

MTX *User-Club Deutschland*

Info 25
01.02.1988

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

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

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

Verpflichtungen: Einsendung unseres Anmeldeformulars.

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

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

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

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

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

Kontaktadressen: (nach PLZ geordnet)

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

Telefon-Sprechzeiten

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

Inhaltsverzeichnis**C L U B**

Lesenswertes	Seite 1
Bestellaktion	Seite 2
Redaktionelles	Seite 3
Wer tut Was / Ports	Seite 4
Kleinanzeigen	Seite 5
Korrektur & Nachtrag	Seite 6
Fragen & Antworten	Seite 6
Fragebogen-Auswertung	Seite 9

C P / M unter R A M 4 . x

Probleme	Seite 10
----------	----------

S o f t w a r e

Neuigkeiten	Seite 11
KLICK.xxx PD's	Seite 15

L e s e r b r i e f

Hartmut Traber, 5270	Seite 16
Jan Bredererke, 2000	Seite 17
Hagen Wenzek, 5300	Seite 36

H a r d w a r e

8 MHz-Erfahrungen von Horst Kupka (4019)	Seite 19
Einige Anregungen	Seite 21
Reparatur	Seite 25
Interrupts und ECB-Bus	Seite 32
80-Zeichen und VS4 angepaßt	Seite 72
RAM-Hochstaperei	Seite 73

D o m e s D O S

Benchmark	Seite 37
-----------	----------

N e w W o r d

Spooler	Seite 38
Druckertreiber von CLUB.009	Seite 38
Bildschirmattribute	Seite 41

S u p e r C a l c

Bildschirmformat	Seite 42
------------------	----------

P a s c a l

Numerische Darstellung von Meßwerten	Seite 49
--------------------------------------	----------

d B A S E

'ne Macke	Seite 52
Bitoperationen	Seite 54

S u p e r T a p e

Für MTX	Seite 57
---------	----------

A s s e m b l e r

Kurs	Seite 61
------	----------

K o m i k

CYBER-Mitteilung Nummer 7411	Seite 24
Computerersetzer Hund	Seite 58
Aufbauorganisation	Seite 75
Reklame - mal anders	Seite 76

Preis für dieses Info: DM 12,-

Redaktionsschluß für Info 26: 29. Februar 1988

Liebe Leserin, lieber Leser,

es ist soweit, das Info Nummer 25 ist in Deinen Händen. Die erste Überweisung auf das Club-Konto 34 8000-200 wurde am 01.08.84 gutgeschrieben. Das erste Info hatte erstaunliche 18 Seiten, was ich damals für viel hielt. Ich fragte mich seinerzeit auch, wie ich es überhaupt schaffen sollte, regelmäßig mindestens 10 Seiten über den MTX zu fabrizieren. Nun, Dank Eurer Hilfe hat es sogar geklappt wesentlich mehr als diese 10 Seiten alle 6-8 Wochen zusammenzubringen. Daß sich der MTX User-Club Deutschland so rasch zu einer Größe von zeitweilig über 450 Mitgliedern mausern würde haben Thomas Wulf, Martin Hofmann und ich zum Zeitpunkt der Gründung nicht erwartet. Euch allen unseren herzlichen Dank, und 'Macht weiter so!'.

Ich wurde wieder gefragt, wie ich eigentlich Infobeiträge am liebsten haben möchte. Nun, mir ist es eigentlich ziemlich gleich, ob die Beiträge auf Papier oder Diskette kommen, solange die Zeilenlänge von 70 Zeichen eingehalten wird, damit ein vernünftiger Rand zum Lochen bleibt! Wenn Ihr allerdings schon ausgedruckte Texte schickt, dann verwendet doch bitte möglichst ein kräftiges Farbband, bei dem die Buchstaben alle vollständig in gleicher Helligkeit auf das Papier kommen. Wenn die Vorlagen mit schlechten Farbbändern gedruckt worden sind, muß ich die in einigen Fällen sogar abtippen, da sich Fotokopierer und insbesondere Druckmaschinen da anstellen - schließlich soll das gedruckte Resultat im Info gut lesbar sein!

Freizeit ist sicherlich eine der schönsten Dinge. Wenn man nun wie Ulrich Hönisch und ich voll im Beruf stehen, wird diese doch deutlich eingeschränkt. Uwe Grass steuert jetzt hart auf sein Diplom zu, um dann auch ins Berufsleben überzuwechseln. Daher ist leider unser Spielraum in Sachen Freizeit deutlich kleiner geworden. Die Folge dieses Umstandes ist, daß wir unsere Aktivitäten bezüglich der Umbauten und Reparatur eingeschränkt und teurer werden. Damit Ihr Euch ggf. selbst helfen könnt, findet Ihr in diesem Info eine Reihe von Hinweisen und Hilfestellungen unter dem Motto 'Hilfe zur Selbsthilfe'. Obendrein sind auch RAMs, EPROMs etc. seit dem letzten Einkauf teurer geworden, was sich auch ungünstig auf die Preise ausgewirkt hat.

In diesem Info findet Ihr auch die Auswertung des Fragebogens. Leider habe ich anscheinend die Fragen 1-4 nicht richtig gestellt oder erläutert. Die Antworten von Fragen 1 & 2 (Anwendungen und Programmiermittel) sowie 3 & 4 (Geplantes/Interessantes und Probleme) waren häufig nicht klar getrennt. Ich glaube, daß ich daran selbst schuld bin. Nichtsdestotrotz habe ich mich an die Arbeit gemacht, diese Bögen alle auszuwerten, wobei ich dann Fragen 1 & 2 sowie 3 & 4 mehr oder weniger Zusammengefaßt habe. Ach so, was so die Probleme der Abgrenzung waren: dBASE kann man eigentlich nicht anwenden, jedoch mittels dBASE eine Adreßverwlatung machen - schließlich ist dBASE lediglich eine Datenbank-Programmiersprache mit Interpreter.

Wettbewerb: Ich habe ein kleines Spiel geschrieben, bei dem es darauf ankommt, durch Verschieben von Steinen eine bestimmte Position zu erreichen. Das klingt einfach, ist es aber nicht. Wer mir als erster einen Lösungsweg zukommen läßt (es gilt das Datum des Poststempels!), der gewinnt den 'Topf'! Wer am Wettbewerb teilnehmen möchte zahlt DM 3.-, und erhält eine 5 1/4"-Diskette mit dem Spiel, welches auf der 80Zeichen-Karte gespielt wird. Falls Du keine geeignete Diskettenstation hast, schicke ich Dir den Spielplan samt Anleitung auf Papier. Ohne den Einsatz von DM 3.- nimmst Du natürlich nicht teil! Von diesen DM 3.- je Teilnehmer gehen DM 1.- in den Topf. Als Michael Möwe dem MTX User-Club Deutschland ade gesagt hat, hat er mir sein Restguthaben von DM 25.- für den Club vermacht, und diese sind schom im Topf!

C L U B: Bestellaktion anläßlich des Jubiläums-Infos Nummer 25:

Im Laufe der Zeit trudeln bei mir immer wieder Anfragen und Bestellungen von älteren Infos und Diskettenlaufwerken ein. Da ich bei TEAC-Diskettenlaufwerken jedoch erst ab drei Laufwerken Mengenrabatt erhalten kann, versuche ich immer zu warten, bis mindestens drei Bestellungen zusammengekommen sind. Werden ältere Infos (1-10) bestellt, so muß ich diese einzeln fotokopieren.

Daher bin ich der Meinung, daß es angebracht wäre, dies mit einer Sammelbestellung für mich zu vereinfachen, und Euch so bessere Angebote machen zu können.

Der beiliegende Bestellzettel beinhaltet nur TEAC-Diskettenlaufwerke. Diese sind zum einen preiswerter als EPSON-Laufwerke zu bekommen, und zum anderen sehr gut und recht leise! Da ich die Erfahrung gemacht habe, daß der eine oder die andere von Euch gerne PD's haben möchte, und mit Diskettenlaufwerken und Infos eine DIN A4-Seite noch reichlich Platz läßt, habe ich auch die PD's aufgenommen.

Wie sieht das Angebot aus, fragt Ihr sicherlich.

Folgendes Angebot gilt nur für diesen Bestellzettel, und damit nur bis zum 01. Mai 1988!

1. Diskettenlaufwerke:

Die Preise sind incl. MWst, zzgl. Porto/Verpackung, und können nach dem jetzigen Stand der Dinge, insbesondere wenn mindestens 3 Laufwerke zusammenkommen, unterboten werden!

TEAC FD55 BR : DM 240,-
5 1/4 ", 40 Spur, 500 kB unformatiert,
Formate 03, 09, ...

TEAC FD55 GFR : DM 275,-
5 1/4 ", 80 Spur, 1000 & 1600 kB (d.h. 1 & 1.6 MB) unformat.,
Formate 03, 07, 09, 0A, ...
umschaltbar auf HighDensity für 8"-Formate: 13, 1B, ...

TEAC FD135 HFN : DM 275,-
3 1/2 ", 80 Spur, 1000 & 2000 kB (d.h. 1 & 2 MB) unformatiert,
Formate 03, 07, 09, 0A, ...
umschaltbar auf HighDensity für 8"-Formate: 13, 1B, ...
vermutlich werden wir ein dann größeres Format basteln

2. Infos

Infos 1 - 20 je DM 6,- Infos 21 - 24 je DM 9,-.
Ab einer Info-Bestellung von DM 25,- gebe ich 5 % Rabatt, ab DM 50,- 10 %, ab DM 75,- 15% und ab DM 100,- 20 %.
Zuzüglich Porto & Verpackung

3. PD's außer KLICK.xxx

je DM 9,-; ab 4 Stück je DM 7,-; ab 10 je DM 6,-

4. KLICK.xxx-PD's

je DM 15,-

Spätestens Anfang Mai 1988 werde ich die Bestellungen bearbeiten: Laufwerke bestellen und Infos kopieren. Wenn's nicht mit dem Teufel zugeht, sollten die bestellten Waren bis Ende Mai alle bei Euch eingetroffen sein.

C L U B: Redaktionelles**Betrifft beiliegende Angebotsliste des MTX User-Club Deutschland**

Im letzten Info stand auf Seite 1:

WENN (Du hast kein Angebot in Liste)
u.s.w.

Das bedeutete, daß Ihr Eure Angebote aus der Angebotsliste bestätigen solltet, und daß alle nicht bestätigten Angebote - sei es nun Hardware oder Software, lösche. Dies ist offensichtlich auch geschehen.

Patch - Public-Domain-Diskette

(Herbert zur Nedden, 2000)

Da ich vor habe, eine Public-Domain mit allen (un)möglichen Patches zusammenzustellen, bitte ich Euch mir dabei zu helfen, indem Ihr mir mal so zukommen laßt, was Ihr so alles gepatcht habt. Die Patches, die im Gesamt-Info-Inhaltsverzeichnis aus Info 23 stehen in der Rubrik 'Reparaturen, ...', in den Infos 23, 24 und diesem auftauchen, oder auf einer der CLUB.xxx-PD's sind, werde ich übernehmen. Aber es ist sicherlich so, daß auch unveröffentlichte Patches herumschwirren oder auch im G-I-I evtl. einige fehlen, die in den ersten 22 Infos genannt sind. Auf der Patch-PD werden dann diverse Patch-Programme, in Turbo-Pascal oder Turbo-Modula geschrieben mit Quelltext stehen. Dabei werde ich natürlich alle Patches für ein Programm wie NewWord mit einem Patch-Programm erledigen, ggf. mit einem Frage-Antwort-Dialog, um jedem individuellen Wunsch entgegenzukommen.

Anmerken möchte ich noch, daß diese Patcher dann davon ausgehen, daß sie mit ungepatchten installierten Programmen gefüttert werden (Installation ist kein Patch!). Wer also auch kleine Sonderpatches in einem Programm vorgenommen hat, läuft Gefahr, daß es zu Kollisionen kommt, falls diese mir nicht bekannt sind.

C L U B: Wer tut Was / Ports**Wer tut Was**

Allgemeines	H. zur Nedden
Info-Inhaltsverzeichnis	H. zur Nedden
(FDX-)BASIC	A. Viebke
BASICODE	H. Gras
CP/M System	H. zur Nedden
Assembler	H. Oppmann
NewWord	U. Grass, H. zur Nedden
Turbo-Pascal	O. Krumnow, T. Wulf
Modula	O. Krumnow, H. zur Nedden
SuperCalc	W. Gieger
Edicta-Grafik	H. zur Nedden, C. Löhrmann, C. Romanazzi
Hardware	H. zur Nedden, P. Kretschmar, U. Hönisch
Reparatur	U. Hönisch, H. zur Nedden, U. Grass

Wer sich auf dieser Liste fehlt am Platz oder vermißt fühlt ... schreibe mir. (Bitte nur ernstgemeinte Zuschriften, d.h. Ihr solltet im genannten Bereich "firm" sein).

Ports (zur Nedden, 2000)

<u>Bereich</u>	<u>Port</u>	<u>Verwendung</u>
MTX	00 - 0F	Grudgerät
	10 - 14	SDX-Floppy-Controller!
	18 - 1B	8255-PIO-Box
	1F	vorgesehen für Cassettenmotorsteuerung
FDX	30 - 33	80-Zeichen-Karte
	38 - 39	6845-Controller der 80-Zeichen-Karte
	40 - 47	FDX-Floppy-Controller
	50 - 5F	4x Memotech SiliconDisc
	70 - 73	EPROM/SRAM-Floppy von J. Marquart und F. Cröll
ECB	80 - 83	EDICTA Grafik-Karte
	8B - 8B	Reserviert für HardDisk
	9B - 9B	c't RAM-Floppy
	A0 - A3	EDICTA RAM-Floppy
	A4 - A7	c't EPROM-Floppy
	AB - AB	c't SRAM-Floppy
	BB - BB	Conitec-Floppy
	BC - BF	Conitec-Floppy
	C0 - C4	Reserviert für Testzwecke !!!!!
	CC - CF	Janich & Klass Programmer
FB - FB	HD 64180 Sub-Prozessor-Karte	

Falls jemand etwas bastelt, und dafür dann Ports belegen möchte, den bitte ich mir diese Pläne möglichst frühzeitig mitzuteilen, damit wir es vermeiden können, daß plötzlich zwei Dinge an der selben Adresse liegen, oder Ports aus einem falschen Bereich verwendet werden. Die adressierbaren Port-Bereiche sind:

MTX	00 - 1F
FDX	20 - 7F
ECB	80 - FF.

Dabei müßt Ihr natürlich beachten, daß in der Tabelle oben einige schon verwendete Port-Adresen genannt sind, die Ihr daher nicht nutzen solltet.

C L U B: Kleinanzeigen**V E R K A U F**

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

(Preise sind ohne Porto & Verpackung, ich gebe ggf. Mengenrabatt)

- Ich vermittele jederzeit gebrauchte/neue Geräte und Teile der selben. Außerdem weiß ich i.a. was es wo am billigsten gibt.
 - Ich habe Apple-Communication-Software: Software für Rechnerkopplung Computer mit einem Apple. Das sind zwei Disketten (1x MTX, 1x Apple), die ich ggf. verleihe, da ich die Apple nicht kopieren kann.
 - Interface für Olympia-Carrera, in eigenem Gehäuse, kann neben die Schreibmaschine gestellt werden. DMX 80-Kabel kann zum Anschluß verwendet werden. 100%-ig Centronics-Kompatibel, also auch für andere Computer geeignet: DM 100,-
 - Rikadenki Plotter RY21, VB DM 1750,-
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, verschieden gestrichelte Linien, Textausgabe in 4 Richtungen und verschiedenen Größen, absolute und relative Koordinaten, Markierungen auf Linien und Graphen.
Kann in Textmodus geschaltet werden, um als Drucker zu fungieren.
 - Netzteil für Bastler (d.h. einige Anschlüsse müssen nach außen geführt werden), längsgeregelt, im Gehäuse, 3-fach Netzsteckdose mit Schalter, nachweislich Kurzschlußfest; DM 75,-
2 Trafos 8V/6A -> 5V/10A;
1 Trafo 15V/18A (mit Zwischenabgriffen) -> 12V/4A,
1 weiterer Trafo 18V/1A.
 - Einbau-Drehspulmeßgerät 0-50uA: DM 15,-
 - Elektor-HiFi-Sinusgenerator, 10Hz - 100kHz, 1,10,100mV, incl. Netzteil, aufgebaut und funktionstüchtig, ohne Gehäuse mit Elektor-Bauanleitung: DM 40,-
 - Taxan Grünmonitor, für 80/40 Zeichen gut geeignet, schwarzes Gehäuse: DM 100,-
 - DMX80-Drucker mit Originalkarton: DM 350,-
- Solange der Vorrat reicht:
- Z80A CPU: DM 2,-, Z80A DART: DM 7,10, Z80A CTC: DM 2,-,
Z80B CPU 3,-, 64k dynamische RAMs 4164 200ns: 8 Stück DM 15,-
 - Platinenstecker für Erweiterungen links am MTX-Grundgerät. Natürlich mit dem Gegenstück zu der Kerbe an Pin 5. je DM 4,-
 - Dynamische RAM's 32k x 1 Bit: 8 Stück DM 1.50
 - Statische RAM's 2k x 8 Bit (6116): je DM 2,-
 - ITL-IC's: 74LS175, 74LS368, 74LS21, 74LS173, 74LS158,
74LS258 je DM 0.50; 74LS10, 74LS11 je DM 0.30
 - Original-Memotech-Spielecassetten: (MENGENRABATT)
Toado, Kilopede, Knuckles, Draughts, Reversi, Snappo, Blobbo, Utilities, Demo für je DM 4.-; StarCommand, Goldmine, je DM 6.-
15 Leercassetten, mind C10 mit 2 Boxen für 5 Cass: Zusammen DM 15.-

Uwe Grass, Wachholtzstr. 8, 3300 Braunschweig, 0531-343167:

- 128k-Speichererweiterung zu verkaufen, auf Wunsch von Ulrich Hönisch aufgerüstet. 100,- DM

Erik d'Hondt, Wilgstraat 25a, B-9440 Erebodegem, 053-774497:

FDX-80Zeichen-Karte, SDX-80Zeichen/RS232-Karte, beide laufen einwandfrei, Preis VHS.

Ulrich Reichart, Huttoi 61, 8961 Wildpoldsried

An alle Grundgerätebesitzer: Verkäufe 128k-Speichererweiterung für MTX 500: DM 150,- und Spektrum-Emulator neuwertig (damit läuft ca. 80% der Spectrum-Cassetten-Software) für DM 100,- (NP 198,-).

SUCHE

C L U B: Korrektur & Nachtrag / Fragen & Antworten**Korrektur & Nachtrag**

Info 24, Seite 7: Patch zu XDIR3: (Zimmermann, 4330)

Der Patch in XDIR3 (C71h und C7Ah) funktioniert bei mir nicht. Da steht bei mir 33h. Deswegen kam mir die zu patchende 42h schon komisch vor.

CLUB.022: PRTNEU.PAS: (Peter Würfel, 7262)

Die drittletzte Zeile:

```
writeln(#27,#91,#88,#27,#91,#71,#27,#91,#84,#01);
```

ist zu ändern in:

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

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

Nur dann wird am Ende des Programms auf Tastaturliste 1 zurückgeschaltet.

Info24, Seite 25: Variables Bildschirmformat unter NewWord

Siehe Leserbrief von Jan Brederke weiter hinten.

Fragen & Antworten

F: Uwe Sanders, Gevedestr. 23, 2400 Lübeck (neue Anschrift)

Rechts sieht Ihr einen Auszug aus der NJW-Leitsatzkartei. In dieser regelmäßig erscheinenden Kartei sind eine ganze Menge von Leitsätzen, die möglichst auf Diskette gespeichert werden sollen, damit sie dann leichter sortierbar und wiederauffindbar sind. Kann man diese kleine Schrift irgendwie einlesen, und wenn ja wie. Das Ergebnis sollte möglichst gleich als Text-Datei vorliegen.

F: Peter Würfel, 7262

Warum wird die Turbo-Pascal-Zeile:

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

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

F: Peter Würfel, 7262

Wo gibt es XLISP in den Versionen 1.4 oder 1.5 als Public-Domain ? Fa. Kotulla hat nur Veriosn 1.1 als PD.

A: Hans Leiter, 6330

Der NEC P2200 ist aufwärtskompatibel zum NEC P6/P7, zumindest aus der Sicht von NewWord, welches den P2200 als P6/P7 angesteuert fehlerfrei bedient.

Wer weitere Informationen hierzu haben möchte, kann sich an Hans wenden - ggf. können wir auch gerne einen ausführlicheren Artikel hierzu veröffentlichen.

F: Herbert zur Nedden, 2000

Wer möchte versuchen den DMTI-Festplattencontroller am MTX über den ECB-Bus anzuschließen ? Ich habe einen Adapter sowie gewisse Software für den älteren DMTI, die ich gerne ausleihe. Bisherige Versuche (wir hatten für kurze Zeit Leih-Controller und Leih-Festplatte) waren erfolglos!

F: Was sind die Attribute 1-8, die ich mit NSWEEP setzen kann, die drei anderen R, S und A sind klar ?

A: Herbert zur Nedden, 2000

Mit 1-8 wird das 7. Bit des entsprechenden Buchstabens des Dateinamens gesetzt, d.h. bei einer Datei mit dem Namen ABCDEFGH.IJK setzt die Auswahl 3 das 7.Bit des C gesetzt. Übrigens kann auch die 0 angegeben werden, um alle 7. Bits zu löschen.

ZPO § 91 Kostenerstattung A/80
Privatgutachten

Die Kosten für Privatgutachten sind als Vorberbeitungskosten des Rechtsstreits nur ausnahmsweise erstattungsfähig. Die Notwendigkeit ist zu bejahen, wenn erhebliche technische Kenntnisse zu dem Nachweis erforderlich sind, daß es sich um einen *fingierten Verkehrsunfall* handelte (Verdacht des Versicherungsbetruges).

OLG Frankfurt, 13. 10. 1986, 6 W 252/86
JurBüro 87, 894

C L U B: Fragen & Antworten

- A:** Herbert zur Nedden, 2000
Mit NSWEEP können übrigens gleich mehrere Dateien umgenannt werden. Dazu wird mit R(ename) die Umbenennungsfunktion aufgerufen, und ein * eingegeben. Anschließend fragt NSWEEP nach dem alten, und dann nach dem neuen Namen. Hier sind Wildcards (* und ?) zulässig. Gebe ich als alten Namen TEST.*, und als neuen Namen PROD.* an, passiert das naheliegende.
Interessant ist übrigens die Möglichkeit, bei der Angabe des neuen Namens entweder bei R(ename) oder auch bei R(ename) * einen anderen USER anzugeben. Benenne ich TEST.PAS in 3:TEST.PAS um, ist TEST.PAS schon im USER-Bereich 3.
- F:** von mehreren Mitgliedern gestellt:
Warum kann nicht die CP/M-TPA über die 64kB-Grenze hinaus wachsen ?
- A:** Herbert zur Nedden, 2000
Dies liegt an CP/M, welches dank der 16Bit-Adressierung der Z80 nur 64kB unterstützt. Unter RAM4 ist auch lediglich das BIOS, d.h. der hardwareabhängige Teil des CP/M auf Bank 1 ausgelagert worden. Sicherlich kann man es irgendwie schaffen, daß auch das BDOS auf eine andere Speicherbank kommt, um dann ein 63k-CP/M-System zu basteln, aber dann werden mit Sicherheit einige Programme nicht mehr laufen, die davon ausgehen, daß das BDOS in den 64kB CP/M-Speicher liegen. Außerdem sind da dann noch diverse Tabellen, wie insbesondere die Belegungstabelle für die Disketten, die von vielen Dienstprogrammen zwingend in diesen 64kB erwartet werden. Mit viel Mühe können wir evtl. ein 60k-CP/M erzeugen, wenn wir viele Kompatibilitäts-Ansprüche unter den Tisch fallen lassen. Um große Dateien untersuchen zu können sollten diese lieber direkt auf der Diskette, und nicht im Speicher untersucht werden. Wer schreibt hierfür mal ein Programm ?
- F:** Herbert zur Nedden, 2000
Die 80-Zeichen-Ausgabe der SDX ist eigentlich recht kontrastarm. Mit meinen Hardware-TIPS weiter hinten läßt sich das auch nicht beseitigen. Wer weiß Rat. Um mit NewWord arbeiten zu können kann als Übergangslösung das entsprechende Patch-Programm von CLUB.024 verwendet werden.
- F:** Dietrich Jerg, 7750
Wie können die RAM4-Funktionstasten unter dBASE genutzt werden ?
- F:** Gerd Rüdiger von Pezold, 3000
Schrift- und Speichererweiterung für Drucker Brother M-1509, wie ?
- F:** Tom Witkowski, 5439
Hat jemand eine Modelleisenbahnsteuerung, oder sogar eine Steuerung für Scheinwerferstellungen ? Was ist dafür notwendig ?
- F:** Peter Lang, 8507
Wer hat Erfahrung mit Mehrplatz-Terminalbetrieb (mind. 3 Tastaturen, z.B. via RS232) ?
- F:** Karlheinz Rößler, 7920
Hat jemand ein kommentiertes NewWord-Listing ?
- F:** Herbert zur Nedden, 2000
Es soll ein Turbo-Pascal-Listing geben. Wer weiß was ?
- F:** Harald Gottwald, 8634
dBASE-Verwaltungsprogramm für Tischtennis-Spieler mit 250 Spielern (Name, Geburtsdatum, Verein) für 18 Turniere.
- F:** Hauke Ahrensfield, 3100 Celle
Hat jemand den Schaltplan des SDX-Controllers ?
- F:** Gerhard Orth, 5600
Kann Port 7 für andere Zwecke genutzt werden, wenn die Hardware-Uhr drauf sitzt ?
- A:** Herbert zur Nedden, 2000
Nein, bis auf evtl. die 4 freien Ausgabe-Bits - mit Vorsicht.

C L U B: Fragen & Antworten

- F:** Helmut Grothe, 8000
Wie wird ein RAM/EPROM-Bereich zur RAM-Floppy ?
- A:** Herbert zur Nedden, 2000
Im System muß ein Treiber für die Floppy und der von CP/M benötigte DiskParameterHeader, DiskParameterBlock muß vorhanden sein. Der Treiber muß in das System eingeschleift werden (d.h. bei Disketten-zugriffen gefragt werden), und folgende Funktionen bereitstellen: Config, ReadSector, WriteSector, und auf die Sektor-, Spur- und DMA-Adressen-Werte aus dem System zurückgreifen können. Falls Du uns mal das Programm schickst, welches Deine RAM-Floppy (aus SDX 3.5"-Lw) realisiert schickst, schauen wir mal nach, was machbar ist
- F:** Rainer Tewes, 4690
Wie kann man einen Strom in eine proportionale Spannung umwandeln. Es geht darum, daß ich direkt an Ventilen fließenden Strom messen und mit einem Einplatinen-Rechner verarbeiten möchte.
- F:** Bernd Fitz, 7136
Rechnerkopplung MTX-RS232 an JoyStick eines anderen.
- A:** Herbert zur Nedden, 2000
MTX-seitig eigentlich völlig banal, das kann z.B. das Modemprogramm von CLUB.001 oder auch CONTACT von der Memotech-Systemdiskette. Peter Kretzschmar (2350) bietet ein Paket an, welches den APPLE-Game-Port (also wohl etwas JoyStick-artiges) nutzt, jedoch beim MTX den Port 7.
- F:** Bernd Fitz, 7136
Wie kann ich die RS232 programmieren ?
- A:** Herbert zur Nedden, 2000
Z.B. gemäß Info 2, Seite 32 und Info 5, Seite 5.
- F:** Wolfgang Daenell, 5000
Kann einer ein Oszilloskop-Programm entwerfen ?
Existieren AD-Wandler-Platinen ?
- F:** Peter Würfel, 7262
Kann jemand die Assembler-routinen für meinen Digitalisierer in Pascal einbinden ?
- F:** Herbert Oppmann, 8520
Welche umschaltbaren HighDensity 80-Spur Laufwerke sind MTX-tauglich ?
Was ist der Wiederverkaufswert eines EPSON 40-Spur-Lw.
- A:** Herbert zur Nedden, 2000
Ich weiß es nicht, empfehle jedoch TEAC FD 55 GFV, da preisgünstig und Spitzenqualität.
- F:** Hans Friz, 6700
Wie kann man den Cursor dauerhaft, oder zumindest in Turbo Pascal und NewWord groß machen, damit man ihn auch findet ?
- F:** Joachim Höft, 4000
Wo kann ich ein Ring-Rom bekommen ?
- F:** Erik d'Hondt, B 9440
Wer kann eine einfache Beschreibung zur Aufrüstung unserer Kiste auf 6 MHz geben ?
- F:** Stefan Hößler, 1000
Wofür sind im KLICK-Calculator die Tasten LONG, BYTE, WORD ?
- A:** Herbert zur Nedden, 2000
Damit wird die Stellenzahl festgelegt. BYTE = 1 Byte, d.h. 8 Bit, WORD = 2 Byte, d.h. 16 Bit (z.B. Adressen) und LONG = max. Genauigkeit, und die ist reichlich.
Diese Optionen sind eigentlich nur für HEX, OCTal und BINär interessant. So kann ich z.B. mit HEX, WORD die Adressrechnung der Z80 nachvollziehen.

