuweisungsbefehlen gibt es zwei verschiedene. In unserem Beispiel möge dieser Befehl so aussehen:

```
.mcA§                oder so:                .mnA64
```

Er bedeutet, in beiden Fällen, daß immer dann, wenn im Text das '§'-Zeichen (ASCII-Code 64 dezimal!) erscheint, nicht dieses Zeichen, sondern der Makrotext ausgegeben wird. Nun reicht also folgende Textzeile:

```
...und nicht nur §, sondern auch....
```

um beim Ausdruck die Zeile

```
...und nicht nur jenes höhere Wesen, das wir verehren, sondern auch...
```

zu erhalten.

Bradford: Befehlsübersicht

Jetzt ist natürlich das §-Zeichen für nichts anderes mehr zu gebrauchen, es sei denn, die Zuordnung zum Makro wird durch

```
.mc0A                oder                .mc064
```

wieder aufgehoben.

In Makros können jedoch nicht nur 'normale' Texte, sondern auch Bradford-Kommandos stehen. Eine Makro-Definition z.B. für 'Auf Tastendruck warten' (in NW ist das ^C = <CTRL><C>) sieht dann wie folgt aus:

```
.mdC "#wk "
```

Und wenn man nun mit dem Befehl

```
.mnC3
```

dieses Makro dem Zeichen mit dem ASCII-Code 3 zuweist, dann unterbricht Bradford immer dann sein Druckwerk, wenn im Text ein ^C auftaucht, bis der geneigte Bediener eine Taste zu drücken geruht.

Wenn man auf diese Art und Weise Umschalten zum Unterstreichen (^S in NW) programmieren möchte, dann muß man sich klar machen, daß ^S ja eigentlich zwei Bedeutungen hat, nämlich beim erstenmal "Unterstreichen einschalten" und beim nächstenmal "Unterstreichen ausschalten" und NW zählt immer mit, was jetzt gerade an der Reihe ist. BRADFORD kann das auch! Die Makrodefinition und die Zuweisung zu ^S sieht dann so aus:

```
.mnS19 Underline
.mdS "#u11 ö#u10"
```

Was zeigt uns dieses Beispiel? Erstens: .mn... kann vor .md... stehen, zweitens: auf den Zuweisungsbefehl kann ein Kommentar folgen, drittens, das kleine 'ö' ist die Markierung, die das Makro in zwei Teile aufteilt; wenn das Makro zum erstenmal aufgerufen wird, wird der linke Teil verwendet, beim zweitenmal der rechte, beim drittenmal der linke usw. und viertens: in der Makrodefinitionszeile ist ein FEHLER (einer, der einem leicht unterkommt), sie muß nämlich so aussehen:

```
.mdS "#u11 ö#u10 "
```

Wer das kleine 'ö' als Textzeichen im Makrotext braucht, muß es halt undefinieren, den Befehl dazu findet Ihr im KLICK-Bildschirm 5.

Und damit man sich nicht immer die Arbeit machen muß, in jedes Dokument die Makros zu schreiben, kann man sie alle in eine Datei zusammenfassen und kann ihr z.B. den Namen NW.BF geben. Man hat dann eine Datei, die ich oben bei meinen Ausführungen zum Aufruf von BRADFORD 'Steuerdateien' genannt habe.

Bradford - Befehlsübersicht

Anm.d.HzN: Peter hat die folgende Befehlsübersicht als KLIX-Overlay sowie als NewWord-Dokument auf CLUB.054 in PD gegeben. Ich habe hier im Info die Trennung in die Bildschirm-Seiten herausgenommen. Peter verweist oben gelegentlich auf Bildschirm 5; das ist der letzte Teil dieser Übersicht.

B r a d f o r d: Befehlsübersicht

Aufruf: [LW:]datei [[LW:]datei2...] [/option1[/option2...]]

Optionen: Zahl d. Kopien	/Cxxx		
Pause zw. d. Seiten	/P	Erste Seite	/Axxx
Draft-Qualität	/D	Letzte Seite	/Zxxx

Textgestaltung:

Zeichensatz wählen	.FNdateiname (ohne .BF2)		
Breite der Zeichen	.HMx	x=0..2..9	
Abstand der Zeichen	.AIX	x=0..3..99	
Fettdruck	.DKx	x=1..4 (^B=DK3)	
Hochstellen	.SUP	(=an)	.SU0 (=aus)
Tiefstellen	.SUB	(=an)	.SU0 (=aus)
Unterstreichen	.UL1	(=1x)	.UL2 (=2x) .UL0 (=aus)
Proportional	.MOO	(=ein)	.MO1 (=aus)
Zeilenvorschub	.NL		
Überschreiben	.OV		
Tabulator	.TA		

Zeilen gestalten: #Sta[b]

a = C	Zentrieren	= R	Rechtsbündig
= L	Linksbündig	= J	Blocksatz
= H	Halber Blocksatz	= A	Ausgeglichener Blocksatz (#RJ Reset Ausgleichszähler)
b = 1	Nur eine Zeile	= C->A	Letzte Absatzzeile

Seite gestalten:

Linker Rand	.LM[+,-]xx	xx = 0...79	xx/ 10 Inch	[ML,PO]
./.	.LXxxx	xxxx = 0..1920	xxxx/240 Inch	
Rechter Rand	.RM[+,-]xx	xx = 0...80	xx/ 10 Inch	[MR]
./.	.RXxxx	xxxx = 0..1920	xxxx/240 Inch	
Oberer Rand	.TMxxx	xxx = 0...254	xxx*1/6 Inch	[MT]
Unterer Rand	.BMxxx	xxx = 0...254	xxx*1/6 Inch	[MB]
Seitenlänge	.PLxxx	xxx = 0...254	xxx*1/6 Inch	
Zeilenabstand	.SPb	b = 1	normal	1/6 Inch
		= 2	doppelt	1/3 Inch
		= H	ein+1/2	1/4 Inch
		= Q	eng	3/24 Inch

