MTX User-Club Deutschland

Info 20
25.05.1987

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's Guthaben nicht reicht! (s.u.)
Schüler, Studenten, Auszubildende, W15-er, Rentner und Arbeitslose erhalten einen Nachlaß von 40% auf die zukünftigen Infos nach Einsendung einer entsprechenden Bescheinigung. Die Bescheinigung gilt nur für den auf ihr genannten Gültigkeitzeitraum.

Geld/Konto: Für jedes Mitglied führt Herbert Herberg ein Konto, von dem die jeweils enstehenden 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 nicht vergessen!)
Postgiroamt Hamburg, BLZ 200 100 20,
Herbert Herberg, Sonderkonto C, Nr. 3480 00-200

Kontaktadressen: (nach PLZ geordnet)

Herbert Herberg	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 Herberg: Do 18 - 22 Uhr, Sa 13 - 16 Uhr

I n h a l t s v e r z e i c h n i s

Preis für dieses

Info: DM 10,40

Redaktionsschluß
für Info 21:

01. Juli 1987

Bitte entschuldigt
den teilweise etwas verschmierten und fetten Druck!
Ich habe ein neues Original-Panasonic-Farbband von K-H. Harter im Drucker, und vor dem Info in das Loch hinten gepiekst, um die Reserven des Bandes zu aktivieren.
Resultat ist dieser Schmierkarm - 50% des Infos habe ich ein zweites Mal gedruckt!

Liebe Leserinnen, liebe Leser,

nun haben wir das Info Nummer 20 herausgebracht. Das ist eine **runde Zahl** (ja, in Hex ist zwanzig = 14, also krumm, aber ganz so verquer bin ich nicht). Wenn ich an die Anfänge des MTX User-Club Deutschland denke, dann finde ich, daß der Club fantastisch gewachsen und gediehen ist, wofür ich Euch allen herzlich danken möchte. Ich weiß, daß leider einige nicht so überzeugt von unseren Leistungen sind; einige Wünsche können wir nicht erfüllen, andere kennen wir gar nicht. Bitte laßt mich auch weiterhin Eure Wünsche, Fragen, Bedürfnisse und Änderungswünsche wissen.

Bitte entschuldigt, wenn ich hin und wieder nicht zu meinen **Telefonsprechzeiten** greifbar bin. Ich versuche immer zu diesen Zeiten zu hause am Telefon zu sein, aber leider kommt auch mal etwas dazwischen. Tut mir leid, ist aber nicht zu ändern.

Ich höre immer wieder etwas von einem **'Sinkenden Schiff'** genannt Memotech-Computer. Und dann auch noch so viele Kleinanzeigen bzgl. Verkauf MTX/FDX. A-B-E-R der Schein trügt. Jedenfalls soweit ich es beurteilen kann. Ich habe immer wieder Anfragen betreffs Kauf einer FDX, und sogar betreffs Kauf eines MTX/FDX-Komplettsystems. Wenn ich mir dann auch noch ansehe, was unser Computer kann, von dem viele Besitzer anderer Computersysteme nicht einmal träumen, dann verstehe ich die Verkäufe kaum. Und Boot-Probleme sind leicht beseitigbar! Und wenn ich bedenke, daß unser NewWord mit dem Calculator, den Bernd für das Klick schreibt Rechnenunterstützung erhält ... was will man mehr. CAD ist auch in Arbeit.

A pro pos: **Bernd Preusing** hat sich als Zweitgerät einen Amiga 2000 zugelegt. Er behält den Memotech, aber in der nächsten Zeit wird er nichts neues für den MTX produzieren. Nichtsdestotrotz hält er den Service für RAM4 aufrecht. Da der Calculator noch nicht fertig ist, und Olaf Prüfungen bevorstehen wird dieser leider noch etwas auf sich warten lassen. Aber er kommt bestimmt!

Da seit meinem letzten Hinweis einige Zeit ins Land gegangen ist, möchte ich an dieser Stelle nocheinmal erläutern, warum ich im Info alle **DU**tze. Anfangs habe ich das Info 'unpersönlich' verfaßt, d.h. immer wieder mit man/frau gearbeitet. Da ich es jedoch wesentlich ansprechender und netter finde, wenn das Info ansprechend ist, stand ich vor der Wahl SIE und DU. Letzteres ist m.E. die schönere Variante, die ich daher auch wahl - in der Hoffnung, daß Sie es akzeptieren.

Auf vielfachen Wunsch habe ich diesem Info mal eine nach Nachnamen alphabetisch sortierte **Mitgliederliste** beigefügt. Eine Änderung hat es auch mal wieder gegeben: Für das **Inhaltsverzeichnis** des Club-Info ist jetzt dipl-ing. Ulrich Hönisch, Wachholtzstr. 8, 3300 Braunschweig, 0531-343167 zuständig. Für Anregungen und Kritik diesbezüglich hat er ein offenes Ohr!

Und nun noch etwas in eigener Sache: Ich werde am 22. Juni 1987 heiraten, und ab dem Tag den Nachnamen **zur Nedden** führen. Ich hoffe, daß die Umstellung des Club-Postgirokontos auf diesen Namen nicht allzulange dauert. Daher mächte ich Euch bitten ab **01.07.1987** für Überweisungen und Schecks meinen zukünftigen Namen zur Nedden zu verwenden - sicherheitshalber nicht eher. Den Namen Herberg akzeptiert die Post noch eine ganze Weile.

Auf das die Bit's so purzeln, wie sie sollen!

Hubert Hesky

C L U B: Wer tut Was / Ports

Wer tut Was

Allgemeines	H. Herberg
Info-Inhaltsverzeichnis	U. Hönisch
(FDX-)BASIC	A. Viebke
CP/M System	B. Preusing, H. Herberg
Assembler	H. Oppmann
NewWord	U. Grass, H. Herberg
Turbo-Pascal	O. Krumnow, T. Wulf
SuperCalc	W. Gieger
Edicta-Grafik	H. Herberg, C. Löhrmann, C. Romanazzi
Was gibt's wo billig	H. Herberg
Hardware	H. Herberg, P. Kretschmar, U. Hönisch
Reparatur	U. Hönisch, H. Herberg, U. Grass

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

Ports (Herbert Herberg)

Bereich	Port	Verwendung
MTX	00 - 0F	Grudgerät
	10 - 14	SDX-Floppy-Controller!
	18 - 1B	8255-PIO-Box, H. Herberg
	1F	vorgesehen für Cassettenmotorsteuerung
FDX	30 - 33	80-Zeichen-Karte
	38 - 39	6845-Controller der 80-Zeichen-Karte
	40 - 47	FDX-Floppy-Controller
	70 - 73	EPROM/SRAM-Floppy von J. Marquart und F. Cröll
ECB	80 - 83	EDICTA Grafik-Karte
	88 - 8B	Reserviert für HardDisk
	98 - 9B	c't RAM-Floppy
	A0 - A3	EDICTA RAM-Floppy
	A4 - A7	c't EPROM-Floppy
	A8 - AB	c't SRAM-Floppy
	B8 - BB	Conitec-Floppy
	BC - BF	Conitec-Floppy
	C0 - C4	Reserviert für Testzwecke !!!!!!
	CC - CF	Janich & Klass Programmer
	F8 - 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

K L E I N A N Z E I G E N

Anzeigetexte und Absender bitte schriftich an Herbert Herberg!

Herbert Herberg, Sonnenau 2, 2000 Hamburg 76, 040 - 2008704:
(Preise sind ohne Porto & Verpackung, ich gebe ggf. Mengenrabatt)
- Ich vermittle 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.
- FDX, ein oder zwei Laufwerk(e), in Top-Zustand (Post-versand-fähig, d.h. rüttelfest) für DM 850.-
- SDX, ein Laufwerk, in Top-Zustand (Post-versand-fähig, d.h. rüttelfest) für DM 750.-
  Was ich weitergebe ist überprüft, FDX bootet dann einwandfrei!
- Farbmonitor mit RGB-, BAS- und Toneingängen, RGB/BAS-Umschalter. Auflösung für 40-Zeichen gut, für 80-Zeichen nicht berauschend, Schrift jedoch lesbar: VB DM 300.-
- Kleinen S/W-Monitor, als Zweitgerät für VS 4 geeignet: DM 60.-
- FDX-Netzteil(e), original, 100%-ig o.k. Uli und ich haben in unserer FDX ein stärkeres Netzteil von wegen des ECB-Busses und der Hauptplatine. Preis je Netzteil: DM 95.-
- Einspaltige Etiketten, 1000 Stück, 8,9 cm x 3,6 cm: DM 16.-
- Sucht jemand eine 100%-ig bestückten FDX-Floppy-Controllerkarte im Tausch gegen die Seine/Ihre ? Ich habe so eine wo alle IC's, DIP-Schalter, ... drin sind, und tausche diese gegen eine 100%-ig funktionstüchtige andere Controllerplatine für DM 90.- (allein die zusätzlichen Bauteile kosten einiges!)

Solange der Vorrat reicht:
- 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.-
- TTL-IC's: 74LS175, 74LS368, 74LS 21, 74LS173, 74LS04, 74LS14, 75LS155, 74LS158, 74LS139, 74LS258 je DM 0.50; 74LS10, 74LS11 DM 0.30
- Z80-Chips: Z80A CPU DM 1.50, Z80 PIO DM 1.50, Z80A PIO DM 3.-, Z80A CTC DM 2.50, Z80A SIO DM 4.-

V E R K A U F

Uwe Grass, Wachholtzstr. 8, 3300 Braunschweig, 0531-343167:
MTX 500 mit 512k, RS232, ECB-Option, Netzteilumbau, verlötete Platinen, 5MHz umschaltbar (also allem was gut und teuer ist), FDX 2 Lw., Monitor DM 2200,--. Auf Wunsch wird auch noch die 80-Zeichenkarte umgebaut, Booteprom getauscht, SRAM-Floppy installiert.

Willi Rowedder, Wiesengrund 4, 2210 Heiligenstedten
DMX 80 incl. Kabel, Handbuch für DM 250.-

Jens Ebert, Otawiweg 15, 2000 Hamburg 50, 040 - 8801267:
MTX 500 mit 128k-Karte, FDX, 2 Lw., voll bestückte Controller-Platine, neuer 80Zeichen-Satz, alle Infos, CP/M-Literatur, DMX 80 für DM 1400.-

Peter Braun, Veldastr. 23, 5000 Köln 1, 0221 - 373490 ab 17.00 Uhr:
MTX 500 (96kB), FDX, 2 Lw., TP200, fahrbares Gestell, Spiele, Disketten, Bücher, Infos, 32k-Karte VB DM 1300.-

Jörg Lengnick, Velberstr. 10, 3000 Hannover 91, 0511 - 4581218:
MTX 500, FDX, TP 200, 22 Disketten (darunter 5 PD, 4 BASIC), etliche Infos, TURBO 3.0 DM 1200.-

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

Korrektur & Nachtrag

CLUB.006/013: Window-Demo und Window-Routinen. Das Byte Attribute wird leider auch mal Attribute genannt. Das e am Ende sollte weg.

CLUB.006, WINDOW3.INC arbeitet mit der aufgerüsteten 80Z-Karte nicht zusammen. Lösung: In den Procedures LoadScreen und SaveScreen kurz nach dem Label Loop2: steht die Zeile

```
 $e6/$07/            (*     and  7                          *)
```

die in die Zeile

```
 $e6/$1f/            (*     and 1f                          *)
```

geändert werden muß.

Info 12, Seite 30: (Peter Würfel, 7262)
.cw6, .cw8, .cw9 geht nicht

Info 19, Seite 36: (Josef Heuper, 5461)
Tastaturbelegung. Die Tasten in den letzten beiden Spalten haben andere Wertigkeit als in der Tabelle. Ist beim ROM-BASIC leicht zu überprüfen:

```
10 PRINT PEEK(64892)
20 GOTO 10
```

SPACE=49, DEL=4D, TAB=4E, BS=4F, F8=42, F4=41, F7=44, F3=43, F6=45, u.s.w.

Info 17, Seite 24: (Josef Heuper, 5461)
Zeile 1030 mit OR und AND sowie Zeile 2030 mit AND geht im ROM-BASIC. Von wem ist dieses Programm?
Das Programm ist von Memotech - bei denen es übrigens auch nicht läuft.
Korrektur: (Herbert Herberg, 2000)

```
1030 LET TEMP=64+MOD(TEMP2,63)
2030 LET TEMP2=MOD(TEMP2,63)
```

Diese einfache Lösung liegt übrigens nur an den Zahlen 63, 64, 128!

Fragen / Antowrten

F: Wie fuktioniert der RESET-Befehl unter dBASE ?
A: Das Laufwertk muß in Großbuchstaben angegeben werden, z.B.:

```
RESET B:
```

F: Sag mal, Kurt-Bernd Rohloff, warum hast Du für den Assemblerkurs einen fiktiven Rechner mit seiner eigenen Assemblersprache entworfen. Ich frage mich, ob nicht ein Z80-Assemblerkurs ....
(Herbert Herberg, 2000)

B A S I C: Grafik einfach

Einfache schnelle Grafiken (Helge Siegloff, 2370)

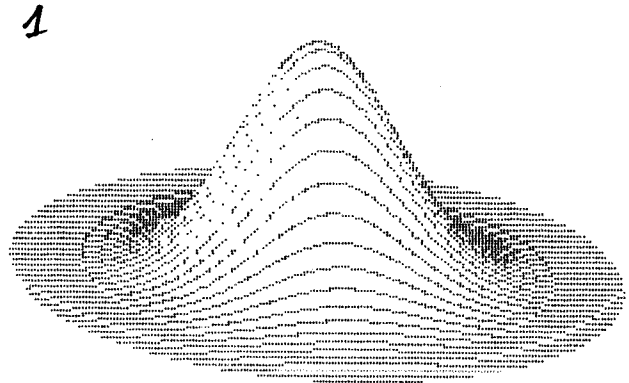
```
100 CLOCK "000000": GOSUB 270
110 FOR Y=-I TO I STEP DY
120 LET KX=Y/DY: LET KY=W*Y: LET ZE=SQR(I*I-Y*Y)
130 FOR X=-ZE TO ZE
140 LET R=SQR(X*X+Y*Y)*AF
150 LET Z=S*EXP(-R*R)
160 LET XO=X+KX+U: LET YO=Z-KY+V
230 PLOT XO*.79,YO*.96
240 ATTR 2,1:LINE XO*.79,YO*.96-1,XO*.79,0: ATTR 2,0
250 NEXT : NEXT : LET Z$=TIME$
260 SOUND 1,200,15: PAUSE 1000: SOUND 1,0,0: PAUSE 1000
261 IF INKEY$="" THEN GOTO 260
262 VS 5: PRINT Z$: STOP
270 LET U=170: LET V=120: LET I=121: LET S=90
271 LET DY=6: LET A=PI*7/8: LET AF=A/I: LET W=.35
280 VS 4: CLS : RETURN
```

Dieses Programm liefert Bild 1.

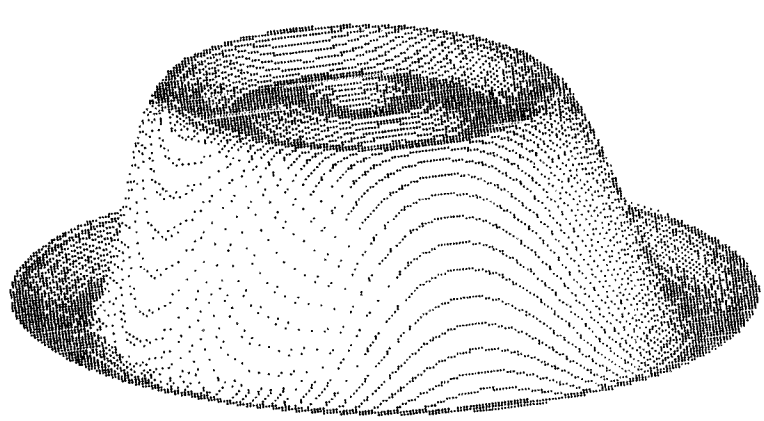
Durch austauschen der folgenden drei Zeilen gibt's Bild 2:

```
150 LET Z=S*(COS(R)-COS(3*R)/3+COS(5*R)/5-COS(7*R)/7)
270 LET U=160: LET V=100: LET I=141: LET S=50
271 LET DY=3: LET A=PI*.75: LET AF=A/I: LET W=.35
```

1



2



B A S I C: Aus ASCII mach Programm

BASIC: AUS ASCII MACH PROGRAMM (ABER DALLI)! *Andreas Viebke*

Da haben wir's wieder mal: es gibt noch Leute, die keine Floppy haben. Und auch nicht soundsoviele Megatonnen - äh Kilobyte. Leute, die einfach nur 'son bißchen fummeln wolln'. Ja.