C L U B: Fragen & Antworten / Fragebogen-Auswertung

F: Philemon Lopatta, 7560

Anschluß eines 'Supercharger' an den MTX. Siehe CHIP 11/87, Seiten 114-116. Supercharger ermöglicht die Verarbeitung von MSDOS-Software auf Atari ST unter Verwendung des Atari-Laufwerks. Kosten ca. DM 500,- (8086 CPU, 1 MB RAM).

F: Hartmut Traber, 5270

Kopplung eines Modems an meine Super-HI-FI-Stereo-Anlage (selbstverständlich mit Super-Super-Kurzwellen-Teil) um Pressedienste o.ä. lesbar und ausdrückbar zu machen. Da CONTACT unter RAM 4 nicht mehr läuft, überdies der 5-Bit-Baudot-Code nicht einstellbar ist, habe ich diverse Schwierigkeiten. Grundsätzlich aber geht dies mit RTTY.COM.

F: Uwe Sanders, 2400

Ist ein späterer Umbau des Memotech in einen AT-kompatiblen möglich? (Austauschen von IC's, ...)

Fragebogen-Auswertung

(Herbert zur Nedden, 2000)

46 Fragebögen gingen beantwortet ein.

Hardware-Bestand

FDX:	44	Umgebaut:	6	Drucker:	alle
SDX:	2	< 512 kB:	12	Typenrad:	4
Laufwerke:	97	> 512 kB:	34	Modem:	7
Hardw-Uhr:	8	80Z aufgerüstet:	6	AD/DA-Wandler:	3
		> 4 MHz:	4		
ECB-Karten:		RAM-Floppy:	5	Edicta-Grafik:	6
		SRAM-Floppy:	11	Prommer:	2
		EPRAM-Floppy:	6		

Software-Nutzung

NewWord:	45	Turbo-Pascal:	30	dBASE:	23
BASIC:	16	SuperCalc:	14	Assembler:	10
Spiele:	5	Koppler-SW:	5	C:	4
MODULA:	3				

Interesse an Erläuterungen/Erweiterungen von Software

RAM4/KLICK:	12	Grafik:	12	Z80:	4
Kopplung:	2				

Interesse an Hardware

HardDisc:	7	Reparatur:	4	> 4 MHz:	3
Laufwerke:	4	Ser Tastatur:	3	OCR:	3
Ring-ROM:	2	DoMessDOS:	2	Port 7:	1

SW-Kurse

Z80:	5	CP/M:	2	Sonst:	2
------	---	-------	---	--------	---

Anregung

- MTX-Schlagwort-Register: 1. Wörterbuch,
 2. Wegweiser durch versch. Themenbereiche
 3. Hinweise auf weitere Literatur

Fragen und Probleme

Siehe oben die Rubrik 'Fragen & Antworten'.

C P / M unter R A M 4 . x : Probleme

FRAGE zu ZEX

(Peter Würfel,7262)

Ich möchte durch eine ZEX-Batch-Datei folgendes leisten lassen:

- NW aufrufen
- eine Datei laden (Bspl.unten: 'test.txt')
- alle Anführungszeichen durch nix ersetzen (d.h. ersatzlos rauswerfen)
- die Datei abspeichern
- NW verlassen

Wie muß die Batch-Datei aussehen?

'Zu Fuß' gebe ich ein:

```
nw<RET>
dtest.txt<RET>
<CTRL>qa<RET>
"<RET>
<RET>
nr<RET>
<ESC>
<CTRL>kd
x
```

Die Batch-Datei:

```
nw
dtest.txt
?
?
?
?
?
?
?
?
x
```

MENUE-TIP

(Peter Würfel,7262)

Der Befehl ?^L^AASX in Menue eingesetzt führt zu Abbruch von Menue. Wenn ich jedoch ein Non-doc-file mit dem Inhalt ^L^AASX schreibe, kann ich das in Menue mit type <filename> aufrufen.

UHR-FRAGE (und KLIX)

(Peter Würfel,7262)

Theoretisch müßte man doch mit dem Befehl:

```
?<ESC>AS <RET> (4 Leerzeichen zwischen S und <RET>)
```

die Uhrenanzeige ausschalten und mit dem Befehl:

```
?<ESC>ASj! 6<RET> (Leerzeichen zwischen ! und 6)
```

die Uhr wieder einschalten können. Doch das klappt nicht. Wenn ich jedoch zwei kleine non-Dokument-files mit jeweils diesem Inhalt schreibe und die mit type<filename> aufrufe wird die Uhr gelöscht bzw. eingeschaltet. Ich habe auch versucht ein KLIX zu schreiben, so daß ich nach dem Aufruf des KlIX wählen kann, ob ich die Uhr einschalten (dann z.B. F1 drücken) oder ausschalten (dann z.B. F2) will. Das klappte jedoch immer nur in einer Richtung (hab leider nicht notiert ob nur Ein- oder nur Ausschalten funktionierte).

Software**FORMSTAR**

(Michael Keßler, 5600)

Was ist FORMSTAR:

Formstar ist ein universelles Formatierprogramm, das nur in Verbindung mit **RAM 4.2** von Bernd Preusing zu benutzen ist. Formstar existiert in zwei Versionen: der **Normalversion** und der **KLIX-version**. Beide Versionen sind von der Benutzung her identisch. Es stehen drei Möglichkeiten der Benutzung zur Verfügung:

- Menuegeführtes Formatieren: aus insgesamt 11 Menus kann gewählt werden
- Direkteingabe: Man gibt den gewünschten Cfig-Code direkt ein
- Automatic: Es wird das derzeit konfigurierte Format formatiert.

Im Gegensatz zu FORM4 formatiert Formstar unabhängig von der jeweiligen Drivekonfiguration. Auch wenn das Laufwerk auf Format 03 konfiguriert ist, kann man darin eine Diskette mit Format 09 formatieren, wobei sich allerdings der Cfig-Codes des Laufwerkes nicht ändert!

Was kann FORMSTAR:

Formstar beherrscht derzeit 83 Formate, von SS/SD 40 Trk. bis hin zu DS/DD 80 Trk. und 8"-Formaten. Hierin sind auch alle Formate von Bernnds FORM4.COM enthalten:

03,07,08,09,0A,0B,0C,0D,0E,0F,10,11,12,13,19,
1A,1B,70,80,83,86,88,89,8B,8D,8F,94,98,9A,9C,
9F,A5,A6,AF,B1,B3,B6,B7,B8,B9,BA,BB,BC,BD,BE,
BF,C0,C1,C2,C3,C5,C6,C7,C8,C9,CA,CB,CC,CD,CE,
CF,D0,D1,D2,D3,D4,D5,D6,D7,DB,DA,DB,DC,DD,
DE,DF,E2,E5,E8,EA,F4,FB

Weitere Formate kann ich unter gewissen Voraussetzungen implementieren.

FORMSTAR ohne RAM4.2

Falls jemand FORMSTAR haben möchte, jedoch kein RAM4.2 hat, möge er/sie sich an mich wenden, und ich werde ggf. eine Version erstellen, die ohne RAM4.2 läuft.

Edicta-Grafiksoftware

(Herbert zur Nedden, 2000)

Es ist wieder eine neue Version dieser Software herausgekommen, und auch schon verteilt. Neu hinzugekommen sind u.a. Balken für Stabdiagramme, Kreisbögen, sowie einige Optionen für die Textausgabe und die Hardcopy-Funktion.

CONNECT

(Herbert zur Nedden, 2000)

Herbert Oppmann schreibt gerade an einem Ersatz für CONTACT. Dieses Programm unter RAM 3.x oder RAM4.x laufen. Da der DART (also die serielle Schnittstelle) über Interrupts läuft - wer hat da gesagt, daß ginge nicht ??? - können leicht Übertragungsraten von 19200 Baud fehlerfrei realisiert werden. Die Vorversion, die Herbert mir mal so zum unverbindlichen Ansehen schickte wirkte recht überzeugend.

Software: Etikettierer

Rainer Tewes (4690)

Hallo Herbert!

Weil ich im Club bisher kein Etiketten-Druck-Programm fand, welches ich wirklich gebrauchen konnte, habe ich mich als blutiger Turbo-Pascal-Laie an die Arbeit gemacht.
Dabei ist E.PAS bzw. E.COM herausgekommen.

Dieses Programm möchte ich nun vorstellen. Anschließend noch eine Beschreibung meines neuen Matrix-Druckers LX 800 von Epson.

Als Basis diente mir das Programm 'ETIKETTE.PAS' von der Club-PD 11 von Martin Wessel (vielen Dank!!).
Ich möchte das Programm, falls es Dir gefällt, als Club-PD-Software freigeben.

E.COM ist ein Programm um Standard-Etiketten des Formats 88mm*36mm zu bedrucken.

Das Programm ist selbsterklärend.
Trotzdem eine kurze Beschreibung.

Es fragt als erstes, ob der auf der Diskette gespeicherte Absender geändert werden soll. Falls 'J' wird der neue Absender verlangt. Anschließend wird gefragt, ob der neue Absender dauerhaft gespeichert werden soll.

Der Absender ist in der Datei 'E.DAT' gespeichert. E.DAT muß sich auf dem aktuellen Laufwerk befinden.

Nun können die Daten (Anrede1,Name;Anrede2,Adresse usw) eingetragen werden.

So sieht die Maske mit einer Adresse aus:

Rainer Tewes; Nordstraße 188; 4690 Herne 1 (max. 57 Z)

Anrede 1	:Firma	.	(max. 33 Z)
Name	:Mannesmann Rexroth GmbH	.	
Anrede 2	:- Personalbüro -	.	
Adresse	:Postfach 340	.	
Postleitzahl	:8770.	.	
Wohnort	:Lohr am Main	.	(max 25 Z)

Adresse in Ordnung ? J/N.

Falls 'N' kann die Adresse nochmal eingegeben werden.

So sieht das Etikett aus:

Rainer Tewes; Nordstraße 188; 4690 Herne 1

**Firma
Mannesmann Rexroth GmbH
- Personalbüro -
Postfach 340**

8770 Lohr am Main

Die Punkte begrenzen auch im Programm die maximale Länge.
Die Fragen nach 'Anrede1' und 'Anrede2' können mit RETURN beantwortet werden, die Zeilen werden dann nicht gedruckt. Die nachfolgenden Zeilen rücken automatisch nach.

S o f t w a r e: Etikettierer

Das Programm fordert jetzt auf das Etikett in Position zu bringen. Nach RETURN wird ein Probedruck gemacht. Der Probedruck wird solange wiederholt, wie die Taste 'N' gedrückt wird. Also erst korrigieren, dann Taste drücken. Ist der Probedruck in Ordnung, wird nach der Anzahl der zu druckenden Etiketten gefragt. Es kann ja sein, daß man öfters an die selbe Adresse schreiben muß. Einen kleinen Schönheitsfehler hat die ganze Sache. Mein neuer Drucker ist ein Epson LX 800. Ich weiß nicht, wie das Programm mit anderen Druckern zusammenarbeitet. Die Etiketten-Druck-Programme aus dem Club liefen auf dem LX 800 einwandfrei. Mein Programmchen ist also 'wahrscheinlichkompatibel'. Außerdem habe ich die Procedure 'drucken' im Programm ausführlich documentiert, ist also einfach zu ändern.

Nun zum Drucker.

Die technischen Daten:

9*9 Matrix	Draft
11*18 Matrix	NLQ
80 Zeichen	pica
137 Zeichen	pica schmal
96 Zeichen	elite
160 Zeichen	elite schmal
bei Breitschrift eben die Hälfte	
180 Zeichen/sec.	Draft
25 Zeichen/sec.	NLQ
2 Schriftarten	Roman und Sans Serif
Grafikfähig	
Selbstdefinierte Zeichen sind möglich	
Tabulatoren (vertikal und horizontal) programmierbar	
13 Zeichensätze (USA, Deutschland, Spanien...)	
Jede Menge Formatierbefehle	
3 kByte Puffer	
Centronics-Parallel-Schnittstelle	
Als Option auch RS 232; IEEE 488; Apple 2 kompatibel	
Endlospapier (Stachelwalze) und Einzelblätter können bedruckt werden.	
Als Option gibt es auch einen Rollenpapierhalter.	
Breite = 400 mm Höhe = 140 mm Tiefe mit Stecker = 350 mm	

Die Handhabung:

Endlospapier

Der Papierlösehebel muß in Stellung 'lösen' gebracht werden. Der Stachelwalzen-Traktor wird nun auf das Gerät aufgesteckt. Der Papierlösehebel ist nun nicht mehr zugänglich. Führe mal bei gelöster Walze neues Papier oder Etiketten ein. Für Einzelblätter muß der Traktor also wieder abgenommen werden, das geht zwar sehr schnell, aber dann fliegt das Ding wieder irgendwo rum.

Die weitere Beschreibung auf der nächsten Seite habe ich als Schriftprobe mißbraucht.

Software: Etikettierer

Schriftprobe

- NLQ und Roman -

Der halbautomatische Einzelblatteinzug (hahaha!):
Er besteht aus einer simplen Kunststoffplatte mit verstellbaren
*Papierführungen. Habe Ihn ausprobiert, und weggelegt, es geht von
Hand besser. Die Walze erwischt das Papier meist nicht, falls doch,
muß man die Lage des Papier's eh von Hand korrigieren.*

- NLQ und Sans-Serif -

Papier-Andruckrollen oder etwas ähnlich gescheites gibt es auch
nicht. Diese Funktion übernimmt die Staubschutzklappe. Bei
Einzelblättern verfängt sich der Blatt-Anfang meistens an dieser ollen
Klappe. Also erst Klappe auf, einige Zeilen drucken lassen, Klappe zu.
*Im Allgemeinen ist die Handhabung des Gerätes also popelig.
Was will man auch für einen Preis von 550,-DM (Versand-Handel) oder
630,-DM (Fachhandel, wenn man ein wenig handelt) verlangen.*

- elite -

Die Mechanik allerdings macht auf mich einen sehr soliden Eindruck.
Ich kann das beurteilen, bin Handwerker. Der Druckkopf wird in stabili-
len Buchsen geführt und von einem Zahnriemen angetrieben. Zahnriemen
garantieren präzieseste Steuerung und lange Lebensdauer. Der Druckkopf
sieht auch recht ordentlich aus, er hat große Kühlrippen. *Ich habe
mal mehrere 100 Seiten an einem Stück gedruckt, er hat tatsächlich
einen kühlen Kopf behalten. Er soll 200.000.000 Anschläge pro Nadel
verkräften können. Ich habe schon Druckköpfe gesehen, denen traute
ich keine 5 Seiten zu, bevor sie auseinanderfallen.*

Da ist noch ein Druckdichten-Hebel, damit kann man den Druckkopf
näher zum Papier bringen um Texte mit mehreren Durchschlägen zu
drucken.

Das sind nur die wichtigsten Schriftarten, es sind noch diverse
Kombinationen möglich. Er druckt doch recht Ordentlich, oder??
Trotz der oben genannten Mängel bin eigentlich zufrieden. Schließlich
ist man ja Kummer gewohnt.
Übrigens benutze ich in NEWWORD den DRAFT-Treiber, mit den Epson-
Treibern hat der LX 800 Schwierigkeiten.

Tschüß Rainer

Ann.d.HzN.: Das hier beschriebene E.COM ist nicht mit meinem MTX-Edit,
welches vielerorts unter dem Namen E.COM eingesetzt wird zu ver-
wechseln.

Software: KCLICK-PD's**KCLICK-Public-Domain**

(Herbert zur Nedden, 2000)

Die neue KCLICK.002 ist fertig, bis auf einige kleine kosmetische Änderungen, die vermutlich mittlerweile auch erledigt sind.

Auf der Scheibe findet Ihr wie vielleicht schon erraten das neue HardCopy, Version 3, welches um einige Funktionen erweitert wurde - u.a. um eine Hardcopy der Edicta-Grafikkarte.

Die Benutzeroberfläche (schönes Wort, gelle) ist ansprechend mit Pull-Down-Menüs realisiert - HardCopy reagiert allerdings auch auf einige F-Tasten!

Außerdem ist auf der KCLICK.002 mein KALENDER in einer fehlerbereinigteren Version (danke Jan Bredereke), bei der man nun auch mal einen Monat oder ein Jahr hin/her blättern kann. Ein neues ASCII (Uwe Grass), und falls noch Platz drauf ist, eine Spoolerumschaltung (Wolfgang Zierenberg) sind weitere KCLICK-Overlays. Damit auch mal etwas aus dem KCLICK sicher entfernt werden kann ist KILL42 anbei und sogar ein KLIX-SWEEP, mit dem die Belegung des KLIX-Heaps angesehen und bearbeitet werden kann.

Die KCLICK-Overlay-Programmierer werden sich sicherlich freuen, daß jetzt auch zwei neue Routinen-Sammlungen und SUBmit-Files dabei sind, die das Erstellen von K-O's vereinfachen. Auch die neue SYSLIB (Version 3.6), KCLICK-Programm-tauglich (bis auf BDOS-CALLs) ist drauf, d.h. insbesondere sind alle DefineSpaces (DS) mit 0 vorbelegt, da anderenfalls die Loader auf Probleme stoßen.

Schließlich unendlich findet Ihr auch hier wieder diverse HLP-Dateien, die mit HELP bequem angesehen werden können.

KCLICK.003 ist schon angefangen, und beinhaltet unter anderem von Jan Bredereke ein KCLICK-Overlay mit dem im laufenden NewWord das Bildschirmformat geändert werden kann, sowie einen Wecker. Vermutlich werdet Ihr hier auch die Routinen für Olaf's Pull-Down-Menüs finden, mit denen die Programmierung von solchen Menüoberflächen für ein Assemblerprogramm sehr schön einfach und pflegeleicht (d.h. änderungsfreundlich) geht.

Hattu 'n KCLICK, dann schicken Olaf!

Leserbrief: Hartmut Traber, 5270

Die Clubleiche, zeitweise scheinot, aber wenn das Info kommt rege, meldet sich zu Wort:

Info 24, Steiten 11 und 12:

Vielen Dank, Herr Dr. Göbel für den ausführlichen Erfahrungsbericht! Er hat mich angeregt, mich animiert, Schwierigkeiten mitzuteilen! (Weiter so!und die anderen Leichen?)

Ich habe 2 c't-CMOS-Disc's am ECB.

Trotz Schreibschutz wurde bei mir die Speicherstelle der Disc I: (Bootlaufwerk) 00h (C3), also 1. phys. Sektor, 1. Byte gelegentlich überschrieben mit 00 oder was anderem.

D.h. das Booten von I: war dann nicht mehr möglich.

Das habe ich natürlich erst herausgefunden, als ich es leid war, immer wieder mit Syscopy4 mein System neu zu kopieren und dann zu booten, wobei dann auf der schlappen Scheibe nicht der richtige Path war, und ich wieder erneut patchen mußte. (Path arbeitet offenbar auch nicht immer richtig!)

Ich betone: **Trotz Schreibschutz!**, und soweit ich weiß, nur die erste Speicherstelle.

Taktfrequenz ?

Mein Rechner läuft mit 6 MHz problemlos. (Aachen!)

Schalte ich einen 8 MHz-Quarz ein, läuft der Rechner auch, aber langsamer als mit 6 MHz, vermutlich mit 4 MHz. Ob das Wait's sind, oder sonst was, ist mir noch zu hoch. Wie programmiert man so was? Wie werden sie eingebaut? Bzw. wieso läuft der Floppy-Controller nur mit 4 MHz? (Ich merke keinen Unterschied, wenn ich von 6 auf 4 MHz umschalte).

Leute, Leute

seid bitte in den kleinen Dingen etwas ausführlicher!!

Sicher, wenn ich nach Monaten alles noch mal durchlese, erkenne ich doch manches, was vorher nicht klar war.

Aber drei Worte zur rechten Zeit können den Scheintoten oder Leichen sehr helfen!

Nach diesem Holzhammer: unser Club ist Spitze, deshalb Dank vor allem an die 1''-Hacker (First-Class-Hacker).

Hartmut

Leserbrief: Jan Brederke, 2000

Korrektur zum Info 24, S. 25: Variables Schirmformat für Newword (Jan Brederke)

Da ist mir doch ein Tippfehler unterlaufen.

HITE (02E6H) ist natürlich die Zeilenzahl des Bildschirms und WID (02E7H) ist die Zahl der Zeichen pro Zeile, nicht umgekehrt. Entsprechend kommt nach 18B8H der Wert von WID minus 1, und nicht von HITE minus 1.

Jürgen Freimuth aus München hat mir zu dem Artikel außer dem Hinweis auf diesen Fehler noch mitgeteilt, daß nach 18B7H der Wert von HITE minus 1 muß. Ausprobieren hat aber ergeben, daß dies noch nicht ausreicht, um den Bildschirmaufbau in Ordnung zu bringen.

PS: Inzwischen hat Jürgen mir noch einige weitere Tips gegeben, so daß ich die Problemursache herausgefunden habe und ich an einer Lösung arbeite. Demnächst mehr.

Auto-Repeat-Tastatur unter RAM4.2? (Jan Brederke, 2000)

Unter dem originalen Tastatortreiber gab es eine Auto-Repeat-Funktion, d.h. ein Tastencode wurde automatisch wiederholt, wenn die Taste etwas länger gedrückt wurde. BP hat diese Funktion zugunsten einer Repeattaste weggelassen. Jetzt wird der Code sehr viel schneller als früher wiederholt, wenn die Taste zusammen mit <LINE FEED> gedrückt wird. Dies ist sicherlich ein Vorteil für Zwei-Finger-Schreiber, aber für diejenigen, die wie ich mit zehn Fingern schreiben, bedeutet es Verrenkungen der Finger oder Zeitverlust, um die Hände von der Tastatur zu nehmen und diese Taste zu drücken. Nicht umsonst gibt es auf der Tastatur zwei SHIFT-Tasten, denn wenn ich mit dem rechten kleinen Finger den '*' tippe, kann ich nicht gleichzeitig die rechte SHIFT-Taste drücken, sondern nehme die linke mit dem linken kleinen Finger. Entsprechendes gilt für die Tasten auf der linken Seite. Das Zehn-Finger-System bietet dem, der es beherrscht, einen großen Geschwindigkeitsvorteil, dafür sind aber jedem Finger fest bestimmte Tasten zugeordnet.

Ich weiß nicht, ob technische Gründe für die Abschaffung der Auto-Repeat-Funktion sprachen, und ich weiß auch nicht, wieviele MTX-Benutzer mit zehn Fingern schreiben, aber ich stelle diesen Punkt hiermit einfach einmal zur Diskussion.

Anmerkungen zum 8 MHz-Artikel von Holger G. im Info 24 (Jan Brederke, 2000)

Holger G. hatte dort festgestellt, daß bei der c't-CMOS-RAM-Floppy beim Directory-Lesen vermutlich ein ganz kurzer Schreibimpuls erzeugt wird. Dazu will ich auch noch eine Beobachtung beitragen. Bei meiner CMOS-Floppy wurden gelegentlich das erste Byte der Systemspur und das erste Byte des Directories beim Einschalten des Rechners verändert, und das, obwohl der Hardware-Schreibschutz eingeschaltet war! Unter anderem diesem Problem habe ich durch einen verlängerten Resetimpuls beim Einschalten abgeholfen. Die Schaltung dazu habe ich schon im Info 23-17 veröffentlicht.

Übrigens noch eine Antwort an Holger, der diese Schaltung gerne noch weiter erklärt gehabt hätte:

Du hast recht, daß man bei jedem Hard- oder Software-Eingriff möglichst auch die Begründungen für diesen mitliefern sollte. Die Werte der Bauteile in meiner Schaltung habe ich allerdings durch reines Probieren herausgefunden, so daß die Angabe von Formeln für RC-Glieder wohl auch nicht hilfreich wäre. Meine Lieblingsformel ist in solchen Fällen immer "pi mal Daumenbreite durch Autonummer". Wenn jemand solch eine Schaltung nachbaut, sollte er zuerst die dort angegebenen Werte nehmen, und wenn es dann nicht gleich funktioniert, diese variieren.

Ein weiterer Punkt ist die Zielgruppe eines solchen Artikels. Ich habe gehofft, daß die Leute, die sich zutrauen, so etwas nachzubauen, auch den Schaltplan lesen können. Ansonsten hoffe ich, daß Holger mit meinem Artikel über die Reparatur des Klux-Programmes KALENDER.COM an anderer Stelle in diesem Info zufrieden ist. Ich habe mich bemüht, sowohl eine Patchanleitung zum Abtasten zu geben als auch die Hintergründe des Fehlers näher zu erläutern.

Leserbrief: Jan Bredereke, 2000

Montageproduktion? (Jan Bredereke, 2000)

Am Montag, den 21.12.87, trudelt bei mir die neue PD-Diskette KCLICK.001 ein, und es sind viele nützliche Sachen darauf. Also wird erst einmal alles ausprobiert. Das **Klick-Programm KALENDER.COM** zeichnet mir sehr hübsch eine laufende Uhr und ein Kalenderblatt vom Dezember, wobei die 22 heller dargestellt wird. Wieso 22? Aha, meine Hardware-Uhr geht falsch. Ich stelle sie auf Montag, den 21. um, und schon geht gar nichts mehr. KALENDER produziert nur noch Bildschirmmüll. Tests ergeben, das dies bei jedem Montag im Dezember passiert. Arbeitet der Autor nie montags?

Zum Glück ist der Quellcode dabei, und ich entdecke, daß das Programm in Wirklichkeit genau dann nicht geht, wenn der Tag des Monats durch 7 teilbar ist, was im Dezember zufälligerweise die Montage traf.

Ursache ist, daß an einer Stelle eine Zahl kleiner als Null wird, woraufhin eine folgende Subtraktion manchmal größere Werte als den Ausgangswert ergibt, was nicht abgefangen wird. Durch eine kleine Befehlsumstellung läßt sich das aber vermeiden, so daß ich hier sowohl die Patches für den Quellcode bringe als auch eine Anleitung, wie sich KCLICK.001-Besitzer ihr KALENDER.COM reparieren können, selbst wenn sie wie ich keinen M80-Assembler haben.

Alt:		Neu:	
NoFeb:		NoFeb:	
	ld a,(ix+0) ; Tag		ld a,(ix+0) ; Tag
	ld (Tag),a ;ergibt 1 <= A <= 31		ld (Tag),a
Cut:	sub 7	DEC A ; *****	
	jp nc,Cut	sub 7	
	add a,7	jp nc,Cut	
	ld c,a ;ergibt 0 <= C <= 6	add a,7	
	ld a,(ix+6) ;0 <= Wochentag<= 6	ld c,a	
	dec c ;ergibt -1 <= C <= 5	ld a,(ix+6) ; Wochentag	
	sub c ;ergibt -5 <= A <= 7	DEC c *****	
	jp nc,Gut	sub c	
	add a,7	jp nc,Gut	
Gut:	;ergibt 0 <= A <= 7	add a,7	
	;statt 0 <= A <= 6		
		Gut:	

So wird KALENDER.COM gepatcht:

Man benötigt das Programm DDT.COM von der Systemdiskette. Zuerst werden die Befehle umgeordnet, wobei eine Sprungadresse korrigiert wird, dann wird die Relokationstabelle korrigiert, da ein Sprungbefehl verschoben wurde. Zum Schluß wird das Programm unter neuem Namen abgespeichert, um bei einem Tippfehler immer noch die alte Version zu haben. Anmerkung: Den m(ove)-Befehl darf man hier nicht verwenden wegen der Überlappung von Start und Ziel.

Eigene Eingaben in Fettdruck. Besonders wichtige Passagen sind unterstrichen.

```

A>ddt kalender.com
DDT VERS 2.2
NEXT PC
OC00 0100
-e0762
D6 3d
07 d6
D2 07
62 d2
07 63
C6 07
07 c6
4F 07
DD 4f
7E dd
06 7e
0D 06
91 .
-e0b17
42 41
00 .
-~C
A>save Obh kalendr2.com
    
```

Hardware: 8 MHz-Erfahrungen von Horst Kupka (4019)

Lieber Herbert !