Grundeinstellung in meinem CONFIG.BF: [(TOP0/BOTTOM9/LEFT8(11)/RIGHT76(73)]

Seitenumbruch:

Neue Seite	.PA	(kein In-line-Befehl)
Restzeilen	.CPxxx	Umbruch, wenn weniger als xxx Zeilen

Header / Footer:

Header definieren	.HE<Text und/oder Befehle>	(kein In-line-Befehl)
Footer definieren	.FO<Text und/oder Befehle>	(kein In-line-Befehl)
H-Zeile festlegen	.HLxxx	xxx = 1...2..254 xxx=0 : kein Header
F-Zeile festlegen	.FLxxx	xxx = 1..64..254 xxx=0 : kein Footer
Schrifttyp	.FFfontname	
Header- bzw. Footer-Zeilen innerhalb der MT- bzw. MB-Zeilen !		

Bradford: Befehlsübersicht**Benutzerhilfen**

Kommentarzeile: .. (wie NW)
 In-Line-Kommentarzeile #XX (keine Textausgabe, Befehle aktiv)
 Bildschirmausgabe .DM text (nur Ganz-Zeilen-Kommando)
 Tastendruck abwarten .WK

Makros:

definieren .MDb "<text1>[!<text2>]"
 b: A..Z Name des Makros
 : Space ein Leerzeichen
 ": zwei beliebige Begrenzungszeichen, die nicht
 in <text1> und/oder <text2> vorkommen dürfen.
 ! : ö "Toggle"-Markierung

ausführen .XMb b = A..Z

Auslöser zuweisen .MCba b = A..Z a = ASCII-Zeichen
 .MNbxxx b = A..Z x = ASCII-code (Dez)

zurücksetzen .MCOa bzw. .MCOxxx

Makros in NW.BF vordefiniert:

A	^A	"Fraktur" (*)	R	^R	Standard "dpiza"
B	^B	Fettdruck (3x) (*)	S	^S	Unterstreichen (*)
C	^C	Druckpause	T	^T	Superscript (*)
D	^D	Fettdruck (2x) (*)	U	31	Soft hyphen
E	^E	"Italic" (*)	V	^V	Subscript (*)
I	^I	Tabulator	W	^W	"dPizaW" (*)
L	^L	Seitenvorschub	X	^X	"dPiza2H" (*)
N	^N	"Select" (*)	Y	^Y	"dPiza2" (*)
Q	^Q	Seitenbreite (*)	Z	141	NW-Blocksatz (*)=Toggle

Nummerieren:

Seiten-Nummern werden beginnend mit 1 durchgehend gezählt
 #PN aktuelle Seiten-Nummer ausdrucken (nur In-Line)
 .PNxxxx xxxx = 0..9999 Seiten-Nummer setzen

Kapitel-Nummern Zähler wird nicht automatisch gesetzt
 #CN Kapitelnummer ausdrucken (nur In-Line)
 .CNxxxx xxxx = 0..9999 Kapitel-Nummer setzen
 .CN+ Kapitel-Nummer um eins erhöhen

Abschnitt-Nummer wie Kapitel-Nummer
 #SN .SNxxxx .SN+

Steuerzeichen ändern:

.-Kommando am Zeilenanfang .DCLx x = neues Zeichen
 \-Inline-Kommando .DCCx ./ (ö=\ ; neu:#)
 !-"Toggle"-Zeichen .DCTx ./ (ö=!)
 nächstes Steuerzeichen drucken .PR

Hardware: 768k-Karte / 512k-Hauptplatine**768k-Karte / 512k-Hauptplatine**

(Herbert zur Nedden, 2000)

Hier findest Du meine Unterlagen zu diesen beiden Aufrüstungen. Bevor Du dich an den Nachbau machst, geh in Dich (insbesondere bei 512k auf der Hautplatine)! Diese Arbeiten erfordern einiges an Geschick und Routine - vor allem beim Auslöten der alten ICs! Wenn Du dabei Leiterbahnen zernagelst, dann hast Du ein Problem. Auch ein kleiner Lötkecks und es läuft evtl. nicht! Ich habe übrigens keine große Lust, Lötversuche zu debuggen, denn das ist eine Saurarbeit!

Packt Dich der Mut, dann kannst Du von mir das entsprechende PROM erhalten.

768k-Karte

(Herbert zur Nedden, 2000)

Bekanntlich kann sich jeder, der Interesse hat, bei mir seine 32k-Karte aus dem Grundgerät auf 768kB aufrüsten lassen. Damit diese Masse an Speicher auf die kleine Platine paßt (es sind 24 IC's HM50256 o.ä. erforderlich), müssen acht davon huckepack auf ebensolche gelötet werden. Die restlichen acht kommen in die i.a. auf der Karte eh schon vorhandenen Sockel. Und so entsteht auch die sogenannte '512k-Options'-Karte, was nichts anderes bedeutet, daß die Platine für 768kB aufgerüstet wurde, jedoch die acht Sockel leer sind, so daß lediglich 512kB Speicher da sind - mit der Option der einfachen Aufrüstung auf 768kB.