Rumgespielt hat da jemand mit VIDEODAT, einer Methode, derer sich der WDR bedient, um Daten (vorwiegend Programme als ASCII-Texte) über den (Fernseh-)Äther zu schicken. So genau kenne ich mich da nicht aus, ist auch nicht nötig, denn ich will gar nicht darüber berichten.

Jedenfalls hat dann dieser Jemand, oder vielleicht auch ein anderer, nachdem er mit VIDEODAT etwas empfangen hat, im Speicher einen ASCII-Text zu stehen, den er zu gerne in ein Programm umwandeln würde. Na ja, eigentlich ist es ja schon eins, nur laufen tut's nicht, weil so ein ASCII-Text dem Herrn Memotech nicht genehm ist. Wie hätten Sie's denn gerne?

Ja, am besten wär's, wenn jede Zeile in die Editier-Zeile (was für'n Tier?) gebracht würde, und wenn Fehler drin sein sollten, müßte ich was ändern können, und dann wird die Zeile als ganz normale Programmzeile, hrhmm, angenommen.

Sehr geehrter Herr Jemand! Wenn Sie folgendes Programm abtippen - am besten hängen Sie's an Ihr VIDEODAT-Dingsda an (muß aber nicht sein) -, und alles zusammen abspeichern, dann können Sie genau das tun.

Im Programm ist eine Stelle, bei der Sie angeben müssen, bei welcher Adresse der ASCII-Text beginnt. In meinem Listing (siehe bitte unten) ist das dort, wo "OVER: LD HL,#A000" steht. Das "#A000" ersetzten Sie durch diese Adresse. Dann ist noch etwas zu beachten: Irgendwie muß mein Prográmmchen ja das Ende des ASCII-Textes erkennen können. Es nimmt an, daß das letzte Zeichen eine 26 ist. Sie wissen schon, was ich meine. Sollte das nicht so sein - ich meine das mit der 26 -, sondern ist das letzte Zeichen ein anderes, z.B. 0, dann tragen Sie diesen Wert bitte dort ein, wo die drei Pfeile ("<<<") sind.

So! Jetzt können Sie starten! Mein Prográmmchen liest nun gewissermaßen die Zeichen aus dem Speicher und schreibt sie in die Kommandozeile, als wären sie von Hand eingegeben worden. Stößt es auf ein RET - jede Zeile muß unbedingt mit RET (Code 13) abgeschlossen sein -, dann tut es so, als ob der Benutzer RET gedrückt hätte. Und an dieser Stelle merken Sie dann, ob in der Zeile ein Fehler ist oder nicht. Ist kein Fehler drin, wird die Zeile übernommen und "Ready" erscheint. Ansonsten können Sie machen was Sie wollen. Korrigieren, sich (und anderen) einen Kaffee kochen... na ja, was Sie wollen. Sind Sie fertig und haben das "Ready" wieder vor Augen, dann drücken Sie auf "LINE FEED", und swupps ist die nächste Zeile zu sehen. (Das geht 25mal schneller, als Sie sie ohne Brille lesen können.) Sie können das so lange machen, bis Sie umkippen oder bis nach dem Drücken von LINE FEED die Kommandozeile leer bleibt und ganz unten, wo sonst immer die Fehlermeldungen erscheinen, nun plötzlich "End of text" steht. Dann ist nichts mehr da! Und wenn Ihnen sogar mitten drin einfällt, daß Sie keine Lust mehr haben, dann drücken Sie bitte auf ESC! Dann erscheint das "End of text", und Prográmmchen hat diesen Modus der "Eingabe aus dem Speicher" wieder verlassen.

B A S I C: Aus ASCII mach Programm

Noch etwas: Wichtig sind für mein Programm nur folgende Zeichen im ASCII-Text: 13 (RET), 10 (LINE FEED), 26 (EOF oder HOME) und die anderen, deren Code größer oder gleich 32 (SPC) ist. Alle nicht genannten Control-Zeichen werden ignoriert. Was 13 und 26 bewirken, sagte ich schon. Was macht die 10? Ganz einfach: Statt ihrer wird HOME zum Schirm gesandt, damit der Cursor an den Anfang der Kommandozeile huscht. Wenn nämlich zufällig der AUTO-Modus an ist, wird keine Zeile mehr angenommen.

Genug geschwafelt. Bleibt nur noch zu sagen, daß die im Speicher stehenden Texte nicht unbedingt von VIDEODAT kommen müssen, und daß das Programm sowohl mit als auch ohne Floppy funkzjohniehrt. Bei mir jedenfalls.

```
8007          RST 10      ;       ;VS 0 anwählen
8008          DB #48      ;       ;und löschen
8009          DB #C3      ;       ;JP
800A SKIP:    DW INIT     ;       ;erst zu INIT, dann zu OVER
800C INIT:    LD HL,(#FD52)       ;Urspr. KBDIO merken
800F          LD (K),HL
8012          LD HL,OVER  ;       ;nie wieder merken
8015          LD (SKIP),HL
8018 OVER:    LD HL,#A000         ;hier Start des Textes eintragen
801B          LD (START),HL
801E CONT:    DI
801F          LD HL,GETKEY        ;KBDIO auf unsere Routine lenken
8022          LD (#FD52),HL
8025          EI
8026          RET
8027 START:   DW #A000    ;       ;Zeiger auf Text
8029 KBD:     DB #C3      ;       ;JP
802A K:       DW #FDF2    ;       ;zum urspr. KBDIO
802C GETKEY:  LD HL,(START)       ;Der Zeichenholer
802F          LD A,(HL)   ;       ;Nächstes Zeichen holen
8030          CP #1A      ;       ;<<< Ende des Textes?
8032          JR Z,END    ;       ;Ja
8034          INC HL      ;       ;Zeiger auf Text erhöhen
8035          LD (START),HL       ;und abspeichern
8038          CP #0D      ;       ;war's RET?
803A          JR Z,CR     ;       ;Ja
803C          CP #0A      ;       ;war's LINE FEED?
803E          JR Z,LF     ;       ;Ja
8040          CP #20      ;       ;Zeichen kleiner als Space?
8042          JR C,GETKEY         ;Ja, dann nächstes Zeichen
8044 EXIT:    AND A       ;       ;ansonsten Z-Flag löschen
8045          LD (#FD7D),A        ;abspeichern
8048          RET         ;       ;und zurück!
8049 LF:      LD A,#1A    ;       ;Wenn LF, dann HOME
804B          JR EXIT     ;       ;und zurück!
804D CR:      LD HL,GETESC        ;Wenn CR, dann KBDIO
8050          LD (#FD52),HL       ;auf ESC warten lassen
8053          JR EXIT     ;       ;und zurück! (CR wird ausgeführt)
8055 GETESC:  CALL KBD    ;       ;Tastatur normal lesen
8058          PUSH AF     ;       ;Zeichen retten
8059          CP #1B      ;       ;Ist's ESC?
805B          JR Z,ABORT  ;       ;Ja
```

B A S I C: Aus ASCII mach Programm

```
805D          CP #0A      ;       ;Ist's LF?
805F          JR Z,LINE   ;       ;ja, nächste Zeile
8061          POP AF      ;       ;ansonsten Zeichen zurückholen
8062          RET         ;       ;und zurückgeben!
8063 LINE:    POP AF
8064          XOR A       ;       ;Zeichen invalidieren
8065          JR CONT     ;       ;und zum Weiterlesen reinitialisieren
8067 ABORT:   POP AF      ;       ;Zeichen zurückholen
8068          XOR A       ;       ;invalidieren
8069 END:     LD HL,(K)   ;       ;Normales KBDIO wiederherstellen
806C          LD (#FD52),HL
806F          LD HL,INIT  ;       ;Entwarnung
8072          LD (SKIP),HL
8075          RST 10      ;       ;Meldung ausgeben
8076          DB #6F,#AB,"End of text",#40
8084          JR EXIT     ;       ;und zurück!
8086          RET
```

Statt der Symbols zwei Anmerkungen für Profis:

Die SKIP-INIT-OVER-Geschichte ist nötig, falls jemand mitten drin das Programm noch mal startet. Dann würde GETESC plötzlich zum Original-KBDIO. Und das kann doch keiner ernsthaft wollen...

Ob DI und EI bei CONT wirklich vonnöten sind, habe ich nicht ausprobiert.

AV

B A S I C: Mal etwas für MTX

JOSEF HEUPER 5461 DATTENBERG TEL. 02644/2802

TOKEN-ADRESSE

```
1 REM TOKEN-Adressen, Start mit RUN
2 REM TOKEN eingeben + RET + Leertaste
3 GOTO 5
4 CODE
4059 INTON:  LD HL,#4087;Interrupt-Adr.KEYS
405C         LD (#FA99),HL;zu User-Interrupt
405F         LD A,#C3;CODE fuer JUMP NN
4061         LD (#FA98),A;zu USER-Interrupt
4064         LD A,(#FD5E);Interrupt-Flag
4067         OR #9F
4069         LD (#FD5E),A;Bit 4 u. 7 setzen
406C         RET
406D INTOFF: LD A,(#FD5E);Interrupt-Flag
4070         AND #0F
4072         LD (#FD5E),A;Bit 4 u. 7 ruecksetzen
4075         RET
4076 TOKEN:  LD BC,(#FD82);COMAND
407A         CALL #1B50;Inhalt v. BC zum Schirm
407D         LD DE,64000;Verzoegerungsschleife
4080 LOOP:   DEC DE
4081         LD A,D
4082         OR E
4083         JP NZ,LOOP
4086         RET
4087 KEYS:   LD A,(#FD7C);LAST KEY pressed
408A         CP #49;Leertaste
408C         JR Z,TOKEN
408E         CP #45;Taste F6
4090         JR Z,INTOFF
4092         RET
Symbols:
INTON4059INTOFF406D
TOKEN4076KEYS4087
LOOP4080

5 CODE
428B         LD HL,#4059
428E         LD DE,#BE00
4291         LD BC,#003A
4294         LDIR
4296         RET
Symbols:


6 LET 0=USR(48640)
7 REM Von Fehlermeldungen nach RET nicht stoeren lassen.
8 REM Einige Adr. wie LLIST,LPRINT usw.lassen sich so nicht pruefen.
```

B A S I C: Mal etwas für MTX

JOSEF HEUPER 5461 DATTENBERG TEL. 02644/2802

AUSDRUCK VON NODDY-SEITEN

```
5 REM  ERKLAERUNG DES PROGRAMMS AB ZEILE 400
10 DIM A$(12000): LET K=1
20 INPUT "39 OD.78 ZEICHEN PRO ZEILE DRUCKEN ? ";B$
30 IF B$="39" THEN  LET B=39 ELSE  LET B=78
40 INPUT "WIEVIEL ZEILEN WOLLEN SIE DRUCKEN ? ";Z$
42 LET Z=VAL(Z$)
50 CLS
100 GOTO 230
110 CSR 0,0: REM OBEN LINKS POSITIONIEREN
120 FOR I=0 TO 23: REM 24 ZEILEN
130 FOR J=1 TO 39: REM 39 SPALTEN
140 LET A$(K)=SPK$: LET K=K+1
150 NEXT : NEXT
160 RETURN
190 REM DRUCKEN
200 FOR I=0 TO Z
210 LPRINT A$(I*B+1,B)
220 NEXT
225 STOP
230 PLOD "1"
235 GOSUB 110
240 PLOD "2"
245 GOSUB 110
250 PLOD "3"
255 GOSUB 110
260 PLOD "4"
265 GOSUB 110
270 PLOD "5"
275 GOSUB 110
390 GOTO 200
400 REM  A$ IN ZEILE 10 GGF. AENDERN.
410 REM  DIE NODDY SEITEN AB ZEILE 230 SOLLEN ENTHALTEN:
420 REM  *D NAME.  PUNKT NICHT VERGESSEN!
430 REM  *R
440 REM  NAME = NAME DER JEWEILIGEN NODDY-SEITE.
450 REM  IM BEISPIEL WERDEN 5 NODDY-SEITEN
460 REM  ZUERST GELESEN UND DANN GEDRUCKT.
```

B A S I C: Mal etwas für MTX

JOSEF HEUPER 5461 DATTENBERG TEL. 02644/2802

AUSDRUCK VON DSI-SEITEN

```
10 DIM A$(2000): LET K=1
20 INPUT "39 OD.78 ZEICHEN PRO ZEILE DRUCKEN ? ";B$
30 IF B$="39" THEN  LET B=39 ELSE  LET B=78
35 IF B$="39" THEN  LET R=17 ELSE  LET R=8
40 INPUT "WIEVIEL ZEILEN WOLLEN SIE DRUCKEN ? ";Z$
42 LET Z=VAL(Z$)
50 CLS
100 DSI
110 CSR 0,0: REM OBEN LINKS POSITIONIEREN
120 FOR I=0 TO 23: REM 24 ZEILEN
130 FOR J=1 TO 39: REM 39 SPALTEN
140 LET A$(K)=SPK$: LET K=K+1
150 NEXT : NEXT
160 GOTO 50
170 STOP
190 REM DRUCKEN
200 FOR I=0 TO Z
210 LPRINT A$(I*B+1,B)
215 NEXT
216 STOP
220 REM ANZEIGE
221 LET P=0
222 FOR I=(P) TO (P+R)
223 PRINT A$(I*B+1,B)
224 NEXT
225 LET P=P+R+1
226 INPUT "W FUER WEITER  ";W$: IF W$="W" THEN  GOTO 222
```

Vor dem Start mit RUN ggf. A$ in Zeile 10 anpassen.
Nach der Eingabe von B$ und Z$ ist der Bildschirm leer.
Mit CTRL und ↑ den Kursor einschalten und bis 24 Zeilen schreiben.
Wenn weniger als 24 Zeilen geschrieben werden, wird der Rest als
Leerzeile gespeichert und auch gedruckt=Luecken beim Druck!!
Seite mit "RET" abschliessen; Inhalt wird bei A$ gespeichert.
Kursor ist wieder oben, und man kann 24 Zeilen zufuegen usw. usw.
Wenn genug geschrieben ist, mit "RET"+"BRK" abbrechen.
Nun kann man mit GOTO 220 das Geschriebene ansehen aber nicht aendern
Mit GOTO 200 wird alles gedruckt, je nach Eingabe bei B$ 39 oder
78 Zeichen pro Zeile. Der Ausdruck kann beliebig oft erfolgen.
Beim SAVE'n ist A$ mit auf dem Band und kann nach dem Laden
wieder mit GOTO 220 angesehen oder mit GOTO 200 gedruckt werden.

B A S I C: Mal etwas für MTX

JOSEF HEUPER 5461 DATTENBERG TEL. 02644/2802
TOOL

1) Nach dem Laden - Start mit RUN
   Programm wird nach #BE00 bis #BFAC geladen.

2) LET 0=USR(48992) bzw. RAND USR(48992)
   USER-Interrupt wird eingeschaltet.

3) LET 0=USR(48640) bzw. RAND USR(48640)
   Die ersten 8 Bytes eines BASIC-Programms und die Systemvariablen
   werden in den Bereich #BE88 bis #BEC9 kopiert.
   Bei eingeschalteten Interrupts hat ein Druck auf die Leertaste
   die gleiche Wirkung wie USR(48640)

4) LET 0=USR(48707) bzw. RAND USR(48707)
   Die nach #BE88 bis #BEC9 kopierten BASIC-Bytes und die System-
   variablen werden an ihren alten Platz gebracht.
   Wenn zwischen 3) und 4) NEW oder RESET eingegeben wurde,
   kann jetzt wieder gelistet werden.
   Bei eingeschalteten Interrupts hat ein Druck auf die Taste F5
   die gleiche Wirkung wie USR(48707)

5) MERGE
   BASIC-Progr. werden ohne Ruecksicht auf Zeilen-Nr. vereinigt!!

   "Oberes"Programm laden und CLEAR als Direktbefehl eingeben.
   LET 0=USR(48800) bzw. RAND USR(48800)
   Die ersten 8 Bytes des Programms und die Systemvariablen
   werden nach #BE88 bis #BEC9 kopiert und eine Kopie des Programms
   wird hinter #C400 gespeichert.
   Jetzt "Unteres"Programm laden.
   LET 0=USR(48820) bzw. RAND USR(48820)
   Falls NODDY-Seiten vorhanden sind, werden diese hinter das
   obere Programm gesetzt. Dann wird das komplette obere Pro-
   gramm an das Ende des unteren Programms gesetzt und die
   Systemvariablen addiert.