Erstmal vielen Dank für Dein offenes Ohr und Deine Hilfe. Endlich, die 8Mhz funktionieren in meinem Rechner. Wohlbemerkt, die 512k--Karte ist mit 150ns-Samsungrams bestückt, und Waits sind nicht nötig. Was habe ich nun gemacht ? Wie fing alles an. Auf dem Clubtreffen sah ich so eine schwarze Kiste, dem äußeren Erscheinungsbild ein MTX-FDX. Was sich aber auf dem Bildschirm tat, nein das konnte kein Memotech sein, viel zu schnell! Das Innenleben war, wie sich später rausstellte, ein aufgemotzter MTX.

Toll dachte ich, und mein Bastlerherz ließ mir keine Ruhe. Zu Hause wird, bedingt durch meine Kinder, der Rechner häufig eingeschaltet. (auch für Spiele haha....) Das Ausschalten wird dann meistens vergessen. Wie die Grundplatine aussah kann sich ja mancher denken. Das schöne Grün sah an bestimmten Stellen aus, wie ein im Sommer vertrockneter und verbrannter Rasen. Hier wollte ich schon lange was ändern. Nun als ich das Netzteil so langsam ausbaute fiel mir das Clubtreffen ein. (8Mhz-Tuning). Och, das erledigst du eben so mit, war eine innere Stimme. Unterlagen gab es ja im Info genug. Nach zweimaligem kompletten Ausstieg des Rechners, war ich soweit. Hätte ich diese Sache bloß nicht angefangen, nichts funktioniert, und 4Mhz tun es doch auch. Also wieder auf 4Mhz und zusammengebaut.

Als ich das Ding dann vor mir sah, kam wieder der Gedanke, wenn andere Rechner auf 8Mhz laufen, dann muß es deiner auch. Also wieder ans Werk.

Hier sind nun die Ergebnisse tage- und nächtelanger Versuche. Mein größtes Problem lagen an der Hauptplatine. Die 32k wollten und wollten nicht richtig laufen. Mein Gedanke ein statisches Ram. Also alle D-Rams raus und umgebaut. Endlich mit 8Mhz klappte es im Minimonitor. Nun den Rest bestückt und die 8Mhz liefen auch unter Ram4. Aber, aber, die Kiste wird ja auch warm, und dann ??????? Bei 8Mhz ja noch ok, nur die 4Mhz hatten auf einmal ihr eigenes Leben entwickelt. Beim Warmstart lief die Kiste zwar bis zum Diskettenzugriff, druckte dann die BIOS-Meldung auf den Bildschirm, und dann legten sich die Elektronen schlafen, aus, sie hatten keine Lust mehr. Einmal könnte ich es verstehen, aber permanent das gleiche : Boot, erste Meldung ,Ruhepause! Und mit 8Mhz alles ok. Mist was konnte das nur sein. War das Gerät kalt lief es (auch mit 4Mhz), war es warm, hatten die kleinen Männchen keine Lust mehr. Zog ich den VS4-Prozessor und das Basicprom heraus, so konnte ich mit etwas Glück auch mit 4Mhz arbeiten. (aber unzuverlässig). Erstmal machte ich meine Änderung rückgängig. Die D-Rams kamen wieder rein und nun funktionierte die Kiste bei 4Mhz, aber nicht bei 8Mhz. So kaufte ich schnellere 4164 120ns(32k gibt's ja nicht). Der Erfolg war anscheinend da, aber der Minimonitor stieg trotz eines Waits zeitweise aus. Mit den Kondensatoren habe ich bis zum gehtnichtmehr herumexperimentiert, bis zum Schluß beide ausgebaut waren. Die kirtischste Stelle war der LS04 MPX und CAS-Generierung. Machte ich ihn mit Kältespray kalt, war der Zeitraum bis zum Ausstieg länger. Einige andere LS04 brachten auch keine Abhilfe. Herbert schickte mir ein neues PAL, aber auch kein

Hardware: 8 MHz-Erfahrungen von Horst Kupka (4019)

Erfolg. Warum das nicht klappte, kann ich nicht sagen. Die Zugriffszeiten waren, wenn das Gerät lief, doch ca. 120ns (mit dem Oszillografen nachgemessen). Kam ich mit der Tastspitze an eine gemultiplexte Adresse, sofortiger Ausstieg, es war eine windige Sache. Also baute ich mein statisches Ram wieder ein. Mit ihm lief die Sache ja auf 8Mhz.(einige Spiele brauchen aber 4Mhz)

Irgend etwas muß doch der Grund dafür sein, daß die 4 Mhz nicht funktionieren. Mit dem Oszi wurde nachgemessen. Nach einiger Zeit stellte ich folgendes fest: 1. Der CS-Ausgang vom PAL (Pin 14 von IC 6A) hat zwischendurch einen kurzen Einbruch, und mein S-Ram könnte dies möglicherweise erkennen und Unsinn bauen. (eine andere Ansteuerung muß her).

2..Der Takt Phi geht über Inverter und Bustreiber zur Floppy-Disk-Station. Hier geht er nochmals über einen Bustreiber und dann endgültig zu den einzelnen Karten. Auf dem Floppycontroller wird dieser Takt gebraucht um mit Hilfe eines Prom's neue WR-, RD-, IORQ- und MREQ-Signale zu erzeugen. Mit diesen Signalen werden dann die Daten von der Floppy in die Rams geschaufelt. Hier entstand also das Problem !

Auf dem Oszi sah man einen Versatz der Signale, wenn die Floppy Daten sendete. Diesen Versatz mochte wohl mein statisches Ram gar nicht. Nach einigem Probieren fand ich heraus: a) war ich mit der Tastspitze an dem Takt, nahm das Ram die Daten. b) überbrückte ich den Inverter, lief die Sache zeitweise, aber nicht zufriedenstellend, und c) ersetzte ich den Bustreiber durch einen HC244 (IC 5) klappte alles. Für zwei Sachen konnte ich mich nun entscheiden. 1. den Oszillografen mit einbauen (fällt aus, der Rechner würde zu schwer), oder 2. den Bustreiber auswechseln. (Ich habe den Bustreiber ersetzt.)

So nun kurz alles zusammengefaßt.

Die dynamischen 32k-Rams sind durch ein statisches ersetzt worden. Die Ansteuerung ist später beschrieben. Zeiten auf der 512k-Karte : über den 680 Ohm-Widerstand ist ein 330 gekommen.(wenn dieser R nicht eingebaut ist gings zeitweise auch). Der zugehörige Kondensator = 33pF. Der andere Kondensator hat den Wert von 27 pF.(mit 10pF gehts auch) Umschalten auf 4Mhz nur beim Booten, sonst nicht mehr. Extra Leitung zur Floppy auch nicht. Hier habe ich 40H-47H dekodiert, das zweite Momoflop vom LS123 genutzt und damit ein Wait erzeugt. Ich brauche nur langsamer werden, wenn ich etwas vom Floppycontroller-IC will. Hierbei entsteht ein großer Vorteil. Bei jeder Umschaltung von 4 auf 8 Mhz vergehen ca. 1 Sekunde (Zeit des Monoflops), diese Zeit habe ich nicht mehr. Die Platine auf der alles aufgebaut wurde, liegt nun an der Stelle, wo ehemals die dynamischen Rams waren.

Zum Netzteil folgendes.

Ich habe im Grundgerät alles rausgerissen und die Buchse von der FDX mit den Spannungen vom FDX-Netzteil belegt. Zu bemerken bleibt noch, das Verbindungskabel FDX-MTX sollte mit mindestens 1 Quadratleitungen verlegt werden. Ich habe ein Herdanschlußkabel (5adrig und flexibel) genommen, so ist der Spannungsabfall

Hardware: 8 MHz

relativ gering. Das Netzteil ist wie im vorherigen Info beschrieben nachgeregelte worden.

Aber sehr vorsichtig an die Sache gehen. Werden die 5 Volt erhöht, so erhöhen sich auch die 12 Volt. Bitte nicht viel höher als 5 Volt, sonst spricht die Crowbar-Sicherung an. (Crowbar deutsches Wort Brechstangensicherung) Es ist ein Thyristor, der bei Überspannung durchschaltet und so einen Kurzschluß vortäuscht bzw. erzeugt.

So nun zu den versprochenen Skizzen:

P.S. Das hier wurde alles mit 8MHz geschrieben.

Tschüß

Einige Variationen

(Herbert zur Nedden, 2000)

Im Laufe der Zeit hat sich noch einiges zu diesem Thema ergeben. Immerhin habe ich bei einigen MTX-en das 74LS02-IC, welches den 74LS123 ansteuert, anders beschaltet. Diese haben sich zum einen daraus ergeben, daß ich statt das Boot-EPROM zu tauschen schlicht und ergreifend mit 4 MHz boote, und zum anderen daher, daß es auch Systeme gibt, die lieber mit einem Wait laufen, welches natürlich nicht auch noch bei 4 MHz kommen soll.

Schließlich und endlich habe ich festgestellt, daß die Memotech-Silicon-Discs, d.h. die RAM-Floppys mit Formaten 40-4F, die in die FDX kommen Aversionen gegen 8 MHz haben.

Auf der nächsten Seite sind drei andere Möglichkeiten, den LS123 anzusteuern:

Probleme mit 8 MHz und den RAMs

(Herbert zur Nedden, 2000)

Horst Kupka (4019) wies mich darauf hin, daß es nicht genügt, die Kondensatoren aus dem RAM-Timing-Bereich (Verzögerung für MPX- und CAS-Signal) herauszunehmen, da die Widerstände zusammen mit den Eingangskapazitäten der dahinterliegenden Gatter immer noch Verzögerungen bewirken, da schließlich auch solche Gatter Eingangskapazitäten haben. Daher kann es sich lohnen die Widerstände zu überbrücken!

Floppy-Zugriff mit 8 MHz

(Herbert zur Nedden, 2000)

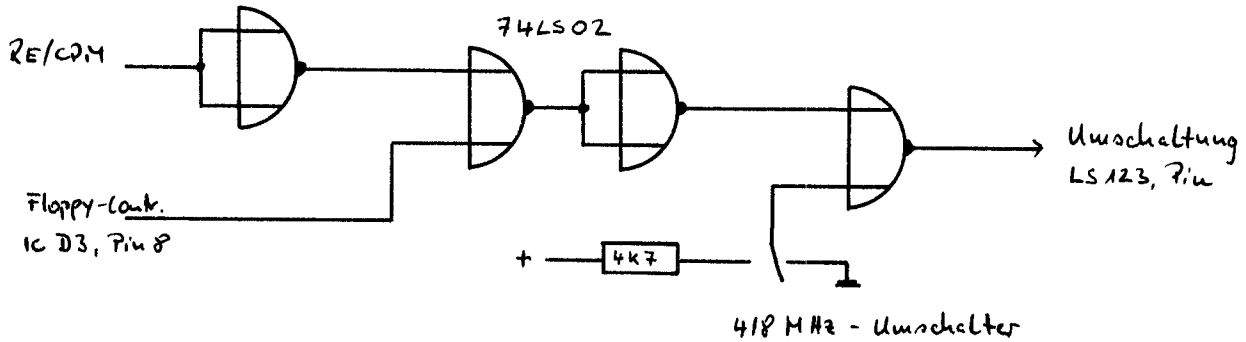
Horst Kupka (4019) hat sich mal der Problematik des Floppy-Controllers angenommen, und festgestellt, warum es zu Problemen kommt. Schuld ist einzig und allein der Controller-Chip, der in Punkto Zeitbedarf zum Übernehmen von Daten mit unserem Video-Controller auf der Hauptplatine einiges gemein hat: Er benötigt sage und schreibe 350 ns um Informationen zu transferieren.

Was ist naheliegender, als diesen Wunsch mittels WAITs zu befriedigen, und dann die Kiste nicht bei Diskettenzugriffen auf 4 MHz umzuschalten. Gesagt, und getan ... LÄUFT!

Horst hat zu diesem Zweck auch gleich den einen freien halben LS123 verwendet. Hierbei handelt es sich um ein MonoFlop, d.h. eine Schaltung, die nach eingehen eines Impulses für eine gewisse Zeit (abhängig vom angelöteten Kondensator und Widerstand) den Ausgang umschaltet. Der Schaltplan von Horsts 8 MHz-Aufrüstung, die übrigens auf der Hauptplatine mit statischen RAMs arbeitet ist auf der übernächsten Seite.

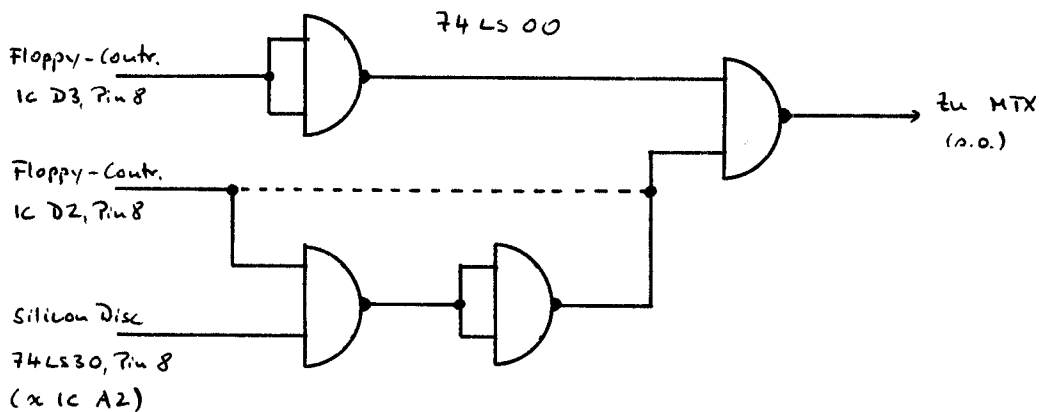
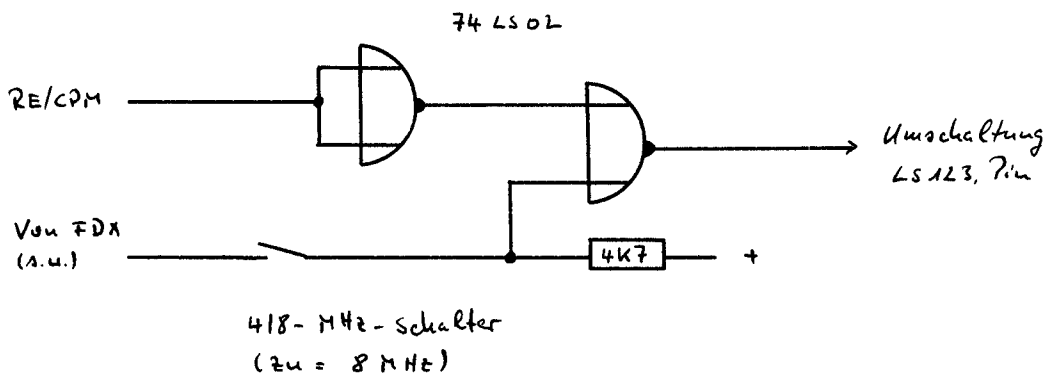
Hardware: 8 MHz

Booten mit 4 MHz



Booten mit 4 MHz

Booten mit 4 MHz, Einbindung der Memotech-SiliconDisc



Falls keine Memotech-SiliconDisc in der FDX ist, können die beiden unteren LS00-Gatter entfallen, und die Leitung von FDX D2/8 wird entsprechend der gestrichelten Linie angeschlossen. Falls das Kabel von der FDX zum MTX in der o.g. Schaltung lang ist, sollte an das MTX-seitige Ende ein Kondensator 22pF gegen Masse.

K o m i k: CYBER-Mitteilung Nummer 7411

Aus der weiten Welt der EDV (entdeckt von Hans-Joachim Ernst - 8554)
Das sind ja tolle Aussichten:

CYBER - Mitteilung Nr. 4711

Wie aus den jüngsten Veröffentlichungen der Amerikanischen Gesellschaft für digitale Impulsunterdrückung (ASPIC) hervorgeht, ist es nach jahrelanger Forschungsarbeit gelungen, ein Software-Programm zur Ausschaltung von Hardware-Fehlern zu entwickeln. Die Routine wird in Fachkreisen als Meilenstein in der Geschichte der Systemtechnik bezeichnet, da sie die immensen, durch Funktionsfehler bedingten Zeitverluste weitgehend ausschaltet.

Der Programmname OREMA ist vom lateinischen 'oremus' (lasset uns beten) abgeleitet. Die Routine ruft in bestimmten Zeitabständen Gebete ab, in denen himmlischer Beistand für die Bewahrung der Speicherinhalte, die Betriebssicherheit der Drucker und solcher Löt- bzw. Wrapstellen erfleht wird, die besonderen mechanischen oder kybernetischen Anfechtungen ausgesetzt sind.

Die liturgische Struktur der OREMA-Routine gestattet den wahlfreien Abruf wirkungsvoller extern gespeicherter Stoßgebete in lateinisch, hebräisch oder Assembler. Durch die OREMA-Unterroutine MISERERE werden dreimal täglich sogenannte zyklische Messen zelebriert (äußeres Merkmal: alle Bandgeräte, Platten- und Diskettenlaufwerke verharren in andächtiger Ruhe). Die ministrierenden Operateure beschränken sich während dieser Zeit auf das Einlegen von Testbändern und das Eingeben sakraler Nachrichten über das Terminal (z.B. "und mit Deinem Geiste" oder "Kyrie eleison").

Litaneien in Hebräisch oder Assembler können direkt über einen Freien Kommunikationskanal (FKK) in die Zentraleinheit eingegeben werden. Lateinische Psalmen (Vorsatz: TE-DEUM) sind jedoch über Band einzuspielen. Litaneien in Hebräisch oder Assembler haben sich als besonders wirkungsvoll erwiesen.

Obwohl von den Herstellerfirmen heute bereits fertige Gebetsbänder für alle denkbaren Maschinenfehler geliefert werden, kann der Operator von sich aus hinter jedem Amen-Block ausblendbare Sätze mit Eigenerfahrungen eingeben. Spezielle Nothelfergebete (IRIS) stehen für wichtige Tests zur Verfügung.

Nach letzten Testläufen mit OREMA wird die durchschnittliche Maschinenausfallzeit um 98,2% reduziert. Wie aus gut unterrichteten Kreisen verlautet, wird OREMA für absehbare Zeit nur für die Ausmerzungen von Hardware- und Fertigungsfehlern eingesetzt werden können.

Es wurde jedoch inzwischen ein kybernetisch-theologischer Arbeitskreis eingesetzt, der sich mit der Entwicklung eines Software-Systems zur Abschaffung des menschlichen Versagens befaßt. Die Fertigstellung dieser Programme (Arbeitstitel MEACULPA) ist wohl nicht vor 1989 zu erwarten.

Das Programm OREMA steht für die CYBER vom 1. Mai 1987 an zur Verfügung und kann bei Bedarf in bereits erstellte Programme mit eingebunden werden.

Am 1. April 1987

(A.E. Neumann)

Hardware: Reparatur**Er will nicht arbeiten**

(Herbert zur Nedden, 2000)

Hier nun einige Kniffe, mit denen der MTX samt Diskettenstation wieder zum vernünftigen Arbeiten bewegt werden kann.

Dies sind keine Patentrezepte, die stets Erfolg garantieren!

Diese Tips können zum einen helfen Boot-Probleme zu beseitigen, und zum anderen die Betriebssicherheit des MTX samt FDX/SDX zu erhöhen.

- Er will nicht einmal als MTX ohne SDX/FDX laufen: Häufig ist in diesem Fall das Netzteil auf der Hauptplatine defekt. Das wiederum kann zweierlei Ursachen haben: 1. & 2. Langt das nicht, versuche 3.
- Er will, aber nicht so recht. Booten nur nach gutem Zureden, ... Hier sind meist schlechte Kontakte die Ursache: 3. - 7.
- Einige Tasten wollen nicht, oder prellen: Versuche 10. & 11.
- Bild auf dem Bildschirm merkwürdig: Versuche 13. - 15.
- Probleme zusammen mit ECB-Bus: Siehe 19. & 20.
- Was sinnvoll ist: Alles unten genannte!
- Wenn die Kiste immer noch nicht will: (040) 200 87 04

Hier nun einige Ideen

1. Die Lötstellen im Bereich des Netzteils auf der MTX-Hauptplatine sind bei einigen MTXen kalt, d.h. nicht anständig gelötet. Das erkennt man daran, daß eine deutliche Trennlinie zwischen den Bauteil-Beinchen und dem umgebenden Lötzinn zu sehen ist, oder gar Bauteile mit etwas Ruckeln herausnehmbar sind. Hier hilft nur sorgfältiges Nachlöten. Insbesondere bei den Dioden ist es recht sinnvoll zuvor die Beinchen mit einer feinen Feile oder Sandpapier etwas zu reinigen. (Ich löte diese immer mit 400 Grad nach, da 350 Grad kapp sind.)
2. Die beiden dicken schwarzen Dioden, die rechts am Rand der MTX-Hauptplatine sitzen sind hinüber und müssen ersetzt werden (Typ 1N5402). In diesem Fall steigt Spannung statt auf 5 Volt häufig nur auf unter 4 Volt - in Glücksfällen sogar auf 4,5 Volt an.
3. Der Takt kommt nicht anständig (Takt = Sinuskurve), falls das IC9D auf der MTX-Hauptplatine ein 74S04 ist. Um diesen Defekt zu beheben ist folgendes zu tun: IC9D durch einen 74HC04 ersetzen, Widerstände R32 und R33 durch je 330 Ohm ersetzen, und Widerstände R35 und R36 entfernen.
4. Das Flachbandkabel zwischen MTX und FDX ist zu lang. Es gibt zwar MTX/FDX-Systeme, die ein 100 cm langes 60-poliges Flachbandkabel gnädig akzeptieren, aber warum diese Geräte nicht regelmäßig abstürzen ist mir schleierhaft!

Hardware: Reparatur

5. Der 60-polige Flachbandkabelstecker, der in der FDX auf der kurzen Interface-Platine ganz unten im FDX-Bus sitzt steckt nicht ganz drauf: Festdrücken! (es genügt hierfür vollkommen, nur die FDX-Rückwand abzunehmen - 4 Schrauben)
6. Der 60-polige Flachbandkabelstecker, der im MTX auf der RS232-Karte sitzt steckt nicht ganz drauf (Dies ist das Kabel zur FDX).
7. Die Karten im MTX (Hauptplatine, Speichererweiterung und RS232-Karte bei FDX-Systemen, bzw. Hauptplatine, RS232/BOZeichen-Karte und Floppycontroller bei SDX) stecken nicht anständig zusammen. Auseinanderziehen, Kontakte evtl. mit TUNER600 einsprühen, und kräftig zusammenstecken. Falls die verzinnten Kontakte richtig schöne schwarze Streifen aufweisen (der Fachmann spricht von Oxidation) ist es ratsam diese mit einem sauberen Radiergummi vorsichtig abzuradieren. Die Verwendung der Finger ist wenig ratsam. Das selbe gilt für die meisten Kontaktsprays, Reinigungssprays, Domestos und ähnliche Produkte.
8. Die BO-Zeichenkarte in der FDX (das ist die Karte in der FDX mit dem Kabel dran, welches zur FDX-Rückwand verläuft) hat keinen anständigen Kontakt. Wackeln, besser noch Herausnehmen, TUNER 600, Hineinstecken hat sich bewährt.
9. Einige MTX/FDX-Systeme danken es einem, wenn das IC6 auf der RS232/Z80-Karte (74LS241) durch ein 74HC241 oder gar 74ALS241 ersetzt wird. Dies halte ich allerdings für eine Notlösung!
10. Falls mehrere Tasten nicht wollen, ist meistens eine Leitung des Tastaturkabels gerissen.
11. Wenn Tasten stark prellen, sollten die Kontakte der betroffenen Taste unter der Tastaturplatine nachgelötet werden.
12. Die Video-RAMs (4116) auf der Hauptplatine sind sehr empfindlich, und überleben den Betrieb ohne die minus 5 Volt nicht. Falls diese hinüber sind (d.h. kein VS4-Bild kommt, oder dieses immer vermurkst ist) sollten diese gegen neue ausgetauscht werden. Am besten werden aber gleich 32k-DRAMs eingesetzt, da diese unempfindlicher sind, und bei mir billigst zu haben sind! Umbauanleitung: siehe Info 22, Seite 12.
13. Auf der BOZeichen-Karte tauchen nach etwa einer Bildschirmseite plötzlich diverse Sonderzeichen oder was auch immer auf. In diesem Fall ist meistens nichts kaputt, sondern lediglich die BOZeichen-Karte auf 8 kB Video-RAM aufgerüstet. Je nach Anwendung kann dem abgeholfen werden: Soll FDX-BASIC laufen ist das wurscht! Einfach aufrufen! Auch der Aufruf von RAM4 wirkt Wunder. In allen anderen Fällen hilft folgendes: (/ = RET)


```

A>DDT/
S100/
3E/1F/32/AB/F6/32/96/F7/32/BO/F7/32/BD/F8/32/CD/F8/C9../
^C
A>SAVE 1 SCREEN.COM
A>SCREEN

```

 Bei späteren Gelegenheiten genügt der Aufruf von SCREEN, falls es nicht inzwischen gelöscht wurde!

Hardware: Reparatur

14. Sind streifen auf dem Bildschirm, oder flackert dieser, wenn ein Laufwerk anläuft, so ist das ein Zeichen, daß das Netzteil zu schwach ist. (Bei der SDX ist dies fast normal!) MTX-Netzteil verstärken, indem der mittlere der drei Bräter auf der MTX-Hauptplatine (7805) gegen einen 78S05 ausgetauscht wird, und der fette grüne Widerstand neben den ROMs auf der Hauptplatine (R62 - auch wenn daneben noch R60 steht!) gegen einen Widerstand mit 2,7 Ohm und 5 Watt ausgetauscht wird.

Falls der MTX von außen mit Gleichspannungen versorgt wird, siehe Punkt 17.

15. Es ist garnicht einmal so schwierig, einer SDX-Anlage zu einer anständigen Stromversorgung zu verhelfen. Hinten am SDX-Controller, und an der rechten MTX-Außenwand sind ja 3,5-mm-Buchsen (d.h. solche wie sie auch der Cassettenport hat). Bleiben diese Buchsen leer, werden der Controller und die 80Zeichen/RS232-Karte von der Hauptplatine aus mit 5 Volt versorgt. Werden jedoch über diese Buchsen von außen 5 Volt zugeführt, wird die Stromversorgung von der Hauptplatine abgetrennt. Damit ist es leicht möglich, den Floppy-Controller bzw. die 80Zeichen/RS232-Karte extern mit Strom zu versorgen, was auch in Anbetracht des schwachen Netzteils der Hauptplatine sinnvoll ist!

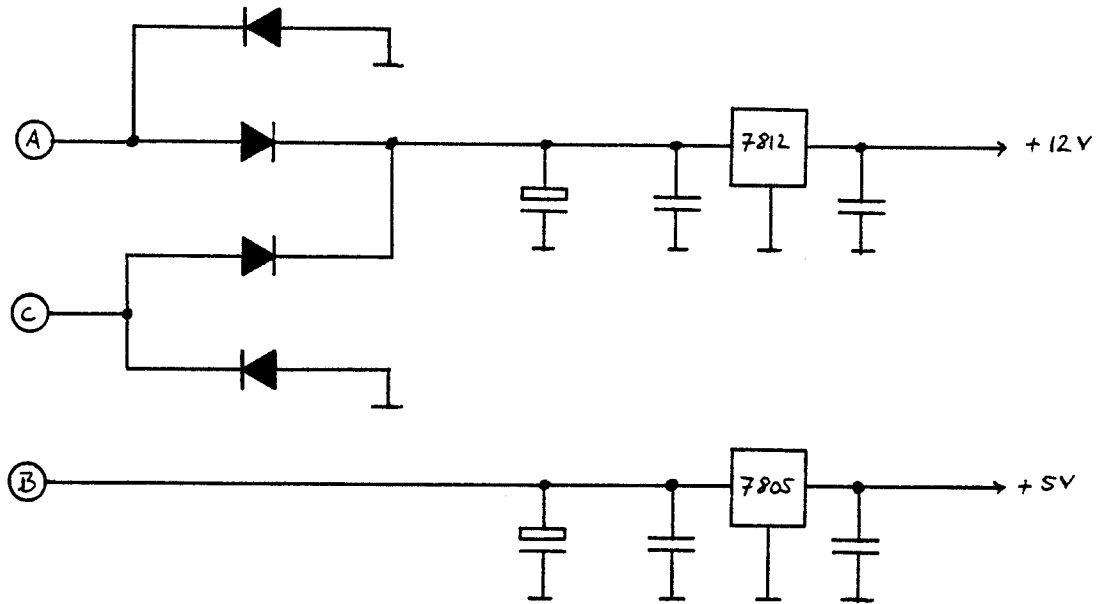
Im Original-Aufbau wird die 80Zeichen/RS232-Karte vom Netzteil der Hauptplatine mitversorgt, was eine reichlich schlappe Versorgung ist. Der Floppy-Controller hingegen erhält seinen Saft aus dem Netzteil des Laufwerkes. Die einfachste Lösung ist es, ein kleines 5-Volt-Netzteil folgender Art zusammenzustricken, und diese 5 Volt über die 3,5-mm-Buchse an der rechten Außenwand des MTX einzuspeisen. Natürlich könnte ich auch einfach die 5 Volt des Netzteils des Laufwerkes statt an den Controller zu geben rechts am MTX auf die 80Zeichen/RS232-karte einspeisen, aber dann habe ich immer schöne dunkle Balken auf dem Bildschirm, wenn das Laufwerk läuft.