Auf der folgenden Seite findet Ihr den Schaltplan dieser Karte. Bei der kleineren Karte (32k bis 512k), die nur mit zwei RAM-Bänken, d.h. zwei Sätzen zu acht RAMs auskommen, liefert das PROM an seinen Pins 9 und 10 das Bank-Enable. Da ich zwei Banks und zwei Enable-Pins habe, liefert das PROM je nach Sachverhalt keinen oder GENAU ein Enable. Bei drei Banks (3x 8 RAMs) brauche ich logischerweise drei Bank-Enables. Mit zwei Bit geht das auch gut, indem ich die Bank binär kodiere. Da die Elektronik aber einzelne Bank-Enables benötigt, wird ein Binärde-koder 74LS155 (das ist so etwas ähnliches, wie der 74LS138 aus meinem Artikel über Ports aus dem vorherigen Info) nachgeschaltet. Er macht aus dem 2-Bit Binärsignal vier einzelne Signale, von denen wenn, dann GENAU ein Signal auf Low liegt. Ansonsten ist diese Schaltung mit der der 512k-Karte identisch.

Für die mutigen Bastler: auf den darauffolgenden drei Seiten habe ich meine Verdrahtungs-Zeichnungen beigelegt. Zuerst die Trennungen, d.h. das, was so an Leitungen verschwinden muß und danach die Legungen, also das, was neu muß. Auf der Zeichnung 'Legungen Oberseite' ist der Block links der 74LS08 mit dem huckepack darauf sitzenden 74LS155 und das IC links ist eines der RAMs, die huckepack oben sitzen; es steht stellvertretend für die acht oberen. Die vier Zahlen im Kreis bezeichnen Leitungen, die von der Ober- zur Unterseite verlaufen (oder umgekehrt, falls Dir das mehr zusagt).

512k auf der Hauptplatine

(Herbert zur Nedden, 2000)

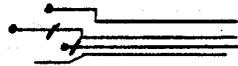
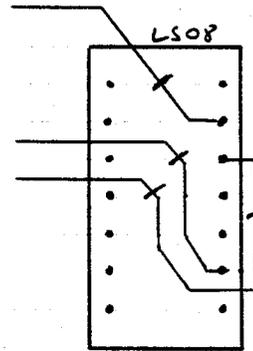
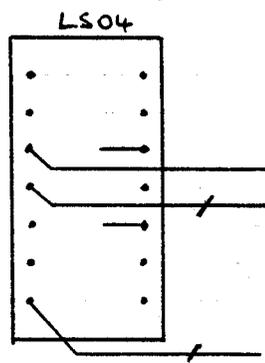
Für ganz Mutige (denn wenn's nicht tut, tut gar nichts mehr) auch meine Unterlagen zur Aufrüstung der Hauptplatine auf 512kB. Die eigentliche Schaltungstechnik ist die der 512k-Erweiterung auf der 32k-Karte - allerdings sitzen die RAMs jeweils Huckepack und die PROM-Pinbelegung wurde verändert, damit es sich besser lötet.

Das PROM wird hier Huckepack auf das PAL 14L4 (links auf der Hauptplatine) gelötet, der LS08 und der LS00 auf die ICs östlich bzw. nord-östlich gelötet, wie aus der Zeichnung zu ersehen.

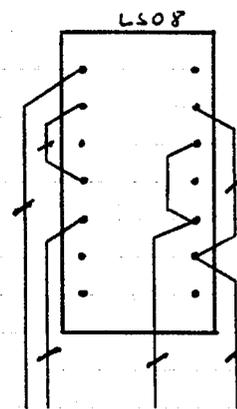
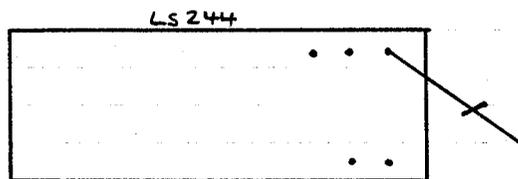
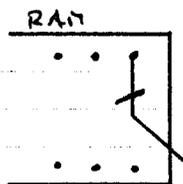
Übrigens solltest Du die Hautplatine nur dann aufrüsten, wenn Du eine SDX hast! Die Aufrüstung der 32k-Karte ist WESENTLICH problemloser!

Hardware: 768k-Karte Trennungen

768k Trennungen Untersseite

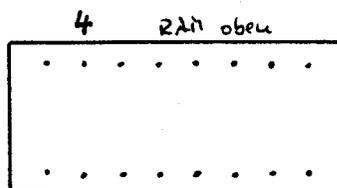
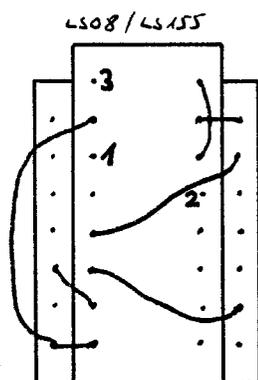