6) RENUMBER
   Es werden nur Zeilennummern umnumeriert. GOTO's, GOSUB's usw.
   werden nicht geaendert.

   LET R=USR(49021) bzw. RAND USR(49021)
   Neunummerierung ab Zeilennummer 100 mit Schrittweite 10.
   Bei eingeschalteten Interrupts hat ein Druck auf die Taste F7
   die gleiche Wirkung wie USR(49021)
   LET R=USR(49029) bzw. RAND USR(49029)
   Neunumerierung ab Zeilennummer 9000 mit Schrittweite 10.
   Bei eingeschalteten Interrupts hat ein Druck auf die Taste F8
   die gleiche Wirkung wie USR(49029)

   Mit PANEL kann man in #BF8C eine andere Schrittweite und in
   #BF7E/#BF7F bzw. #BF86/#BF87 eine andere Startzeile eingeben.

7) LET 0=USR(49012) bzw. RAND USR(49012)
   USER-Interrupt wird ausgeschaltet.
   Ein Druck auf die Taste F6 hat die gleiche Wirkung wie USR(49012)

B A S I C: Mal etwas für MTX

TOOL — LISTING

```
5 GOTO 12
6 REM ********* BYTE PACK OCT 1984 **********
10 CODE
403B SAVEB:  LD HL,(#4000);1.+2. STELLE VON BASIC
403E         LD (#BE88),HL
4041         LD HL,(#4002);3.+4. STELLE VON BASIC
4044         LD (#BE8A),HL
4047         LD HL,(#4004);5.+6. STELLE VON BASIC
404A         LD (#BE8C),HL
404D         LD HL,(#4006);7.+8. STELLE VON BASIC
4050         LD (#BE8E),HL
4053 SAVESV: LD HL,(#FAA4);OBERGRENZE VON NODDY AUF AKT. SEITE
4056         LD (#BE90),HL
4059         LD HL,(#FAA7);OBERGRFNZE VON BASIC AUF AKT. SEITE
405C         LD (#BE92),HL
405F         LD HL,(#FAAA);ANFANG VON BASIC
4062         LD (#BE94),HL
4065         LD HL,(#FAAC);BASIC-OBERGRENZE ALLER RAM-SEITEN
4068         LD (#BE96),HL
406B         LD HL,(#FACC);OBERGRENZE DER ARRAYS
406E         LD (#BE98),HL
4071         LD HL,(#FACF);NR DER GERADE AUSGEF. BASIC-ZFILE
4074         LD (#BE9A),HL
4077         LD HL,(#FAD6);OBERGRENZE DER AKT. BASIC-SEITE
407A         LD (#BE9C),HL
407D         RET
407E RESTB:  LD HL,(#BE88)
4081         LD (#4000),HL
4084         LD HL,(#BE8A)
4087         LD (#4002),HL
408A         LD HL,(#BE8C)
408D         LD (#4004),HL
4090         LD HL,(#BE8E)
4093         LD (#4006),HL
4096 RESTSV: LD HL,(#BE90)
4099         LD (#FAA4),HL
409C         LD HL,(#BE92)
409F         LD (#FAA7),HL
40A2         LD HL,(#BE94)
40A5         LD (#FAAA),HL
40A8         LD HL,(#BE96)
40AB         LD (#FAAC),HL
40AE         LD HL,(#BE98)
40B1         LD (#FACC),HL
40B4         LD HL,(#BE9A)
40B7         LD (#FACF),HL
40BA         LD HL,(#BE9C)
40BD         LD (#FAD6),HL
40C0         RET
40C1         NOP;DATENPUFFER  (BE86)
40C2         NOP;             (BE87)
40C3         NOP;             (BE88)
40C4         NOP;             (BE89)
40C5         NOP
40C6         NOP
40C7         RET
```

B A S I C: Mal etwas für MTX

```
40C8          NOP
40C9          NOP
40CA          NOP
40CB          NOP
40CC          NOP
40CD          NOP
40CE          NOP
40CF          NOP
40D0          NOP
40D1          NOP
40D2          NOP
40D3          NOP
40D4          NOP
40D5          NOP
40D6          NOP
40D7          NOP
40D8          NOP
40D9          NOP
40DA          NOP;DATENPUFFER  (BE9F)
40DB SUB:     CALL #BE00;8 STELLEN BASIC + SYST.VAR. ZU BF88....
40DE          LD HL,(#FAAA);BASIC-ANFANG
40E1          LD DE,#C400
40E4          LD BC,(#FACC);OBERGRENZE DER ARRAYS
40E8          LDIR
40EA          LD (#BE86),DE;DE ZUM PUFFER
40EE          RET
40EF MAIN:    LD HL,(#FAA4);OBERGRENZE VON NODDY AUF DER AKT. SEITE
40F2          LD DE,(#FAA7);OBERGRENZE VON BASIC AUF DER AKT. SEITE
40F6          AND A
40F7          SBC HL,DE
40F9          JR Z,NONOD
40FB          PUSH HL
40FC          POP BC
40FD          LD HL,(#FAA7);OBERGRENZE VON BASIC AUF DER AKT. SEITE
4100          LD DE,(#BE86)
4104          LDIR
4106          LD (#BE86),DE
410A NONOD:   LD HL,(#BE86)
410D          LD BC,#C400
4110          PUSH BC
4111          AND A
4112          SBC HL,BC
4114          PUSH HL
4115          POP BC
4116          POP HL
4117          LD DE,(#FAA7);OBERGRENZE VON BASIC
411B          LDIR
411D MERGE:   LD HL,(#BE90)
4120          LD DE,(#FACC);OBERGRENZE DER ARRAYS
4124          ADD HL,DE
4125          PUSH HL
4126          LD (#FAAC),HL;BASIC-OBERGRENZE ALLER RAM-SEITEN
4129          LD HL,(#FAA7);OBERGRENZE VON BASIC AUF AKT. SFITE
412C          LD DE,(#FAAA);BASIC-ANFANG
4130          AND A
4131          SBC HL,DE
4133          PUSH HL
4134          LD BC,(#BE92)
4138          ADD HL,BC
4139          LD (#FAA7),HL;OBERGRENZE VON BASIC AUF AKT. SEITE
```

B A S I C: Mal etwas für MTX

```
413C          POP HL
413D          LD BC,(#BE9C)
4141          ADD HL,BC
4142          LD (#FAD6),HL;OBERGRENZE DER AKT. SEITE
4145          LD HL,(#FAA4);OBERGRENZE VON NODDY AUF AKT. SEITE
4148          AND A
4149          SBC HL,DE
414B          LD BC,(#BE98)
414F          ADD HL,BC
4150          LD (#FACC),HL;OBERGRENZE DER ARRAYS
4153          POP HL
4154          LD (#FAA4),HL;OBERGRENZE VON NODDY AUF AKT. SEITE
4157          RET
4158          NOP;DATENPUFFER  (BF1D)
4159          NOP;             (BF1E)
415A          NOP;             (BF1F)
415B CHECK:   LD HL,(#FACC);OBERGRENZE DER ARRAYS
415E          LD A,H
415F          OR L
4160          RET Z
4161          LD BC,(#BF1D)
4165          LD B,0
4167          LD HL,(#BF1E)
416A RENUMR:  LD IX,(#FAAA);ANFANG VON BASIC
416E REPEAT:  LD E,(IX+0)
4171          LD D,(IX+1)
4174          LD (IX+2),L
4177          LD (IX+3),H
417A          ADD HL,BC
417B          PUSH HL
417C          ADD IX,DE
417E          PUSH IX
4180          POP DE
4181          LD HL,(#FAAC);BASIC-OBERGRENZE ALLER RAM-SEITEN
4184          AND A
4185          SBC HL,DE
4187          JR C,END
4189          LD A,H
418A          OR L
418B          JR Z,END
418D          POP HL
418E          JR REPEAT
4190 END:     POP HL
4191          RET
4192          NOP;FREIRAUM     (BF57)
4193          NOP
4194          NOP
4195          NOP
4196          NOP
4197          NOP
4198          NOP
4199          NOP
419A          NOP;FREIRAUM     (BF5F)
```

B A S I C: Mal etwas für MTX

```
419B INTON:   LD A,#C3;JUMP NN
419D          LD (#FA98),A;ZU USER-INTERRUPT
41A0          LD HL,#BF93;INTERRUPT-ADRESSE ->KEYS
41A3          LD (#FA99),HL;ZU USER-INTERRUPT
41A6          LD A,(#FD5E);INTERRUPT-FLAG
41A9          OR #9F
41AB          LD (#FD5E),A;BIT 4 UND 7 SETZEN
41AE          RET
41AF INTOFF:  LD A,(#FD5E);INTERRUPT-FLAG
41B2          AND #0F
41B4          LD (#FD5E),A;BIT 4 UND 7 RUEKSETZEN
41B7          RET
41B8 REN1:    LD HL,#64;RENUMBER AB ZEILE 100
41BB          LD (#BF1E),HL;HL IN PUFFER
41BE          JR STEP
41C0 REN9:    LD HL,#2328;RENUMBER AB ZEILE 9000
41C3          LD (#BF1E),HL;HL IN PUFFER
41C6 STEP:    LD A,#0A;ZEILEN-ABSTAND =10
41C8          LD (#BF1D),A;A IN PUFFER
41CB          JP #BF20;CHECK (41B5)
41CE KEYS:    LD A,(#FD7C);LASTKEY
41D1          CP #49;LEERTASTE -> ZU SAVE BASIC
41D3          JP Z,#BE00;SAVEB
41D6          CP #47;TASTE F5 -> ZU RESTORE BASIC
41D8          JP Z,#BE43;RESTB
41DB          CP #45;TASTE F6 = USER-INTRRUPT AUSSCHALTEN
41DD          JR Z,INTOFF
41DF          CP #44;TASTE F7 = RENUMBER STEP 10 AB ZEILE 100
41E1          JR Z,REN1
41E3          CP #42;TASTE F8 = RENUMBER STEP 10 AB ZFILE 9000
41E5          JR Z,REN9
41E7          RET
```

```
12 CODE
4BD1          LD HL,#403B
4BD4          LD DE,#BE00
4BD7          LD BC,#01AD
4BDA          LDIR
4BDC          RET
Symbols:
```

H A R D W A R E: Digitizer / 512kiloByte

Digitizer-Schaltung (Otmar Rücker, 5102)

Den Digitalisierer können Clubmitglieder bei mir für DM 100.- incl. Demodiskette bestellen.
Für Versandkosten und Verpackung rechnie ich DM 5.-
Für insgesamt DM 110.- gibt es das Gerät mit Netzteil (Aufpreis 5.-)
Es gibt zwei Ausführungen: a) für Parallelport
b) für Joysitckport
Beide sind getestet und laufen.
Bei Bestellunen bitte telefonische Absprache, da ich keine Massenproduktion machen kann.
Konto-Nr. 7662034, BLZ 39150100, Kreissparkasse Aachen.

512 kiloByte auf der Hauptplatine (Herbert Herberg, 2000)

Ich habe mir Gedanken über das Design der 512kB auf der Hauptplatine des MTX gemacht. Am 16. Mai war Ulrich Hönisch bei mir zu Besuch, und ich habe ihm meine Gedanken unterbreitet. Es war einverstanden, fing an zu löten, und ich habe das Prom gebrannt (was hinein soll habe ich schon vorher überlegt). Dann haben wir das Prom in den Sockel gesteckt, und den MTX angeschaltet. Und oh, welch Freude: es lief auf Anhieb.
Das klingt einfach, aber ich glaube Ulrich Hönisch sieht das etwas anders. Es hat eine unwahrscheinliche Erfahrung und Routine beim Löten!
Diese Aufrüstung gibt's jetzt für DM 290.-

512 kiloByte auf der 32k-Karte (Herbert Herberg, 2000)

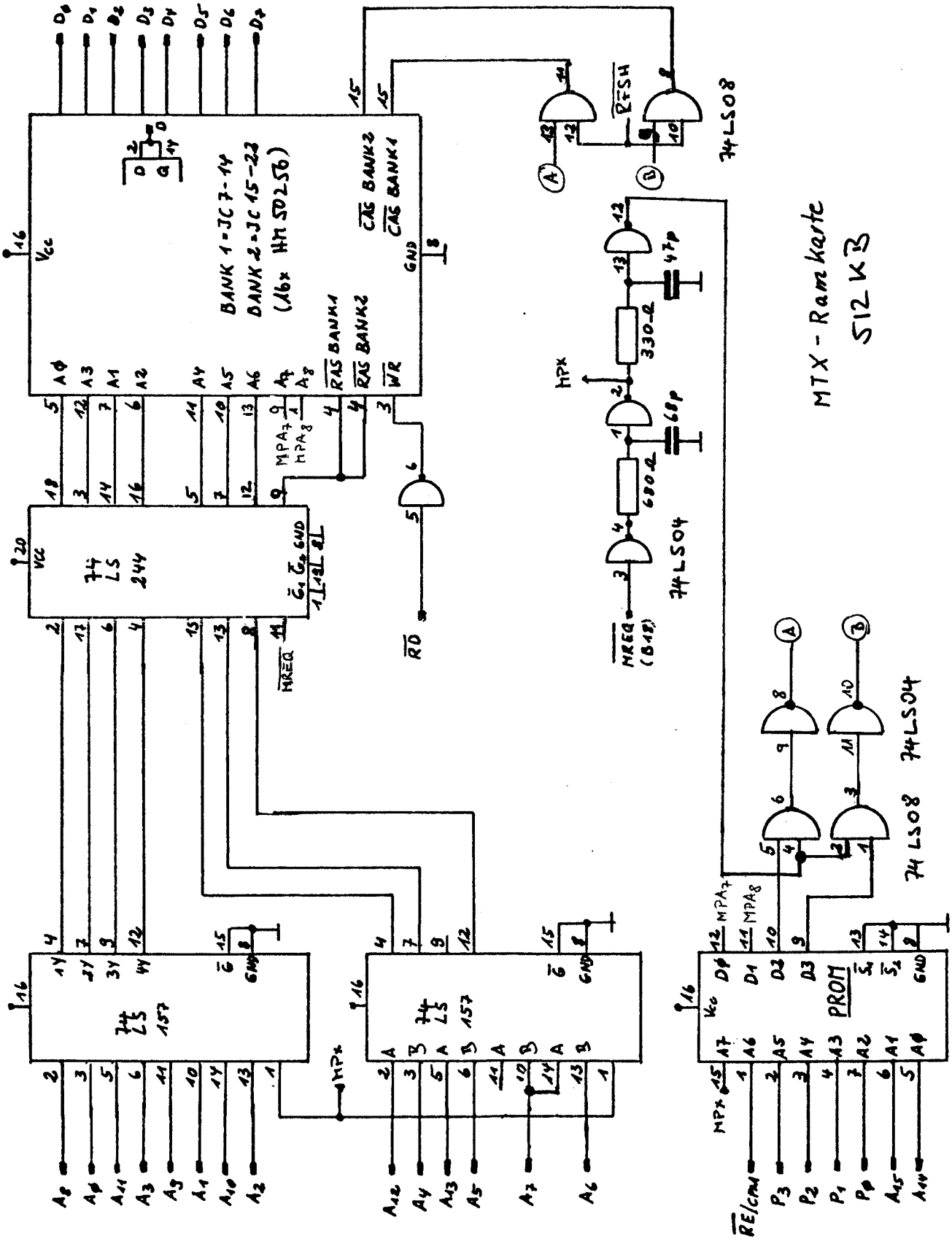
Auf der nächsten Seite findet Ihr den Schaltplan für die 512k-Karte. Wer nun also seine Lötkünste ausnutzen will muß nur die paar Änderungen der 32k-Platine vornehmen, um das Layout an die 512kB anzupassen, das Prom von mir kaufen und löten.
Damit Ihr die Arbeit nicht unterschätzt:
Ulrich Hönisch hat bei seinen Arbeiten um 768k-Karten zu löten festgestellt, daß fast alle (d.h. über 90%) 512k-Karten, die er nicht selbst gelötet hat dermaßen verhunzt sind, daß er einiges an Mühe hat, die Platinen überhaupt zum Laufen zu bringen.
Eine Ähnliches kann ich aus meinen Erfahrung bei den Reparaturen an boot- und laufunfreudigen MTX/FDX-Systemen berichten, wo eine nicht Hönisch-Karte häufig der Grund für die Probleme war. Viele Computer entwickeln erst nach dem Einbau einer schlechten 512k-Karte ihre Allüren.