16. Wer will kann natürlich auch das Netzteil im Laufwerk aufpeppen, indem dieses mittels eines externen Trafos (am einfachsten einen "Campingwagen", also ein MTX-Trafo) verstärken. Dazu müssen allerdings auf der Netzteil-Platine im Laufwerksgehäuse kleine Umbauten vorgenommen werden. Hier ist der Schaltplan des Originals und der der geänderten Version. Die 5 Volt dieses Netzteils sind dann der 80Zeichen-Karte über die Buchse rechts am MTX zuzuführen. Der Floppy-Controller hingegen wird dann von der Hauptplatine mitversorgt. Es ist zu empfehlen ein dickes 5-Volt-Kabel vom Netzteil der Hauptplatine zu der Controller-Platine zu verlegen, damit dort der Saft besser ankommt. Natürlich muß die Buchse hinten an der Controller-Platine leer bleiben!

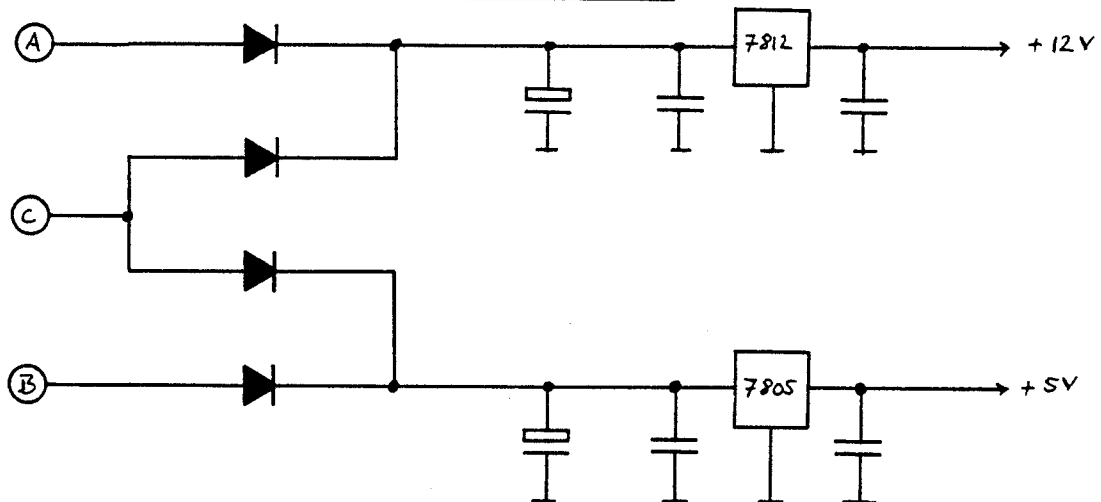
Schaltpläne auf der nächsten Seite (Herbert zur Nedden, 2000).

Hardware: Reparatur

Original-Netzteil im SDX-Laufwerk



Verstärktes Netzteil aus dem SDX-Laufwerk



Der MTX-Campingwagen wird wie folgt angeschlossen:

Campingwagen Pin	1	2	3	4	5
Netzteil	A	B	Masse		C

Beachte bitte, daß zwei Dioden jetzt gedreht sind!!!

Hardware: Reparatur

17. Wenn die Hauptplatine von außen mit fertigen Gleichspannungen versorgt wird, d.h. die Netzteilgeschichte ist aus dem MTX entfernt, dann müssen ggf. diese Spannungen überprüft werden. Als erstes sollte sichergestellt werden, daß das Kabel mit dem die Spannungen zugeführt werden dick ist (Netzkabel), da sonst leicht viel Leistung im Kabel verbraten wird.

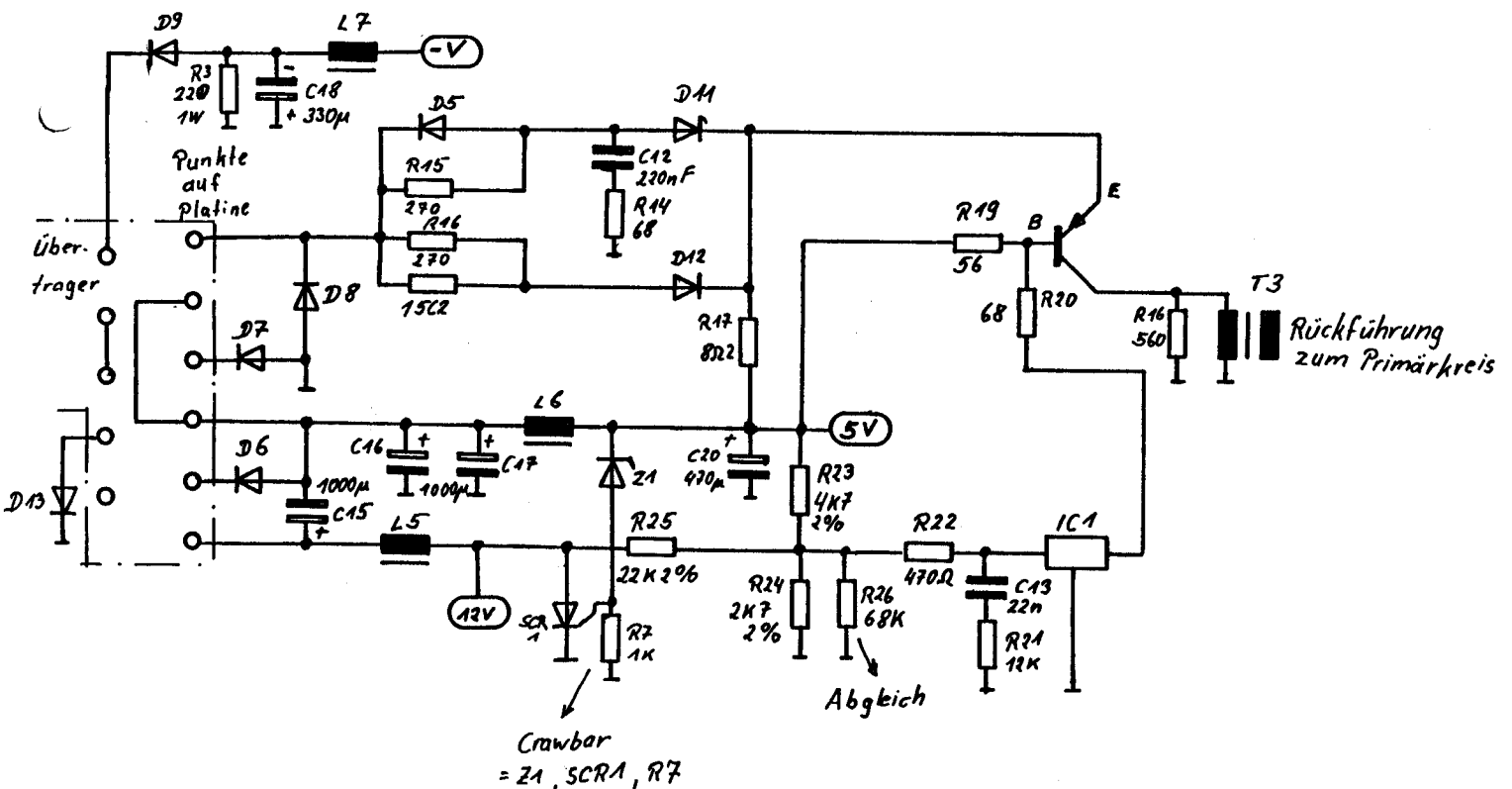
Bei den meisten Netzteilen kann die Ausgangsspannung nachgeregelt werden, was allerdings mit Vorsicht und nur unter Zuhilfenahme eines guten Meßgerätes in Angriff genommen werden darf. Versorgt dieses Netzteil auch andere Teile des Computers, darf die Spannung nicht nur aufgrund der im MTX ankommenden Werte nachgeregelt werden - sonst sind evtl. anderswo 5,5 Volt, wenn der MTX gerade seine 5 Volt bekommt.

Falls das FDX-Netzteil schuld ist, wird es knapp! Dieses kann gemäß Info 21, Seite 34 aufgepeppt werden.

ACHTUNG: Das FDX-Netzteil darf nicht auf eine Spannung von ca. 5,2 Volt oder mehr hochgeregelt werden, da sonst die Spannungsüberwachung (mittels Crawbar) im Netzteil dieses Kurzschließt, um den Computer zu schonen. Dadurch fliegt i.a. lediglich die Sicherung des Netzteils raus.

Hier ist der Schaltplan der Niedervolt-Seite des FDX-Netzteils:
(Horst Kupka, 4019)

NV - Seite des FDX-Netzteils



Hardware: Reparatur

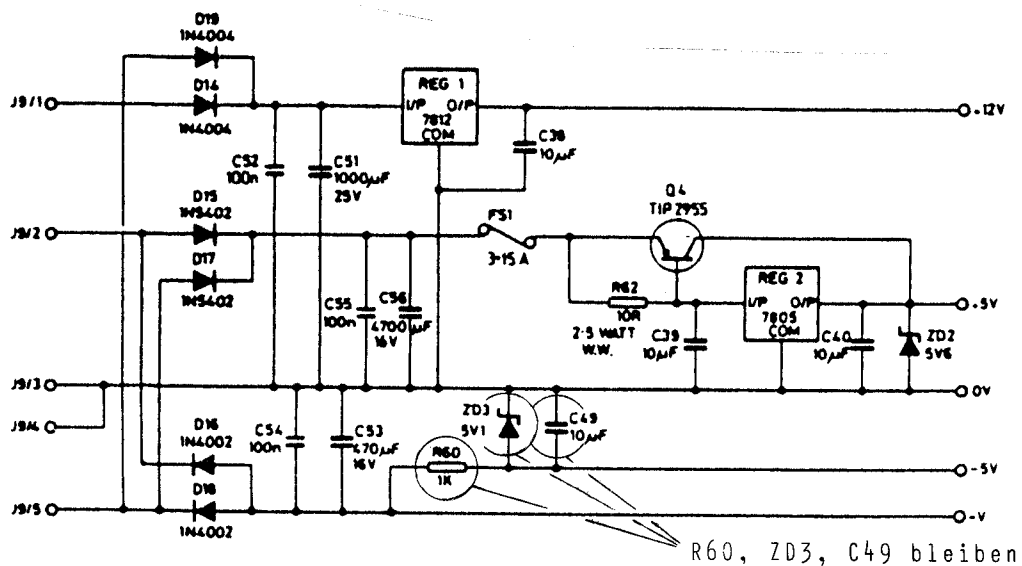
18. Fast jede MTX/FDX- und MTX/SDX-Anlage weiß eine neue kräftige Stromversorgung aus einer einzigen Quelle zu schätzen! Dazu sollte ein getaktetes Netzteil mit folgenden Daten her:

- 5V, nicht unter 6 A, bei über 2 Laufwerken, vollem ECB-Bus mehr
- 12V, 0.5 A plus je Laufwerk bis zu 1 A (je nach Hersteller)
- 12V, 0.5 A

Dieses sollte dann alle Verbraucher versorgen, also auch insbesondere die Hauptplatine, deren Stromversorgung demzufolge zu entfernen ist (siehe Schaltplan unten). Wichtig ist, daß die Leitungen für die Stromzufuhr nicht mit obskuren Steckkontakten versehen werden, alle Stromkabel mit dickem Draht (am besten Netzkabel) versehen werden, und diese möglichst kurz und gehalten werden müssen! Verwende ich das Original-Memotech-Kabel, welches den MTX mit Saft aus der FDX versorgt, um hierüber Gleichspannungen zu führen, kommen statt der 5 Volt problemlos lumpige 4.5 Volt im MTX an. Ja, ich weiß, daß man die Netzteile hochregeln kann, aber ob die Karten in der FDX die dann dort ankommenden 5.5 Volt mögen ?

Schaltplan des MTX-Netzteils

Die eingekreisten Bauteile müssen bleiben (der Rest weg), damit die -5 Volt aus den -12 Volt erzeugt werden können.



19. Die Busterminierung von Uwe Grass erwartet, daß ihr ziemlich genau 5 Volt spendiert werden. Die c't-Statik-RAM-Floppy übrigens auch! Wie das erreicht werden kann steht etwas weiter oben.

Evtl. kann es einiges bringen, die beiden Widerstände, die die 5 Volt auf der Busterminierung teilen etwas zu verändern, um die Spannung an den Eingängen der Operationsverstärker zu verringern. Die beiden Widerstände weisen im Original 3,3 kOhm gegen +5 Volt, und 4,7 kOhm gegen Masse aus.

Schaltplan: Info 16, Seite 25.

Hardware: Reparatur

20. (Uwe Grass, 3300)

Es gibt doch tatsächlich so merkwürdige Phänomene, wie das folgende: Eine FDX wollte nicht formatieren sowie die Terminierung auf dem auf den ECB-Bus gesteckt wurde.

Nun hat Jan Brederke in mühseliger Fummelei festgestellt, daß diese Probleme verschwinden, wenn die Adressen A8bis A15 nicht mehr terminiert werden. Das sind auf den V664-Steckern die Anschlüsse c8, a30, c18, c17, c27, a29, a18, c28. Ebenfalls bleiben folgende Anschlüsse von den meisten Karten ungenutzt: RFSH a28, PHI c29, NMI c20, INT c21, MREQ c30. Wenn also nach Anschluß des Adapters Fehler auftreten sollten, hilft es oft, wenn die nicht benötigten Leitungen von der Terminierung getrennt werden.

Aber kneift diese nicht gleich ab! Es gibt auch ECB-Karten, die dies überhaupt nicht mögen!!!! Also nur dann, wenn es anders nicht geht, und Euch den Rückweg offen halten!

21. Platinen, die im MTX (Hauptplatine, RS232-Karte, Speichererweiterung, 80Zeichen/RS232-Karte, SDX-Floppycontroller) stecken fest verlöten. Hierzu müssen lediglich die Stecker zerbrochen werden, was sich mit einer Spitzzange gut bewerkstelligen läßt: Einfach die Plastik-Backen nach außen wegbrechen. Ihr könnt sie natürlich auch absägen - aber vorsichtig, damit die Metallkontakte nicht auch gleich weg sind. Dann die Karten so zusammenstecken, daß ein halber Millimeter Spiel bleibt. Nun die beiden zu verlötenden Karten in die Bodenwanne des MTX legen, und die beiden äußeren oberen Kontakte verlöten - damit sind die Karten schon einmal zusammen. Jetzt alle Kontakte auf der Unterseite verlöten, dann die Karten wieder in die MTX-Bodenwanne legen, und erst jetzt die oberen Kontakte verlöten.

Warum so: Stecken die Karten ganz dicht aneinander gibt es Spannungen (nicht in Volt, sondern in Biegung). Löte ich erst die oberen Kontakte, ist es schwierig, die unteren so zu löten, daß die Karten gerade hintereinander sitzen! Dabei muß sichergestellt werden, daß keine der angelöteten Kontakte auf der Unterseite der Platinen so weit abstehen, daß sie Bodenberührung haben!

22. Im MTX sollte der Widerstand R14, der links neben den RAM's sitzt 330 Ohm haben (orange, orange, braun) und nicht 680 Ohm (blau, grau, braun). Falls es ein 680-er ist, kann einfach ein zweiter 680-er darüber gelötet werden, oder natürlich dieser auch gegen einen 330-Ohm Widerstand eingetauscht werden.

23. Auf der 80Zeichen-Karte vom IC 1C (74LS74) Pin 5 über der Platine abkneifen, und unterhalb der Platine die Pins 5 und 8 verbinden. Für die 80Zeichen/RS232-Karte der SDX-Stationen ist dies irrelevant.

Hardware: Interrupts und ECB-Bus

Der Interrupt Modus 2 am ECB - Bus im MTX / FDX

(Wolfgang Zierenberg, 3451)

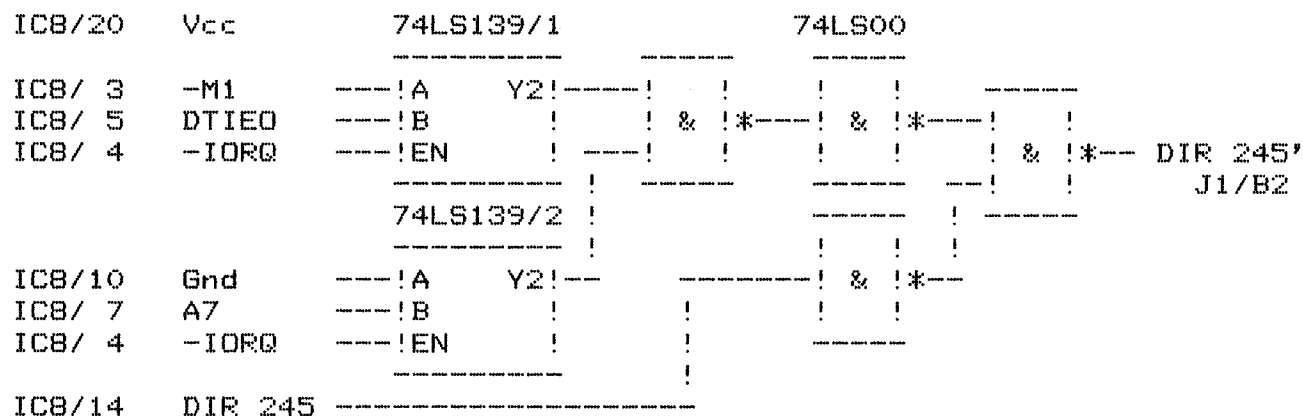
Hardware

Im IM 2 kann ein Peripherie - Baustein aus der Z80 - Serie durch Setzen der INT - Leitung auf 'low' der CPU seine Interruptanforderung mitteilen. Gleichzeitiges Auftreten der Signale M1 und IORQ unter Berücksichtigung der IEI/IED - Leitungen fordern ihn dann auf, das untere Byte des Interruptvektors auf den Datenbus zu legen. Das obere Byte des Interruptvektors ist im I - Register der CPU enthalten. (Eingehendere Information in c't 84, Heft 9, Seite 56 ff.)

Wie bei der ECB - Option sind die notwendigen Signale an dem PAL auf der RS232- Platine zu erreichen. Der DART stellt durch das Signal DTIED das letzte Signal der IEI / IEO - Kette zur Verfügung, d.h. alle IO - Bausteine auf der Grundplatte und der RS232 - Platine haben eine höhere Priorität als die auf dem ECB -Bus, was auch erwünscht ist (Betriebssystem, RAM4 ..). Durch eine Verknüpfung der Signale -M1, -IORQ und DTIED einerseits und A7 und -IORQ (wie bereits in der ECB - Option) andererseits muß der Datenbustreiber auf der FDX - Treiberplatine so geschaltet werden, daß die Datenrichtung bei Interruptanforderungen unter Berücksichtigung ihrer Priorität und Portzugriffen bei I/O Adressen >#7F vom ECB -Bus auf den FDX - Bus ist.

Ausserdem muß verhindert werden, daß bei einer Interruptanforderung durch Herabziehen der INT - Leitung mehrere Ausgänge von TTL - Bausteinen gegeneinander geschaltet werden. Durch Einfügen einer Diode zwischen Ausgang und Flachkabel auf der Treiberplatine der FDX ist das möglich (technisch nicht einwandfrei, aber praktikabel). Dabei wäre zu überlegen, ob auch die -WAIT - und -BUSREQ - Leitungen auch mit Dioden versehen werden sollten.

Bus Pin 50 -Wait die Anoden der Dioden müssen
 Bus Pin 51 -BUSREQ auf das Flachkabel zeigen
 Bus Pin 53 -INT



Bei Verwendung der Platine (ist genügend Platz vorhanden ?) kann das PAL auf die Platine gesteckt werden, die IC - Beinchen der Fassung müssen verlängert werden und können in die Fassung des PAL gesteckt werden.

(zur Vereinfachung des Layouts liegt Eingang A des 74 LS 139/2 nicht an GND sondern auch an -IORQ).

Hardware: Interrupts und ECB-BusSoftware

Dieses Beispiel setzt RAM4 im Speicher voraus, kann aber auch allgemein verwandt werden. Zunächst muß in der Interrupttabelle nach freiem Platz für den Zeiger auf die Interruptroutine gesucht werden, der mit den Hardwareeigenschaften des I/O - Bausteines übereinstimmt (CTC - 4 Worte, PIO ev. nur 1 Wort, siehe Datenblätter). Dann müssen bei der Initialisierung des Programmes entsprechende Einträge in die Tabelle gemacht werden. Das kann z.B. so aussehen:

```
ld a,i           ;das Register I der CPU enthält das höherwertige Byte
ld h,a          ;des Zeigers auf die Tabelle
ld a,xx         ;jetzt enthält A das niederwertigere Byte xx
ld l,a          ;und HL zeigt auf den Eintrag
ld a,low intr
ld (hl),a
inc hl
ld a,high intr  ;der Eintrag ist in der Reihenfolge niederwertiges -
ld (hl),a       ;höherwertiges Byte gemacht
```

Das eigentliche Interruptprogramm muß in der common bank liegen, d.h. zwischen C000 und FFFF Hex, sinnvollerweise unterhalb von TOAM, die benötigten Register rettet und nach Beendigung wiederherstellen.

Anm.d.HzN.: Da die Routine im Common-Bereich liegt, ist eine Umschaltung auf Bank 0 unnötig - das gerade ist ja der Witz des Common-Bereiches!

Da Lage von TOAM je nach Rechner unterschiedlich sein kann, ist die absolute Adresse stackp mutig! Klappt nicht immer, da nicht jeder Rechner an der gleichen Stelle Platz hat!

```
intr: ld (stackp),sp ;einen eigenen Stack einrichten (in Bank 0, >#C000)
      ld sp,stackp-1
      push af        ;die benötigten Register retten
      .
      .
      push iy
      nop           ;das Programm
      pop iy       ;die Register wiederherstellen
      .
      .
      pop af       ;alte Speicherbank wieder anwählen
      ld sp,(stackp) ;auch SP mit dem ursprünglichen Wert füllen
      ei          ;Interrupts wieder zulassen
      reti        ;und zurück !
```

Hardware: Interrupts und ECB-Bus**Interrupt-Routine samt Loader**

(Herbert zur Nedden, 2000)

(Idee für Loader aus Edicta-Routinen von Olaf Krumnow, auch 2000)

```

TOAM      equ      0046h
FREE      equ      0040h

Loader:   ; Lade-Programm

          ; pruefe, ob zwische TOAM und FREE genug Platz ist
          ld       hl,(TOAM)
          ld       de,(FREE)
          sbc      hl,de                      ; Groesse des Freiraums
          ld       de,EndOvr-Overlay
          sbc      hl,de
          jp       c,NoPlace                 ; nicht genug Platz

          ; setze Aufruf-Routine unter TOAM
          ld       hl,(TOAM)
          ld       bc,EndOvr-Overlay
          sbc      hl,bc
          ld       (TOAM),hl
          ; berechne alle erforderlichen Adressen
          ex       de,hl
          ld       hl,Stack-Overlay
          add      hl,de
          ld       (StkAdr1),hl
          ld       (StkAdr3),hl
          ld       hl,OwnStack-Overlay
          add      hl,de
          ld       (StkAdr2),hl
          ld       hl,(StartAdr)
          ld       (CallAdr),hl
          ld       a,(StartBnk)
          ld       (ZielB2),a

          ; Routine verschieben
          ld       hl,Overlay
          ld       bc,EndOvr-Overlay
          ldir

          ; Erfolgsmeldung
          call     putstr
          db       13,10
          dc       'erfolgreich eingebunden.'

PrgEnde: call     putstr
          db       13,10
          dc       'Programm beendet.'

          jp       0                          ; Warmboot

NoPlace: call     putstr
          db       13,10
          dc       'Kein Platz zwischen TOAM und FREE'
          jp       PrgEnde

```

H a r d w a r e: Interrupts und ECB-Bus

```

; PutStr erwartet, daß der String ab der Speicherstelle
; nach dem CALL PutStr - Befehl steht, d.h. ab der durch
; den CALL auf den Stack gePUSHTen Adresse.
; Das Ende des Strings wird durch ein gesetztes 7. Bit
; im letzten Zeichen gekennzeichnet. Der MACROBO-Assembler
; erledigt dies durch die Verwendung von DC statt DB bei
; der Definition des letzten Teils der Nachricht.

```

```

PutStr:   ex    sp, (hl)           ; Adresse des Strings
PutLoop:  ld    c, (hl)
          res   7, c              ; Lösche Bit 7
          call Offd3h            ; System-Sprung
          bit   7, (hl)          ; Teste Bit 7
          inc   hl               ; Nächstes Zeichen
          jp    z, PutLoop       ; Next
          ex    sp, (hl)        ; Rücksprungadresse
          ret

```

```

Overlay: ; Dies ist die eigentliche Interrupt-Routine

```

```

          ld    (0), sp          ; rette Stackpointer Original
StkAdr1  equ   #-2
          ld    sp, 0           ; setze eigenen Stack
StkAdr2  equ   #-2
          push  ...             ; Rette alles
          nop                   ; Programm
          pop   ...             ; Entrette alles
          ld    sp, (0)         ; hole alten Stack zurueck
StkAdr3  equ   #-2
          ret
Stack:   ds    2, 0
          ds    20, 0          ; Stack-Platz
OwnStack:
EndOvr:  ; Ende des Overlays

          end

```

Das Interrupt-Programm darf keine LD-, CALL- und JP-Befehle auf Teile von sich selbst haben, es sei denn auch diese Adressen werden vom Loader richtig umgerechnet. Die Umschaltung auf Bank 0 ist wie gesagt völlig überflüssig.

Leserbrief: Hagen Wenzek, 5300

Na ja, nachdem mich das letzte Info erreicht hatte und ich in der Kurzmeldung zum letzten Clubtreffen gelesen habe, daß ihr an eine Tastaturanschluß an der RS232 denkt, wollte ich mich auch mal wieder melden.

Vor einiger Zeit habe ich eine KeyTronic Tastatur von Bühler für ca. 8,- DM erstanden, nur mit dem Anschluß haperte es immer.

Nachdem ich endlich die Anschlußbelegung mit Schaltplan (mehr Abfallprodukt) fertig hatte und die Tastatur netterweise mit 9600/8/1/N sendet, habe ich eine Tastaturdekoder-Tabelle ins SysIO einbinden wollen. Natürlich habe ich erst(!) das Programm geschrieben und #dann# gesehen, daß dort überhaupt kein Platz mehr ist und ich mir meinen eigenen alten Tastatortreiber überschrieben hatte.

Seitdem habe ich aus Mangel an Zeit (und Lust) nicht mehr weitergemacht.

Was habt ihr denn bisher so gemacht, gedacht, geplant, geschrieben? Was für eine Tastatur soll an den Port, IBM-Kompatibel (XT,AT,PS/2) ?

Aufgebaut sind die Tastaturcodes folgendermaßen:

Zuerst ein Byte, das den Status der Zusatztasten (SHIFT...) angibt, dann den Code der Taste selber.

Das erste Byte sieht so aus:

```

1 0 1 X   X X 0 X
      |   | |   |___SHIFT
      |   | |___FCTN (<=> CTRL)
      |   |___REPEAT (Taste gedrückt halten)
      |___LOSLASS (Taste nach Repeat loslassen)

```

Nur die Cursorstasten haben immer den "LOSLASS" Code nochmal hinter dem Tastencode und ein NUL.

Ansonsten habe ich bis jetzt nicht viel gemacht. Einen Eprommer, den aus der mc, gebaut, nur habe ich das Programm dazu erst diese Woche zum Laufen bekommen, so muß ich jetzt erst mal Testen.

Weiterhin werde ich in nächster Zeit einen IC-Tester aus der Byte nachbauen. Ich warte nur noch das nächste Heft ab, um mir die Software dafür schicken zu lassen.

Mein Rechner hat dazu noch irgendeinen blöden Wackler rumliegen, der es vermeidet, daß das Motherboard mit dem FDC kommuniziert.

Es dauert immer eine viertel Stunde bis die Kiste läuft.

