

# MTX *User-Club Deutschland*

Info 19  
01.01.1987

**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 Gültigkeitszeitraum.

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

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

**Kontaktadressen:** (nach PLZ geordnet)

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

**Telefon-Sprechzeiten**

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

Inhaltsverzeichnis**C L U B**

Lesenswertes	Seite 1
Wer tut Was / Ports	Seite 2
Kleinanzeigen	Seite 3
Memotech gibt's noch	Seite 5
Betrug	Seite 7
Diverses von Thomas Wulf aus Nürnberg	Seite 8
Kritik	Seite 11

**d B A S E**

Dunkle Stunde von Ashton Tate	Seite 7
Patch für OSBORNE-dBASE	Seite 7

**G R A F I K**

Ein Fractal	Seite 11
-------------	----------

**B A S I C**

Kamikaze	Seite 12
SAVE, LOAD, VERIFY in Assembler	Seite 12
Micro-Assembler	Seite 13
Interrupt-Hardcopy	Seite 13
Relozier-Routine	Seite 15
REM-Entferner	Seite 16

**F O R T H**

auf Cassette	Seite 17
--------------	----------

**S P I E L E**

Kritik	Seite 18
--------	----------

**N e w W o r d**

Adreßverwaltung	Seite 19
Tips	Seite 21
NWPRINTN.OVR-PrinterOverlay von CLUB.009	Seite 22

**L e s e r b r i e f**

Uwe Groos, 6393	Seite 25
-----------------	----------

**T U R B O**

Installation	Seite 28
Interrupts	Seite 31

**H A R D W A R E**

Ton Aus	Seite 34
Tastaturbelegung	Seite 35
Digitizer - On Stage	Seite 37
Datenerfassung mit BASIC-EMUF	Seite 48
Temperaturwächter für Netzteil	Seite 49
Immer noch nicht schnell ?	Seite 50
Mut zum Löten ...	Seite 51
Leserbrief	Seite 52
SRAM-Floppy	Seite 53

**A S S E M B L E R**

Kurs Teil 1	Seite 54
-------------	----------

**S O F T W A R E**

EDICTA-Grafik	Seite 53
MTX-Edit	Seite 58
RAM 4 - Update	Seite 58
KLICK - Tips	Seite 58

Preis für dieses Info: DM 11,20

Redaktionsschluß für Info 20: 1. Mai 1987!

Vielen Dank für die reichlich eingetroffenen Beiträge zu diesem Info

Liebe Leserinnen, liebe Leser,

auf zum nächsten Info. Doch zuerst möchte ich bekanntgeben, daß mir meine Arbeit bei Hermes Kreditversicherungs AG in Hamburg Spaß macht - und mir weder die Computerei vermiest, noch meine Freizeit zusehr einschränkt. Abends, wenn ich meinen MTX anschalte genieße ich immer wieder die geringen Antwortzeiten - schließlich habe ich meine CPU ja auch für mich alleine!

Das **Clubtreffen** steigt am 11.04.87 unter Regie von Christian Löhrmann in Hannover. Weitere Info von ihm. Ich war doch recht irritiert, als ich hörte, daß sich so wenige am 22.03.87 angemeldet hatten. Kurz danach hörte ich von einem Mitglied "Also dann bis zum 11. in Hannover". Frage von mir: Hast Du Dich angemeldet? A: "Nein, muß man das ???". **Anmeldung** muß sein, damit Christian disponieren kann - auch einkaufen genannt! Woher soll er wissen, mit wievielen er rechnen soll. Und eine Anmeldung 1 Tag vor dem Treffen (ohne zwingende Gründe für die Verzögerung) finde ich ausgesprochen unfair! Gelle ?

Bitte entschuldigt vielfach die schlechte **DRUCKQUALITÄT** des Info 18. Die Druckmaschine meiner Druckerei war defekt. Ich stand vor der Wahl eine Woche zu warten - evtl. auch länger - oder diese lesbare, aber nicht allzu ansprechende Druckqualität zu akzeptieren. Nun habe ich mich für die rasche Informationsverbreitung entschieden. Das war eine Ausnahme!