Ich schreibe das nicht, um Euch etwa zu demoralisieren - aber wir wollen Euch deutlich vor den Problemen warnen, die auf Euch zukommen!
Wer nicht die unten unter 1-4 aufgeführten Dinge hat sollte tunlichst die Finger von den Versuchen des Selberlötens lassen:

1. Feine professionelle Lötspitze
2. Elektronik-Lötdraht 'amasan' (gelbe Rolle)
3. kein Lötfett
4. Erfahrung im Löten
5. mögl. IC-Auslötspitze, da nur dieses Teil 'es bringt'

Nun jedoch das wichtigste: Ulrich Hönisch und ich werden versuchen vermurkste 512k-Karten zum Laufen zu bringen. Da das jedoch in der Regel bedeutet, daß wir u.a. auch die IC-Sockel, die die ausgelöteten IC's ersetzen alle auslöten müssen, um festzustellen, welche Leiterbahnen hops gegangen sind - und auch noch diese nachverdrahten müssen. D.h. das Reparieren einer solchen Karte macht uns mehr Arbeit, als das Aufrüsten neuer Platinen. Daher könnt Ihr Euch sicherlich denken, daß das nicht gerade billig ist: DM 180.-

H A R D W A R E: 512 k-Karte Schaltplan (Herbert Herberg, 2000)



R A M 4 . 2: Patches

RAM 4.2: Fehler!! Fehler!!Fehler!! Fehler!! Fehler!! (BP 2200)

Leider gab es eine sehr dWP (dunkle Woche von Preusing), bei der ich unter dem Zeitdruck des Clubtreffens bzw. der Fertigstellung von RAM 4.2 einige grobe Fehler gemacht habe. Ich möchte mich hierfür in aller Form entschuldigen und Korrekturen aufzeigen.

Diese Fehler betreffen RAM42.COM, P2DOS.MAC, P2DOS.PR, das System und MS.COM. Ich bitte alle inständig, die folgenden Korrekturen durchzuführen, denn die 'gepatchten' Programme bilden ab nun die offizielle RAM 4.2-Version und die Fehler sind nicht zu verachten!!!!!

Alle Korrekturen bitte zunächst nur auf Kopien machen, die Original-Diskette erst nach einem Erfolgstest patchen!!

In **RAM42.COM** ist ein furchtbarer Fehler, eine Blödheit und eine Unschönheit.

1.: Der schlimme Fehler führte dazu, daß die Systeme auf allen Update-Disketten im Format 09, die vor dem 16.4.87 ausgeliefert wurden (DDIR !), fehlerhaft sind und erst nach längerer Zeit zu einem Absturz führen. Dies kam, weil ich die Update-Disketten mit SYSCOPY4 B:=A: unter einem fehlerhaften RAM 4.2 erstellt habe. Wer sich das neue System mit der WRITESYS-Prozedur erstellt hat, hat Glück gehabt und sich nur 2 kleinere Fehler eingehandelt!
   Er besteht darin, daß bei einem SYSCOPY4 von A: der erste P2DOS-Sektor zweimal auf verschiedene Sektoren geschrieben wird, und der zweite überschreibt einen wichtigen Teil des P2DOS. Dies kommt, weil in der Blockread-Routine ein falscher Sprung ist. Korrektur bei #1093: anderes Sprungziel.
2.: Die 'Blödheit' ist die Meine und betrifft nur die Formate 12 und 13 und besteht darin, daß man die Systemspuren (die ja eine andere Sektorlänge haben) nicht mit DU2 beschreiben kann, sondern immer die Fehlermeldung 'falsche Sektorlänge' kommt. Korrektur bei #40D4: Aufruf eines Unterprogrammes zur Längenbestimmung
3.: Dieser Fehler ist zwar momentan noch nicht von Bedeutung, aber hoffentlich in Bälde! Er hängt mit 2. zusammen und tritt dann auf, wenn bei einem Test auf die richtige Sektorlänge eine Interrupt-Routine 'dazwischenfegt' und die Bank umschaltet. Dann werden nämlich evtl. einige Bytes in ebendieser Bank zerstört. Dies kann z.B. beim Kopieren von Programmen zu **sehr schlimmen** 'schleichenden' Fehlern führen. Korrektur bei #40B0: Beim auf Sektorlänge Test Interrupts sperren

Für **SDX-Besitzer** ist nur der erste Fehler bei #1093 zu korrigieren, die anderen treten nicht auf!

Also nehmen wir uns nun DDT.COM (oder auch DDTZ oder MONI oder PATCH, wer es hat) und RAM42.COM (oder wie es gerade heißt) und korrigieren die 3 Fehler: (Benutzereingaben sind fett gedruckt) Bei einigen neueren Versionen ist der erste Fehler bereits korrigiert, ab 10.5.87 ist alles ok, auch MS.COM und das System!

R A M 4 . 2: Patches

```
H>ddt ram42.com
DDT VERS 2.2
NEXT  PC
5900 0100
-s1093
1093 BD e2
1094 AF .
-s40b0
40B0 21 3e
40B1 9F 9f
40B2 40 d3
40B3 7D 46
40B4 D3 3e
40B5 46 40
40B6 7C d3
40B7 D3 47
40B8 47 f3
40B9 3E .
-s40d4
40D4 20 cc
40D5 02 c1
40D6 3E 3d
40D7 01 00
40D8 47 .
-^C
A>save 88 ram42.com
```

Im **P2DOS** ist seit jeher ein Fehler, der die Funktion 40 (write random with zero fill) immer ins 'Nirwana' führt. Im P2DOS.MAC steht in der Zeile (ca.) 2182 ein **LD HL,DIRBUF**, was durch **LD HL,(DIRBUF)** zu ersetzen ist. Daß der Fehler noch nie bemerkt wurde, zeigt, daß diese Funktion zwar völlig überflüssig ist, aber eben doch von mindestens einem Programm benutzt wird.

Ein weiterer Fehler existiert erst seit RAM 4.2 und wurde von mir 'verbrochen' und im RAM42.DOC für gut angepriesen. Dies war allerdings ein Flop, der u.U. beim Arbeiten mit DU2 zu Fehlern führen kann und ansonsten das System sehr viel langsamer macht. Im P2DOS.MAC muß also ca. in Zeile 1079 vor dem Label SELDK2: das NOP wieder durch RET Z ersetzt werden.

Auch hier müssen wir das P2DOS.PR und das System patchen:

```
A>ddt p2dos.pr
DDT VERS 2.2
NEXT  PC
1100 0100
-s678
0678 00 c8
0679 77 .
-sdea
0DEA 21 2a
0DEB 12 .
-^C
A>save 16 p2dos.pr
Erase P2DOS   .PR ?y
```

Nun kann man entweder das System direkt mit DU2 patchen oder einfach mit **WRITESYS 58 B:** (und dem neuen, gepatchten P2DOS.PR) ein neues System generieren, wobei dann aber die vielleicht gatätigten Änderungen des Boot-Sektors verloren gehen. All, die sich das System mit SYSCOPY4 von der Update-Diskette geholt haben, müssen WRITESYS benutzen und können sich das Patchen mit DU2 sparen!

R A M 4 . 2: Patches

Hier eine leicht gekürzte und kommentierte DU2-Sitzung:

```
A>du2
DU2 - Disk Utility II, Version 1.1
Type ? for Help
DU2  A0? lb,t1,s1,d            (Laufwerk B:,Spur 1,Sektor1, anzeigen)
 Track = 1, Sector = 1, Physical Sector = 1
 Track = 1, Sector = 1, Physical Sector = 1
00  46756E63 74696F6E  203D2420 46696C65  *Function =$ File*
50  CB1B10FA 2137E0CB  432802BE 0077D54F  *K..z!7'KC(.>.wUO*
60  CD1BE17C B528355E  235623ED 530AE022  *M.aö5(5^#V#mS.'"*
70  0CE02323 220EE023  232210E0 23231112  *.'##".'##".'##..*

DU2  B0? ch5c c8,d          (ändere 5C auf C8 und zeige Sektor)
00  46756E63 74696F6E  203D2420 46696C65  *Function =$ File*
50  CB1B10FA 2137E0CB  432802BE C877D54F  *K..z!7'KC(.>HwUO*
60  CD1BE17C B528355E  235623ED 530AE022  *M.aö5(5^#V#mS.'"*
70  0CE02323 220EE023  232210E0 23231112  *.'##".'##".'##..*

DU2  B0? w,s14,d     (schreibe Sektor, lies Sektor 14 und zeige)
 Track = 1, Sector = 14, Physical Sector = 14
00  A5D8CD9C D93A2FE0  FE14C0DD 3420C9CD  *¥XM.Y:/'..§Ü4 IM*
70  0E023A2F E0D62820  33D52112 E0068077  *..:/'V( 3U!.'..w*

DU2  B0? ch7a 2a,d          (ändere 7A auf 2A und zeige Sektor)
00  A5D8CD9C D93A2FE0  FE14C0DD 3420C9CD  *¥XM.Y:/'..§Ü4 IM*
70  0E023A2F E0D62820  33D52A12 E0068077  *..:/'V( 3U*.'..w*

DU2  B0? w             (schreibe Sektor)
DU2  B0? ^C            (Warmboot, DU2-Ende, System ok)
```

In **MS.COM** wird die Länge eines MSDOS-Files falsch berechnet und angezeigt, wenn es größer als 64K ist. Infolgedessen wird falsch kopiert. Dieser Fehler läßt sich gottlob leicht beheben:

```
A>ddt ms.com
DDT VERS 2.2
NEXT  PC
6080 0100
-s3316
3316 07 09                      (war falsches SHL-Argument)
3317 00 .
-^C
A>save 96 ms.com
Erase MS       .COM?y
```

T U R B O, d B A S E: Patches

Patches für Overlay-Laufwerke

(Uwe Grass, 3300)

Für alle, die die Vorteile des neuen Betriebssystems nutzen, stellt sich immer wieder die Frage, wie ein Programm, das über den Path aufgerufen wurde, seine Overlays findet. Im Newword ist es noch einfach, dort werden die Overlays in dem Laufwerk gesucht, das als erstes "LEGAL DRIVE" eingegeben wurde. Anders sieht es in Turbo Pascal oder dBase aus.

1. Turbo Pascal
   Im TURBO.COM steht einmal "TURBO Pascal system". Dieser Text muß durch "**H:**TURBO Pascal syst" ersetzt werden, dann sucht TURBO seine Overlays auf Laufwerk **H:**. Selbstverständlich muß hier der Laufwerksbuchstabe eingegeben werden, wo üblicherweise die Overlays abgelegt werden. Im Turbo Pascal, Version 3.00a ist die Adresse dieser Meldung #217D. Da ich selbst noch kein Turbo Pascal besitze, konnte ich dies nur in einer Version ausprobieren, dort hat es aber einwandfrei funktioniert.

2. dBASE II
   Hier besitze ich die Version 2.41. Dort ist an der Stelle #165 ein Laufwerkseintrag vorgesehen. Hier kann man die Laufwerkskennnr. eintragen. Das heißt, hier wird **#08** eingepatcht, wenn in Laufwerk **H:** das Overlay ist.

   Außerdem kann dBASE das Funktionstastenmodul nicht umschalten. Hier kann man sich damit behelfen, daß an den Stellen, an denen Systemmeldungen ausgegeben werden, die Sequenz zum Umschalten der Tabelle eingetragen wird. Beispiel: nach dem Start fragt dBASE das Datum ab. Hier kann statt des Textes der String "1B 5B 47 34 00" stehen. Der Datumstext wird dadurch allerdings gekürzt. Man kann sich aber zusätzlich Platz schaffen, da dBASE Texte immer bis zum ersten "00" ausgibt. Das zurückstellen der Tabelle kann mittels der "Abschiedsmeldung" geschehen. Die Meldungen findet man im dBASEOVR.COM.

Anm.d.HH.: Uwe, ich finde, dieser Artikel ist einfach fantastisch! Ich glaube auch, daß Du hier mal für die sog. 'Experten' eine Überraschung in Petto hast.

N e w W o r d: Patches

PATCHES für NewWord-Hilfstexte (Uwe Grass, 3300)
(Version 2.16 oder später)

Newword gibt dem Benutzer recht großzügig Hilfstexte aus, die die Control-Commands erklären. Dem geübteren Benutzer ist dies aber oft etwas lästig, da der Text dabei verschoben wird (z.B. Hyphen Help). Nun kann man aber sehr einfach die Nw-Messages verändern, damit so etwas nicht mehr passiert. Dazu kurz eine Erläuterung, wie diese neuen Messages aufgebaut sind.
Die Menüs und Texte sind auf den ersten Blick etwas konfus. Um das Overlay kurz zu halten, ist zu Beginn eine Tabelle mit Ausdrücken, oder Teilen davon, die öfter auftauchen. Wenn nun in einer Meldung so ein Ausdruck benötigt wird, springt das Programm in diese Tabelle und holt sich den String. Die Tabelle beginnt mit mehreren Leerfeldern an #49D und endet bei #6FB. Der Ausdruck "Newword" wird mit dem Aufruf #8D angesprungen, von dort aus wird durchgezählt.
Wenn in den Texten #80 auftaucht, schaltet der Rechner auf hellleuchten um, bei #81 wird wieder dunkel geschaltet. Eine größere Anzahl von Leerfeldern kann dadurch erzeugt werden, daß #06 und die Anzahl der Leerfelder eingegeben wird. Beispiel: 10 Leerfelder werden durch #06 #0A angegeben.
Die Anzahl der Zeilen, die eingeblendet werden sollen, wird dadurch gezählt, daß zu Beginn der Sequenz die Anzahl hex eingegeben wird. Beispiel: Adr.:#1476, dort beginnt der Text für Hyphen Help. Wenn man hier eine #01 eingibt, wird nur noch die Überschrift in den Text eingeblendet (also eine Zeile, statt sonst sechs Zeilen). Die Zeilenzahl wird dadurch ermittelt, daß die #0D gezählt werden.
Es folgt eine Tabelle, die alle Änderungen zeigt, die ich in meinem Overlay NWMSGS.OVR gemacht habe:

Hex.-Adr.	Neuer Inhalt	Bedeutung
07CF	00	Langer Text bei Eingabe von D im Opening Menue entfällt
0A9A	00	s.o. Eingabe N
0F15 und	02	s.o. Eingabe M
0F17 bis 0F9A	alle 00	
0B6E und	02	s.o. Eingabe E
0B70 bis 0B8A	alle 00	
0D18 und	02	s.o. Eingabe P
0D19 und	0D	
0D1A bis 0D99	alle 00	
114A	00	s.o. Eingabe Y
1496	01	Text "Hyphen help..." nur eine Zeile
160D	00	Text "While entering..." entfällt

Im alten Newword ist das Overlay leicht lesbar. Die Zeilenzählung funktioniert aber genauso. Der Text Hyphen Help steht dort ab #172C (dort also 01 eintragen). Die Texte für Doc bzw. Non-Doc stehen ab #31A bzw #617. Dort kann eine 0 eingetragen werden.

UG

Anm.d.HH.: Ich habe diese Patches mal gemacht, und mußte feststellen, daß das NWMSGS.OVR zu NewWord 2.16/2.17 ziehmlich unlesbar ist. Dadurch, daß viele häufig vorkommende Worte/Texte in einer Tabelle (s.o.) abgespeichert sind. Das spart Platz, macht aber das .OVR unlesbarer, und die Ausgabe der Texte langsamer. Im neuen NewWord müßt (und könnt) Ihr Uwe glauben. Im alten hat man anscheinend auf diese Platzsparende und Zeitkostende Tabelle verzichtet, muß also daher nicht bei einem #8D wissen, daß das NewWord bedeutet.

A S S E M B L E R: Kurs Teil 2

ASSEMBLERKURS

1.2 Warum heute noch Assembler? (Volker Griener, 8581)

Berechtigte Frage, werden einige von euch sagen, wo es doch so schöne "höhere" Programmiersprachen gibt! Und der Aufwand, Assembler zu lernen! Aber wo viel Schatten, da ist viel Licht (oder umgekehrt?). Assembler hat natürlich schon seine Vorteile, denn sonst bräuchte es diesen Kurs ja nicht.