Das wars schon wieder

Tschüß

Hagen

DoMessDOS: Benchmark

(Manfred FLume, 6500)

Hier ein Leistungs-Vergleich eines IBM-Kompatiblen mit dem MTX. Natürlich auch gleich für verschiedene Taktfrequenzen. Damit Ihr auch wißt, was sich so abgespielt hat (in Turbo-Pascal):

1. Integer-Arithmetik:
`x:=0; y:=3;
 For i:=1 to 10000 Do x:=x+(y*y-y) Div y;`
2. Real-Arithmetik:
`x:=0.0; y:=9.9;
 For i:=1 to 10000 Do x:= x+ (y*y-y) / y;`
3. Trigonometische Funktionen:
`x:=0.0; y:=9.9;
 For i:=1 to 1000 Do x:=x+sin(arctan(cos(ln(y))));`
4. Text-Ausgabe auf den Bildschirm:
`For i:=1 to 1000 Do WriteLn ('1234567890qwertyuiop',i);`
5. Grafik-Ausgabe (MTX: VS 4):
`For x:=1 to 100 Do
 For y:=1 to 100 Do
 Plot (x,y);`
6. Speichern auf Diskette:
`x:= '1234567890qwertyuiop';
 For i:=1 to 1000 Do Write (Testfile,x);`

Bench- mark- Test	IBM	Memotech	MTX/FDX
	4.77	8.00	4 oWait
Int	1.2	0.9	7
Real	39.1	27.3	56
Trig	56.3	39.5	98
Text	64.2	46.0	17
Graf	6.6	4.7	3
Disc	10-18	10.0	19
" F:50			2
" B:09			8

Anm.d.HzN.: Die 8 MHz-Systeme sind natürlich bei reinen Diskettenzugriffen auf ein echtes Diskettenlaufwerk noch so langsam wie ein 4 MHz-System, da die Taktfrequenz umgeschaltet wird. Dank der Untersuchungen und der Schaltung von Kurt-Bernd Rohloff wird auch dieses Manko der 8MHz-Kisten behoben werden (s.u.).
 Deutlich zu sehen ist, daß das Format 09 von Bernd Preusing wesentlich schneller ist, als 03, was in Anbetracht des Murks von Memotech nicht verwunderlich ist.

Übrigens hat Andreas Burmester, der Schöpfer von 'DoMessDOS', der mit der Z280 schon arbeitet mal mit Integer-Arithmetik unter C eine 6 MHz Z80 mit einer 10 MHz Z280 ohne Cache verglichen: 0,42 zu 0,06 Sekunden. Mit Cache wird die Z280 jedoch noch deutlich schneller!

NewWord: Spooler / Druckertreiber von CLUB.009**NewWord und Spoolerinhalt**

(Herbert zur Nedden, 2000)

Angeregt wurde ich durch die folgende Frage von Peter Würfel,7262:

Wenn ich den Spoolerinhalt mit dem KLIX-Hilfsprogramm in eine Datei schreibe, erscheint, wenn ich mir diese Datei mit NW ansehe, in manchen Zeilen auf dem Bildschirm in der letzten Spalte rechts, wo sonst z.B. das '<'-Zeichen für CR oder ':' für Dot-Befehls-Zeile steht ein 'J'. Was bedeutet das?

NewWord verwendet einige sogenannte 'Flag-Characters', die am rechten Bildschirmrand einige Informationen anzeigen. Diese haben folgende Bedeutung:

Blank	Zeile, die innerhalb eines Fließtext-Abschnitts ist.
<	Zeile, die mit einer Zeilenschaltung (RET) beendet wurde, d.h. hier ein Absatzende bzw. eine Zeile ohne Blocksatz ist.
+	Zeile geht über den rechten Bildschirmrand hinaus
^	Ende des Dokuments erreicht
P	Seitenumbruch via .pa
L	Seitenumbruch via Control-L im Text
?	hää ? NewWord kennt das Punkt-Kommando nicht
.	Gültiges Punkt-Kommando
:	Punkt-Kommando, welches nur den Ausdruck, nicht den Bildschirmaufbau verändert.
!	Punkt-Kommando, welches am besten am Anfang des Dokumentes stehen sollte.
-	Zeile, die über die nächste gedruckt wird
J	Zeile, die nur mit einem Line-Feed endet, aber ohne CR.

NewWord schickt beim Drucken eine ganze Menge merkwürdiger Zeichen auf den Drucker, falls es sich nicht um einen der simplen wie DRAFT handelt, damit unabhängig vom Zustand des Druckers ein anständiges Druckbild gewährleistet ist. Daher kann es anscheinend auch mal vorkommen, daß eine Zeile ohne CR (Hex 0D) endet.

Einbau des NWPRINTN.OVR von CLUB.009

(Herbert zur Nedden, 2000)

Es muß der erste Sektor von NWPRINTN.OVR von der PD CLUB.009 durch den ersten Sektor des NWPRINT.OVR zum eigenen NewWord ersetzt werden.

Das geht mit DU oder DU2 wie folgt:

1. Directory-Eintrag von NWPRINT.OVR suchen, und auf die erste Group (hier 1B) positionieren.
2. Diesen Sektor mit < save
3. Nun den Directory-Eintrag von NWPRINTN.OVR suchen, und auf die erste Group positionieren (hier 02).
4. Ansehen, ob es richtig aussieht
5. Nun den gesaveten Sektor mit > zurückholen, und damit den alten von NWPRINTN.OVR überschreiben.
6. Fertig

New Word: Druckertreiber von CLUB.009

Hier nun das ganze etwas anschaulicher:

(Eingaben **fett**, wichtige Informationen unterstrichen, Erläuterungen und ähnliches *kursiv*)

A>du2

DU2 - Disk Utility II, Version 1.1
Type ? for Help

DU2 A0? **g0** *Erst einmal ins Direktory sehen*

Group = 0000:00, Track = 8, Sector = 1, Physical Sector = 1

DU2 A0? **d**

```
00 004E5750 52494E54 4E4F5652 00000080 *.NWPRINTNOVR....*
10 02000300 04000500 06000700 08000900 *.....*
20 004E5750 52494E54 4E4F5652 01000080 *.NWPRINTNOVR....*
30 0A000B00 0C000D00 0E000F00 10001100 *.....*
40 004E5750 52494E54 4E4F5652 0200005A *.NWPRINTNOVR...Z*
50 12001300 14001500 16001700 00000000 *.....*
60 004E5750 52494E54 204F5652 0000005C *.NWPRINT OVR...ö*
70 1B001900 1A001B00 1C001D00 00000000 *.....*
```

*Falls in diesem Sektor NWPRINT.OVR nicht zu sehen ist, muß mit den Kommandos **+** und **d** weitergesucht werden. Außerdem ist nur der Eintrag im Direktory interessant, der an der oben hervorgehobenen Stelle eine 00 hat!*

DU2 A0? **g1B** *Die 1B ist die oben gefundene 1B*

Group = 001B:00, Track = 22, Sector = 21, Physical Sector = 21

DU2 A0? **d**

```
00 00034E65 77776F72 6420322E 31362020 *..Newword 2.16 *
10 4F766572 6C617920 52656C65 61736564 *Overlay Released*
20 206F6E20 3231204D 61722038 3520200D * on 21 Mar 85 .*
30 0A436F70 79726967 68742028 43292031 *.Copyright (C) 1*
40 39383320 4E657773 74617220 536F6674 *983 Newstar Soft*
50 77617265 20496E63 6F72706F 72617465 *ware Incorporate*
60 642E0D0A 416C6C20 7269676B 74732072 *d...All rights r*
70 65736572 7665642E 0D0A0A00 1A000000 *eserved.....*
```

Falls dieser Sektor nicht etwa diesen Inhalt hat, so hast Du etwas falsch gemacht!

DU2 A0? **<** *Sektor in den Puffer schreiben*

New Word: Druckertreiber von CLUB.009

DU2 A0? g0 Zurück zum Direktory

Group = 0000:00, Track = 8, Sector = 1, Physical Sector = 1

DU2 A0? d

```
00 004E5750 52494E54 4E4F5652 00000080 *.NWPRINTNOVR....*
10 02000300 04000500 06000700 08000900 *.....*
20 004E5750 52494E54 4E4F5652 01000080 *.NWPRINTNOVR....*
30 0A000B00 0C000D00 0E000F00 10001100 *.....*
40 004E5750 52494E54 4E4F5652 0200005A *.NWPRINTNOVR...Z*
50 12001300 14001500 16001700 00000000 *.....*
60 004E5750 52494E54 204F5652 0000005C *.NWPRINT OVR...ö*
70 18001900 1A001B00 1C001D00 00000000 *.....*
```

*Falls in diesem Sektor NWPRINT.OVR nicht zu sehen ist, muß mit den Kommandos * und d weitergesucht werden. Außerdem ist nur der Eintrag im Direktory interessant, der an der oben hervorgehobenen Stelle eine 00 hat!*

DU2 A0? g02

Group = 0002:00, Track = 9, Sector = 7, Physical Sector = 7

DU2 A0? d

```
00 00034E65 77776F72 6420322E 30322020 *..Newword 2.02 *
10 4F766572 6C617920 52656C65 61736564 *Overlay Released*
20 206F6E20 3331204D 61792038 3420200D * on 31 May 84 .*
30 0A436F70 79726967 68742028 43292031 *.Copyright (C) 1*
40 39383320 4E657773 74617220 536F6674 *983 Newstar Soft*
50 77617265 20496E63 6F72706F 72617465 *ware Incorporate*
60 642E0D0A 416C6C20 72696768 74732072 *d...All rights r*
70 65736572 7665642E 0D0A0A00 1A000000 *eserved.....*
```

Dieser Sektor sollte genauso aussehen, schließlich ist er so auf der CLUB.009

DU2 A0? > Eignen Sektor hervorkramen

DU2 A0? d Dein eigener

```
00 00034E65 77776F72 6420322E 31362020 *..Newword 2.16 *
10 4F766572 6C617920 52656C65 61736564 *Overlay Released*
20 206F6E20 3231204D 61722038 3520200D * on 21 Mar 85 .*
30 0A436F70 79726967 68742028 43292031 *.Copyright (C) 1*
40 39383320 4E657773 74617220 536F6674 *983 Newstar Soft*
50 77617265 20496E63 6F72706F 72617465 *ware Incorporate*
60 642E0D0A 416C6C20 72696768 74732072 *d...All rights r*
70 65736572 7665642E 0D0A0A00 1A000000 *eserved.....*
```

DU2 A0? w Sichern

DU2 A0? x Fertig

NewWord: Bildschirmattribute**Bildschirmattribute in NewWord**

(Herbert zur Nedden, 2000)

In NewWord liege ab der Adresse 03FB Hex eine Tabelle, die die verschiedenen Bildschirmattribute der diversen Anzeigefelder festlegt.

Beim Einsprung in die Routine, die die Attribute setzt, wird im A-Register die Art des Textes, der ausgegeben werden gekennzeichnet. Dazu werden die Bits 0 - 6 verwendet. Die Bedeutung eines gesetzten Bits ist wie folgt:

Bit	Bedeutung
0	Text, der durchgestrichen werden soll (^P^X)
1	Fehlermeldungen und Warnungen
2	Markierter Text, d.h. inneres eines Blocks
3	Unterstrichener Text (^P^S)
4	Tiefgestellter Text (^P^V)
5	Hochgestellter Text (^P^T)
6	Fettgedruckter Text (^P^D und ^P^B)
7	Reserviert für zukünftige Erweiterungen

Falls kein Bit gesetzt ist, deutet das auf ganz normalen Text hin, ansonsten müssen alle Bits von 0 bis 6 untersucht werden.

NewWord setzt das Attribut mit der Bildschirmsteuersequenz ^F gefolgt vom Attribut-Byte. Das Attribut-Byte ist auch bitweise kodiert, wobei die verschiedenen Bits folgende Bedeutung haben:

Bit	Farbbildschirm	Monochromschirm
0	Vordergrund rot	Unterstreichen
1	Vordergrund grün	kein Effekt
2	Vordergrund blau	Helle Zeichen
3	Hintergrund rot	kein Effekt
4	Hintergrund grün	Invers
5	Hintergrund blau	Hintergrund
6	Blinken	Blinken
7	Grafikzeichen	Grafikzeichen

Erfreulicherweise ist in NewWord die Umsetzung der Art der Anzeige in die Ausgabeattribute für das ^F-Kommando an den Bildschirm recht einfach kodiert: (Die Notation <xxxx> bedeutet 'Inhalt der Speicherstelle xxxx Hex')

```
Attr = <03FB>
If A = 0 Then Goto Los

For I = 0 to 6
  If Bit I von A gesetzt,
    Then Attr = (Attr AND <03FC+2*I>) OR <03FD+2*I>
Next I
```

Das Attribute-Byte wird erst einmal mit dem Inhalt von 03FBH geladen. Es wird das Text-Art-Byte aus dem A-Register bitweise von Bit 0 bis 6 untersucht. Ist das Bit I gesetzt, so wird das Attribute-Byte mit dem Inhalt der Speicherstelle 03FCH+2*I maskiert (AND). Damit werden ggf. einige Bits des Attribut-Bytes gelöscht. Anschließend werden im Attribut-Byte zusätzlich die im Byte an der Adresse 03FDH+2*I gesetzten Bit in das Attribute-Byte übernommen.

SuperCalc: Bildschirmformat

Zum zweiten: Jan Brederke hat im letzten INFO (24/25) seinen Versuch und dessen begleitende Schwierigkeiten beschrieben, in NEWWORD ein variables Schirmformat zu implementieren. Vielleicht klappt's ja noch. Gerade in SUPERCALC aber, wo man doch einmal lange Zahlenkolonnen untereinander aufgereiht hat, ist so eine Umstellung auf ein variables Schirmformat mit viel mehr Zeilen sicher sinnvoll. Ich habe mich also daran gemacht, einen zusätzlichen Befehl in SUPERCALC einzubauen, der zwischen den beiden installierten Schirmen hin- und herschaltet. Dieser Befehl ist mit "&" dann zu erreichen, wenn SUPERCALC ein "Initial Command" erwartet (wenn also der Cursor in der Kommandozeile ganz links steht und auf /, !, ;, ? oder = wartet). Tippt man hier "&", dann wechselt das Schirmformat so, wie es in RAM42 initialisiert ist.

Für Interessierte: Ich hänge mich bei der Tastaturabfrage ein, prüfe, ob ein "&" eingegeben wurde und werte dieses nur dann aus, wenn die Kommandozeile leer ist. Zum Schluß übergebe ich der Tastaturabfrage eine "0", so, als ob keine Taste gedrückt worden wäre.

Den String zum Umschalten des Schirmformates habe ich hinter den zur HOME-Funktion gelegt (der hat 9 Bytes frei, braucht aber nur "01" als Zählbyte und "1A" als übergebener Wert). Er lautet ja bekanntlich ESC "A?",t, wobei ?=G für Format 1 und ?=H für Format 2. t bedeutet die Nummer der angewählten Funktionstastentabelle (+30H), bei mir hat SUPERCALC die Nummer 4. Bei 01ED steht also in meinem SUPERCALC: 1B 5B 47 34 24 (24 ist das Abschlußbyte für den BDOS-Aufruf Nr. 9 und bedeutet "\$").

Es genügt beim Schirmformatwechsel nicht, bei 01A6 und 01A7 die neuen Zeilen- bzw. Spaltenzahlen einzutragen (da, wo INSTALL.COM sie hinsetzt). Glücklicherweise besorgt die Routine bei 52CB die Anpassung der anderen nötigen Variablen. Diese Routine darf man aber nur aufrufen, wenn man kein Horizontal- oder Vertikalsplitting durchgeführt hat (den Schirm also in zwei Teile geteilt hat), sonst würde das einem CLEAR des Splittings entsprechen. Das ist zwar lästig, aber nicht deletär, denn man kann ja neu splitten. Wer also keinen Wert auf Erhalt des Splittings legt, kann sich eine recht lange Routine (s.u.) und damit eine Menge Patch-Platz sparen, sofern er noch andere Patches einzufügen gedenkt. Dann sollte er allerdings nicht 52CB sondern 52C2 aufrufen, damit die Splitting-Tabellen gelöscht werden (s.u.).

Apropos Platz zum Patchen: Manche nehmen ja den Speicherplatz, wo der Name des Programmes, die Seriennummer usw. eingetragen sind. Ich bin mir aber nicht sicher, ob das im Rahmen des Copyright statthaft ist (SUPERCALC scheint da gutmütig zu sein. Als ich z.B. 'mal in NEWWORD 2.17 meinen verhunzten Namen "Holger Gobel" in "Holger Göbel" umändern wollte, quittierte dieses schlicht mit einem Abbruch). In SUPERCALC habe ich zwei Bereiche entdeckt, die man zum Patchen heranziehen kann:

Der erste ist eigentlich ein Bereich, in dem die GOTO XY Routine für das spezifische Terminal angepaßt wird. Sie stellt 79 Bytes zur Verfügung, wobei unsere implementierte Routine auf 17 Bytes umgeschrieben werden kann. Zum Schluß der Routine müssen HL und DE noch mit 0010H bzw. 0013H geladen werden. Dieser Schluß (jetzt neu bei 02E6) wird von 02C9 und 02CD angesprungen, das RET überflüssigerweise auch noch von 02CF und 02D2, so daß diese Sprungadressen geändert werden müssen. Wenn das getan ist, dann stehen 62 Bytes zur Verfügung. Die reichen genau aus, um einen Patch einzubauen, der die Schirmformate umschaltet, aber das Splitting nicht berücksichtigt. Bei 0337H sind glücklicherweise weitere 137 Bytes frei (bis 03BFH), von denen meine Routine trotz sparsamer Programmierung nur noch 11 übrigläßt. Allerdings funktioniert dann auch Splitting einwandfrei.

SuperCalc: Bildschirmformat

Eine kleine Einschränkung: Wenn man auf dem "großen" Bildschirm ein Splitting-Border einführt, das auf das "kleine" nicht paßt, dann sieht man es auch auf dem kleinen nicht; z.B.: Auf dem kleinen Schirm kann ich 21 Zeilen darstellen. Ich definiere auf dem großen Schirm ein horizontales Splitting bei Zeile 33: dann kann ich dieses Border auch nur auf dem großen Schirm sehen. Trotzdem kann ich Zahlen auch im kleinen Schirm eingeben - nur sehe ich sie nicht, da das aktive Feld nach unten "weggetaucht" ist. Bei Vertikalsplitting gibt es dazu auch noch komische Bildschirmeffekte. Diese Nebenwirkungen abzustellen, bräuchte noch weit mehr Speicherplatz zum Patchen, der halt leider nicht zur Verfügung steht. Im Grunde genommen besteht ja auch keine Notwendigkeit. Wenn man schon so großzügiges Splitting ausführen will, dann soll man sich das Ergebnis eben auf dem großen Schirm angucken. Ansonsten empfiehlt es sich, Splittings immer auf dem kleinen Schirm zu definieren, dann gibt es beim Hin- und Herschalten keine Probleme.

Zur Realisation des Horizontal- bzw. Vertikal-Splittings ins SUPERCALC selber: Für den Teil unten bzw. rechts steht eine 3 Byte lange Tabelle bei 55A1 und eine 24 Byte lange Variablen-Tabelle bei 55A6, für den Teil oben bzw. links bei 55D6 bzw. 55BE. Diese Tabellen werden in die aktiven Tabellen umgeladen, die bei 530C bzw. 52F4 stehen. Aktiv ist immer die Seite, in der sich die aktuelle Zelle befindet.

Mein Patch-Programm ändert die Variablen in der aktiven Tabelle und lädt sie dann in die "Reservetabellen" um.

Ganz zum Schluß wird dann der Bildschirm neu aufgebaut (die entsprechende Routine steht bei 2354H; wenn man 234BH aufruft, wird der Cursor gleichzeitig auf die HOME-Position gerückt.).

Für noch mehr Interessierte drucke ich die beiden Patchprogramme in Mnemonics ab und erkläre vorher die entscheidenden Variablen in der aktiven Tabelle (in Klammern in Hex ein Beispiel für Vertikalsplitting zwischen D und E, aktive Zelle links/rechts bei E3/F3, Normalschirm (80x25), H = Horizontal-, V = Vertikal-Splitting, r = rechts, l = links, o = oben, u = unten):

52F2: Spalten (wenn 40d Spalten, dann 41d) (50/50)
 52F3: Zeilen-3 (16/16)
 52F4: Zeile des neuen Borders bei Hu, sonst 1 (01/01)
 52F5: Spalte des neuen Borders bei Vr, sonst 1 (01/2A)
 52F6: Zeilen-3, bei Ho Zeilen des oberen Teils inkl. Border (16/16)
 52F7: Spalten, bei Vl Spalten des linken Teils inkl. Border (29/50)
 52F8: ?weitmöglichste Linksposition-1 der Zelle? (00/04)
 52F9: ?weitmöglichste Hochposition-1 der Zelle? (00/00)
 52FA: Anzahl der darzustellenden Felder (z.B. A...H=07) (03/06)
 52FB: Anzahl der Zeilen-1 im aktiven Teil (wichtig für H) (14/14)
 52FC: 63-(auf den Schirm passende Felder); 63 entspr. "BK" (3C/3C)
 52FD: Nummer der aktiven Zelle-1 (von li nach re) (03/04)
 52FE: Zeile der aktiven Zelle-1 (02/02)
 52FF-5308: ? (00)
 5309: Größe der Zelle (Width) (z.B.09/09)
 530A: ? (01/01)
 530B: ? (04/04)

55A5: Splittingvariable: 00=kein Spl., 01=Ho, 02=Hu, 81=Vl, 82=Vr

Die Tabellen bei 55A6 bzw. 55BE entsprechen der obigen ab 52F4 (s.o.). Die 3-Byte-Tabellen bei 530C (bzw. 55A1 oder 55D6) sind mir in ihrer Bedeutung unklar, sie enthielten immer 00.

SuperCalc: Bildschirmformat

Nun zum Listing:

```

02C0 JP 02D5H ;Einsprung zur GOTO XY-Routine
      JP 032BH
      JP 0331H ;bis hierher wie gehabt
02C9 JP 02E6H ;4 geänderte Einsprungadressen
02CC JP 02E6H
02CF JP 02E6H
02D2 JP 02E6H
02D5 PUSH DE ;hier beginnt die GOTO XY - Routine,
02D6 LD A,03 ;X wird in D, Y in E übergeben, voraus
      CALL 017DH ;wird 03 ausgegeben (Ausgabe bei 017D)
      POP DE
      PUSH DE
      LD A,E
      CALL 017DH
      POP DE
      LD A,D
      CALL 017DH
02E6 LD HL,0010H
      LD DE,0013H
02EC RET
02ED CALL 0005H ;BDOS-Aufruf (Tasteneingabe), der Patch
      CP "&" ;wird von 5129H aufgerufen
      RET NZ ;nur weiter, wenn "&" gedrückt
      LD C,A
      LD A,(5864H) ;Zählerbyte für den String in der Kommando-
      AND A ;zeile
      LD A,C
      RET NZ ;nur weiter, wenn Cursor ganz links
      PUSH HL
      PUSH BC
      LD HL,(OFFC6H) ;VARPTR bei RAM42
      LD BC,004BH ;Offset für Form1x
      ADD HL,BC
      LD A,(01EFH) ;hier steht "G" oder "H" für Schirm 1 oder 2
      XOR OF ;G und H toggeln (für ESC "A","G oder H",t),
      LD (01EFH),A ;t=Nr. der SC-Funktionstastentabelle + 30H
      RRA
      JR C,0310H ;springe, wenn G
      INC HL
      INC HL ;sonst nimm Form2x
0310 LD DE,01A7H ;hier ist die Spaltenzahl einzutragen
      LD A,(HL)
      LD (DE),A
      INC HL ;Form1y bzw. Form2y
      DEC DE ;01A6, hier muß die Zeilenzahl hin
      LD A,(HL)
      LD (DE),A
      LD DE,01EDH ;hier steht der String ESC "A?",t; ?=G/H
      LD C,09 ;BDOS-Funktionsnummer
      CALL 0005H ;und raus damit
      CALL 0337H ;hier CALL 52C2H, wenn Splitting zerstören
      CALL 2354H ;Schirm neu aufbauen
      POP BC
      POP HL
      XOR A ;A=0, als ob keine Taste gedrückt wurde
032A RET ;und zurück

```

SuperCalc: Bildschirmformat

Jetzt folgt die Patch-Routine zur Behandlung des Splittings:

```

0337 LD   A,(55A5H)      ;Splittingvariable
      OR   A            ;wird überhaupt gesplittet?
      JP   Z,52CBH      ;nein --> dann Var. einfach initialisieren
      LD   C,A          ;in C merken
      LD   A,(01A6H)    ;Zeilen
      SUB  03           ;für 3 Statuszeilen
      LD   (52F3H),A
      LD   A,(01A7H)    ;Spalten
      CP   28H          ;bei 40 Zeichen komischerweise +1
      JR   NZ,034FH
      INC  A
034F LD   (52F2H),A
      LD   IX,(0FFC6)   ;VARPTR bei RAM42
      LD   A,(IX+4EH)   ;Form2y (Zeilen bei Schirm 2)
      LD   E,(IX+4CH)   ;Form1y (Zeilen bei Schirm 1)
      SUB  E            ;Differenz der beiden
      LD   E,A
      LD   A,(01EFH)    ;G oder H (auf Schirm 1 oder 2 schalten)
      RRA
      LD   A,E
      JR   NC,0367H     ;Sprung, wenn auf Schirm 2 geschaltet wird
      NEG                ;sonst später die Diff. subtrahieren
0367 LD   E,A          ;Diff. in E merken
      LD   B,02         ;zwei Durchgänge
036A BIT  7,C          ;vertikal?
      JR   NZ,0372H     ;ja, dann immer ändern
      BIT  0,C          ;horizontal oben?
      JR   NZ,0380H     ;ja, dann so lassen
0372 LD   A,(52F6H)    ;Variablen s.o.
      ADD  A,E          ;neuen Wert bilden
      LD   (52F6H),A
      LD   A,(52FBH)
      ADD  A,E
      LD   (52FBH),A
0380 BIT  7,C          ;horizontal?
      JR   Z,038BH      ;ja, dann immer ändern
      BIT  0,C          ;vertikal links?
      JR   NZ,038EH     ;ja, dann so lassen
0388 LD   A,(01A7H)    ;Spalten
      LD   (52F7H),A
038E PUSH  BC
      CALL 2746H        ;(52FCH) ausrechnen lassen
      POP  BC
      BIT  7,C          ;horizontal?
      JR   Z,039BH      ;ja, dann immer ändern
      BIT  0,C          ;vertikal links?
      JR   NZ,03A7H     ;ja, dann so lassen
039B LD   H,A          ;(52FC) nach H
      LD   A,(52FBH)
      LD   L,A
      LD   A,3EH        ;maximale horizontale Felderzahl (BK)
      SUB  H
      ADD  A,L
      LD   (52FAH),A    ;(52FA)=3E-(52FC)+(52FB)

```

SuperCalc: Bildschirmformat

```

03A7 PUSH BC
      PUSH DE
      CALL 2403H      ;in die entsprechende Splitting-Tab. umladen
      POP DE
      POP BC
      LD A,C          ;Splitting-Variable
      XOR 03          ;aktives Feld auf die andere Seite
      LD C,A
      DJNZ 036AH      ;und für die andere Seite nochmal
      RET             ;Ende Splitting-Manager, noch 11 Byte frei

```

So, jetzt kann's losgehen, wir nehmen wieder DDT (ich selber arbeite ja viel lieber mit MONI von Olaf Krumnow, dieses Programm besitzt aber nicht jeder):

DDT SC.COM

Zuerst das Einschleifen in die Tastaturabfrage:
S512A, dann nacheinander eingeben
ED 02 . (nach jeder Zahl natürlich wieder RETURN)

Der String zum Umschalten des Schirmformates:

S1EB, dann

01 1A 1B 5B 47 34 24 .

Wessen SUPERCALC-Funktionstastentabelle z.B. Nr. 3 trägt, muß **33** anstelle von **34** eintragen (Nummer +30H).

Jetzt die geänderten Einsprungadressen, die neue GOTO XY-Routine und die "&"-Befehlsauswertung:

S2CA, dann

```

E6 02 C3 E6 02 C3 EC 02 C3 EC 02 D5 3E 03 CD 7D
01 D1 D5 7B CD 7D 01 D1 7A CD 7D 01 21 10 00 11
13 00 C9 CD 05 00 FE 26 C0 4F 3A 64 5B A7 79 C0
E5 C5 2A C6 FF 01 4B 00 09 3A EF 01 EE 0F 32 EF
01 1F 38 02 23 23 11 A7 01 7E 12 23 1B 7E 12 11
ED 01 0E 09 CD 05 00 CD 37 03 CD 54 23 C1 E1 AF
C9 .