768k Trennungen Oberseite



Hardware: 768k-Karte Legungen Oberseite

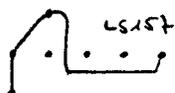
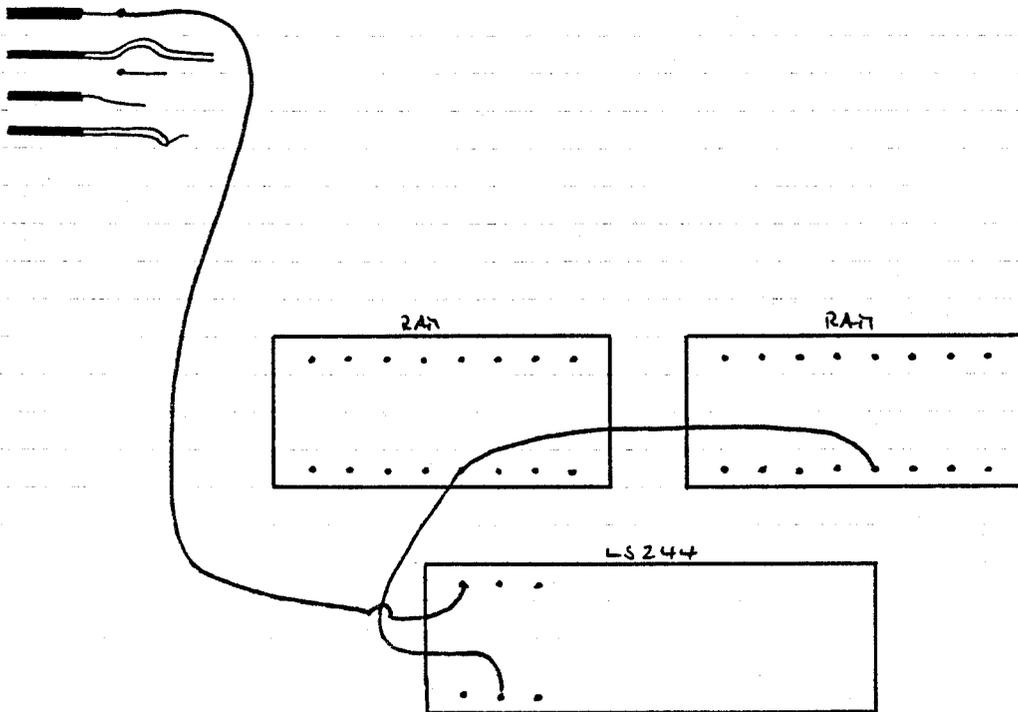
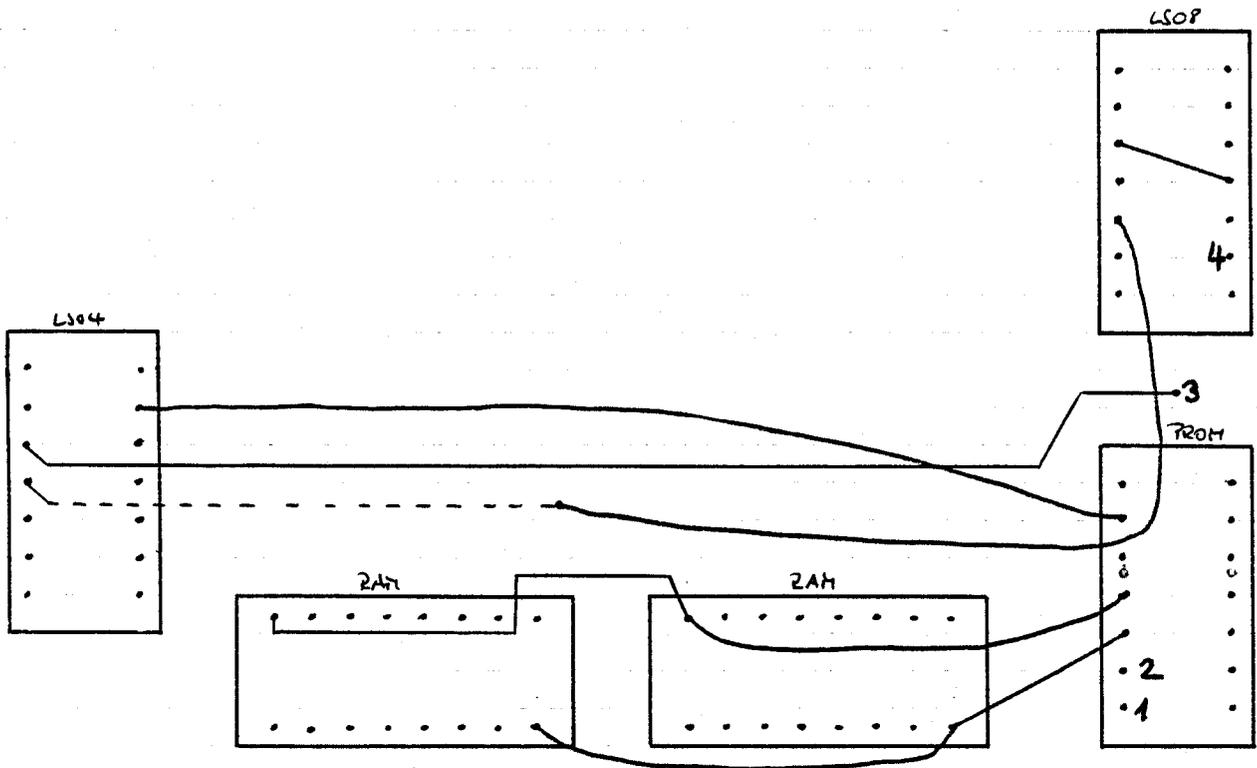
768k Legungen Oberseite



für 512k-Optionskarte:
 die beiden Drähte, die quer über
 Rücken des LS155 laufen nicht
 "parallel" sondern über Kreuz.