1.2.1 Vorteile

1.2.1.1 Die Rechengeschwindigkeit

BASIC mag für manche Anwendungsgebiete ausreichend schnell sein, aber wer hat sich noch nicht über lange Wartezeiten in Programmen geärgert? Hier schafft die Assemblersprache Abhilfe. Maschinensprache ist für die CPU die eigentliche primäre Programmiersprache. (Anm.: Assembler ist - nach entsprechender Assemblierung - für die CPU das gleiche wie Maschinensprache; für den Benutzer nur einfacher, da jedem Befehl ein "verständlicher" Name zugeteilt wird anstelle der Zahlenkolonnen aus 0 und 1.)

Ein BASIC Befehl wiederum kann sich ja nur aus vielen einzelnen Maschinenbefehlen zusammensetzen. Es leuchtet schon ein, daß Assembler ja viel schneller sein müßte. Stimmt! Ist in BASIC die Rechengeschwindigkeit bei einigen tausend Befehlen am Ende alles Machbaren, so kann die CPU in Assembler bis zu 1 Million Befehle pro Sekunde verarbeiten. Dieser Wert ist allerdings theoretisch, in der Praxis sind es nur einige 100000 pro Sekunde, da nicht jeder Befehl die gleiche Ausführungsdauer besitzt. Man kann also durchaus sagen, daß Assembler einige 100 mal schneller ist als BASIC.

1.2.1.2 Der Speicherbedarf

Viele Probleme lassen sich in Assembler mit erheblich weniger Speicheraufwand programmieren. Ist man in BASIC gezwungen, auf Fließkommavariablen (also Zahlen der Art $a*10^b$) zurückzugreifen, welche 6 Speicherzellen belegen, so kommt man in Assembler häufig mit Zahlen ohne Nachkommastellen (Ganzzahlvariable) aus, welche nur eine oder zwei Speicherzellen, je nach ihrer Länge (und benötigtem Wertebereich) belegen. Dies ist meistens ausreichend, denn um Adressen von Speicherzellen anzugeben, reichen solche "kurzen" Zahlen völlig aus. Auch fällt in Assembler die Speicherung von Variablennamen völlig weg. Man gibt einfach nur die Speicheradresse an. Dies kann in Assembler (im Gegensatz zur reinen Maschinensprache) allerdings auch durch symbolische Namen geschehen. Auch sind in Assembler einige sehr sinnvolle Befehle vorhanden, die man in BASIC erst umständlich programmieren müßte, z. B. "kopiere n Speicherzellen, beginnend mit Adresse A nach Adresse B". In Assembler ein Problem, was mit insgesamt vier Befehlen gelöst werden kann, während in BASIC schätzungsweise über 50 nötig wären.

1.2.2 Nachteile

1.2.2.1 Maschinenabhängigkeit

Assembler hat auch große Nachteile. Einer davon ist die Maschinenabhängigkeit. Leider konnten sich die Computerhersteller nicht auf eine einheitliche Assemblersprache für ihre verschiedenen CPU's einigen. Da jeder sein eigenes Süppchen kocht, haben viele unterschiedliche Compu-

ter auch verschiedene CPU's und dementsprechend verschiedene Assemblersprachen. Als gängige CPU's haben sich dabei im Homecomputerbereich der Z80 von Zilog (im MTX) und der 6502 von Motorola (in solchen Computern, die es für'n Apple und 'n Ei gibt) erwiesen. Es gibt aber noch einige dutzend anderer Typen. Und jede dieser CPU's hat einen anderen Befehlssatz! Die Menge, Art und Bezeichnung der Befehle kann jeweils grundverschieden sein. Für den Assembler-Programmierer hat das den Nachteil, daß er für jeden Computertyp eine eigene Programmiersprache lernen muß. Programme lassen sich daher auch nicht zwischen verschiedenen Systemen austauschen. Bei Maschinenprogrammen ist das einsichtig, es gilt aber leider auch für den reinen Text eines Assemblerprogramms (den sog. Quelltext), da dieser von dem Übersetzungsprogramm auf einem anderen Computer (mit verschiedener CPU) nicht "verstanden" wird. Im Vergleich zu BASIC, wo viele Programme, meist nur mit wenigen Änderungen, ausgetauscht werden können, ein großer Nachteil. Man kann in Assembler also nur für einen bestimmten Computertyp programmieren. Versuche, das Problem zu verringern (wie CP/M, MSX, ...) möchte ich hier nicht näher erläutern.

1.2.2.2 Programmierfehler

Ein weiterer Nachteil ist die drohende Gefahr eines Systemabsturzes. In einem "normalen" BASIC Programm (d. h. ohne ASSEM und POKE) ist es normalerweise nicht möglich, den Computer "aussteigen" oder sich "aufhängen" zu lassen. Gemeint ist damit jener gefürchtete Zustand, wo "nichts mehr geht". Das Ende vom Lied ist dann der Ausschalter oder die Reset-Tasten. Ein Daten- und Programmverlust ist meistens die Folge. In BASIC erscheint bei einem Fehler im Programm eine Fehlermeldung oder man hat die Möglichkeit, zur BREAK-Taste zu greifen. In Assembler gibt es eine BREAK-Funktion normalerweise nicht. (Anm.: Unser MTX-Assembler hat eine BREAK-Funktion, die einem vielleicht weiterhilft.) Sollte das Maschinenprogramm sich "aufhängen", so ist eine Unterbrechung meist nicht mehr möglich. Gelingt unter günstigen Bedingungen doch eine Rückführung in den Dialog-Betrieb, so ist bei einem kleinen Fehler durchaus damit zu rechnen, daß der Computer seinen Speicher so durcheinandergebracht hat, daß das Programm oder die Daten mehr oder weniger vollständig zerstört sind. Deshalb: Bei Assembler-Programmem immer erst abspeichern, dann RUN!

1.2.2.3 Primitivität

Ein gescheiteres Wort ist mir leider nicht eingefallen. Ich will damit folgendes sagen: Assemblerbefehle sind sehr primitiv in dem Sinne, daß sie nur einen kleinen Beitrag zur Lösung eines Problems leisten. Hierzu ein kleines Beispiel in BASIC. Die Anweisung

```
LET A=B*C+10
```

ließe sich in einem (hypothetischen) Assembler etwas so realisieren: Wir vereinbaren zunächst (willkürlich), daß die Variable A, B und C in den Speicherzellen 0, 1 und 2 abzulegen sind. Das Assemblerprogramm könnte dann so aussehen (Kommentare dazu in *kursiv*):

```
LOAD    (1)    hole B in die CPU
MUL     (2)    multipliziere mit C
ADD     10     addiere dazu 10
STO     (0)    speichere Ergebnis in A
```

Dabei haben wir vereinfachend angenommen, daß die CPU einen Maschinenbefehl zur Multiplikation hat (beim Z80 nicht der Fall) und daß A und B ganzzahlig sind. Müßten wir hier mit Fließkommazahlen rechnen, wie BASIC es immer tut, käme allein dadurch schon ein anständiges Programm zustande.

Daher werden Assemblerprogramme auch bei einer einfachen Aufgabe in

der Regel doch recht lang und dadurch unübersichtlich. Höhere Programmiersprachen stellen dagegen mächtige, vom jeweiligen Computer weitgehend unabhängige Befehle zur Verfügung und bieten dem Programmierer dadurch eine "geraffte" Sicht des Problems. (Wir brauchen uns in BASIC beispielsweise nicht darum zu kümmern, wo die Variablen abgespeichert werden.) Es kann allerdings, wie oben erwähnt, in Ausnahmefällen auch mal den gegenteiligen Effekt geben.

1.3 Die I2DITVM - ein etwas anderer Computer (Kurt-B. Rohloff, 8000)

Um euch nun nicht gleich ins kalte Wasser zu stürzen, haben wir weder Kosten noch Mühen gescheut und extra für diesen Kurs einen Lehrcomputer entwickelt. Mit diesem werden wir einige der grundlegenden Konzepte erarbeiten. Er hat für euch einige Vorteile:

+ sehr einfache Architektur
+ er arbeitet im Dezimalsystem
+ er fängt Programmierfehler weitgehend ab
+ man hat ihn ständig und völlig im Blick
+ er hat alle vier Grundrechenarten "drauf"
+ er spricht deutsch mit euch

Es sollen allerdings auch die Nachteile nicht verschwiegen werden:

- er kann nur in Maschinensprache programmiert werden
- Programme können nicht gespeichert werden

Um euch nicht mit umständlichen Löt- und Bastelarbeiten zu belästigen, werden wir unseren MTX Rechner benutzen, um die I2DITVM zu emulieren (= nachbilden). Das entsprechende BASIC Programm ist leider zu lang, um hier auch noch abgedruckt zu werden. Ihr könnt es gegen Einsendung eines Datenträgers inklusive selbstadressiertem, frankiertem Rückumschlag beziehen bei

Disketten (Typ 03): Kurt-Bernd Rohloff, Kafkastr. 14, 8000 München 83
Kasetten: Kai-Uwe Pleban, Dr.-Heinrich-Köhler-Str. 8, 6968 Walldürn

Dieser exotische Rechner arbeitet wie gesagt im Dezimalsystem. Jede Zelle seines Speichers faßt zwei Dezimalziffern. Ebenso werden auch alle Berechnungen mit zwei Ziffern durchgeführt. Da er interaktiv bedient wird, habe ich ihn Interactive 2 Decimal Digit Virtual Machine genannt.

1.3.1 Architektur der I2DITVM

Der Computer ist sehr einfach aufgebaut. Seine CPU enthält ein Leitwerk, das die Befehle interpretiert und die entsprechenden Aktionen auslöst. Die englische Bezeichnung dafür ist control unit, abgekürzt CU. Außerdem enthält die CPU ein Rechenwerk, in dem die arithmetischen Operationen durchgeführt werden (welche das sein soll, wird ihm vom Leitwerk "gesagt"). Es wird nach der englischen Bezeichnung Arithmetic and Logic Unit meist kurz ALU genannt. Seine Resultate werden in einem Register mit der Bezeichnung Akkumulator (kurz: Akku) abgelegt. (Als Register bezeichnet man besonders schnelle und leistungsfähige Speicherzellen, die sich innerhalb der CPU befinden, im Gegensatz zu den normalen Speicherzellen des Hauptspeichers.) Dieser Akkumulator ist auf die ALU zurückgekoppelt. Dadurch steht sein Inhalt als erster Operand der nächsten Berechnung automatisch zur Verfügung (s. Bild 1.3.1-1). Dies erklärt, daß z. B. bei einer Addition nur noch die zweite Zahl angegeben werden muß, wie bereits in 1.1.2 erwähnt.

A S S E M B L E R: Kurs Teil 2

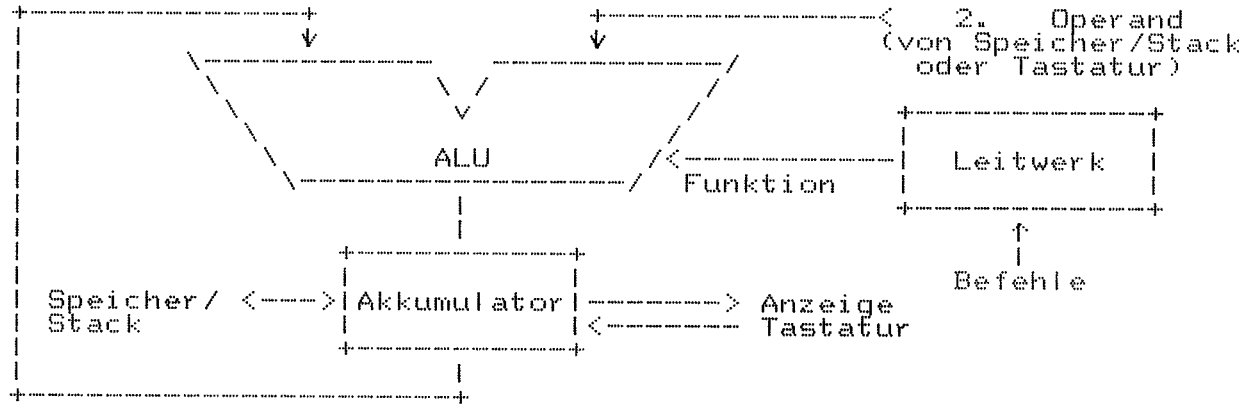


Bild 1.3.1-1 Die CPU der I2DITVM

Wir wollen das gleich mal ausprobieren. Der Befehl 40xy dient dazu, die Zahl xy (immer zwei Ziffern!) zum gegenwärtigen Inhalt des Akkumulators zu addieren. Ebenso kannst du mit 50xy subtrahieren. Die ersten beiden Ziffern stellen den sog. Operationscode dar, durch die die Art des Befehls festgelegt wird. Danach kann noch eine oder zwei Ziffer(n) als Operand folgen. Dies ist von Befehl zu Befehl verschieden. Lasse das Emulator-Programm laufen und gib dann das folgende Programm ein.

Hinweis zur Programmeingabe: Die Ziffern werden hintereinander ohne RETurn eingegeben. Daß wir immer nur einen Befehl auf einer Zeile abdrucken, dient nur der Übersichtlichkeit. Wenn nach einer Ziffer noch eine zweite erwartet wird, fragt das Programm mit "und?" nach dieser zweiten Ziffer. Da der Operationscode immer zweistellig ist, erfolgt diese Frage immer nach der ersten Ziffer eines Befehls. Meine Kommentare zum Programm sind in *kursiv* gedruckt.

Programm 1.3.1-1

```
4003      addiere 3
4012      addiere 12, ergibt 15
5008      subtrahiere 8, ergibt 7
4071      addiere 71, ergibt 78
991       Programm beenden
```

Der Hauptspeicher der I2DITVM besteht aus 10 (Dezimalcomputer!) Zellen, die jeweils 2 Ziffern speichern können. Außerdem besitzt die I2DITVM noch einen Stack von ebenfalls 10 Zellen, der genau genommen mit zum Hauptspeicher gehört. Wir werden ihn erst weiter unten besprechen. Damit ist die Architektur dieser "Maschine" schon beschrieben.

1.3.2 Programmierung der I2DITVM

Wir wollen nun die Maschinensprache dieses Computers kennenlernen. Außerdem werden wir eine Assemblersprache dazu angeben. In Ermangelung eines geeigneten Assemblers (das Umsetzprogramm meine ich jetzt) wirst du diese Arbeit allerdings selbst übernehmen müssen. Wir werden zuerst anhand einiger Beispiele ein paar Befehle kennenlernen, um danach den gesamten Befehlssatz anzugeben.

Die Befehle zur Addition und Subtraktion kennen wir schon. Wir wollen den Additionsbefehl 40 symbolisch mit ADD bezeichnen, den Subtraktionsbefehl 50 mit SUB. Allgemein schreiben wir dann statt 40xy ADD xy. Wenn wir eine konkrete Zahl, sagen wir 3, im Auge haben, ersetzen wir xy durch diese (auf zwei Ziffern aufgefüllte) Zahl, also z. B. ADD 03 statt 4003. Entsprechend verhält es sich mit der Subtrak-

tion. Mit ADD und SUB können wir jedoch immer nur den vorherigen Inhalt des Akkumulators erhöhen bzw. erniedrigen. Manchmal möchte man aber den Akku mit einem neuen Wert unabhängig vom alten laden. Dazu dient der Befehl 80xy, den wir symbolisch mit LOAD xy bezeichnen. Wenn wir noch den Multiplikationsbefehl 60xy, symbolisch MUL xy, hinzunehmen, können wir ein kleines Programm schreiben, um die Formel (3+7)*6 zu berechnen. In BASIC wäre das ja kein großes Problem:

```
10 PRINT (3+7)*6
```

Das ist schon alles. In der Sprache der I2DITVM sieht das so aus:

Programm 1.3.2-1

```
8003            LOAD 03         lade den Akku mit 3
4007            ADD 07          addiere 7 dazu
6006            MUL 06          multipliziere das Ergebnis mit 6
```

Starte den Emulator (hört sich schon gut an, nicht?) und führe das kleine Progrämmchen aus. Das Ergebnis steht am Schluß im Akku. Wenn du willst, kannst du es auch noch mit anderen Zahlen versuchen. Wenn du genug gespielt hast, beende das Programm mit dem Befehl 991, den wir EXIT taufen werden.