```

Wer also auf die Splitting-Verwaltung verzichten will, der braucht folgende Routine nicht einzutippen, muß dann aber **C2 52** anstelle der oben unterstrichenen **37 03** eintragen.

S337, dann

```

3A A5 55 B7 CA CB 52 4F 3A A6 01 D6 03 32 F3 52
3A A7 01 FE 28 20 01 3C 32 F2 52 DD 2A C6 FF DD
7E 4E DD 5E 4C 93 5F 3A EF 01 1F 7B 30 02 ED 44
5F 06 02 CB 79 20 04 CB 41 20 0E 3A F6 52 83 32
F6 52 3A FB 52 83 32 FB 52 CB 79 28 04 CB 41 20
06 3A A7 01 32 F7 52 C5 CD 46 27 C1 CB 79 28 04
CB 41 20 0C 67 3A FB 52 6F 3E 3E 94 85 32 FA 52
C5 D5 CD 03 24 D1 C1 79 EE 03 4F 10 B6 C9 .

```

Und wieder ^C und **SAVE 96 SCTEST.COM**

Man kann dann auch noch in den HELP-Bildschirm den neuen Befehl eintragen (falls man einmal das "&" vergessen hat, ist es mit einem einfachen "?" abrufbar):

SuperCalc: Bildschirmformat

DDT SC.HLP

SIDC, dann

26 20 20 2D 2D 3E 20 20 54 6F 20 63 68 61 6E 67

65 20 73 63 72 65 65 6E 20 66 6F 72 6D 61 74 0D

0A 0D 0A 46 6F 75 72 20 61 72 72 6F 77 73 20 6F

72 20 43 54 52 4C 2B 6B 65 79 20 73 63 72 6F 6C

6C 20 61 72 6F 75 6E 64 20 74 68 65 20 63 65 6C

6C 73 3A 20 20 .

^C und SAVE 40 SC.HLP

Zum Schluß noch ein paar Anmerkungen: Für die Analyse und das Patchen von SUPERCALC habe ich etwa 14 Tage gebraucht, seit einigen Tagen läuft es im Test ohne Beanstandungen. Trotzdem kann sich natürlich ein unvorhergesehener "Bug" noch auftun (nachdem ich eigentlich nur Bildschirmvariable verändere, ist er dann hoffentlich nicht so schlimm), sei es auch nur aufgrund eines Tippfehlers in diesem Beitrag. Ich wäre Euch dankbar, wenn ihr ihn dann mitteilen könntet, damit ich eine Lösung suchen kann.

Werkzeuge zum Disassemblieren sind Monitore (z.B. MONI.COM) und spezielle Programme wie REZILOG.COM. Da diese immer nur "ruhende" Programme betrachten können, muß ein quasi "ONLINE"-Monitor die Arbeit ergänzen. Viele Programmteile werden nämlich beim Aufruf des Programms erst irgendwohin verschoben, Variablen werden gesetzt und, das Ärgste, Overlays werden zu- und ausgeladen. So ein Monitor steht uns ja mit dem KLICK-Monitor zur Verfügung. Da vermißt man halt aber doch so ganz wichtige Funktionen wie Assemblieren, Steppen, Suchen, Verschieben usw. Vielleicht kann Bernd seinen Monitor doch noch erweitern oder, einfacher, vielleicht kann ein leistungsfähiger Monitor wie MONI als KLICK-Programm modifiziert werden.

Herbert hat angedeutet, daß er eine FD-Diskette mit lauter Patch-Programmen zusammenstellen will. Wenn sich diese SC-Patches bewähren, werden sie dort Eingang finden. Wer also die Tippmühe scheut, kann ja noch ein bißchen warten.

Das wär's.

P a s c a l : Numerische Darstellung von Meßwerten**Numerische Darstellung von Messwerten** (Kurt-Bernd Rohloff, 8000)

Gemessene Daten sind stets mit einem Fehler behaftet. Die Darstellung numerischer Messwerte sollte daher auf eine "sinnvolle" Stellenzahl beschränkt werden. Beispielsweise ist die Angabe 123.456789 unsinnig, wenn der Fehler (üblicherweise die Standardabweichung einer statistischen Auswertung) bereits 4.6 beträgt. Daher habe ich eine PASCAL Prozedur geschrieben, die den Messwert (x) immer auf zwei fehlerbehaftete Stellen genau ausgibt. Ebenso wird der Fehler (dx) auf zwei signifikante Stellen ausgegeben. Für obiges Beispiel würde meine Prozedur Show_x_dx

1.234 +/- 0.046 E 2

ausgeben. Zwei fehlerhafte Stellen erschienen mir aus 3 Gründen sinnvoll:

- Man hat noch die Freiheit, die Rundung auf eine fehlerhafte Stelle selbst vorzunehmen (oder auch nicht).
- Muß mit dem Ergebnis noch weitergerechnet werden, können durch zu frühzeitiges Runden unakzeptable Fehler im Endergebnis entstehen.
- Häufig geht im Fall b auch die Fehlergröße dx noch in Folgerechnungen ein (Fehlerrechnung!). Dann wäre eine Stelle zu wenig.

Die Prozedur versagt, wenn einer der beiden Werte x, dx gleich Null ist. Dann wird auf das standardmäßige REAL Ausgabeformat von TURBO-PASCAL umgeschaltet. Die Ausgabe wird nicht mit einem Zeilenwechsel abgeschlossen, so daß gegebenenfalls noch eine Maßeinheit o. ä. angehängt werden kann.

Der dritte Parameter ist eine Dateivariablen, die das Ausgabegerät bestimmt. Gedacht ist hier in erster Linie an Bildschirm und Drucker, jedoch wäre auch Ausgabe in eine Textdatei möglich. Im aufrufenden Programm muß dazu eine Dateivariablen vom Typ TEXT vereinbart werden, der dann via ASSIGN ein logisches Gerät zuzuordnen ist. Ein kleines Beispielprogramm verdeutlicht dir am besten, wie's geht.

Beispielprogramm:

```

Program TESTDX;
{ testen von show_x_dx }
VAR      p, q                : REAL;
         device              : TEXT;
{$I SHDX.INC }
{$I INKEY.INC }
BEGIN
WRITE(#6, #2, 'Ausgabe auf ', #6, #4, 'B', #6, #2, 'ildschirm oder ',
      #6, #4, 'D', #6, #2, 'rucker? ');
CASE In_Key OF
  'b', 'B' : Assign(device, 'CON:');
  'd', 'D' : Assign(device, 'LST:');
  ELSE     WRITELN; WRITELN('Bildschirm wird genommen');
           Assign(device, 'CON:');
END; { case }
Rewrite(device);
REPEAT
  WRITELN; WRITE('Enter x dx ');
  READLN(p,q);
  WRITE(device, 'Ergebnis: ');
  Show_x_dx(p,q, device); WRITELN(device);
  WRITE('Weiter (j/n)? ');
UNTIL UpCase(In_Key) <> 'J';
END.

```

P a s c a l: Numerische Darstellung von Me~wertenDie Include-Datei INKEY.INC sieht wie folgt aus:

```

FUNCTION In_Key:Char;
{ liest ein Zeichen von der Tastatur mit Echo }
BEGIN
  In_Key:=Chr(BDOS(1));
END;

```

Die Include-Datei SHXDAX.INC sieht wie folgt aus:

```

PROCEDURE Show_x_dx(x, dx : REAL; VAR device : TEXT);
{ Anzeige von x (Messwert) und dx (sein Fehler) auf zwei fehlerhafte
  Stellen genau. Die Ausgabe erfolgt auf das der Dateivariablen device
  zugewiesene Gerat.
}
  VAR      a1, a2, tmp      : REAL;
           b1, b2, f       : INTEGER;
  { normalisierte Darstellung: x=a1*10**b1 mit 1 <= a1 < 10
    dx=a2'*10**b2 = a2*10**b1, wobei a2=dx/10**b1
    dargestellt wird a1 +/- a2 E b1 wobei a1,a2 mit f=b1-b2+1
    Nachkommastellen ausgegeben werden
  }

  FUNCTION Lg(x : REAL) : REAL; { Zehnerlogarithmus }
    BEGIN
      LG:=0.43429448 * LN(X);
    END;

  FUNCTION zehn_hoch( loga : INTEGER) : REAL;
    VAR      h, faktor      : REAL;
           i                : INTEGER;
    BEGIN
      h:=1.0;
      IF loga <> 0
        THEN BEGIN
          IF loga > 0
            THEN faktor:=10.0
            ELSE faktor:=0.1;
          FOR i:=1 TO Abs(loga)
            DO h:=h*faktor;
          END;
          Zehn_hoch:=h;
        END;
    END;

```

P a s c a l : Numerische Darstellung von Meßwerten

```

FUNCTION Zehner(x : REAL) : INTEGER;
  { Bestimmen des Zehnerexponenten der normalisierten Darstellung
  }
  CONST      diff      =1.0E-4;
  VAR        tmp, rund  : REAL;
            h           : INTEGER;

  BEGIN
    tmp:=lg(Abs(x));
    { naechst kleinere ganze Zahl bestimmen }
    IF tmp > 0
      THEN rund:=INT(tmp + 0.5)
      ELSE rund:=INT(tmp - 0.5);
    IF Abs(tmp - rund) <= diff
      THEN { tmp liegt schon nahe einer ganzen Zahl, nimm diese }
           h:=Round(rund)
      ELSE BEGIN
           h:=Trunc(tmp);
           IF tmp < 0
             THEN h:=Pred(h);
           END;
    Zehner:=h;
  END;

BEGIN { show_x_dx }
IF (x <> 0) AND ( dx <> 0)
  THEN BEGIN
    b1:=Zehner(x);
    b2:=Zehner(dx);
    f:=b1 - b2 +1;
    IF f < 0
      THEN f:=0;
    tmp:=Zehn_hoch(-b1);
    a1:=x*tmp; a2:=dx*tmp;
    WRITE(device, a1:f+4:f, ' +/- ', a2:f+4:f);
    IF b1 <> 0
      THEN WRITE(device, ' E ', b1);
    END
  ELSE WRITE(device, x, ' +/- ',dx);
END; { of show_x_dx }

```

d B A S E: 'ne Macke

dBASE hat 'ne Macke!

(Peter Würfel,7262)

Herbert hatte ja schon immer mal wieder Ärger mit dBASE und den Adres-

senlisten. Immer mal wieder hat dBASE eine Reihe von Datensätzen bei der Ausgabe unterschlagen. Auch mir ist das schon vorgekommen und ich habe dabei folgende Macke entdeckt, die ich Euch hier an einem kleinen Beispiel-Programm demonstrieren möchte.

Feld	Name	Typ	Länge
001	FEINS	C	006
002	FZWEI	C	004
**	Gesamt	**	00011

00001	eins	5000
00002	zwei	5000
00003	drei	5000
00004	vier	5000
00005	fünf	1000
00006	sechs	2000
00007	sieben	6000
00008	acht	7000
00009	neun	8000

Ich erzeuge z.B. eine Datenbank 'test.dbf' mit nebenstehenden Daten und Inhalt. Ein solcher Dateiaufbau könnte z.B. in der Praxis in einer Adressendatei vorkommen: in Feld 'feins' stehen Namen, in Feld 'fzwei' stehen Postleitzahlen. Um mir jetzt alle Namen in der Reihenfolge der Postleitzahlen ausgeben zu lassen, erzeuge ich eine Indexdatei mit dem Namen 'test.ndx', eröffne meine Datenbank mit

diesem Index und gebe die Daten über eine Schleife aus (Programm: TEST1.CMD); und das klappt dann auch.

```

*** TEST1.CMD
USE TEST
INDEX ON fzwei TO test
USE test INDEX test
*
STOR 3 TO mzeile
SET TALK OFF
ERAS
DO WHILE .NOT. EOF
*
*
$ mzeile, 0 SAY feins
$ mzeile,10 SAY fzwei
STORE mzeile+1 TO mzeile
SKIP
ENDDO
*
*
*
*
USE
SET TALK ON
RETURN
    
```

```

*** TEST2.CMD
USE TEST
INDEX ON fzwei TO test
USE test INDEX test
STOR ' ' TO meingabe
STOR 3 TO mzeile
SET TALK OFF
ERAS
DO WHILE .NOT. EOF
$ 1, 0 SAY 'beliebige Eingabe'
$ 1,30 GET meingabe
$ mzeile, 0 SAY feins
$ mzeile,10 SAY fzwei
READ
STORE mzeile+1 TO mzeile
SKIP
ENDDO
* hier TEST3.000 einfügen
* Sternchen vor 'DO'
* löschen:
* DO test3.000
USE
SET TALK ON
RETURN
    
```

Interessant wir die Chose jedoch, wenn ich aus irgend einem Grund einen 'READ'-Befehl einsetze (Beispiel 2). Wie dann das Ergebnis aussieht, zeigt die nebenstehende Hardcopy: Das Programm unterschlägt die Datensätze 2,3 und 4! Wenn wir wissen wollen, warum dem so ist, sollten wir uns einfach mit test3.000 den Aufbau der Indexdatei anschauen. Wir sehen, daß dBASE die Index-Datenbank umsortiert hat. Während ursprünglich die Rei-

beliebige Eingabe : :

fünf	1000
sechs	2000
eins	5000
sieben	6000
acht	7000
neun	8000

d B A S E: 'ne Macke

dBASE hat 'ne Macke! (Fortsetzung)

```
*** TEST3.000
STORE mzeile+1 TO mzeile
$ mzeile, 0 SAY 'NEU:'
STORE mzeile+1 TO mzeile
GO TOP
DO WHILE .NOT. EOF
  $ mzeile, 0 SAY feins
  $ mzeile, 10 SAY fzwei
  STORE mzeile+1 TO mzeile
SKIP
ENDDO
RETURN
```

Reihenfolge der Datensätze in der Index-Datei 5-6-1-2-3-4-7-8-9 war, ist durch den READ-Befehl die Reihenfolge innerhalb der Datensätze mit identischem Eintrag in Feld 'fzwei' umsortiert worden. Ein Grund dafür ist nicht zu erkennen, man muß jedoch zugestehen, daß prinzipiell die Index-Datei immer noch korrekt ist, denn die Reihenfolge der Datensätze 1 bis 4 ist ja in der Indexdatei eigentlich beliebig.

Abhilfe:

Durch Zufall hab ich dann in einem dBASE-Buch einen Hinweis entdeckt, der im Markt&Technik-Handbuch nicht zu finden ist, aber klappt: Der Befehl READ kann durch NOUPDATE ergänzt werden. Das bewirkt, daß die Indexdatei nicht verändert wird. Also ersetzt in TEST2.CMD die Zeile READ durch READ NOUPDATE und alles klappt wie gewünscht.

Noch 'ne dBASE-Macke

(Peter Würfel, 7262)

Mir ist noch ein dBASE-Fehler aufgefallen, der zwar nicht so gravierend, aber doch störend ist. Probiert einfach aus:

```
*** NULLZEILEN-MACKE
ERASE
STORE ' ' TO meingabe
$ 0, 0 SAY '0123456789Z123456789Z123456789D'
$ 10, 10 GET meingabe
READ
RETURN
```

Daß durch den Befehl '\$ nn,nn GET <Variable>' einige Zeichen in der obersten Zeile gelöscht werden ist zwar nur ein optischer Fehler, da ich in Programmen jedoch gerne eine Kopfzeile in der Art hätte, wie wir sie aus NW kennen, stört er mich doch sehr. Wer findet die Wanze? Der Fehler ist übrigens auch im Original-M&T-dBASE, wie ich bei einem Freund sehen konnte, der dBASE für seinen Schneider Joice besitzt.

Und 'ne dBASE-Absonderlichkeit

Bei der Bildschirmausgabe mit dem ?-Befehl, interpretiert dBASE den Strichpunkt auf absonderliche (weil zwar interessante, aber nicht konsequente) Art und Weise. Ein Strichpunkt veranlaßt, daß das, was nach ihm folgt, in einer neuen Zeile steht; außerdem werden beide Zeilen zentriert, aber leider nicht bezogen auf die Mitte des 80-Zeichen-Schirms, sondern viel zu weit links. Schade, sonst wär das 'ne brauchbare Möglichkeit der Bildschirmgestaltung. Probierts mal aus:

? 'Dies ist eine Zeile;mit einem Strichpunkt'

d B A S E: Bitoperationen**DBFLAG**

(Kurt-Bernd Rohloff, 8000)

Das Programm DBFLAG kann einzelne Bits setzen, löschen und prüfen. Ferner können die Bits mit einer Maske logisch verknüpft werden. Das Programm erwartet eine Variable mit folgendem Aufbau:

1. Byte: Das Flagbyte, dessen Bits manipuliert werden sollen
2. Byte: Aktionscode, der angibt, was gemacht werden soll

Danach folgen je nach Aktionscode noch weitere Bytes. Aktionscode kann sein (nur bei S und R wird zwischen Groß/Kleinschreibung unterschieden):

- a) kein weiteres Byte:
 - C Komplementiere das Flagbyte (complement)
- b) 3. Byte gibt die Bitnummer an (als ASCII Ziffer "0".."7"):
 - s setze das Bit, dessen Nummer Byte 3 angibt
 - S setze und teste das Bit ...
 - r lösche das Bit ... (rücksetzen)
 - R lösche und teste das Bit ...
 - T teste das Bit ...
- c) Bytes 3 bis 10 sind die Bitmaske als String aus "0" und "1":
 - A Test, ob alle gesetzten Bits der Maske auch im Flagbyte gesetzt sind, d. h. ist Flagbyte \geq (numerischer Wert der Bitmaske) ?
 - E Test, ob wenigstens ein "1" Bit der Maske im Flagbyte gesetzt ist, d. h. (Flagbyte UND Bitmaske) \neq 0 ?
 - G Test, ob Flagbyte gleich Bitmaske ist. In diesem Fall darf die Bitmaske auch "?" als Wildcard enthalten, "111100??" z.B. prüft, ob das obere Nibble nur 1 enthält und das untere Nibble in Bit 2 und 3 Null ist. Bits 0 und 1 sind egal.
 - U Flagbyte \leftarrow Flagbyte UND Bitmaske
 - O Flagbyte \leftarrow Flagbyte ODER Bitmaske
 - X Flagbyte \leftarrow Flagbyte EX-ODER Bitmaske (exklusives ODER)
 - N Flagbyte \leftarrow Bitmaske (lade Bitmaske nach Flagbyte)
 - V Flagbyte \rightarrow Bitmaske (lade Bitmaske von Flagbyte). Es ist zu beachten, daß auch in diesem Fall eine gültige Bitmaske an das Assemblerprogramm übergeben wird, auch wenn dessen Wert egal ist.

Alle Tests hinterlegen in Adresse 41987 ein Byte, u. z. 0 für FALSE und 255 für TRUE. (Die gleiche Zuordnung benutzt dBASE für logische Variablen.) Man kann das Ergebnis mit PEEK(41987) abfragen. Zusätzlich wird in Adr. 41988 ein Fehlercode hinterlegt. Dabei bedeuten:

- 0 alles ok
- 1 Länge ungültig (nur 2, 3 oder 10 ist erlaubt)
- 2 ungültiger Aktionscode
- 3 ungültige Bitnummer (Fall b)
- 4 ungültige Bitmaske (Fall c)

Der Fehlercode sollte zumindest in der Entwicklungsphase abgefragt werden.

Nun ein paar Beispiele dazu. Wir nennen die Übergabevariable FLAGVAR. Das Programm steht in der Datei DBFLAG.HEX. Laden und Startadresse setzen:

```
LOAD DBFLAG
SET CALL TO 41984
```

FLAGVAR mit Nullen initialisieren:

```
STORE CHR(0) TO FLAGVAR
```

Initialisierung mit dem Bitmuster 11110011:

```
STORE "XN11110011" TO FLAGVAR
* Aktionscode N, X ist willkürlich
CALL FLAGVAR
```

d B A S E: Bitoperationen

Nach der Initialisierung dürfen wir das erste Byte von FLAGVAR nicht zerstören. Wollen wir nun beispielsweise Bit 7 löschen, gehen wir so vor:

```
STORE $(FLAGVAR,1,1)+"r7" TO FLAGVAR
CALL FLAGVAR
```

Den aktuellen Wert des Flagbytes anzeigen:

```
STORE $(FLAGVAR,1,1)+"V00000000"
CALL FLAGVAR
? "Flagbyte:",$(FLAGVAR,3,8)
```

Bit 0 testen und verzweigen:

```
STORE $(FLAGVAR,1,1)+"TO" TO FLAGVAR
CALL FLAGVAR
IF PEEK(41987) > 0
  * Bit 0 war 1
ELSE
  * Bit 0 war 0
ENDIF
```

Nun noch ein etwas komplizierteres Beispiel aus einem meiner Programme. Wir haben eine Datei, die durch zwei Felder IDX1 und IDX2, beide vom Typ C 2, indiziert ist. Wir wollen nun innerhalb der durch IDX1 gegebenen Gruppe von Sätzen den nächsten suchen, der im Feld STEREO (C 1) den Eintrag "S" hat, sofern es einen solchen überhaupt gibt. Das Auftreten dieses Satzes soll in Bit 1 gemerkt werden, weil spätere Anweisungen davon abhängig sind. Auf gehts:

```
STORE IDX1 TO GRUPPE
STORE $(FLAGVAR,1,1)+"R1" TO FLAGVAR
* mit Test, Ergebnis ist sicher 0
CALL FLAGVAR
DO WHILE .NOT. EOF .AND. IDX1=GRUPPE .AND. PEEK(41987)=0
  IF STEREO="S"
    STORE $(FLAGVAR,1,1)+"S1" TO FLAGVAR
    * mit Test!
    CALL FLAGVAR
  ENDIF
  SKIP 1
ENDDO
GOTO TOP
* oder irgendwoanders hin
* in diesem Teil keine Test Operation!
IF PEEK(41987) > 0
  * Satz war da
ELSE
  * Satz trat nicht auf
ENDIF
```

DBFLAG.HEX Zeilen 1 - 12

```
:10A40000C305A40000ED738CA5318CA5F5C5D5E579
:10A410007E23231129A4D5FE02CA37A4FE03CA490C
:10A42000A4FE0ACAA6A4D13E013204A4E1D1C1F11E
:10A43000ED7B8CA5C90000E57EE65FFE433E022071
:10A44000052B7E2F77AFA7E1C9C5E5237E0E01D688
:10A4500030280C47FE083E03D284A4CB2110FC2BED
:10A460007E2B1184A4D5FE72CA92A4FE52CA98A46F
:10A47000FE73CAB8A4FE53CABDA4E65FFE54CA9B2D
:10A48000A4D13E02A7E1C1C979B677AFC9CD88A4EE
:10A490001809792FA677AFC9CD92A47EA128023ED4
:10A4A000FF3203A4AFC9C5E5231EFF0100087ED615
:10A4B00030FE02DABBA4FE0F2047AFCB1B1FCB112F
```

DBFLAG.HEX Zeilen 13 - 23

```
:10A4C0002310EBE1E57EE65F432B1103A5D5FE47A4
:10A4D000CA19A5042803D11828FE41CA07A5FE45BC
:10A4E000CA11A5FE55CA27A5FE4FCA2BA5FE58CAFC
:10A4F0002FA5FE4ECA34A5FE56CA37A53E02D11876
:10A50000023E04A7E1C1C97EB93E0038023EFF18F1
:10A51000117EA128023EFF18097EA0B93EFF280245
:10A520003E003203A4AFC97EA118067EB118027E98
:10A53000A977AFC971AFC9C5E54E23230608AFCBD4
:0CA540001117C630772310F6AFE1C1C937
:02A58C0000000C
:0000000000
```


Super Tape**Supertape unter CP/M für unseren Memotech**

(Dr. Holger Göbel, 8630)

Warum ein Kassetten-Speicherprogramm, wo doch sowohl Laufwerke als auch Disketten wirklich erschwinglich sind?

Supertape auf dem Memotech zu implementieren, entsprang der Not, unseren Rechner mit einem Schneider 6128 koppeln zu wollen. Dieser besitzt keine V24-Schnittstelle und verwendet 3"-Disketten (von denen eine einzige zwischen 6 und 10 DM kostet!). Die Zeitschrift c't hat nun seit 1984 ein Kassettenabspeicherverfahren mit dem Namen "Supertape" für eine ganze Reihe (auch Nicht-Z80-)Rechner entwickelt, u.a. auch für CP/M-Rechner (von J. Assenbaum und A. Stiller, C 1985 bei c't). Dieses Listing habe ich mir besorgt, abgetippt und angepaßt. Da jeder dieser Rechner den Supertape-Code "verstehen", kann er Daten mit einem anderen Rechner austauschen (entweder über den Umweg der Kassetten oder - über einen kleinen Signalverstärker - durch direkte Koppelung).

In Kürze sei das Verfahren vorgestellt: "Supertape" verwendet den sog. "Manchester II -Code" (modifiziert, s. elektor 10/82): Nach jedem übertragenen Bit wechselt das Signal seinen Pegel. War das Bit eine "0", so wechselt es seinen Pegel auch in der Mitte. Ein Beispiel:

```

_ | _ _ _ | _ _ _ | _ _ _ | _ _ _ _ _ | _ _ _ | _ _ _ _ _ | _ _ _ _ _ | _ _ _ _ _ |
,  0  ,  0  ,  1  ,  0  ,  1  ,  1  ,

```

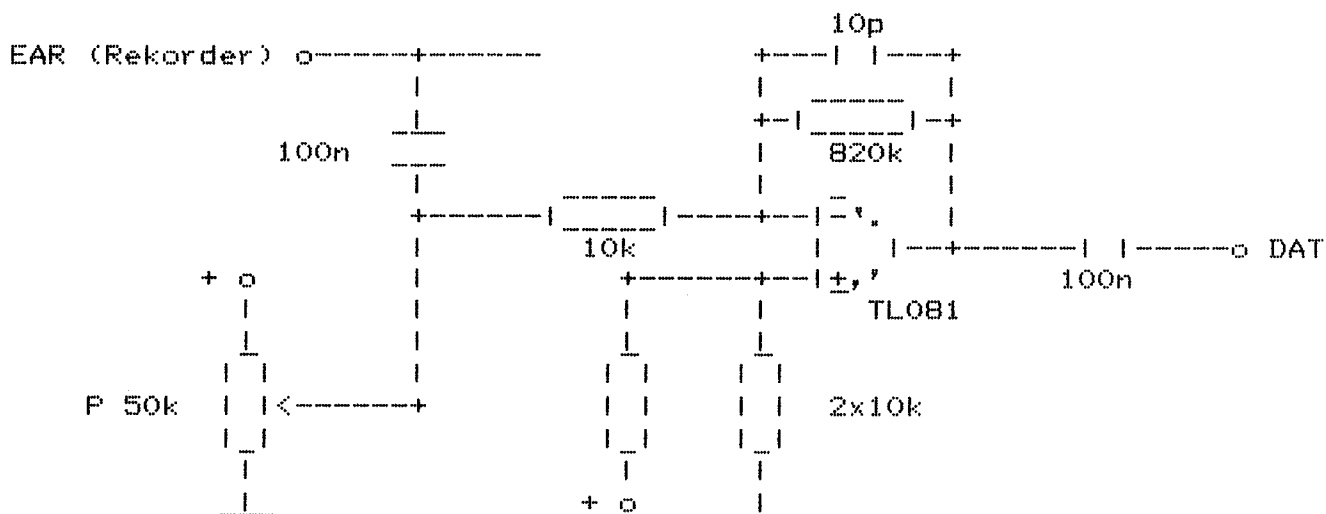
Dieses Verfahren hat u.a. den Vorteil, daß die Übertragung einer "1" genauso lang dauert wie die einer "0" (die in unserem MTX-Basic verwendete Routine erzeugt für eine "1" eine doppelt so lange Schwingung wie für eine "0").

Nicht ganz so sicher ist das Supertape-Verfahren deswegen, weil eine "1" das Band längere Zeit auf einem Pegel magnetisiert und deshalb der Wechsel auf den anderen Pegel nur verzögert erfolgt (jedenfalls bei hoher Baudrate).