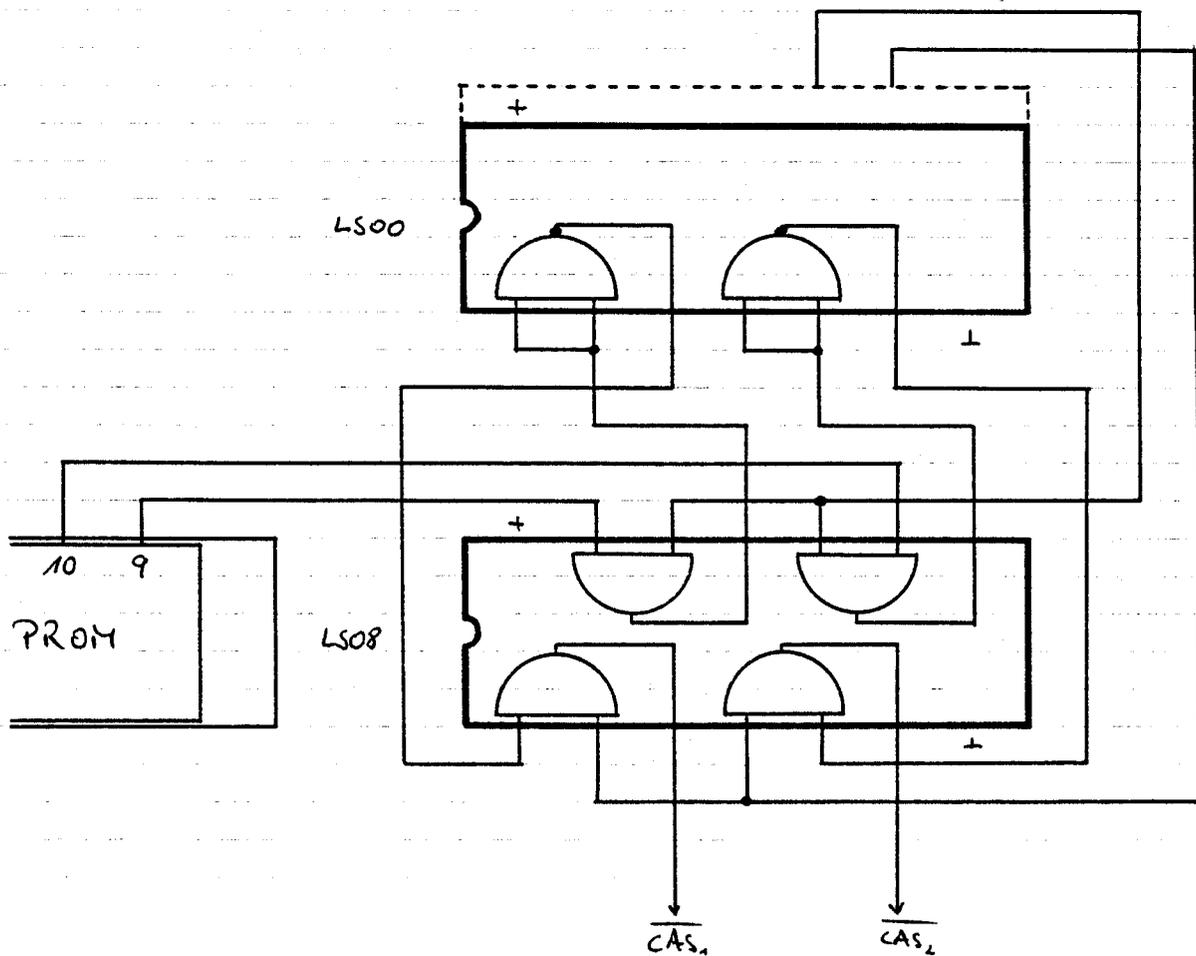
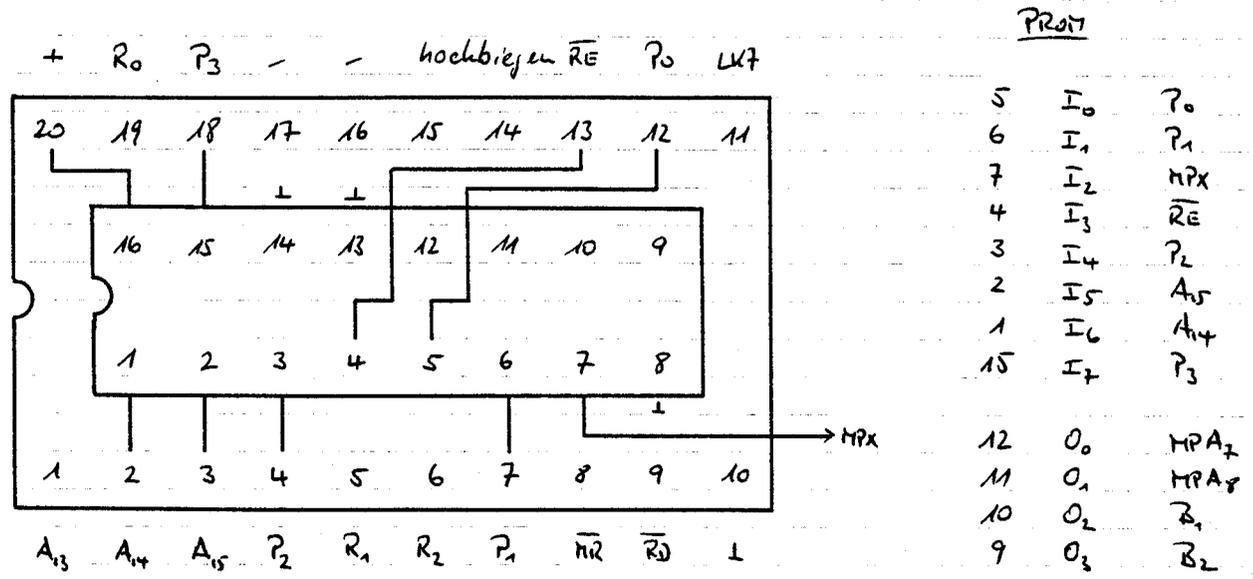
Hardware: 768k-Karte Legungen Unterseite

768k Legungen Unterseite



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

Hardware: 512k-Hauptplatine Layout



(Pin 15, RA04s)