Kürzlich erhielt ich von der **Post** meine **Kontoauszüge**, und stand auf dem Schlauch. Aber fangen wir am Anfang an. Wenn ich Schecks einreiche lege ich einen blauen Einzahlungsschein anbei, und vermerke auf dem Empfängerabschnitt den Namen des Scheckausstellers. Neuerdings schickt die Post aber die Empfängerabschnitte nicht mehr zurück, sondern druckt auf den Kontoauszug 'SCHECKEINR 0000 .....'. Telefon her, und folgendes habe ich erfahren: Auf den Empfängerabschnitt könnt Ihr eine 5-stellige Zahl schreiben, die dann anstelle der o.g. Nullen auf dem Auszug erscheint. Das klappt sogar mittlerweile. Der Nachteil ist, daß ich nun hier eine Liste mit Zahlen von 00001 - 00??? liegen habe, und jedem Scheck eine Nummer zuordnen muß, und nach Rücklauf diesen Vermerken muß, damit ich weiß, welcher Scheck zu welcher Buchung gehört. - Klappt nicht immer! (oft erscheint immer noch 0000)

Hallo **SDX-Beitzer**: Meine SDX sieht zwar merkwürdig aus, weil ich eine 768k-Karte eingelötet habe, und daher ein Stück Platine zwischen MTX und SDX-Controller in der Luft hängt, und dahinter mein altes FDX-Netzteil als Stromspender tut, ABER ABER ABER: Sie läuft unter **RAM 4**. Leider ohne die 8"-Formate, aber alles andere funktioniert problemlos! Ich habe angefangen zu überlegen, wie ich auf eine MTX-Hauptplatine 256k bzw. 512k RAM setzen kann. Dabei will ich versuchen die 512k-Lösung zum gleichen Preis wie eine 512k-Karte hinzukriegen. Vielleicht auf dem Clubtreffen ....

Werner Steiner mußte sein Oume **Laufwerk nachjustieren** (lassen). Er machte den Fehler das Laufwerk ohne den **Schutzkarton** (d.h. die Papp-Diskette) der Post zu überlassen, und da die Köpfe bei offenen Laufwerken frei in der Luft hängen ist das ggf. fatal! Wenn der Karton schon im Müll gelandet ist, dann könnt Ihr als Ersatz ein Stück glatte Pappe wie eine Diskette zuscheiden (mit dem großen Mittelloch).

**MEMOTECH** gibt's noch, und die produzieren sogar neue Computer. Christian Wöhlbier hat in seinem MUCS-Info einiges dazu geschrieben, was ich hier übernommen habe. Ich bleibe am Ball!

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

*Herbert*

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

Allgemeines	H. Herberg
(FDX-)BASIC	A. Viebke
CP/M System	B. Preusing, H. Herberg
Assembler	H. Oppmann
NewWord	U. Grass, H. Herberg
Turbo-Pascal	D. Krumnow, B. Preusing, 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 fehlt am Platz oder vermißt fühlt ... schreibe mir. (Bitte nur ernstgemeinte Zuschriften, d.h. Ihr solltet im genannten Bereich "firm" sein).

**Ports** (Herbert Herberg)

<u>Bereich</u>	<u>Port</u>	<u>Verwendung</u>
MTX	00 - 0F	Grundgerät
	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
	AB - AB	c't SRAM-Floppy
	B8 - BB	Conitec-Floppy
	BC - BF	Conitec-Floppy
	C0 - C4	Reserviert für Testzwecke !!!!!
	CC - CF	Janich & Klass Programmer
FB - FB	HD 64180 Sub-Prozessor-Karte, C. Romanazzi	

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

## KLEINANZEIGEN

**Redaktionelles dazu:** (Herbert Herberg, 2000)

Ich habe bei diesem Info noch einmal ein Auge zugeedrückt, aber ...  
(Siehe auch letztes Info)

Wenn Ihr Kleinanzeigen ins Info haben wollt, was Ihr sehr gerne machen könnt - solange es keine Kontaktanzeigen sind ..., dann schickt mir bitte **SCHRIFTLICH** oder auf **DISKETTE** den Text der Anzeige - am besten in der Form, wie unten zu sehen. Ich übernehme in Zukunft nur solche Dinge, die ich dann auch dort vorfinde (schaue also nicht in meine Mitgliederliste wegen Anschrift oder Telefonnummer)!