Wir wollen jetzt die Formel so umstellen, daß die Multiplikation zuerst ausgeführt wird: 6*(3+7). Dabei taucht ein Problem auf: Wenn wir den Akku mit 6 geladen haben, können wir diesen Wert nicht sofort verarbeiten, sondern müssen, bevor die Multiplikation ausgeführt wird, erst die Addition von 3 und 7 vornehmen. Dazu brauchen wir aber auch den Akku! Wir müssen also die 6 vor dem Überschreiben retten und "merken" sie uns daher im Speicher, sagen wir in Zelle 0. Der Befehl, um den Inhalt des Akkus in den Speicher zu kopieren, ist 91x, wobei x die Adresse der Speicherzelle ist. Da wir nur 10 Zellen mit den Adressen 0 bis 9 haben, ist zur Angabe einer Adresse immer nur eine Ziffer nötig. Wir wollen den Befehl symbolisch mit STO (x) bezeichnen (von to store = speichern). Die Angabe einer Adresse werden wir immer in Klammern setzen, um Verwechselungen mit der Zahl selbst zu vermeiden. Damit können wir den Anfang des Programms schon niederschreiben:

Programm 1.3.2-2

```
8006            LOAD 06         lade den Akku mit 6
910             STO (0)         zwischenspeichern in Zelle 0
8003            LOAD 03         lade Akku mit 3
4007            ADD 07          addiere dazu 7
```

Jetzt fehlt uns nur noch die Multiplikation. Wenn wir nun die 6 aus dem Speicher in den Akku zurückholen würden, wäre ja unser Zwischenergebnis weg. Die I2DITVM bietet daher eine andere Möglichkeit an. Man kann den Inhalt des Akkus direkt mit dem Inhalt einer Speicherzelle multiplizieren (ebenso addieren usw.). Der Befehl dazu ist 61x, den wir MUL (x) nennen werden. Beachte, daß gegenüber dem bisherigen Multiplikationsbefehl 60xy hier an zweiter Stelle eine 1 erscheint (ebenso verhält es sich auch bei ADD und SUB). Und natürlich besteht der Operand, der hier ja eine Adresse meint, nur aus einer Ziffer. Damit können wir das Programm vervollständigen:

```
610             MUL (0)         multipliziere mit dem Inh. von Zelle 0
991             EXIT
```


Damit hast du schon mal einen kleinen Einblick in die Assemblerpro-

grammierung bekommen. Du merkst sicher schon, daß man sich hier um weit mehr Details kümmern muß als etwa in BASIC. Dort hätten wir von den Schwierigkeiten bei der Umstellung der Formel gar nichts gemerkt. Außerdem werden Programme in Assembler meist sehr lang, da jeder Befehl nur einen ganz kleinen Fortschritt bei der Lösung eines Problems erbringt. Vielleicht kannst du dir jetzt auch schon denken, woher der Begriff Akkumulator (akkumulieren=anhäufen, aufsammeln) kommt: das Ergebnis einer längeren Rechnung wird allmählich im Akku aufgebaut ("aufaddiert").

Übungsaufgabe 1.3.2-1:
Schreibe ein Programm, um die Fläche eines Rechtecks mit einem rechteckigen Loch (z. B. Papierfläche eines Fensterbriefumschlags) zu berechnen. Die Maße seien (L x B): äußeres R.: 11 x 8, inneres R.: 4 x 3. Hinweis: mit LOAD (x) kannst du den Inhalt der Zelle x in den Akku kopieren, mit XCHG (x) kannst du die Inhalte von Akku und Zelle x vertauschen.

Vertiefungsstoff
Beim MUL Befehl haben wir gesehen, daß der Operand zwei Bedeutungen haben kann: bei MUL 07 (6007) bedeutet er unmittelbar die Zahl 7, bei MUL (7) (617) bedeutet er eine Adresse, in der die gesuchte Zahl zu finden ist. Die verschiedenen Möglichkeiten, die Zahl, mit der multipliziert werden soll, anzugeben, bezeichnet man als Adressierungsarten. Die erste Art nennt man unmittelbare Adressierung, die zweite direkt (auch absolut). Weitere werden wir noch kennenlernen.
Ende Vertiefungsstoff

1.3.3 Der Befehlssatz der I2DITVM

Obwohl wir noch nicht alle Befehle kennengelernt haben, möchte ich an dieser Stelle schon den vollständigen Befehlssatz in Form einer Tabelle präsentieren. Die Wirkung eines jeden Befehls ist dabei in einer formelartigen Kurzschreibweise angegeben. Dabei steht A für den Akku und MEM(x) für den Inhalt der Speicherzelle mit der Adresse x, der Pfeil (<-) steht für die Wertzuweisung (LET in BASIC). Es ist nicht beabsichtigt, daß du aus der Tabelle heraus schon alle Befehle verstehst (obwohl es auch nichts schaden würde).

INTERACTIVE TWO DECIMAL DIGIT VIRTUAL MACHINE

Befehlssatz und Vorschlag für eine Assembler-Notation

Maschinen-befehl	Mnemonic	Wirkung
00	CLR A	A <- 0
01	CLR M	MEM(i) <- 0 (i=0..9)
02	CLR S	Stack <- 0, SP <- STACKSIZE+1
10	INC A	A <- A+1
11x	INC (x)	MEM(x) <- MEM(x)+1
12	INC SP	SP <- SP+1
20	DEC A	A <- A-1
21x	DEC (x)	MEM(x) <- MEM(x)-1
22	DEC SP	SP <- SP-1
30	SWAP	(SP) <-> (SP+1)
31x	XCHG (x)	A <-> MEM(x)

A S S E M B L E R: Kurs Teil 2

32	XCHG (SP)	A <-> (SP)
40xy	ADD xy	A <- A+xy
41x	ADD (x)	A <- A+MEM(x)
42	ADD (SP)	A <- A+(SP)
50xy	SUB xy	A <- A-xy
51x	SUB (x)	A <- A-MEM(x)
52	SUB (SP)	A <- A-(SP)
60xy	MUL xy	A <- A*xy
61x	MUL (x)	A <- A*MEM(x)
62	MUL (SP)	A <- A*(SP)
70xy	DIV xy	A <- A/xy (ohne Rest)
71x	DIV (x)	A <- A/MEM(x) (")
72	DIV (SP)	A <- A/(SP) (ohne Rest)
80xy	LOAD xy	A <- xy
81x	LOAD (x)	A <- MEM(x)
82	POP	A <- (SP); SP <- SP+1
90	CPL	A <- 99-A
91x	STO (x)	MEM(x) <- A
92	PUSH	SP <- SP-1; (SP) <- A
99	EXIT	Programmende

Wir werden des weiteren nur noch die Assemblerschreibweise benutzen. Die Maschinenbefehle kannst du dir aus obiger Tabelle heraussuchen (dann kannst du später in deiner Autobiographie schreiben, du seist selber mal Assembler gewesen).

S u p e r C a l c: Kurs Teil 5

Kurs-05.cal **Fortsetzung** **Wolfsburg, den 15.02.87**

Weiter geht es mit dem Befehl **/O = O(utput,** Ausgabe Printer/CPM.
Wir können auch den Arbeitsbogen ausdrucken lassen, mit der Eingabe:

/L ---> Den erforderlichen File von der Disk laden

Oder den augenblicklichen Bogen,der gerade bearbeitet wird, ausdrucken. " /L entfällt dabei.

/G B --> Den Rand des Bogen abschalten, oder belassen

/O ---> Aufruf für Ausgabe: Monitor,Disk, Drucker

a) "D(isplay) or C(ontents) Report?"
"D(arstellung oder C(Zellinhalt) Report?"

Hier haben wir die Wahl, die auf dem Bildschirm vorhandene Darstellung oder die tatsächlichen Zellinhalte mit Texten, Zahlen und Formeln ausgeben zu lassen.

b) Die Eingabe "D" oder "C" führt zur weiteren Anzeige:

" Enter Range "
" Bereich eingeben"

Erforderlich sind die Eckzellen, oben links und unten Rechts, des Bereiches der gedruckt wird.

c) Mit A1:D13, definieren wir den Bereich der zur Ausgabe kommen soll, führt zur weiteren Anzeige:

" Enter Device: P(rinter) , S(etup) , C(onsle) or D(isk)"
" Ausgabe nach: P(Drucker), S(Druckformat), C(Moni) oder D(isk)"

d) "P" --> Mit diesem Befehl, auch ohne "S", starten Wir den Ausdruck.Drucker auf bereitschaft:on line

"S" --> Nun können Wir entsprechend der Bildschirmvorgabe die Formatierung für Zeilen pro Seite (Standart 66) und Zeichen pro Zeile (Standart 132), ändern.

"C" --> Mit der Eingabe "C" können Wir den Bogen auf den Monitor ausgeben.

e) "D" --> Wenn hier "D" eingegeben wird, muß eine neue Datei angegeben werden. Es wird eine ".PRN" Datei erzeugt. Wir können aber auch "txt" ein geben als Endung. Aber eine Kontrolle ist nur noch mit Newword oder CPM möglich, Supercalc kann auf diese Datei nicht mehr zugreifen.

Das Eröffnet für Newword neue Möglichkeiten, weil Tabellen aus SC. dann leicht in Newword-Texte übernommen werden können. Diese legt man in SC. an, speichert eine PRN-Datei und übernimmt diese ins Newword.

S u p e r C a l c: Kurs Teil 5

Bevor die Eingabe "P","S" gemacht wird kann durch das ?-Zeichen eine Zusatzfunktions-Erklärung aufgerufen werden, es erscheint die Anzeige:

```
"Enter Device: P(rinter) , S(etup)        , C(onsole),  or D(isk)
"Ausgabe nach: P(Drucker), S(Druckformat), C(Monitor), or D(isk)
```

"P" --> Mit diesem Befehl starten Wir den Ausdruck,auch ohne "S"

"S" --> Wir können entspechend der Bildschirmvorgabe die Formatierung für Zeilen pro Seite (Standart ist 66) und Zeichen pro Zeile (Standart ist 132) ändern.

"C" --> Console schaltet auf den Monitor.

"D" --> Druckt nicht aus, sondern legt einen File auf Diskette an, der von SC. nicht mehr erreichbar ist.(CPM-Ebene)

Bei der Eingabe "S", ist es zweckmäßig die richtigen Ausdruckformate einzustellen, da sonst nur Kaos entsteht. (66 und 132) Es kann verkleinert oder vergrößert (Bildschirmformat) werden.

Ende zusatzerklärung: /O --> Output
Ende des Befehls : /O --> Druckausgabe

Die Befehlseingabe: /P ----> Protect

Daten schützen:

Wir können einzelne Zellen aber auch Zeilen, Spalten und Blöcke vor versehentlichem löschen schützen. Wir können aber damit auch verhindern, daß Dritte, die auch mit dem Arbeitsbogen arbeiten, bestimmte Eingaben verändern (solange sie nicht /U kennen).

/P führt zur Anzeige:

```
" Enter Range "
" Bereich eingeben "
```

Achtung:
Sie haben hier die Auswahl

einzelne Zellen (G45)
Zeilen (A4:F4)
Spalten (F1:F25)
oder auch Blöcke (B1:G20)
für die Schutzfunktion zu definieren.

Mit der Eingabe: A3:F3 Return

werden Zellen, Zeilen, Spalten und ganze Blöcke mit den darin enthaltenen Daten (Texte, Zahlen und Werte) geschütz.

Die geschützten Zellen werden mit anderer Helligkeit angezeigt.Die Löschbefehle /B und /D sind nicht wirksam.

Eine Veränderung des Zelleninhaltes ist auch nicht möglich und es wird mit einer Fehlermeldung, nach versuchter Eingabe, Quittiert.

S u p e r C a l c: Kurs Teil 5

****
Achtung: DER BEFEHL ----> /Z <----LöSCHT WEITERHIN DEN BILDSCHIRM.

Das heißt: Alle Eingaben im Bogen gehen für immer verloren. Die Anwendung /Z sollte nur angewand werden, wenn der Bogen auch wirklich gelöscht werden soll. (vorher noch sichern). Falls die Eingaben gebraucht werden.
****
MIT DER EINGABE: /U können wir mit derselben Reihenfolge, wie oben, den Schutz wieder unwirsam aufheben.

Ende des Befehls: /P ----> Protect.

Quit Options: /Q

Y(es) exits SuperCalc. Y(Ja) exit Supercalc.

<--- Wollen Sie Supercalc verlassen, --->

<--- dann mit Y quittieren, --->

<--- wenn nicht, --->

<--- dann mit N für nein Antworten, --->

<--- falls Sie vergessen haben, --->

<--- Ihre Daten vorher abzuspeichern. --->

<---------->

Die Befehlseingabe: /R -----> Replicate

Mit diesem Befehl können Eingaben die sehr häufig vorkommen, oder sich wiederholen, ganz einfach vervielfältigt werden.

a) /R ---> " Von? (bereich eingeben, Zelle, Spalte, und Zeilen "
Also kurz: Das was Wir vervielfältigen wollen.

b) -------> dann RET

c) ---> " To? (Enter cell) , then Return, or "," for Options
"Nach? (Zelle eingeben) dann Return, oder "," für Auswahl

d) ---> Bei der Eingabe " Von? ", also einer Zelle, wollen Wir eine Zeile vervielf.,definieren Sie dann den Bereich in dem Wir die Eingabe bei a)Von? gemacht haben, z.b: (A1:A11).

---> Etwa so: Text = Zelle A1 nach Zeile B1:E1, dann wird der Text viermal wiederholt und steht in B1 bis E1.

	" A "	B "	C "	D "	E "
1"	Text	Text	Text	Text	Text

Dieses ist auch Spaltenmäßig durchführbar, indem dann der Text nur untereinander steht. Das ist die einfache Form der Verfielfältigung.

S u p e r C a l c: Kurs Teil 5

Bei der weiteren Eingabe: z.b.: D1:F1
führt zur Anzeige:

" N(o adjust), A(djust), V(alues) "
" N(icht anpassen),A(npassen erfragen),V(Nur Werte)"

Die Anzeige sagt schon hinreichend klar, was bei " N " bzw. " V " geschiet. Die Eingaben der Zellen werden nicht angepaßt bzw. nur mit Werten übernommen. Das sollte im kleinen Rahmen unbedingt geübt werden

Bei der Eingabe: A ----> Adjust

" Source Cell A10 Adjust A2 (Y or N) ? "
" Ausgangszelle A10 Anpassen A2 (J oder N) ? "

Das Programm prüft, welche Zelleninhalte (A10) eine Anpassung verlangen könnten und bietet deren Inhalt zum Anpassen (oder nicht Anpassen) an. Überprüfen Wir nun, ob Wir im Zielbereich weiterhin die Zelle "A2" als Begin des Inhaltes haben wollen (Eingabe "N"), oder ob Wir diesen Wert an die neue Position anpassen wollen (Eingabe "Y"). u.s.w

Ende des Befehls: /R ---> Replikate

Bei der Eingabe: /T

Bringen wir den Cursor in die Zelle, Zeile oder Spalte, die wir auf jeden Fall noch auf dem Bildschirm haben wollen, auch wenn wir am anderen Ende des Arbeitsbogens sind.

Die Eingabe: /T führt zur Anzeige

" H(oriz.), V(ert.), B(oth), or C(lear)"
" H(oriz.), V(ert.), B(eides), oder C(Aufheben)"

Achtung: Wenn wir mit dem Cursor nicht in der beabsichtigten Zeile Stehen, müssen wir zwangsläufig den Befehl verlassen, weil die Cursorsteuerung nun auf die Eingabezeile wirkt.

Bei der Eingabe: "H" --> wird bewirkt, daß die Zeile, in der der Cursor steht,sowie alle darüber liegenden, als Titelzeilen fixiert werden.

Bei der Eingabe: "V" --> wird die Spalte fixiert, sowie alle Spalten links davon als Titel.

Bei der Eingabe: "B" --> werden die Zeilen und Spalten fixiert und der restliche Arbeitsbogen in diesen Winkel hineingeschoben.

Bei der Eingabe. "C" --> wird die Zeilen- und Spaltenfixierung wieder aufgehoben.

Das ganze erfordert ein wenig übung. Man sollte diese erst in kleinen fertigen, um zu sehen, wie das ganze so vor sich geht.