Hardware: 512k-Hauptplatine Lötanweisungen

512 KB- HAUPTPLATINE

- 1 ALLE RAMs auslöten
- 2 IC 6C (LS157): Pin 4 trennen
- 3 IC 5B (LS00): Pins 8, 9, 10 trennen
- 4 PAL: Pins 14, 15 trennen
- 5 PROM auf PAL verdrhten und einsetzen
- 6 RAMs stapeln, dabei Pins 15 einzeln hochbiegen (\overline{CAS})
- 7 RAMs einlöten und Pins 1 (MPA8) sowie Pins 15 ($\overline{CAS_1}$, $\overline{CAS_2}$) durchschleifen
- 8 Auf IC 5B (LS00) einen LS00, auf IC 6B (LS04) einen LS08 aufsetzen und Pins 7, 14 (L, +) verlöten, restliche Pins hochbiegen
- 9 Neuen LS00, LS08 verdrhten
(gestrichelt: IC 5B (unterer LS00)).
- 10 RAM, Pin 9 mit PROM, Pin 12 verbinden (MPA7)
RAM, Pin 1 mit PROM, Pin 11 verbinden (MPA8)
- 11 IC 2B (LS04 o. LS14), Pin 3 mit PAL, Pin 8 verbinden (\overline{MREQ})
- 12 IC 6C (LS157), Pin 1 mit PROM, Pin 7 verbinden (MPX)

Wenn ROM-BASIC-Betrieb alle Speicher erkennt, d.h. bei nicht-RESET-festem ROM A in OFA7AH eine OFH steht, aber bei Disketten-Boot kein RAM-Test-OK: RAM-Timing anpassen!

H a r d w a r e: 8"-Laufwerke / 12 Volt-Bräter**Zweiseitige 8"-Laufwerke oder noch 'e dSM in 1990!**

(Hans Gras, 1506)

Leider has Herbert vergessen, eine (sehr) alte Frage von mir über 8" Laufwerke ins Info zu packen. Ich habe nichts in den Infos gefunden was mir weiter helfen konnte, so löste ich das Problem selbst.

Die Frage war: Bei 2-seitigen 8" Laufwerken kann ich nur 1 Seite ansprechen.
Warum? Und wie lösen wir das Problem?

Die Antwort läßt sich fast raten: Auch 1990 entdeckten wir noch dSM (die bekannte dunkle Stunden von Memotech). Dieses Mal war der Fehler auf dem Floppy-Disk-Controller-Board - und zwar an einer Stelle, die anscheinend von keinem benutzt wird. (Oder haben wir Mitglieder nur 1-Seitige 8-Zöller?) Connector J2 (fürs 8-Zoll-Flachbandkabel) hat Pin 35 (Memotech-Numerierung) nicht angeschlossen. Bei 8-Zoll-Laufwerken wird auch eben dieser Pin 35 als Side Select genutzt. Nicht nur Pin 37 wie im FDX Technical Manual vermerkt ist. Das ist seit 1978 (Neunzehnhundert-Achtundsiebzig) bekannt. Z.B. beim Morrow Decision und Exidy Sorcerer. Ich habe dafür eine 8-Zoll Floppy-Controller-Disk-Board von MD genau angesehen. Guess What? Diese Platine heißt "DIJEY 2". Das bedeutet das ein Fragezeichen in der Software-Liste (beim SIG/M 235) auch gestrichen werden kann. Diese Hersteller haben das Pinning-Problem einfach mit einer Lötbrücke zwischen den genannten Pins gelöst. Auf unserer Platine müssen wir es mit einer Drahtbrücke lösen weil eine schmale Leiterbahn zwischendurch geht.

Der Bräter: 12 Volt-Lastwiderstand

(Hans Gras, NL-1506)

Beim meinem (vor)letzten Umbau habe ich den Bräter-Widerstand (39 Ohm, 4 Watt) von der "BackPlane" (FDX-Busplatine) entfernt, weil er in meinen Augen keinen nützlichen Zweck mehr hat. Es war zwar im Info vermerkt, daß dieser Widerstand als Last für den 12 Volt-Zweig dient, aber ich habe mehrere Dinge an den 12 Volt hängen, also ist reichlich Last vorhanden. FALSCH! Ich habe zwar das Schaltnetzteil etwas hochgeregelt, weil meine 5 Volt rund 4,6 Volt waren, aber meine 12 Volt waren ohne Lastwiderstand 12.0 Volt. So ein Bißchen Hochregeln schadet nichts. Wie das geht ist bereits mehrmals im Info vermerkt. Die Spannungen sind Heute 4.8 und 12.2 Volt. Aber beim Anlauf der Floppies und/oder beim Steppen des Kopfes senken sich die 12.2 auf 11.5 Volt, aber die 4.8 gehen auf 5.0 Volt hoch, da das Schaltnetzteil primair taktet, und daher alle Spannungen gemeinsam beeinflußt werden. Diese Schwankungen von 4.8 - 5.0 - 4.8 Volt sind auf der 80-Zeilen-Karte gut zu sehen. Das Bild flackert sehr stark. Das heißt, daß der Lastwiderstand NICHT entfernt werden darf! Ich habe einen neuen 39 Ohm, 9 (neun) Watt Widerstand an die FDX-Rückseite mit einem Bügel montiert und über 2 Drähte an einen Floppy-Stecker gelötet. Kein Gefflacker mehr auf dem Schirm. Ich habe dieses Phänomen mit 2 Memotech-Usern besprochen und was hat Memotech uns wieder ins Nest gelegt? Ja, wieder eine dSM. Es gibt doch FDX-Stationen ohne diesen Widerstand so, bei denen der Schirm von Anfang an geflackert hat.

>>>> Für alle Leute mit einen Flacker-Schirm: Widerstand irgendwo montieren wo die Wärme nicht schadet und an den 12 Volt-Zweig hängen! Aber wenn Dein Schirm nicht flackert, auch keinen Widerstand montieren.