Ich habe mir nun zu Testzwecken einen einfachen Mono-Rekorder (PHILIPS D6260, KAUFHOF, DM 79,-, mit Bandzählwerk) gekauft. Bis 3600 BAUD war dieser (unter bestimmten Voraussetzungen, s.u.) mit einfachen Kassetten sicher zu verwenden. Wenn man Chromdioxid-Kassetten (MAXELL) einlegt, laufen auch 7200 BAUD ohne Probleme. Zur höheren Datensicherheit habe ich dennoch mit Hilfe einer kleinen Komparatorschaltung das Wiedergabesignal aufbereitet (sie hat im Rekorder Platz gefunden):

(Der TLOB1 wird mit + (Pin 7) und Masse (P 4) versorgt (in meinem Rekorder sind 7V zu gewinnen), der invertierende Eingang (-) steht an P 2, der nicht invertierende (+) an P 3 und der Ausgang an P 6 zur Verfügung. Das Poti P dient zur Offseteinstellung des Eingangssignals und kann sinnvoll nur unter Kontrolle des Ausgangssignals an einem Oszilloskop eingestellt werden: Dazu lädt man einen größeren Speicherbereich mit einem abwechselnden Bitmuster (also OAAh) auf Kassette und schaut sich die Wiedergabe auf dem Oszilloskop an. Man wird einen Kompromiß finden, bei dem der mittlere Flankenwechsel einer "0" etwa in der Mitte liegt.)

Super Tape



Die Supertape-Laderoutine mußte ich größtenteils umschreiben: Wenn 2/3 bis 3/4 einer Bitperiode vorbei sind, fragt Supertape normalerweise den Eingangsport ab und schaut nach, ob der Pegel gewechselt hat. Unser MTX stellt aber nicht den an der EAR-Buchse anliegenden Pegel zur Verfügung, sondern generiert bei jedem Flankenwechsel einen kurzen Impuls, der dann an den CTC (Kanal 3) geführt wird, welcher wiederum einen Interrupt anfordert. Bei meinem Supertape werden deshalb Flankenwechsel interruptgesteuert ausgewertet.

Da RAM3 und RAM4x den Kanal 3 des CTC als Interruptgenerator für die Tastaturabfrage und RAM4x Kanal 0 für die Uhrdarstellung gebrauchen, untersucht das Programm, ob RAM3 oder RAM4x geladen sind, verstellt in diesem Falle die Interrupt-Vektoren und restauriert sie am Ende wieder.

Als Ladekontrolle gibt das Programm zusätzlich im Takt der Flankenwechsel Impulse auf den Soundchip aus (ähnlich wie in der MTX-Kassettenroutine).

Da billige Rekorder eine gewisse "Einschwingzeit" bei der Wiedergabe brauchen, habe ich die Anzahl der Synchronisationsbytes auf 500 erhöht.

Um beim Laden (STLZ.COM) kurze von langen Impulsen unterscheiden zu können, muß die Dauer zwischen zwei Impulsen mit einer bestimmten Schwelle verglichen werden. Diese mag bei einem anderen Rekorder evtl. anzupassen sein. Sie steht für 3600 BAUD bei 0109H und beträgt 13 (es funktionieren auch 12 und 14), für 7200 BAUD bei 010AH, hier lautet sie 4 (sehr unzuverlässig funktioniert bei mir auch 5).

In der c't 9/86 wurde ein BACKUP-Programm veröffentlicht, das das Sichern einer ganzen Diskette (evtl. auch mit Maske ...) sehr einfach macht. Telefonisch habe ich von der c't-Redaktion (Herr Assenbaum) die Zusage erhalten, die CP/M-Supertape-Routinen und dieses BACKUP-Programm als PD dem Club zur Verfügung zu stellen.

Alle Programme arbeiten weitgehend selbsterklärend, bei STSZ (Sichern) und STLZ (Laden) führt der Aufruf des Programmes, gefolgt von einem Leerzeichen und einem / (z.B. STSZ /) zu einem HELP-Bildschirm. STLZ <RET> ist dasselbe wie STLZ *.* , d.h., die nächstfolgende Datei wird eingelesen.

Das Schöne an den Programmen ist, daß sie beim Sichern automatisch CP/M-Dateien von Diskette lesen und dann ausgeben. Beim Zurücklesen werden die Dateien auf Wunsch als CP/M-Datei auf Diskette gespeichert.

Super Tape

Einen kleinen Haken hat die Sache: die Datei muß ganz in den Speicher passen, größere Dateien (ab etwa 48k beim Wiedereinladen) werden zurückgewiesen. Die gängigen Programme sind alle klein genug, ich kann mir eigentlich nur vorstellen, daß einmal eine Text-Datei (z.B. eine mit MBO erzeugte .PRN-Datei) zu groß ist. Diese kann man aber z.B. mit NEWWORD teilen.

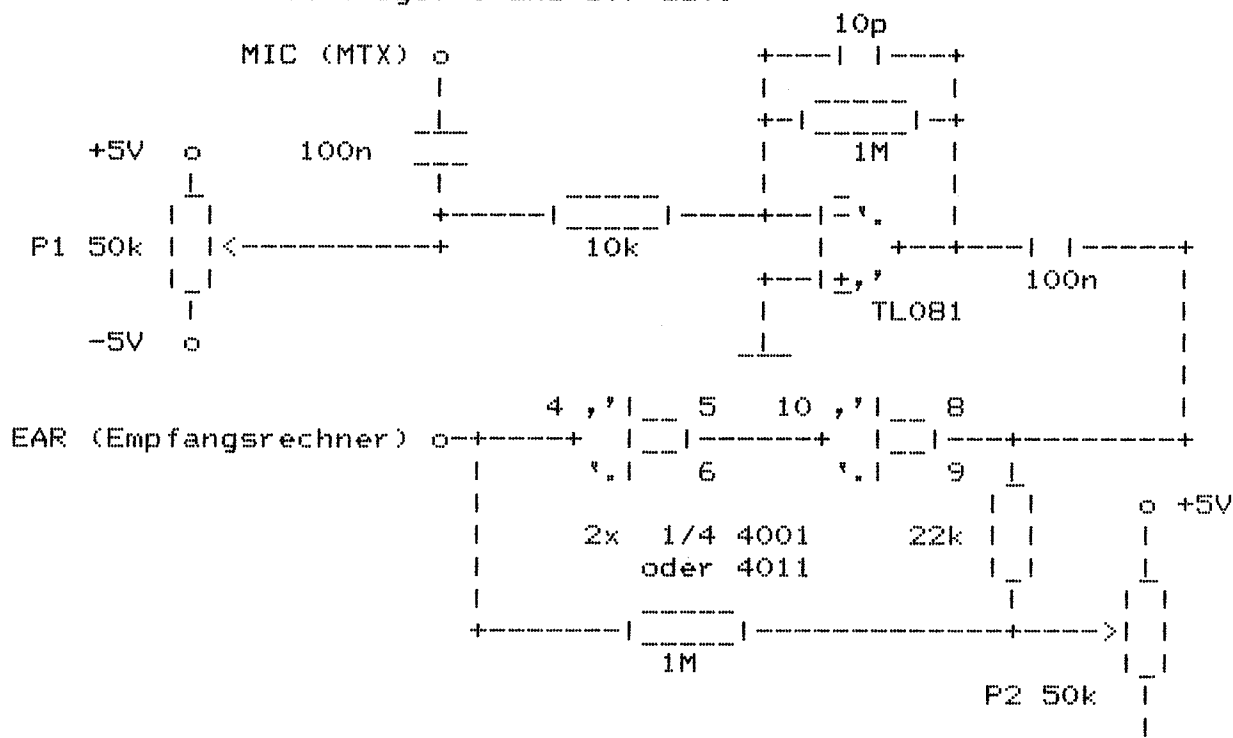
Mit Hilfe des BACKUP-Programms wird das Sichern von wertvollen Programmen auf Kassette auch für Disk-Besitzer interessant, v.a. für Single-Disk-Kopierer entfällt damit das ständige Wechseln der Scheiben. Und mit 7200 BAUD ist das Abspeichern recht flott (für eine volle Diskette etwa 5-6 Minuten Abspeicherungszeit + Zeit zum Einlesen von der Diskette). So lange braucht man auch bei der Erstellung einer Backup-Diskette mit nur einem Laufwerk.

Und noch ein Gesichtspunkt: Ich boote von einer CMOS-RAM-Floppy. Auf dieser und einer EPROM-Floppy sitzen alle wichtigen Programme. Notfalls kann ich somit auch mit dem Computer arbeiten, wenn die Floppy-Laufwerke einmal defekt sein sollten.

Wenn großes Interesse besteht, würde ich Supertape auch für den MTX in der Grundversion (also ohne Floppy) installieren, immerhin ist es doch um einiges schneller als die implementierten Kassettenroutinen.

Und zum Schluß noch einmal zu meiner Rechnerkopplung:

Der MTX gibt an der MIC-Buchse das Datensignal über einen Tiefpaßfilter aus (zu steile Flanken würden die Aufnahme- und Wiedergabeverstärker einfacher AUDIO-Rekorder "zustopfen"). Die Amplitude dieses Signals beträgt nur etwa 50 mV; das ist zu wenig für einen angeschlossenen "Empfangsrechner". Folgende Schaltung "möbelt" das Signal zu sauberen Rechtecken (Pegel 0 und 5V) auf:



Der TL081 braucht +/-5V (P7/4, s.o.), der 4001 (oder 4011) +5V (P14) und Masse (P7). Man gleicht zunächst mit Hilfe des Potis P1 wieder den Ausgang am TL081 ab, indem man sich vom Rechner ständig OAAh an der MIC-Buchse ausgeben läßt, möglichst mit 7200 BAUD. Anschließend wird der Ausgang zum Empfangsrechner genauso mit P2 abgeglichen. (Die Justierung des P1 ist besonders heikel (vielleicht sollte man hier lieber ein 10k-Poti nehmen).

K o m i k : Computerersetzter Hund

Klage eines computerersetzten Hundes

"Wir leben in einem freien Land," sagt Herrchen immer. "Hier kann jeder seine Meinung sagen." Das sagt er immer dann, wenn er abends schlechtgelaunt nach Haus kommt. Hat wohl Ärger mit seinem Boss gehabt.

"Ich lebe auch in einem freien Land!" sagt Frauchen dann ...

... Irgendwann hat Herrchen dann die Leine in der Hand und brüllt: "Bei Fuß!", und die Haustür schlägt krachend ins Schloß. Dann übt Herrchen mit mir: "Lauf! Faß! Aus! Bei Fuß!" Und ich tue oft, was er sagt. Schließlich ist er mein Schlappigeber.

In letzter Zeit spricht er allerdings oft von Veränderungen, ja, es hat sich schon einiges geändert.

Eine neue fernseherartige Maschine steht jetzt im Wohnzimmer. Statt nur hineinzuschauen, hämmert Herrchen an ihr herum und murmelt Befehle: "Go to 100! Run! Break! Go sub 25!" Und die Maschine tut oft, was er will.

Auch sie bekommt ihr "Menü", aber ihr "Output" sei besser, sagt Herrchen.

Frauchen meint, Hund sei gesünder, wegen der Bewegung.

Herrchen widerspricht: "Bei dem Sauwetter?? Davon muß man ja krank werden!"

Frauchen ist auf meiner Seite. Aber wie lange noch?

"Denk' doch nur mal an den Urlaub. Wer nimmt dann wieder den Hund? Der Computer braucht keinen Hüter! Und die viele Hundesteuer, die wir sparen! Davon kauf' ich Dir zu Weihnachten ein Videospiel. Dann kannst Du vormittags auch 'mal 'ran."

"Ja," seufzt sie mit verklärtem Blick und fügt, an mich gewendet hinzu: "Und ein Computer macht auch keine schmutzigen Tapsen auf dem Teppich!"

Sie vergaß wohl, daß sie immer das Altpapier hinausschleppt. Und das ist jetzt, na, über die Pfote, drei Mal so viel wie früher.

Und warum sagt keiner, was für ein ruhiger Hausgenosse ich bin? Ich belle nur, wenn jemand klingelt, jault nicht, also die anderen Hausbewohner finden mich vorbildlich. Hingegen diese Maschine... Und vor allem dieser fiepene Anhang, unbeschreiblich. Dauernd rattert und summt er, und der Teppich ist übersät von seinem Abfall. Das sollte mir einmal passieren! Prügel würde ich kriegen! Aber dort schaut Herrchen sich alles interessiert an...

Na, ich muß ohnehin mal nach ihm schauen. Ja, da sitzt er wieder und hämmert.

K o m i k : Computerersetzter Hund

Jau!! - Keine Reaktion. Ich versuch's mal mit der Leine.

JAUL!! - "Verdammter Köter, nun gib' endlich Ruhe!"

Jau!? - "Aus! Pfui! Go to Korb! Run!"

Das habe ich jetzt davon. Jahrelang treu gedient, Familie ausgeführt und unterhalten, das Haus gehütet; und das alles für ein bißchen Schlappi und ein wenig Streicheln. Und nun steht da die Maschine, neu, gehorsam und noch anspruchsloser.

Neulich fragte Herrchen schon, wo das Tierheim ist.

A s s e m b l e r k u r s

ASSEMBLERKURS

4 Die Z80 Befehle (Claudio Romanazzi, 3070)

Wir kommen nun zur eigentlichen Z80-Programmierung. Zum Glück hat unsere Maschine einen Assembler eingebaut. Er wird durch

Assem (oder A.) Zeilennummer <RET>

aufgerufen. Ich empfehle, Teil 5, Seite 125 im deutschen, Seite 129 im englischen MTX-Handbuch **jetzt** aufzuschlagen und die Bedienung des Assemblers zu studieren. Aus eigener Erfahrung kann ich sagen, daß es ein guter Assembler ist. Einige Befehle, die der Prozessor versteht, kennt er zwar nicht. Diese fehlenden Befehle sind aber für unsere Zwecke ohne Bedeutung. Gleich mitbehandelt wird hier das Panel. Die in meinen Augen wichtigste Funktion ist der Einzelschrittsimulator. Wir können mit seiner Hilfe jede Veränderung in jedem Register nach jedem einzelnen Ablauf eines Befehls nachverfolgen. Die Bedienungsanleitung steht auf Seite 156 im deutschen, auf Seite 161 im englischen Handbuch.

Tip: Wer Englisch versteht, sollte, wenn vorhanden, auch die englische Version des Handbuches lesen. Manchmal stehen da mehr Informationen drin, als im deutschen.

Bevor wir mit der Praxis beginnen noch eine Warnung: **Bitte**, bevor ihr ein Assemblerprogramm laufen laßt, macht euch die Freude, und speichert euer Werk ab. Für den Anfänger sind fehlerfreie Programme nicht allzu häufig, was dazu führt, daß der Computer während des Ablaufs in einen undefinierbaren Zustand fallen kann, aus dem er nicht mehr durch <BRK> herausgeholt werden kann. Als letzte Möglichkeit bleiben dann nur noch die Resettasten, --> das Programm ist verloren. Beim Eingeben von Programmen schadet ein Zwischenspeichern auch nicht. Gewöhnt euch das **gleich** an, ihr werdet es euch danken.

Nun, wie kann eigentlich eine Assemblerzeile aussehen? Hier die Lösung:

(Marke:) Befehl (;Kommentar)

Die in Klammern gesetzten Teile sind optional, d.h. sie werden nur bei Bedarf gesetzt. Links von Marke kommt noch die Adresse. Diese wird jedoch vom Assembler automatisch erzeugt und hochgezählt. Diese dürft ihr nicht überschreiben!

Die Marke, auch (englisch) Label genannt, ist ein Hilfsmittel für den Assembler. Sie bezeichnet einen Punkt, den der Assembler sich 'merkt' und auf den er sich bei Bedarf bezieht. Zum Beispiel kann man die Marke als Sprungziel verwenden. In Basic würde man GOTO ZEILENNUMMER sagen, in Assembler sagt man JP (=jump, englisch für springe) MARKE. Andere Funktionen benutzen eine Marke, um eine Speicherstelle zu markieren. Davon jedoch später.

Als Marke kann jeder Namen verwendet werden, der einem einfällt. Gewöhnlich benutzt man aber Worte, die im Programmzusammenhang einen Sinn haben. **Das solltet ihr auch tun.** In unserem Assembler kann dieser Name fast beliebig lang sein. Ich habe es mal ausprobiert: Er ist nur durch die Zeilenlänge des Editors beschränkt, d.h. er kann bis zu 157 Zeichen lang sein, ausgenommen das Leerzeichen. Eine solche Länge ist natürlich Unsinn. Wir beschränken uns am besten freiwillig auf maximal 8 Zeichen, einmal weil das genügt und weil, haben wir mal was Besseres

Assemblerkurs

an Assembler, diese Beschränkung sowiso auf uns zukommt. Nach der Marke folgt immer ein Doppelpunkt. Das zeigt dem Assembler, daß hier die Marke zuende ist.

Nach der Marke folgt ein Befehl. Die Befehle werden in den folgenden Kapiteln und Absätzen erklärt. Auch hier liefert der Assembler schon automatisch einen voreingestellten Befehl, nämlich RET, den ihr dann durch den von euch gewünschten überschreiben müßt, außer bei der letzten Programmzeile.

Nach einem Befehl **kann** ein Kommentar folgen. Er wird durch ein Semikolon angeführt (in BASIC kennen wir ja auch die REM Zeilen). Das weist dem Assembler an, alles bis zum nächsten Return, das die Zeile abschließt, zu ignorieren. Ein Kommentar soll den vor ihm stehenden Befehl näher erklären. Der Sinn ist, Fremden, die das Programm lesen, Hilfen beim Verstehen zu geben. Nach einigen Monaten ist der Schreiber selbst meistens ein Fremder, der das Programm neu verstehen muß. Deshalb sollte der Kommentar kein Kauderwelsch sein, den nur der Programmierer selbst, und das nur zu diesem Zeitpunkt, versteht, sondern aus wenigen, klaren Worten, die nur das hier Geschehnde kommentieren, bestehen. Wer meint, Kommentar sei gleichbedeutend mit Luxus, auf den man auch getrost verzichten kann, wird es früher oder später selbst bereuen. Obwohl Assemblerbefehle sehr einfach sind, kann ihre Wirkung manchmal doch verblüffend trickreich sein. An solchen Stellen ist Kommentierung ein Muß!

4.1 Der Lade-Befehl (Kai-Uwe Pleban, 6869)

Endlich - das lange Warten hat sich gelohnt - kommt der erste Z-80 Befehl. Wie ihr schon aus der Überschrift messerscharf geschlossen habt, handelt es sich um den Lade-Befehl. Das Mnemonic heißt LD, aber nicht wegen dem deutschen 'LaDen' sondern wegen des englischen 'to Load' (das bedeutet aber dasselbe). Zwischen Klein- und Großschreibung wird übrigens nicht unterschieden.

Der Prozessor läd auf diesen Befehl Daten in ein Register oder eine Speicherzelle. Tatsächlich handelt es sich bei LD nämlich nicht nur um einen einzigen Befehl, sondern um eine sehr umfangreiche Befehlsgruppe. (Für die anderen Befehle, die wir später noch besprechen, gilt das übrigens zumeist auch.) Für den Anfang will ich mich auf die einfachsten Lade-Befehle beschränken, nämlich auf Lade-Befehle mit Einzelregistern (also 8 bit Register). Wir werden uns im folgenden an die Konvention halten, daß r und r' ein Register (außer F) bezeichnet, also durch A, B, C, D, E, H oder L ersetzt werden kann. n soll eine Zahl bezeichnen, die in 8 bit "paßt", also kleiner 256 ist, während wir mit nn eine 16 bit Zahl meinen. Sehr häufig wird es sich dabei um eine Adresse handeln.

Da gibt es zunächst den Befehl

```
LD r,n      Lade Register r mit der Zahl n
```

Das ist eine allgemeine Schreibweise dieser Befehlsgruppe. Alle Befehle, die so aufgebaut sind werden in dieser Form zusammengefaßt, obwohl jeder Befehl mit einem bestimmten Register einen eigenen Befehl darstellt und auch einen eigenen Code hat (bei LD a,... ist es z.B. 3Eh). Aber um die verschiedenen Codes braucht ihr euch nicht zu kümmern; dazu ist das Assembler-Programm da!

Mit diesem Befehl könnt ihr eine Zahl im angegebenen Register speichern. Die vorher gespeicherte Zahl wird überschrieben, geht also

Assemblerkurs

verloren. Um z. B. die Zahl 45 in den Akku zu laden, müßt ihr den Befehl LD A,45 eingeben. Er ist vergleichbar mit dem Basic-Befehl LET A=45. Der Vergleich gilt jedoch nur bedingt: in Basic könnt ihr Anzahl und Namen der Variablen frei wählen, in Assembler habt ihr nur einen bestimmten Vorrat an Registern mit vom Assembler festgelegten Bezeichnungen. In Basic habt ihr noch andere Variablentypen wie Stringvariablen. In Assembler wird nicht so unterschieden. Eine Zahl in einem Register kann eine Zahl sein, oder ein Zeichen, eine Adresse (siehe auch Kapitel 2). Die Zahl n ist normalerweise in dezimal gemeint, z. B. wird durch den Befehl LD B,12 das B-Register mit der Zahl 12 geladen. **Wenn ihr n in hex angeben wollt, müßt ihr ein Lattenzaun '#' voranstellen.** LD B,#C hätte also das gleiche Ergebnis wie eben, da 12d =Ch ist. (Hier im Text werden wir häufig auch die Schreibweise mit dem nachgestellten 'h' für hex bzw. 'd' für dezimal verwenden.)

Nächster Befehl:

LD r,r' Lade Register r mit dem Inhalt von r'

Dieser Befehl läd den Inhalt von Register r' in das Register r. Der Inhalt von r' bleibt dabei gleich, so daß es sich im Endeffekt um eine Kopieroperation handelt. Um z. B. Register C nach E zu kopieren, gebt ihr LD E,C ein.

MERKE:

Die Datenbewegungsrichtung der Befehle ist immer von hinten nach vorn, also LD ZIEL,QUELLE.

In Basic wird in derselben Richtung zugeordnet, vgl. LD A,H mit LET A=H.

Und nun noch die Befehle

LD a,(nn) a ist der Akkumulator (NICHT mehr irgendein Register wie eben!)

LD (nn),a nn steht in diesem Fall für eine Adresse, und die Klammern geben an, daß der Inhalt dieser Speicherzelle gemeint ist.

Wenn ihr den Inhalt einer Speicherzelle in den Akkumulator laden wollt, bzw. umgekehrt, sind diese Befehle genau die richtigen. Was mit welchen geht müßt ihr als angehende Profis mittlerweile schon herausgefunden haben? Prima, ihr habt gut aufgepaßt! LD a,(nn) lädt den Inhalt der Speicherzelle nn in den Akkumulator, LD (nn),a vom Akkumulator in den Speicher. Auch hier gilt wieder: hexadezimalen Werten wird ein '#' vorangestellt.

LD a,(nn) entspricht beim I2DITVM dem Befehl LOAD (x); LD (nn),a entspricht STO (x). In BASIC schreibt man LET A=PEEK(34515) bzw. POKE 7345,A. Ihr seht: die Bezeichnungen der Mnemonics ????? ist willkürlich, und bei anderen Prozessoren werdet ihr andere Mnemonics vorfinden. So, nun wollen wir das Gelernte anwenden. Ein Super-Programm kann man mit diesen paar Befehlen noch nicht machen, aber ihr sollt nur mal sehn, wo's lang geht. Doch vorher will ich euch in die Assembler-Bedienung entführen.

4.1.1 EXKURS: WIE BEDIENE ICH DEN ASSEMBLER UND PANEL ?

Zunächst müssen wir den Assembler aktivieren. Dazu tippt ihr von Basic aus den Befehl (eure Eingaben sind unterstrichen)

ASSEM 100 <RET>

A s s e m b l e r k u r s

ein. Es erscheint dann am unteren Bildschirmrand:

```
Assemble>
```

Jetzt seid ihr im Assembler (vergl. dazu BASIC Handbuch Teil 5, S. 125). Um nun etwas einfügen zu können (= 'to insert'), drückt <RET>; ihr seht dann

```
8007      RET
```

Das bedeutet, daß in Adresse 8007h der Assemblerbefehl 'RET' steht. Ich werde ihn bald näher erklären... Beim MTX512 fangen die Adressen übrigens mit 4 an. Wenn ihr diesen Rechner habt, müßt ihr bei allen Adressen #8xxx durch #4xxx ersetzen.

Jetzt könnt ihr Zeile für Zeile das folgende Programm eingeben. Ihr müßt den alten Inhalt löschen (mit <EOL> und neuer Befehl eingeben) oder überschreiben (und den evtl. verbliebenen Rest des alten Befehls mit <EOL> löschen). In jedem Fall darf die Adresse nicht gelöscht werden. Das Programm selbst braucht für den MTX512 nicht verändert zu werden.

PROGRAMM 4.1-1

```
8007 START:  LD A,(40000)
800A        LD C,A
800B        LD A,(#9C41)  ;=40001
800E        LD B,A
800F        LD D,#FF
8011        LD E,255
8013        LD A,C        ;alter Wert zurück
8014        LD (40000),A
8017        RET
```

Das 'RET' am Ende braucht ihr nicht einzutippen; es bleibt stehen, wenn ihr den INSERT-Modus (also den Einfüge-Modus) mit <CLS> und <RET> verlaßt. Dieses 'RET' (steht für RETURN = kehre zurück) bewirkt den Rücksprung aus einem Unterprogramm; in diesem Fall ist euer Assembler-Programm ein Unterprogramm und die Programmabarbeitung wird durch dieses RETURN an BASIC zurückgegeben.

Um vom Assembler zurück in Basic zu kommen, braucht ihr nur nochmal <CLS> und <RET> einzugeben. Wenn ihr nun das Programm mit 'RUN' startet, seht ihr, daß ihr nichts seht. Warum auch? Im Programm ist kein Ausgabe-Befehl und in die Register könnt ihr nicht reinsehen. Es gibt allerdings so eine Art 'Fenster' in die CPU und den Speicher: PANEL.

In PANEL kommt ihr, wenn ihr in BASIC den Befehl PANEL (Abk. PAN.) eingibt (vergl. dazu auch PANEL in der Reference Section des BASIC Handbuchs, S. 156). Es meldet sich (beim MTX500) mit folgendem Bild:

A s s e m b l e r k u r s

```

AF 0000 C3
BC 0000 C3
DE 0000 C3
HL 0000 C3
IX 0000 C3
IY 0000 C3
SP 0000 C3
PC 0000 C3

```

```

JP #6F03

```

```

FFF0: 80 07 00 00 11 1C 00 00
FFF8: 00 00 00 00 00 00 00 00
0000: >C3 03 6F 00 00 C3 0B 63
0008: 5E 23 56 23 C9 FF FF FF
0010: E3 F5 7E FE 40 C3 FA 06
0018: C3 74 5B D7 2D 0A C9 00

```

Oben rechts seht ihr den Inhalt der Register. Die zweistellige Hex-Zahl rechts davon ist der Inhalt der Speicherzelle, auf die das 16-Bit-Register zeigt, in diesem Fall also der Inhalt von Adresse 0. Darunter ist der Befehl, der als nächstes ausgeführt wird (JP #6F03). Der Block ganz unten ist ein Auszug aus dem Speicher. Links steht die Adresse und rechts daneben die Inhalte dieser und der sieben folgenden Speicherzellen in Hex-Code. Durch ein '>' ist die aktuelle Speicherzelle markiert. Diesen Display-Cursor könnt ihr mit den Tasten <RET> (eine Adresse weiter), '-' (eine Adresse zurück), <Pfeil nach oben> (acht Adressen zurück) und <Pfeil nach unten> (acht Adressen vorwärts) in der näheren Umgebung positionieren. Wir wollen ihn jetzt aber auf eine weit entfernt liegende Adresse setzen, nämlich auf Adresse 9C40h (=40000d). Dazu gibt es den Befehl 'D' (wie Display). Er fragt nach einer Adresse.

MERKE

Die Antworten auf PANEL Befehle sind immer in hex einzugeben und das führende '#' entfällt hier!

Gib also nun ein

```

Display> 9C40<RET>

```

PANEL zeigt jetzt den Inhalt der angegebenen Adresse an und wartet. Worauf? Nun, das tolle (und gefährliche!) ist, daß du jetzt den Inhalt der Speicherzelle gleich ändern kannst. Damit wir merken, was das Programm 4.1-1 tut, wollen wir hier einmal die Adresse selber reinschreiben. Gib also ein:

```

9C40: 00 9C<RET>
9C41: 00 41<RET>

```

Hier brechen wir mit <BRK> ab. Wenn ihr bei 'D' keine Adresse angebt, wird einfach die aktuelle genommen, d. h. die, auf die der Display-Cursor zeigt.

Nun zurück zu unserem Programm. Damit wir es sehen können, hat PANEL den Befehl 'L' (wie List) parat. Er fragt wieder nach der Adresse:

```

List> 0007<RET>

```

Wenn ihr keine Adresse angebt, wird einfach weitergeblättert. Ihr seht nun das Programm genau so, wie Ihr es eingegeben habt, oben links neben dem Registerblock. Auch das Label 'START:' ist mit dabei!