Ende des Befehls: /T --> Titel fixieren

S u p e r C a l c: Kurs Teil 5

Zum Schluss noch ein Beispiel der Fakturierung:

Dieses Beispiel zeigt, wie man mit SC. Werte aus Tabellen selektieren und diese Werte bei der Loesung eines Problems verwenden kann. Außerdem zeigt das Beispiel, wie man eine Rabattstaffel programiert und eine umsatzabhaengige Provision ermittelt.
Menge, Artikelnummer und Bezeichnung werden manuell eingegeben. der Einzelpreis wird aus Tabellen entnommen, der Gesamtpreis, der Rabattbetrag und die Mehrwertssteuer werden ermittelt. Vom Nettobetrag wird die Verkäuferprovision berechnet.
Arbeitsschritte:

Festlegen des Arbeits-Formates (Formatbogen)
Eingabe der mathematischen Formeln
Verblocken der Formeln (Schützen durch Protect)
Eingäbe der Werte
Abspeichern auf Platte (Saven auf Diskette)
Ausdrucken (Formatbogen über Output mit oder ohne Rand)

Benutzte Formeln:

Lookup = Tabellensuche	Max = Maximalwert
Min = Minimalwert	Sum = summieren

Benutzte Befehle:

Format = Formatieren	/ Global = Global	/ Insert = Einfügen
Output = Ausgabe	/ Protect= Verblocken	/ RepaetText=Textwiederhl
Replicate=Duplizieren	/ Save = Sichern	/ Unprotect =Entblocken

Eingabe:
------- **/FG14 <RET> = Format,Global,Spaltenbreite,Return**

Nun Schalten wir die automt. Sprungrichtung des Feldcursors ab.
Eingabe:
--------- **/GN = Global und Next**

S u p e r C a l c: Kurs Teil 5

Als nächstes geben wir die Daten im Arbeitsblatt ein:

```
 :      A      :      B      :      C      :      D      :      E      :      F      :
 1:Rechnungsnr.:
 2:Datum:
 3:Kundenname:
 4:Anschrift:
 5:PLZ:
 6:Ort:
 7:
 8:Verk.Nr.:
 9:
10:Menge        Artikelnr.     Bezeichnung    Stckpr.DM Gesamt DM
11:-------------------------------------------------------------
12:                                                .00        .00
13:                                                .00        .00
14:                                                .00        .00
15:                                                .00        .00
16:                                                .00        .00
17:                                                .00        .00
18:                                                .00        .00
19:                                                .00        .00
20:=============================================================
21:                                          Fracht:
22:                                    Zwischensumme:
23:                                    0%     Rabatt:
24:                                      Nettobetrag:
25:                                  13,5%       MWST:
26:                                                     ----------
27:                                       Gesamtbetrag:        .00
28:                                                     ==========
29:Verkäufer-Provisionsbericht:
30:----------------------------
31:      Verk.Nr:          0
32:  Rechnungsnr:          0
33:  ProvisionDM:        .00
34:
35:-------------------------------------------------------------------------------
36:                Preistabelle                                  Rabattstaffel
37:  Papierwaren  Preis DM   Glaswaren    Preis DM     Betrag      Prozent
38:-------------------------------------------------------------------------------
39:           0          0          0           0          0            0
40:         100       9.23        200        7.56        100           10
41:         125       8.55        225        9.54        200           12
42:         128       5.65        226        5.12        300           15
43:         129       2.96        230        6.32        500           18
44:         130       2.75        255        1.25
45:         131       1.58        275        4.68
46:         132       2.36        276        5.32
47:         133          0        280           0
48:-------------------------------------------------------------------------------
```

Die Daten wie hier im Arbeitsblatt in den Arbeitsbogen ihres Gerätes übernehmen. Zeilen und Spalten genau einhalten.

S u p e r C a l c: Kurs Teil 5

Die Texte auf unsren Arbeitsblatt werden jetzt Rechtsbündig angeordnet Dazu die folgende Eingabe:

/F	**für Format**
E	**für Entry (Eingabe des Bereiches)**
A1	**erstes zu formatierendes Feld**
:	**Doppelpunkt für von - bis**
B8	**letztes zu formatierendes Feld**
,	**Komma oder Return**
TR	**Text Right (Text rechtsbündig)**
RET	**führt den Befehl aus**

Diesen Befehlsvorgang für alle Texte im Bogen wiederholen, die rechtsbündig angelegt werden sollen.

Anschließend wollen wir einige Felder als Betragsfelder formatieren. Bitte folgende Eingabe:

/FE E21:E25,$ RET
/FE E27 ,$ RET

Eingabe der mathematischen Formeln: **in Zelle D12**

MAX(LOOKUP(B12,A39:A47),LOOKUP(B12,C39:C47)) RET

/FE,$ RET

Die erste Formel sucht anhand der Artikelnummer in zwei Preistabellen und überträgt den entsprechenden Einzelpreis in die Spalte Stückpreis. Die beiden Tabellen wurden eigens dafür aufgebaut, um zu zeigen, wie eine Suche in meheren Tabellen erreicht werden kann.

Die erste Formel wird in Feld D12 eingegeben, welches das erste Feld der Spalte Stückpreis ist. Zweimal wird die Funktion LOOKUP verwendet, um zu erreichen, daß nach der entsprechenden Artikelnummer in beiden Tabellen gesucht wird.

Normalerweise, wenn bei LOOKUP der Suchbegriff größer ist, als der größte Listenwert, wird der Wert rechts neben dem größten Listenwert als Ergebnis ermittelt. Ist der Suchbegriff jedoch kleiner als der kleinste Listenwert, wird ein Fehler (ERROR) angezeigt. Um dies zu vermeiden, wurde je Tabelle eine eigene LOOKUP-Funktion verwendet und Null als kleinster Wert jeder Tabelle angenommen. Damit wird, wenn der Suchbegriff kleiner ist, als der kleinste Listenwert, der Wert Null anstelle der Fehlermeldung erreicht. Dasselbe gilt auch für größer, wenn der Wert größer ist, als der größte Listenwert.

Von der Tabelle, die den gesuchten Wert enthält, wird der Preis rechts neben demselben in die Formel übernommen. Von der anderen Tabelle wird durch LOOKUP der Wert Null übernommen. Der größere der beiden Werte wird durch MAX selektiert und als Stückpreis angezeigt.

Die Zweite Formel multipliziert den Stückpreis mit der Menge und zeigt das Ergebnis in Betragsform unter gesamt DM an. Dazu bringen wir den Cursor nach **E12** und geben ein:

A12*D12 RET und **/FE RET $ RET**

S u p e r C a l c: Kurs Teil 5

Diese beiden Formeln wollen wir nun in die darunterliegenden Zeilen kopieren. Dazu geben wir ein:

/R	**für Replicate (Duplizieren)**
D12:E12	**zu duplizierender Bereich**
,	**Komma oder Return**
D13:D19	**Zielbereich (jeweils Anfangsfeld)**

ACHTUNG !

SC- ist so entworfen, daß Formeln, welche auf neue Positionen kopiert werden, die Feldadressen in Relation zur neuen Adresse setzen. Um Feldadressen beim Duplizieren von Formeln unverändert zu belassen, ist es notwendig, die Optionen von Replicate durch Eingabe eines Kommas anzuwählen und dann jeweils anzugeben, ob eine Feldadresse angeglichen werden soll (Adjust) oder nicht (Antworten: Y/N.
In unserem Falle ist es notwendig, die Adressen der beiden Tabellenbereiche unverändert zu lassen, Wir müssen deshalb die folgenden Eingaben machen:

,	**zur Auswahl der Replicate-Optionen**
A	**für die Funktion Adjust (Angleichen)**
Y	**Eingabesequenz auf die Frage, ob Adressen angeglichen**
N	**werden sollen oder nicht. Mit Y werden die auf die**
N	**jeweilige Zeile bezogenen Adressen angeglichen, mit N**
Y	**werden die Adressen des Tabellenbereiches unverändert**
N	**übernommen.**
N	
Y	
Y	

Die nächste Formel addiert die Werte in der Spalte Gesamt DM zwischen den beiden gestrichelten Linien. Die Summe wird als Zwischensumme in Betragsform angezeigt. Plazieren Wir den Cursor in **E22** und geben ein:

SUM(E11:E20) Return

Nun geben wir eine Tabellensuch-Funktion ein, welche die soeben ermittelte Zwischensumme dazu benutzt, den entsprechenden Rabattsatz aus der Rabattstaffel zu wählen. Die Rabattstaffel besteht aus aufsteigenden Beträgen, die progressiv steigende Rabattsätze abhängig von der Höhe der Zwischensumme ausweisen. Der Rabattsatz wird links vom Feld "% Rabatt:" angezeigt. Bringen wir den Cursor nach **C23** und geben wir ein:

LOOKUP(SUM(E11:E20),E39:E43) RETURN

Mit der nächsten Formel berechnen wir den Rabattbetrag. Bringen wir den Cursor nach **E23** und geben ein:

-SUM(	**Summiere die nachfolgende Liste und zeige das Ergebnis negativ an.**
E11:E20)	**Liste und Ende von SUM**
*****	**Multiplizieren**
C23	**Feld mit Rabattsatz**
/	**Divisionszeichen**
100	**Divisor und RETURN**

S u p e r C a l c: Kurs Teil 5

Der Nettobetrag wird mit einer einfachen Summierungsformel ermittelt. Bringen wir den Cursor nach **E24** und geben ein:

SUM(E22:E23) RETURN

Den Mehrwertsteuersatz geben wir in Feld (Feld steht immer gleich bedeutend für Zelle, also Zelle = Feld) **C25** ein. Dann wollen wir die Formel zur Berechnung des Steuerbetrages in **E25** eingeben.

E24*C25/100 RETURN

Jetzt müssen wir noch den Frachtbetrag, den Nettobetrag und den Mehrwertsteuerbetrag addieren und als Gesamtsumme anzeigen. Bringen wir den Cursor nach **E27** und geben ein:

SUM(E21,E24,E25) RETURN

Schließlich wollen wir noch die Provision für den Verkäufer ermitteln und anzeigen. Dies geschiet, indem wir den Nettobetrag mit einer abgestuften Liste von Beträgen vergleichen und dann mit einem entsprechenden Prozentsatz multiplizieren. Als Provisionssätze gelten: 10 Prozent für die ersten 100 DM, 12 Prozent für die nächsten 200 DM, und 15 Prozent für die Beträge über 300 DM. Bringen wir den Cursor nach **B31** und geben ein:

B8 übernahme der Verkäufernummer RETURN

Bringen wir den Cursor nach B32 und geben ein:

B1 übernahme der Rechnungsnummer RETURN

Bringen wir den Cursor nach B32 und geben ein:

(MIN(E24,100) wählt den Mimimalwert von E24 bzw. 100 aus

***** **Multiplikationszeichen**

0.10) **Provisionsatz 10%, Abschluß des Ausdruckes**

+ **Additionszeichen**

(MAX(0,MIN(E24-100,200))

ergibt den Maximalwert aus dem Wert 0 oder dem Wert E24 minus 300

***** **Multiplikation**

0.15) **15% Provisionssatz, Abschluß des Ausdruckes**

>RETURN<

/FE,$ **RETURN**

S u p e r C a l c: Kurs Teil 5

Verblocken der Formeln

Um in einem Feld (Zelle) eine geschriebene Formel nicht zu überschreiben, können wir den Inhalt der Zelle schützen. Dazu benutzen wir den Befehl Protect.

/P **für Protect**

D12:E27 **zu schützender Bereich (wird heller dargestellt)**

RETURN **führt den Befehl aus**

Mit dieser Eingabe haben wir auch das Feld, in das der Frachtbetrag manuell eingegeben werden soll, verblockt. Dieses Feld müssen wir nun wieder entblocken. Dazu benutzen wir den Befehl Unprotect.

/U E21 RETURN führt den die Befehle aus

Das Arbeitsblatt ist jetzt soweit fertig, um Daten eingeben zu können.

Dieses ist also ein Arbeitsblatt mit Besprechung aller Eingaben, die im einzelnen den Aufbau zeigen. Es dürfte nicht schwierig sein dieses Arbeitsblatt zum laufen zu bringen. Viel Spaß dabei!

Ende Kurs-05.cal W.Gieger

L e s e r b r i e f: Peter Würfel (7262)

Peter Würfel - Rosenstr11. - 7262 Althengstett - 07051/4375

09.04.87

- Leserbrief:

Den SuperCalc-Kurs von Wolfgang fand ich duft, die Kritik daran schoß m.e. e bißle übers Ziel raus. Wenn ich (und das ist mein Beruf) die Beiträge zu den Infos nach der Sprachqualität beurteilen würde, dann müßte ich 80-90% wegwerfen (Passagen meiner Artikel eingeschlossen). Aber was solls. Fact ist, daß ich z.B. mein SuperCalc über ein Jahr im Safe habe liegen lassen, weil mich 1. die englische Anleitung abgeschreckt hat und weil 2. mir überhaupt nicht klar war, was ich damit praktisch anfangen soll. Erst ein Telefonpalaver mit einem Clubmitglied hat mich dazu gebracht, mich durch das englische Manual durchzuquälen; und heute möchte ich dieses Programm nicht mehr missen (in meinem Beruf als Pauker für meine Notenlisten und privat zur Kontrolle meiner Einnahmen (weniger) und Ausgaben (sehr viel mehr), die über zwei Konten laufen). Und dann gehts mir auch in SuperCalc so wie in NewWord (und deshalb bin auch ich als SuperCalc-User für den Artikel von Wolfgang dankbar): ich bin froh, daß ich (mehr oder weniger elegant) mit einem Programm umgehen kann, merke aber, daß es noch Feinheiten gibt, die mir das Leben erleichtern könnten. Nur -- im Manual nachschlagen ? -- oh Horror, wie das bei meinen mageren Englischkenntnissen! Bin ich doch froh, daß ich das einmal geschafft habe. Ganz konkret: die Möglichkeiten z.B. des '/X'-Befehls in SuperCalc... das war dann wirklich etwas, wo auch ich als SuperCalc-User noch was (und zudem gut lesbar, weil 'teutsch') von Wolfgang mitgekriegt habe. Meine Aufforderung an alle SuperCalc-, NewWord-, dBASE- und TURBO-PASCAL-User: Wenn Ihr was ausgeknobelt habt, was nicht auf den ersten Seiten der Bedienungsanleitung steht, und was euch Gehirnschmalz abverlangt hat, um es auszutüfteln, dann nehmt Euch doch die Zeit, darüber einen kleinen Artikel zu verfassen und nach Hamburg an die Redaktion zu schicken. Viele sind froh, wenn sie die Zeit, die Ihr für eine 'Kleinigkeit' gebraucht habt, nicht auch noch aufwenden zu müssen. (Meine Frau meint eh' ich würde zu viel Zeit am Computer verbringen, und meine Kinder fangen deshalb langsam auch schon zu meckern an...)

===== TIPS UND ENTDECKUNGEN ==========

- NewWord: zwei Laufwerke:

Folgender Versuch wurde gestartet: Dokument von A: (das ist meine RAM4-Floppy) aufgerufen. Dann wollte ich mit ^KS auf Diskette (B:) sichern bzw. mit ^KD das von Laufwerk A: geladene Programm auf Diskette in Laufwerk B: abspeichern und habe deshalb folgendes eingegeben: ^KL, dann 'B' und dann ^KS (bzw.^KD). Das Ergebnis war Schrott: bei einem Versuch kam auf der in Laufwerk 'B:' liegenden Diskette nichts an, bei einem anderen hats Teile der Diskette gelöscht! Bei weiteren Versuchen hat mein NW 'ne Macke bekommen und bei nem anderen Versuch war nur noch der Kaltstart eine Möglichkeit, wieder vernünftig arbeiten zu können. Wie ichs jetzt mache? ganz einfach: 'HOME' (d.i. ^QR), dann ^KB, dann ^QC, dann ^KK, dann ^KW und nun: 'B:file.ext' (diese Befehlssequenz habe ich auf eine Funktionstaste gelegt) und wenn ich anschließend ^KH tippe, sieht mein Bildschirm aus wie zuvor, aber auf der Diskette in Laufwerk B: befindet sich ein Backup, das mir kein Stromausfall mehr nehmen kann!

L e s e r b r i e f: Peter Würfel (7262)

Anm.d.HH.: Es geht natürlich auch anders, nämlich mit der Kommandofolge: ^K^S Speichern & Weitermachen, d.h. einmal sichern
^K^O Kopieren
Mit ^K^O kannst Du Deine Datei auf eine andere, auch auf einem anderen Laufwerk, kopieren. Vielleicht ist das auch nicht schlecht. Vorteil ist, daß ich nicht erst mit dem Cursor durch die Gegend (Dateianfang und -ende) muß.

- NewWored: Patch

Das in Info 12-38 beschriebene Patchen, um nach Ausstieg aus NW in Bright-(Bildschirmdarstellung) weiterzumachen, findet sich im neuen NW in Patch-Menue #1, Buchstabe U (also dort 03 1B 42 33 pachen).

- CP/M: SUB.COM

Von Sub aufgerufen PIP LST:=LPT.INI : Ergebnis: das funktioniert nur, wenn LPT.INI nicht als SYSTEM-file im directory steht.