Anm.d.HzN: Normalerweise sitzt dieser Widerstand auf der FDX-Busplatine unten neben der FDX-Außenwand auf der Seite der Platine, auf der auch die Stecker sitzen - er heizt folglich munter die Platinen. Die einfachste Abhilfe ist es, den Widerstand einfach auf die andere Platinenseite zu löten. Soll er doch die Abluft, die einem eh nur die Finger kühlt etwas aufwärmen...

S o f t w a r e: Quick Copy 'n Compare / C P / M: BDOS-Dateifunktionen

BDOS-Funktion OPEN oder Warum MFT nicht mehr als ca. 548kB kopiert und Quick Copy 'n Compare (Herbert zur Nedden, 2000)

Es gibt zwei verschiedene recht praktische Kopierprogramme - zumindest in der ZCPR 3.3-Umgebung, d.h. unter RAM 6.x. Das erste ist MFT zusammen mit dem Dateivergleicher MFC, die auch unter CP/M laufen. Der Aufruf ist:

```
A>MFT [Ziel:=][Quelle:][Dateien] [Dateien] [Dateien] ...
```

Dabei werden die Dateien (können auch Wildcards enthalten) von dem Quell-Direktory (Laufwerk/User) auf das Ziel-Direktory (ebenfalls Laufwerk/User) kopiert. MFC hat die selbe Syntax, vergleicht jedoch statt zu kopieren. Das Schöne an MFT ist, daß ich mehrere Dateiangaben angeben kann. Alternativ gibt es ACOPIY 2.2, welches unter RAM 6.x (genauer ZCPR 3.3) läuft. Dieses Programm wird wie folgt aufgerufen:

```
A>ACOPY [Ziel:=][Quelle:][Dateien] [Nicht-Dateien] [/Optionen]
```

Ich kann hier zwar nur eine Dateiangebe machen, um zu sagen, was kopiert werden soll, aber dafür kann ich zusätzlich eine weitere Dateiangebe machen, um Dateien vom Kopieren auszuschließen, also z.B. alles außer den .COMs kopieren. Interessant wird es eigentlich durch die Optionen. Ich kann ACOPIY sagen, daß nur Dateien kopiert werden sollen, die das Archiv-Bit nicht gesetzt haben. Ach so Archiv-Bit: Das ist ein Bit im Dateinamen ähnlich wie das System- oder Read/Only-Bit, dessen Witz ist, daß das BDOS von RAM 6.x dieses Bit löscht, wenn die Datei verändert wird. Will ich also die veränderten Dateien kopieren, sage ich ACOPIY, daß ich nur die mit nicht gesetztem Archiv-Bit kopieren will. Zusätzlich kann ACOPIY nach dem Kopieren und ggf. auch Vergleichen das Archiv-Bit der Quell- und/oder Ziel-Dateien setzen. Klarerweise kopiert ACOPIY auch unsere Zeiteinträge mit (zumindest das ACOPIY 2.2P, KLICK-PD).

Und ich wollte nun so eine Art eierlegende Wollmilchsau: Ein Programm, welches so, wie MFT/MFC kopieren und vergleichen kann und so wie ACOPIY das Archiv-Bit und die Zeiteinträge behandelt. Das Ergebnis heißt QCC (Quick Copy 'n Compare) und ist auf KLICK.012. Aufruf:

```
A>QCC [Ziel:=][Quelle:][Dateien] [Quelle:][Dateien] ... [/Optionen]
```

QCC kann von einen Laufwerk auf ein anderes (oder das selbe) Dateien kopieren, wobei auch die User-Angabe '?', d.h. Wildcard zulässig ist. Stelle ich vor eine Quell-Dateienangabe ein 'B', so heißt das, 'die Dateien nicht'. Durch die Angabe von Optionen kann ich QCC auch nur kopieren oder nur vergleichen lassen, das Archivbit so wie ACOPIY verwalten lassen und die Bildschirmausgabe auf ein sinnvolles Maß reduzieren.

Für die Kopier-Routine in QCC habe ich folgenden Algorithmus verwendet:

```
LeseSchleife:   Öffne Datei
                 Lies bis Dateiende oder Puffer voll
                 Wenn Puffer nicht voll und nicht letzte Datei,
                   dann gehe zu LeseSchleife
```

```
SchreibSchleife: Wenn neue Datei, dann lege Datei an
                  sonst öffne Datei
                  Schreib soviel, wie gelesen wurde
                  Schließe Datei
                  Wenn nicht Puffer-Ende und nicht letzte Datei,
                    dann gehe zu SchreibSchleife
```

S o f t w a r e: Quick Copy 'n Compare / C P / M: BDOS-Dateifunktionen

Daß die Leseschleife in etwa so aussieht, ist wohl klar. Die Dateien brauche ich nicht zu schließen, da ich sie nicht verändert habe. In der Schreibschleife muß ich die geschriebene Datei natürlich schließen. Das tue ich auch dann, wenn die Datei nicht ganz in den Puffer gepaßt hat, ich also den nächsten Teil in der nächsten Runde einlese und schreibe, sprich die Datei portionsweise kopiert wird, weil das sicherer ist, zumal ich laufend Laufwerk/User zwischen Quelle und Ziel umschalte. Folglich muß ich die Datei, wenn ich die nächste Portion schreibe auch wieder öffnen, ist's eine neue (also die nächste Datei), dann lege ich sie halt an.

Dieser Algorithmus funktioniert auch so! Tückisch wird es erst, wenn ich eine Datei kopieren will, die groß ist. Eine 720kB-Datei war auf dem Ziel-Laufwerk nur 548kB klein und QCC wurde und wurde nicht fertig! Nach langem Suchen fand ich heraus, was los war. Die BDOS-Funktion OPENFILE (Funktions-Nr. 15) ist Schuld! Sie setzt die Extend-Group immer auf 0. Und jenseits der 512kB-Linie beginnt bei unseren üblichen Diskettenformaten die Extend-Group 1! Und das steht auch so in einem CP/M-Handbuch! (dSDR = dunkle Stunde von Digital Research ?)