Ihr könnt euch nun Schritt für Schritt - und nicht so rasend schnell, wie es in Echtzeit geht - anschauen, was euer Programm eigentlich tut.

Ihr MTX-Info-CopyRight © 1983-1992 Herbert zur Nedden, dieses PDF darf nur auf www.mtx-club.de/Online-Stiftung/ dieses PDF may only be online on www.mtx-club.de Dazu

Assemblerkurs

schreibt ihr in den Befehlszähler (engl. Program-counter abgek. PC) die Anfangsadresse: steuert den Register-Cursor '>' zu 'PC' (mit '.' abwärts, und von ganz unten springt er wieder hoch zu AF) und tippt 'R' (wie Register) ein. Ganz unten erscheint

```
Register>
```

und da tippt ihr die Anfangsadresse in Hex-Code ein, also wieder 8007. Unter dem Listing steht jetzt der erste Programmbefehl. (Es ist, kurz gesagt, immer der, auf den das PC Register zeigt, der also als nächstes zur Ausführung kommt.) Dann drückt ihr die Taste 'S' für 'SINGLE STEP'. Nun wurde der erste Befehl ausgeführt: der Inhalt von der Speicherzelle mit der Adresse 40000 dezimal (= 9999h) wurde in den Akkumulator geladen. Seht euch jetzt jeden Befehl an. Achtet dabei insbesondere darauf, wie sich die Register verändern. Wenn ihr genug mit PANEL gespielt habt, kommt ihr mit 'B' (wie BASIC) wieder heraus. Ihr müßt dazu die Frage 'Exit?' mit 'Y' beantworten.

PANEL ist gerade für Assemblerprogrammierer eine unschätzbare Hilfe und, ebenso wie der Assembler, in anderen Homecomputern so bald nicht wiederzufinden. Die C64'er dürfen sich so einen Monitor selber basteln. (Monitor hat in diesem Zusammenhang nichts mit Bildschirm zu tun, sondern ist ein Oberbegriff für ein Überwachungs- oder Debugging-Programm, wie es PANEL bei uns darstellt.) Hierin ragt der MTX Computer etwas aus der Masse heraus. Dementsprechend werden wir in Zukunft PANEL häufiger benutzen und daher ist es gut, wenn du dich mit seiner Bedienung jetzt schon etwas vertraut machst. Unter CP/M gibt es übrigens etwas ähnliches: VDEB und MONI.

4.2 Verbindung mit BASIC (Claudio Romanazzi,3070)

Da wir bis jetzt erst wenige Assembler-Befehle kennen und es außerdem sehr mühselig und unökonomisch wäre, alles in Assembler zu programmieren, werden wir in Zukunft unsere Assembler-routinen in BASIC Programme einbetten. Insbesondere die Ein- und Ausgabe läßt sich in BASIC viel einfacher durchführen. In diesem Abschnitt möchte ich zeigen, wie man die Daten zwischen BASIC Programm und Assembler Programm austauschen kann. Der typische Aufbau wird etwa folgender sein:

```
INPUT    Daten in BASIC eingeben
POKE     Daten in Assembler-routine abspeichern
GOSUB    Assembler-routine aufrufen
PEEK     veränderte Daten von dort abholen
PRINT    Daten in BASIC ausgeben.
```

Nachdem jetzt die Bedeutung von Peek und Poke klar ist, müssen wir natürlich einen Ort vereinbaren, der als Datenaustauschplatz fungieren soll. Zu ZX81-Zeiten hat man einfach eine REM-Zeile mit einer Anzahl von beliebigen Buchstaben geschrieben, sie an Anfang des Programms gesetzt und einfach den Inhalt mit Daten überschrieben. Der Ort war also definiert und der benötigte Platz durch die Buchstaben reserviert. Das könnten wir natürlich auch machen, müßten aber dann den Aufbau einer BASIC-Zeile im RAM kennen, um die richtigen Adresen errechnen zu können, mit einem Wort, es geht auch eleganter.

Unser Assembler stellt uns drei Pseudobefehle zur Verfügung. Diese Befehle werden also der CPU nicht übermittelt, sondern richten sich an den Assembler selbst und dienen dem alleinigen Zweck, Platz für Daten zu reservieren. **DB** (englisch: -> define Byte) reserviert ein Byte und **DW** (englisch: -> define word) reserviert zwei Bytes (ein Wort). Der

A s s e m b l e r k u r s

Inhalt dieses Bytes/Wortes muß immer auf den Befehl folgen. Durch Kommata kann der Befehl um weitere Bytes/Worte erweitert werden. Z.B.:

```
DB 1                ein Byte reservieren, Inhalt ist 1
DB 34              .. .. .. .. .. .. .. .. 34
DB "X"            .. .. .. .. .. .. .. .. ASCII Code von X
DB 1,34           zwei Bytes .. .. .. .. .. .. .. .. erstes Byte ist 1
                                      .. .. .. .. .. .. .. .. zweites .. .. 34
DB "ABCDEFGH"     sieben .. .. .. .. .. .. .. .. "ABCDEFGH"
(Buchstaben brauchen also keine Trennung durch Kommata!!)
usw.
```

```
DW 456             ein Wort reservieren, Inhalt ist 456
DW 65432          .. .. .. .. .. .. .. .. 65432
DW 456,65432     zwei Worte .. .. .. .. .. .. .. .. erstes Wort ist 456
                                      .. .. .. .. .. .. .. .. zweites .. .. 65432
usw.
```

Statt dezimale können natürlich auch hexadezimale Werte geschrieben werden, dann muß aber die Kennung '#' für hexadezimal vor die Zahl geschrieben werden, z.B. #10 bedeutet dezimal 16, #FB bedeutet 241.

Die dritte Art Platz zu reservieren heißt **DS** (englisch -> define space). Nach DS folgt eine Zahl, die angibt, wieviele Bytes reserviert werden sollen und 254 nicht überschreiten darf. Will man mehr Platz reservieren, muß man den ganzen Befehl entsprechend wiederholen. Im RAM werden die Bytes, die durch DS abgedeckt werden, auf Null gesetzt (gelöscht). Z. B.:

```
DS 55              reserviert 55 Bytes und setzt sie auf 0.
DS 254            .. .. 254 .. .. .. .. .. .. .. ..
DS 254
DS 254
DS 34             .. .. 542 .. .. .. .. .. .. .. ..
```

Da wir einen definierten Platz brauchen, müssen wir darauf achten, daß wir während des Programmierens immer von demselben Platz ausgehen, das bedeutet, daß die reservierten Bytes nicht verschoben werden dürfen. Wir geben deswegen als Zeile 0 immer ein GOTO 100 und als Zeile 1 eine Assemblerzeile ein. Genauso wie wir mit dem GOTO Befehl den Anfang des BASIC Programms überspringen, überspringen wir auch in Assembler den ersten Teil des Assemblerprogramms, in dem wir unsere Daten ablegen wollen. Der Befehl dazu heißt in Assembler JP von jump (springe). Wohin? Natürlich zu einer Adresse. Aber welche? Da wir die Daten ja noch nicht eingegeben haben, kennen wir die nächste freie Adresse ja noch gar nicht. Hier hilft ein Trick: wir nennen die Adresse START. Daß wir im Moment auch noch kein Label namens START definiert haben, stört den Assembler nicht weiter. Wir können es später nachholen. Das ganze sieht dann so aus:

A s s e m b l e r k u r s

```

0 GOTO 100
Assem 1 <RET>           ; errichte die Assemblerzeile 1
8010                   RET           ; ist voreingestellt
8010                   JP START      ; überspringe Datenbereich
8013  DATA1:         DS 100 <RET>    ; reserviere 100 Bytes mit Wert 0
8077  DATA2:         DB 234,234,234 <RET> ; res. 3 Bytes mit Wert 234
807A  DATA3:         DW #FFFF,#FFFF <RET> ; res. 2 Worte mit Wert hex FFFF
807E  START:         RET <RET>       ; RET stehenlassen
                               ; hier würde die Assembleroutine folgen
<CLS> <RET>           ; beende Eingabe
<CLS> <RET>           ; beende Assemblerzeile
2 RETURN

```

Hier können jetzt BASIC Kommentare folgen. In Zeile 100 fängt dann das BASIC Programm an. Das kann z. B. so aussehen:

```

100 GOSUB 1
110 STOP

```

Wenn wir jetzt listen, dann erscheint folgendes Bild:

```

0 GOTO 100
1 CODE

8010                   JP START      ; überspringe Datenbereich
8013 DATA1:         DS 100           ; reserviere 100 Bytes mit Wert 0
8077 DATA2:         DB 234,234,234   ; res. 3 Bytes mit Wert 234
807A DATA3:         DW #FFFF,#FFFF   ; res. 2 Worte mit Wert hex FFFF
807E START:         RET
807F                   RET

```

Symbols:

```

START      807E      DATA1      8013
DATA2      8077      DATA3      807A

```

```

2 RETURN
100 GOSUB 1
110 STOP

```

Wenn wir später noch mehr Speicherplätze für Daten brauchen, müssen diese hinter den letzten Speicherplatz, also gerade vor START eingefügt werden. Sonst würden wir ja die Adressen der folgenden Speicherplätze ändern, die wir aber von BASIC aus bei PEEK und POKE dezimal (**NICHT** als Label, leider!) angeben müssen. Das bedeutet, daß man sich immer genau bewußt sein muß, was man da gerade gemacht hat, sonst findet man seine eigenen Adressen nicht mehr wieder. Mit dem POKE-Befehl beziehen wir uns ja immer auf eine bestimmte Speicherzelle und wenn die Reservierung verschoben ist, dann schreiben wir im Basic (oder Assembler) herum und das Programm stürzt ab.

Wenn wir unsere Hausaufgaben von 2.2 gemacht haben, dann können wir anhand der Zahlen, die vor den Reservierungen stehen, die dezimale Adresse eines gewünschten Bytes errechnen. In Adresse #8077 steht wie oben angegeben 234. #8077 entspricht dezimal 32887. PEEK (32887) ergibt somit 234. Mit POKE 32887,X, wobei X eine beliebige Zahl zwischen 0 und 255 ist, kann ich diese, und nur diese Speicherstelle ändern. Dasgleiche gilt für alle anderen reservierten Plätze.

Assemblerkurs

dir nicht schwerfallen herauszufinden, was TAUSCH macht. Dieses Programm sollte auf MTX500/MTX512 lauffähig sein (wenn doch nicht, Kurt-Bernd anmosern).

Programm Tausch

```

0 GOTO 100
1 CODE

8010          JP START
8013 PLATZ1:  DB " "
8014 PLATZ2:  DB " "
8015 START:   LD A,(PLATZ1)      ;hole Byte aus PLATZ1
8018          LD H,A            ;merken in H
8019          LD A,(PLATZ2)     ;hole Byte aus Platz2
801C          LD (PLATZ1),A     ;dieses zurück nach PLATZ1
801F          LD A,H            ;erstes Byte zurückholen
8020          LD (PLATZ2),A     ;und nach PLATZ2 schreiben
8023          RET

```

Symbols:

```

START      8015      PLATZ1  8013
PLATZ2     8014

```

```

2 RETURN
10 REM Programm TAUSCH
20 REM Demo-Programm für Verbindung BASIC -- Assembler
30 REM von Kurt-Bernd Rohloff, 09.11.87
40 REM
50 REM DISC SAVE "TAUSCH.BAS"
60 REM LP.:LP.:LP.      "Programm TAUSCH":LP.:LL.
100 REM Hier ist der logische Anfang
110 DIM ZEICHEN$(2)
120 INPUT "Gib zwei Buchstaben ein ";ZEICHEN$
130 IF LEN(ZEICHEN$)<>2 THEN GOTO 120
140 REM Daten nach PLATZ1 und PLATZ2 laden:
150 REM Adressen für MTX512      MTX500
160 REM PLATZ1      4013      8013 hex
170 REM              16403     32787 dez
180 REM PLATZ2     4014      8014 hex
190 REM              16404     32788 dez
200 POKE 32787,ASC(ZEICHEN$(1))
210 POKE 32788,ASC(ZEICHEN$(2))
220 GOSUB 1: REM Assemblerprogramm aufrufen
230 PRINT "Vertauscht: ";CHR$(PEEK(32787));CHR$(PEEK(32788))
240 STOP

```

Wenn du den Assemblerteil mit dem PANEL untersuchen willst (was ich dir empfehle), solltest du die beiden Speicherplätze PLATZ1/2 vorher mit zwei verschiedenen Werten laden ('D' Befehl im PANEL), sonst siehst du nichts. Viel Spaß!

4.3 Inkrementier-/Dekrementierbefehle (Claudio Romanazzi, 3070)

In Basic kennen wir den Ausdruck LET A=A+1 oder LET A=A-1. Selbstverständlich kann man auch in Assembler einen Wert um eins vergrößern oder verkleinern. Inkrementieren bedeutet um eins vergrößern, dekrementieren um eins verkleinern. Der Befehlswortlaut ist INC r und DEC r, wobei r eines der Register A,B,C,D,E,H oder L ist.

A s s e m b l e r k u r s

Da Register nur Werte von 0 bis 255 enthalten können und man einen bequemen Weg brauchte, um festzustellen, wann 0 erreicht ist, hat man das Zeroflag eingeführt. Dieses Flag ist das sechste Bit des Flagregisters (F). Das F-Register wird bei jedem Programmschritt, soweit dieser ein Flag setzt oder löscht, aktualisiert. Das Zeroflag zeigt an, ob nach einer Rechenoperation (und bei unserem Thema handelt es sich um eine solche) 0 erreicht ist oder nicht. Ist 0 erreicht, ist das Zeroflag gesetzt, ist 0 nicht erreicht, ist es nicht gesetzt. Z.B.

A enthalte 0 und B enthalte 1.

Schreibe ich jetzt DEC A, erhalte ich 255 (= -1). 255 ist ungleich 0, also ist das Z-Flag nicht gesetzt.

Schreibe ich jetzt INC A, erhalte ich wieder 0 und das Z-Flag ist gesetzt.

Schreibe ich jetzt DEC B, erhalte ich wieder 0 und das Z-Flag bleibt gesetzt.

Schreibe ich dann INC B, erhalte ich 1 welches ungleich 0 ist und das Z-Flag wird gelöscht.

Das Z-Flag macht man sich zum Beispiel bei Schleifen, die den Wert 256 nicht überschreiten mit dem mächtigen Befehl DJNZ zunutze. DJNZ bedeutet **d**ecrement and **j**ump if **n**ot **z**ero, was bedeutet, ziehe eins ab und springe, wenn 0 nicht erreicht ist, also wenn das Z-Flag nicht gesetzt ist. Gemeint ist hier immer das B-Register. Um eine Schleife zu errichten, muß also zuerst immer das B-Register mit einem Wert geladen werden. Danach wird eine Marke gesetzt, auf die der erste Schleifenbefehl folgt. Auf diese Marke springt der DJNZ-Befehl solange, bis das B-Register 0 ist. Z.B.

```

ld    b,10          ; Schleife 10 mal durchlaufen
marke: nop          ; oder sonst was, das das B-Register
      djnz marke    ; nicht verändert
      ret

```

Hier wird zuerst das B-Register mit 10 geladen, danach die Marke mit dem ersten Schleifenbefehl gesetzt und dann der DJNZ-Befehl, der die Abbruchbedingung enthält, gesetzt. Beim ersten 'Wahrnehmen' von DJNZ wird das B-Register dekrementiert und dann geprüft ob das Z-Flag gesetzt ist. Ist es nicht gesetzt, springt der Befehlszähler zu Marke und macht da weiter. Das geht so lange, bis B gleich 0 ist und deshalb das Z-Flag gesetzt ist. Danach wird die Schleife verlassen und mit dem nächsten Befehl weitergemacht. Neben der Registergrenze von 256 gibt es für den DJNZ-Befehl nur noch eine Einschränkung: die Sprungweite. Man kann zwischen Marke und DJNZ nicht unbegrenzt viele Bytes schreiben, sondern ist auf etwa 126 Bytes beschränkt. Die Gründe werden wir später behandeln.

Nun zu dem Befehl, der in unserem kleinen Beispielprogramm vor DJNZ steht: NOP. Das bedeutet **n**o **o**peration, zu deutsch tu nichts. Der Sinn dieses Befehls liegt im Zeitablauf. Manchmal möchte man eine kleine Pause zwischen Befehlen haben. Da auch Nichtstun Zeit verbraucht, bei unserer Maschine 1 Microsekunde, konnten wir diesen Befehl gut gebrauchen und ihn gleich mit DJNZ zu einer Warteschleife verquicken. Dieses kleine Programmchen tut zehnmal nichts, wartet also.

Um das alles richtig zu verstehen solltet Ihr mal den Einzelschritt-simulator hinzuziehen. Gebt das Programm mal ein und dann ins PANEL.

Also:

A s s e m b l e r k u r s

```
<RET>
LD A,10 <RET>
MARKE: NOP <RET>
DJNZ MARKE <RET>
<CLS> <RET>
<CLS> <RET>
```

Jetzt listen und es müßte folgendes erscheinen (Zeilennummern für MTX 512, für MTX 500 #8xxx):

```
10  CODE

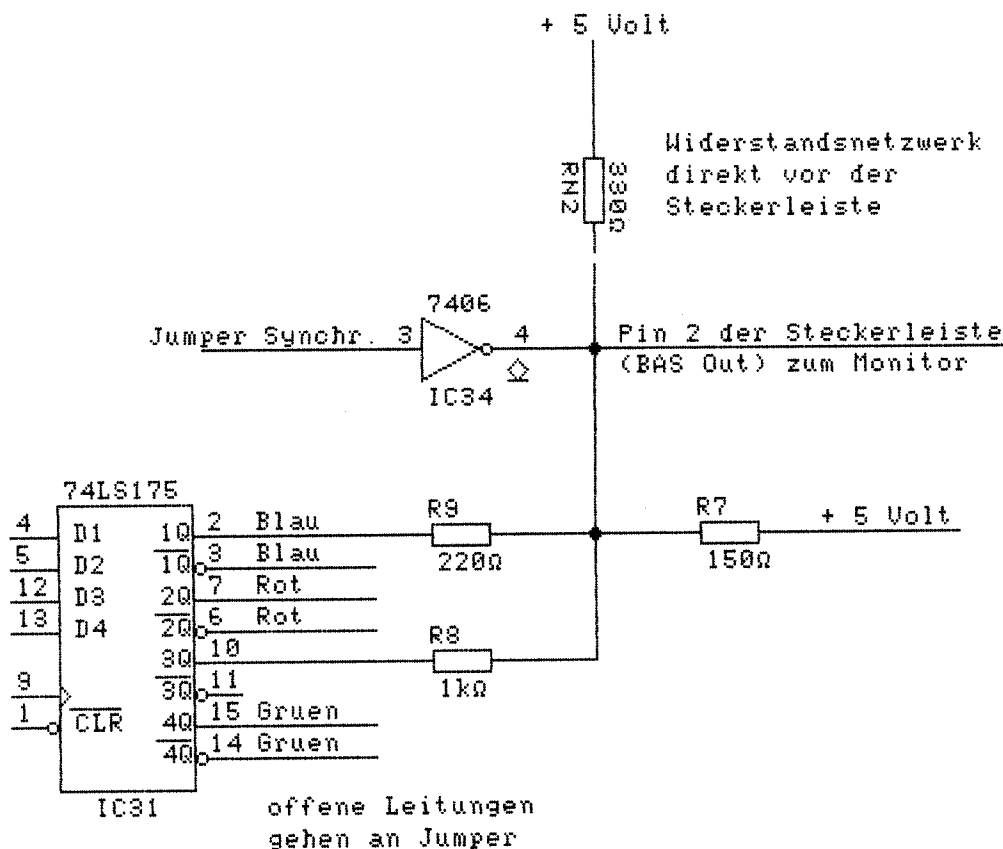
4007          LD    B,10
4009 MARKE:   NOP
400A          DJNZ MARKE
400C          RET
```

Symbols:
MARKE 4009

Jetzt PANEL im Direktmodus eingeben, dann L#4007 <RET> und es erscheint wieder unser Listing. Der Befehlszähler (PC, englisch --> program counter) steht, da alle Zahlen hier in hex angezeigt werden, auf 4007. Die Register stehen auf 0. Drücken wir jetzt auf S, wird der Befehl, auf den der PC zeigt, ausgeführt. In Register B steht nun 0A, der PC zeigt 4009. Sonst hat sich nichts verändert. Wieder S drücken und der Befehl NOP wird abgearbeitet, PC steht auf 400A. Jetzt kommts: drücken wir nochmal auf S, springt PC auf 4009, also auf MARKE und das B-Register ist um 1 kleiner geworden. Da aber 0 noch nicht erreicht ist, können wir das noch neunmal machen. Jetzt erscheint ein Z im F-Register, weil das B-Register 0 ist und der PC steht auf 400C. Die Schleife ist abgearbeitet.

Hardware: 80-Zeichen und VS4 angepaßt

80-Zeichen-Videoausgang an PAL-Board angepaßt (Joachim Keiser, 4925)
 Ewig habe ich mich über die unterschiedlichen Helligkeitspegel dieser beiden Videoausgänge geärgert. Also machte ich mich auf die Suche. Zuerst wurde ein gleiches Bild auf beiden Schirmen eingestellt und mit dem Oszilloskop die Ausgangssignale gemessen. Dabei stellte ich fest, daß die Spannungen der 80-Zeichenkarte wesentlich größer waren, als vom PAL-Board. Die sogenannte "Schwarzschar" oder auch Austastsignal genannt, lag einmal bei ca. 0,7 V und zum anderen bei 1,7 V. Das mußte doch irgendwie anzupassen sein. Da bei dem PAL-Board kaum etwas zu machen war, untersuchte ich die 80-Zeichenkarte näher. Mir war bekannt, daß sich das Videosignal aus drei Teilen zusammensetzt, den Bild-, Austast- und Synchronsignal (BAS-Signal). Auf der Platine verfolgte ich die Leitungen und fand die beiliegende Schaltung heraus. Die Bezeichnungen gelten für die 80-Zeichen/RS232-Karte der SDX-Version. Der Synchronpegel spielt eine untergeordnete Rolle. Wichtiger sind die Pegel des Austast- und Bildsignals, welche durch das Widerstandsnetzwerk R7, R8 und R9 gebildet werden. Ersetzt man diese drei Widerstände durch Potentiometer von ca 1 bis 2,5 KOhm, kann man seine persönliche Helligkeitspegeleinstellung vornehmen. ACHTUNG: R7 darf nicht Null werden, deshalb sicherheitshalber 100 Ohm in Reihe zum Poti legen. Man kann nach Einstellung die Widerstandswerte messen und die Potis wieder durch Festwiderstände ersetzen. Ich habe für mich folgende Widerstände eingebaut: R7 = 560 Ohm, R8 = 100 Ohm und R9 = 390 Ohm.



Hardware: RAM-Hochstapelei

(Joachim Keiser, 4925)

Hallo Herbert,

zwischen den Feiertagen hatte ich mal wieder ein wenig mehr Zeit und konnte mich mit dem MTX beschäftigen.

Die 512k-Speichererweiterung war mein erstes Ziel und es hat geklappt. Die 2 * 8 RAMs habe ich einfach auf die RAMs des Motherboards gelötet. Den Inverter IC2B durch eine Buchsenleiste ersetzt, in die eine Wrapfassung gesteckt wird. An der Fassung hängt eine kleine Platine, auf der sich vier weitere normale Fassungen befinden. In die Wrapfassung kommt der Inverter IC2B, in die Übrigen kommen das PROM der 512k-Speichererweiterung, ein 74LS32, ein 74LS08 und eine weiterer 74LS04. Die Verdrahtung kannst du dem beiliegendem Plan entnehmen. Pin 8 der Wrapfassung wurde oberhalb der Platine durchgesägt und von der IC-Fassung geht es an Pin 13 des 74LS32 mit einem Draht. PIN 11 geht an den Stift 8 der Wrapfassung und somit zum IC5B.

Dieses in der CAS-Signalerzeugung des Hauptspeichers (/CAS0) eingefügte OR verhindert, daß bei Zugriffen auf die 512k-Speichererweiterung der Hauptspeicher mit angesprochen wird. Die übrigen CAS-Signale werden wie bei der Schaltung für die Speichererweiterung erzeugt. Alle Signale, die nicht an der Wrapfassung zur Verfügung stehen, wurden mittels Wrapdraht hergestellt. Ich verwende also für die Ansteuerung der RAMs (alles 120ns-Typen) nur eine Laufzeitkette. Das ist aber nur möglich, wenn das PAL entsprechend geändert wird, d.h. das /RAM-Signal muß in jeder Bank erzeugt werden. Das PAL läßt sich aber umprogrammieren. Die Gleichung für den Ausgang PIN 14 lautet nun:

```

/RAM =                /RECPM * /MREQ *                /LK7
+ A14 * /A15 * /RECPM * /MREQ * /P3 * /P2 * /P1 * /P0 * /LK7
+ A14 * A15 *                /MREQ *                /LK7
+ /A14 * A15 * /RECPM * /MREQ * /P3 * /P2 * /P1 *    /LK7;

```

Damit konnte ich nun auch RAM42 installieren und ich habe zwar noch nicht alles richtig ausgeschöpft, aber ich bin begeistert. Fragen dazu werde ich wohl am Besten mit Bernd Preusing klären.

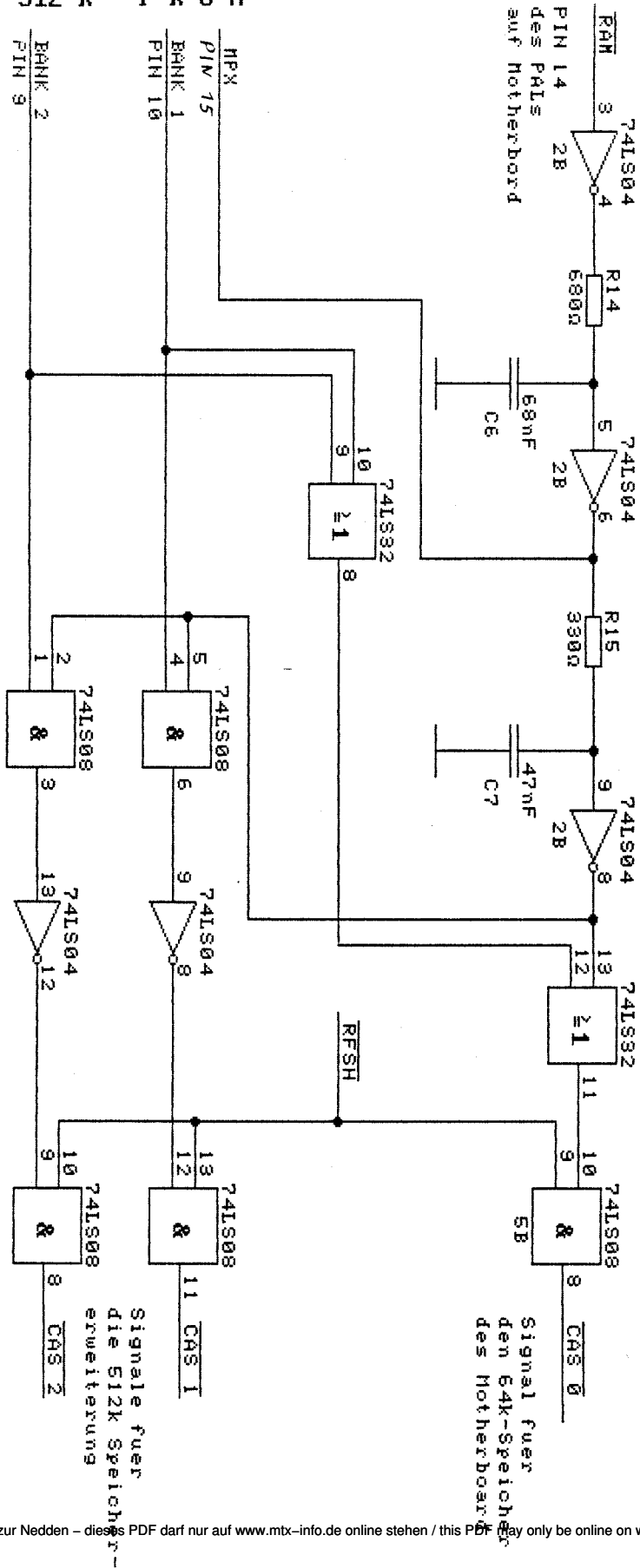
Soweit zum Speicher, ob's fürs Info brauchbar ist, könnt Ihr ja selbst entscheiden, aber das Beiliegende ist bestimmt dafür brauchbar. Es geht um die unterschiedlichen Helligkeitspegel zwischen der 80-Zeichen Karte und dem PAL-Board.

Grüß Joachim



Hardware: RAM-Hochstapelei

512 K - P R O M



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.