Anm.d.HH.: Nimm einfach PIP LST:=LPT.INIÄRÜ bzw. mit anderem Zeichensatz halt PIP LST:=LPT.INI[R].

- dBASE:

Die !-Funktion (Umwandeln von Klein- in Großbuchstaben) funktioniert nicht bei deutschen Umlauten (ä,ö,ü).

Anm.d.HH.: Übrigens auch nicht bei ß.

- BASIC:

Wenn direkt nach dem Rücksprung aus einem Unterprogramm (GOSUB) in der nächsten Programmzeile eine nicht definierte Variable folgt, dann erfolgt Fehlermeldung. Bei dieser Fehlermeldung wird jedoch nicht die Zeile angezeigt, in der die nicht definierte Variable steht, sondern die Zeile, die vorausgeht und das RETURN enthält.

- Turbo-Pascal: Installation:

In Info 7-10ff sowie Info 8-45 wurden Turbo-Patches dargestellt, die verschiedene Turbo-Fehler (bzw. Ungeschicklichkeiten) beseitigen. Ein weiterer Patch dieser Art wurde in Info 15-47 vorgestellt. Als Turbo-Pascal 3.0 und Ram4 - Benutzer habe ich die Patches 'Fehlermeldung automatisch laden' und 'Mod reparieren' aus Info 7 bzw. 8 durchgeführt. Der in Info 15 vorgeschlagene Patch hat bei mir nicht funktioniert, wenn ich in 20D4 den Call nach 0103 aufgerufen habe. Wenn ich jedoch CALL 0105 mit den in Info 15 vorgeschlagenen Patches aufgerufen habe, gab es keine Probleme. Doch beim Testen stellte ich fest, daß bei einem Fehler in einem Window zwar (durch ^X) der gesamte Bildschirm wieder zur Verfügung steht, da ich aber durch ein Pascal-Programm auch die Funktionstastentabelle umstelle, stellte ich fest, daß nach Abbruch nicht wieder auf die Funktionstastentabelle 1 zurückgestellt wurde. Ich habe deshalb den Patch (aus Info 15-47) so geändert, daß bei Abbruch eines Programms automatisch auch die Funktionsstastentabelle 1 wieder angewählt wird:

L e s e r b r i e f: Peter Würfel (7262)

```
20D4     CD
20D5     05
20D6     01
und dann an anderer Stelle:
0105     0E
0106     18
0107     CD
0108     00
0109     F4      (bis hierher wie Info 15-47)
anschließend jedoch ab 010A:
         0E/1B/CD/00/F4/0E/5B/CD/00/F4/0E/54/CD/00/F4/0E/31/CD/00/F4/
         (damit wird auf Funktionstastentabelle 1 zurückgestellt)
und dann erst weiter wie in in Info 15-47 beschrieben:
         3A/D8/00/C9
```

Dadurch wird dann also beim Auftreten eines Fehlers in einem Window nicht nur auf den Gesamtbildschirm, sondern auch in die Funktionstastentabelle 1 zurückgeschaltet!

- Turbo-Pascal und ZEX

ZEX (von RAM4) ist ja ne feine Sache, nur beim Versuch, ein compiliertes Pascal-file aufzurufen, brach ZEX mit der Fehlermeldung 'kein ausreichender Speicherplatz' ab, obwohl das Programm nur 10 k lang war. Was tun? Wenn man die Compileroptionen aufruft und 'C' angewählt hat, muß man anschließend noch die Endadresse herabsetzen, dadurch schafft man dann Platz für ZEX und kann später das .com-file ohne Probleme mit ZEX laufen lassen.

- Umgang mit DU2.COM:

Leider unterscheidet sich DU2.COM das mit RAM4 geliefert wird von dem von Ulrich Hönisch in Info 13-33ff beschriebenen DU etwas; hier eine Gegenüberstellung:

Ulrich: Info 13-33ff:	DU2.COM (RAM4)
I&<RET>	I&<RET>
m<RET>	m<RET> zeigt jedoch keine gelöschten Dateien, sodaß man sich diesen Schritt sparen kann
g00<RET>	g0<RET>
d<RET>	d<RET>
+<RET>	+d<RET>
d<RET>	
Rettungsaktion	
ch??,00<RET>	ch?? 00<RET> (Rest wie bei Uli)

Anmerkung: Wenn mehrere Dateien wieder reaktiviert werden sollen, dann muß die erfolgte Änderung mit w<RET> abgespeichert werden bevor mit +d<RET> 'weitergeblättert' wird.

Anm.d.HH.: Der Unterschied zwischen den beiden ch??-Befehlen ist, daß das ältere DU ein Komma zwischen Adresse (??) und Wert (00) haben will, während das DU2 das Komma nicht akzeptiert (dort muß eine Leerstelle hin).

L e s e r b r i e f: Peter Würfel (7262)

==== FRAGEN UND WÜNSCHE ==========

- Umwelttraktorpapier ?

Wen hat es nicht auch schon gestört, daß er solche Mengen hochwertiges weißes Traktorpapier verbraucht? Ich such deshalb eine Bezugsquelle für Traktor-Umweltschutzpapier. Wer kennt eine Bezugsadresse? (Übrigens, noch nie wurde so viel Papier verbraucht, wie im 'papierlosen' computerisierten Büro!)

- RAM4: LDIRZ.COM:

Bei der Bildschirmausgabe mit LDIRZ.COM wird nach jeder Zeile eine Leerzeile ausgegeben. Daß dadurch der Bildschirm, besonders bei umfangreicheren Libraries nicht besonders sinnvoll genutzt wird, hoffe ich, einer der Club-cräcks findet den passenden Patch, das zu ändern.

- RAM4: RCHEK:

Was bedeuten die Ziffern? Warum kann ich eine '9' durch nochmaliges Formatieren zum Verschwinden bringen? Sind solche Disketten, die u.U. erst beim drittenmal einwandfrei RCHEK durchlaufen genauso datensicher wie die, die schon beim ersten Formatieren fehlerfrei gemeldet werden?

Anm.d.HH.: Ein ':' heißt, daß kein Fehler auftrat, eine '9' bedeutet, daß alles beim zweiten Versuch klappte, eine '8' bedeutet, daß schon drei Versuche gebraucht wurden, ..., und schließlich eine '0', daß zehn Versuche gemacht wurden. Da CP/M nicht mehr als 0 Versuche macht, bedeutet also diese '0', daß die Diskette vermurkst ist.

- RAM4: Cursor

Manchmal ist mir der Autorepeat von RAM4 zu schnell. Kann das geändert werden?

Anm.d.HH.: Das sieht schlecht aus, es sei denn Du nimmst einen 2 MHz-Quartz. Mir z.B. ist der Autorepeat bei der Cursorpositionierung noch zu langsam.

- RAM4: Klick-Wünsche!

Unersättlich ist der Mensch, andere hilfreich und gut! Nachdem nun der Notizzettel für RAM4 da ist, noch eine weitere Bitte an alle, dies blicken und/oder auch selber brauchen können: Ein Taschenrechner fürs Klick-Menue! Brauchen würde ich nur die einfachsten Rechenfunktionen: + - * /. Könnte man das Window für diesen Rechner außerdem noch auf die linke Bildschirmhälfte beschränken, dann wäre das besonders praktisch, da bei z.B. Bestellungen die zu addierenden Zahlen ja rechts auf dem Bildschirm stehen.
Und wenn dann schon einer dabei ist, könnte er sich ja an einem Rechner zum Umrechnen Hex-Dez und umgekehrt im Bereich bis FFFFh versuchen. Auch diese Klick-Erweiterung wäre sicher auch für andere brauchbar.

L e s e r b r i e f: Peter Würfel / C L U B: Clubtreffen

Anm.d.HH.: Peter, stell Dir vor, da Bernd da schon lange dabei ist. Da ihn nun sein Amiga zeitlich auch in Anspruch nehmen wird, mal sehen. Vielleicht vervollständigt Olaf Krumnow die Sache.

- RAM4: Tastaturbelegung:

CTRL+SHIFT ergibt ja, wenn man die linke SHIFT-Taste benutzt bei einigen Funktionstasten etwas anderes als gewünscht; die rechte SHIFT-Taste funktioniert jedoch einwandfifi. Spricht was dagegen, die Tastaturplatine so umzubauen (entspr. Leiterbahnen durchtrennen und nötige Kabel verlegen), daß die linke SHIFT-Taste parallel zur rechten SHIFT-Taste liegt?

Anm.d.HH.: Nein

Clubtreffen des MTX User-Club Deutschland
Am 11. April fand ja in Hannover ein Clubtreffen, welches Christian Löhrmann vorbereitet hat statt. Den meisten, die dort waren hat es sehr gut gefallen - ein Kritikpunkt war, daß es recht plötzlich zu Ende ging, ohne anschließendes gemütliches Beisammensitzen. Dem gebe ich gerne recht - nur war ich z.B. gezwungen, gen Norden zu fahren, da meine Großmutter am Sonntag 89 wurde. Nächstes Mal habe ich mehr Zeit!

Was gab's neues/sehenswertes ? Highlights (engl. für Höhepunkte):
- Digitalisierte Grafikten von Otmar Rücker, auf der Edicta-Grafikkarte sichtbar und editierbar gemacht!
  Bis Ende Juni hat Cluadio zugesagt, das Turbo-Pascal Grafik-Paket für die Karte fertig zu haben (derzeit ist sein MTX/FDX zur Reparatur in Braunschweig)
- RAM 4.2 mit noch mehr Können! U.a. auch meine SDX unter RAM 4.2
- Einige Informationen und Tips von den Hardwareleuten
- Persönliche Kontakte

Ein großes Dankeschön gebührt Christian Löhrmann und seinen Helfern, die das Clubtreffen bis hin zu guten belegten Broten und genügend Getränken hervorragend vorbereitet haben.

Wir haben uns darauf geeinigt, daß dort ein geeigneter Ort für ein Clubtreffen ist, weil es immerhin den aus dem Süden anreisenden Mitgliedern 2 Stunden Fahrt erspart - und noch nicht allzu weit im Süden liegt. Desweiteren haben wir dort einen Raum. In Hamburg ist das derzeit etwas schwierig, da Herbert Gollnik, unser Raum-Beschaffer, der IMB-Kompatiblen-Welle verfallen ist. Außerdem ist es für mich sehr angenehm, wenn ich nicht zusätzlich zu dem Info, einigen Reparaturen und meinen eigenen MTX/FDX-Gelüsten noch ein Clubtreffen vorbereiten muß.

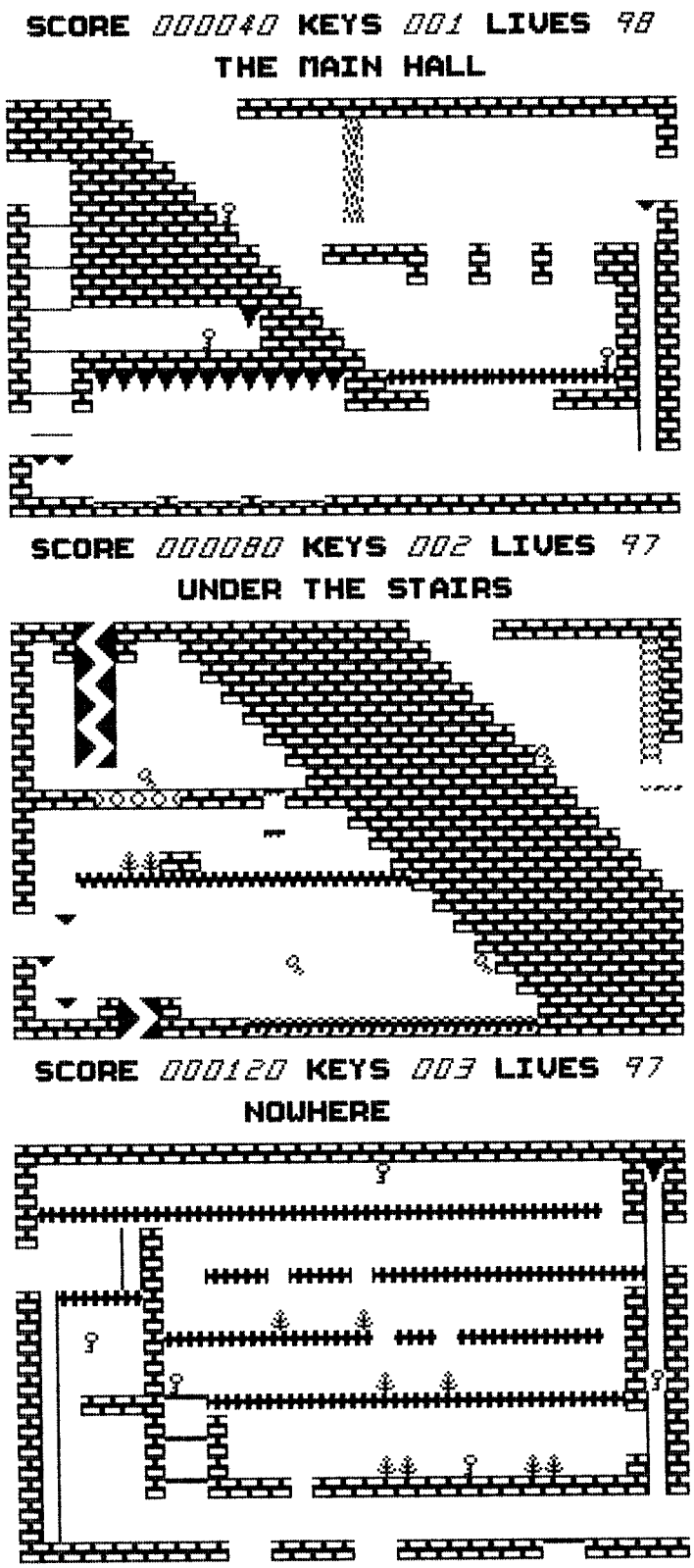
Planung nächstes Clubtreffen
Zeit und Ort: im Oktober in Hannover.
Leitung und Vorbereitung: Christian Löhrmann.

Es ist schwer es allen Recht zu machen - insbesondere, wenn wir nicht wissen, was Ihr mögt! Bitte schickt uns (Christian und mir) darum Eure Vorstellungen, Wünsche, Ideen - insbesondere zu folgenden Punkten:

- Dauer: von-bis; 1 od. 2-tägig, Übernachtung (was darf's kosten/wo),
- Wie soll es ablaufen (große Runde, kleine Themengesprächskreise ...)
- Was wollt Ihr sehen bzw. zeigen (Thema, benötigte Anlagen)

S P I E L E: Game-Hardcopy

Game-Hardcopy als RAM4-KLICK-Erweiterung (K.Muerling,8702)



Ein beliebtes MTX-Game ist Son of Pete. Es hat zahlreiche Kammern, die miteinander verbunden sind. Ich überlegte, wie man die einzelnen Kammern ausdrucken könnte, um eine bessere Übersicht zu bekommen. Durch Versuche bekam ich heraus, daß man das Game, wenn man es unter RAM4 betreibt, mit SHIFT ESC abbrechen und mit ESC weiterlaufen lassen kann. Leider sind die Bilder nicht auf dem VS 4, sonst hätte man in KLICK mittels HARDCOPY diese ausdrucken können. Immerhin kam dabei der Zeichensatz des Spiels heraus. Als ich mir das Programm SONOPETE mal mit dem MONI2 anschaute, daß das Spiel im Grafikmodus 2 läuft und daß die Namenstabelle ab 3C00 steht. Ich schrieb mir nun ein Programm, das die Namenstabelle und den Mustergenerator ausliest. Dann werden die jeweils zu einem Muster gehörenden 8 Bytes in eine neue Tabelle geschrieben und diese dann mit einem modifizierten VS 4-Hardcopy-Programm ausgedruckt. Das gesamte Programm habe ich dann als KLICK-Erweiterung, wie im Handbuch für RAM4 beschrieben, übersetzt.
Ich starte dann das Spiel unter RAM4. Wenn ich ein Bild ausdrucken will, halte ich das Game mit SHIFT ESC an, drücke F4, auf der das Hardcopyprogramm liegt und warte den Ausdruck ab. Dann drücke ich ESC und das Spiel läuft an der unterbrochenen Stelle weiter. Auf dem Ausdruck erscheint, wie nebenstehend ersichtlich, nur die jeweilige Kammer, die Schlüssel und die Schätze. Die bewegten Sachen sind alles Sprites und auf einer anderen Bildebene. Man kann die einzelnen Bilder zusammenkleben und erhält so einen guten Überblick.
Das Ziel des Spieles, die [illegible] habe ich leider noch nicht geschafft .....