Da z.B. ACOPY diese Dateien fehlerfrei schafft, habe ich mal getraced, was ACOPY so an OPENFILE-Aufrufen macht: KEINE!

Folglich habe ich mir genauer angesehen, was OPENFILE eigentlich macht. Es macht nichts anderes, als den passenden Direktory-Eintrag zu suchen und den FCB damit zu füttern. Mehr nicht! Folglich habe ich meinen Algorithmus etwas vereinfacht:

LeseSchleife: Wenn neue Datei, dann öffne Datei
 Lies bis Dateiende oder Puffer voll
 Wenn Puffer nicht voll und nicht letzte Datei,
 dann gehe zu LeseSchleife

SchreibSchleife: Wenn neue Datei, dann lege Datei an
 Schreib soviel, wie gelesen wurde
 Schließe Datei
 Wenn nicht Puffer-Ende und nicht letzte Datei,
 dann gehe zu SchreibSchleife

Also öffne ich jede Quell-Datei nur ein Mal. Die Ziel-Dateien werden nur angelegt (und damit automatisch einmal geöffnet) und nie wieder geöffnet - und das, obwohl ich sie zwischendurch evtl. mal schließe. Allerdings habe ich daher getrennte FCBs fürs Lesen und Schreiben. Schließlich stehen da die für's Lesen bzw. Schreiben erforderlichen Informationen drin. Die entfallenen Datei-Öffnungen waren völlig unnötig, da sie eh nichts am FCB veränderten - außer dem o.g. Verhunzen der Extend-Group.

Die 548kB kommen dadurch zu Stande, daß in den ersten Runden ca. 500kB kopiert wurden, und in der letzten 48kB dazukamen. Danach gings wieder ans Lesen, und beim erneuten Öffnen der Datei durch BDOS die Extend-Group auf 0 gesetzt, d.h. statt hinter 548kB wurde bei 36kB weitergelesen und natürlich auch geschrieben. Irgendwann wurde die 512kB-Grenze wieder überschritten und es ging von vorne los usw.

Es ist immer wieder erstaunlich, was einem so für Steine in den Weg gelegt werden, nicht wahr?

S o f t w a r e: StopCB4 / A s s e m b l e r: Tastatur

StopCB4 und Tastenabfrage in Assembler

(Herbert zur Nedden, 2000)

Jan Brederke hat ein kleines Progrämmchen gebastelt, welches auf Tastendruck die laufende Kommandozeile abbricht und ggf. gar von B: bootet. Den Zweck des Programms beschreibt er wie folgt:

Zweck des Programms

"Wenn bei mir Laien von der Diskette mit den Basic-Spielen rebooten wollen, weil sie ein Spiel zuende gespielt haben und in mein Basic-Menü-Programm zurückwollen, das wie die Spiele auf dem 40-Zeichen-Schirm läuft, müssen sie nur alle unbeschrifteten Tasten gleichzeitig drücken, also die beiden Resettasten und die Leertaste.

STOPCB ist außerdem auch sehr nützlich, wenn ich zu Testzwecken gelegentlich eine ältere Betriebssystemversion booten will."

Ist während des Laufes von STOPCB die Leertaste gedrückt, so wird in das Boot-EPROM gesprungen und ein Coldboot auf Laufwerk B: ausgeführt. Allerdings werden nur die Boot-EPROMs V 1.4 und V 1.6 von Bernd Preusing unterstützt. Jan hat allerdings den Source beigefügt, so daß Du STOPBC auch andere Boot-EPROMs beibringen kannst.

Mittels STOPBC kannst Du jedoch nicht nur mal gelegentlich von B: booten, sondern auch Aufruf von RAMxx oder nach dessen Aufruf das Einbauen der Klixen verhin-derbar machen.

Die Abfrage der Tastatur wird direkt über die Ports gemacht, um irgendwelche Tastaturpuffer zu umgehen. Neugierig, wie ich war, habe ich gleich mal in den Source geschaut, um mir die Tastaturabfrage anzusehen, da Jan erwähnte, daß er die Tastatur direkt über Ports abfragt, damit nichts dazwischenkommt.

Hier die Abfrage, ob die Leertaste gedrückt ist:

```

DI                ; erst mal die Interrupts abschalten
LD      A,0111111b ; Nur Bit 7 auf 0, alle anderen auf 1
OUT     (5),A      ; Das auf Port 5 (Tastatur) ausgeben
IN      A,(6)      ; und anschließend Port 6 (auch Tastatur)
                ; einlesen, um zu sehen, was los ist
EI                ; jetzt können die Interrupts weiterlaufen
BIT     0,A        ; Ist die Leertaste gedrückt?
JP      NZ,NOSPACE ; nein
                ; oder doch

```

Diese Routine prüft nur, ob die Leertaste gedrückt ist. Wenn Du an einer anderen Taste interessiert bist, dann kannst Du entweder in Info 4 (ich glaube jedenfalls, es ist Info 4 - nur liegen meine alten Infos auf dem Dachboden meiner Schwiegereltern, da ich einiges dort zur Vereinfachung meines baldigen Umzuges abgestellt habe) oder auf CLUB.004 im Listing des Tastaturtreibers nachsehen, oder etwas experimentieren. Ggf. muß Du aber statt Port 6 für einige Tasten Port 5 einlesen. Bei der Ausgabe auf Port 5 darfst Du aber immer nur ein Bit auf 0 setzen.