Herbert Herberg, Sonnenau 2, 2000 Hamburg 76, 040 - 2008704:

- Ich vermittele jederzeit gebrauchte/neue Geräte und Teile der selben. Außerdem weiß ich i.a. was es wo am billigsten gibt.
- Ich habe Apple-Communication-Software: Software für Rechnerkopplung Computer mit einem Apple. Das sind zwei Disketten (1x MTX, 1x Apple), die ich ggf. verleihe, da ich die Apple nicht kopieren kann.
- Ich habe FDX und MTX mit/ohne Monitor, Drucker, ... zu verkaufen. Preis ist Verhandlungssache!
- FDX, ein Laufwerk, MTX 512 mit 2764-EPROMs auf der Hauptplatine (umgeändert für RESET-Feste RAM-Floppy), RS 232C, ECB-Option, neuer 80-Zeichensatz, 10 PD nach eigener Wahl in Top-Zustand (Post-versand-fähig, d.h. rüttelfest) für DM 1800.-  
Was ich weitergebe ist überprüft, FDX bootet dann einwandfrei!
- Verschiedene WordStar BASIC, CP/M-Bücher gebraucht für je DM 10.-
- Einspaltige Etiketten, 1000 Stück, 8,9 cm x 3,6 cm: DM 16.-

Solange der Vorrat reicht:

- Platinenstecker für Erweiterungen links am MTX-Grundgerät. Natürlich mit dem Gegenstück zu der Kerbe an Pin 5. je DM 5.-
- Dynamische RAM's 32k x 1 Bit: 8 Stück DM 2.-
- Statische RAM's 2k x 8 Bit (6116): je DM 2.50
- TTL-IC's: 74LS175, 74LS368, 74LS 21, 74LS173, 74LS04, 74LS14, 75LS155, 74LS158, 74LS139, 74LS258 je DM 0.60; 74LS10, 74LS11 DM 0.40
- Z80-Chips: Z80A CPU DM 2.-, Z80 PIO DM 2.-, Z80A PIO DM 3.50, Z80A CTC DM 3.-, Z80A SID DM 5.-
- Z805 (Festspannungsregler 5V): DM 1.-

## **D I V E R S E S**

Klaus Peter Kielbassa, Kolpingstr. 56, 4400 Münster, 0251-296526:

Terminal-Programm, welches 1200/75-Baud Full-Duplex packt, d.h. das Programm muß mit 1200 Baud senden, und gleichzeitig mit 75 Baud empfangen, oder umgekehrt.

Michael Keßler, Röntgenweg 6, 5600 Wuppertal 1, 0202-436649:

kann echte 8"-Disketten lesen/schreiben und auf MTX konvertieren.

C L U B: Memotech gibt's noch

### **Aktivitäten von Herbert Herberg**

Memotech hat sich aufgerafft, einen MTX mit 256k auf der Hauptplatine herauszubringen: MTX512 2S. Ich werde mich um genaue Informationen dazu bemühen! ABER: Ich habe schon angefangen, mir ersthaft Gedanken darüber zu machen, wie man die Hauptplatine auf 512kB aufrüsten kann - damit es sich lohnt. Ich habe festgestellt, daß 512k und 256k keinen allzu großen Unterschied in der Konzeption machen. Ob ich bis zum Clubtreffen Resultat vorweisen kann weiß ich nicht. Jedenfalls soll die Aufrüstung der Hauptplatine auf 512k etwa soviel wie eine 512k-Aufrüstung der 32k-Karte kosten. Hat jemand eine Idee ???

Wer von Euch Interesse an den neuen Memotech-Produkten hat, kann sich gerne an mich oder an Christian Wöhlbier wenden.

### **Geschichte von Memotech** von Christian Wöhlbier

Memotech musste am 14 März 1986 Rekurs einlegen. Kurz darauf begann Memotech mit einer neuen Firma zu verhandeln: MEMOTECH COMPUTERS LTD (kurz: MCL). Diese Firma übernahm dann die ganze Produkte-Palette auf Lizenzbasis. Dieses Unternehmen steht unter der Führung von Mr Geoff Boyd, einem vorherigen Research Director von Memotech Ltd. Alle Produkte werden weiter produziert und der Service klappt einwandfrei.

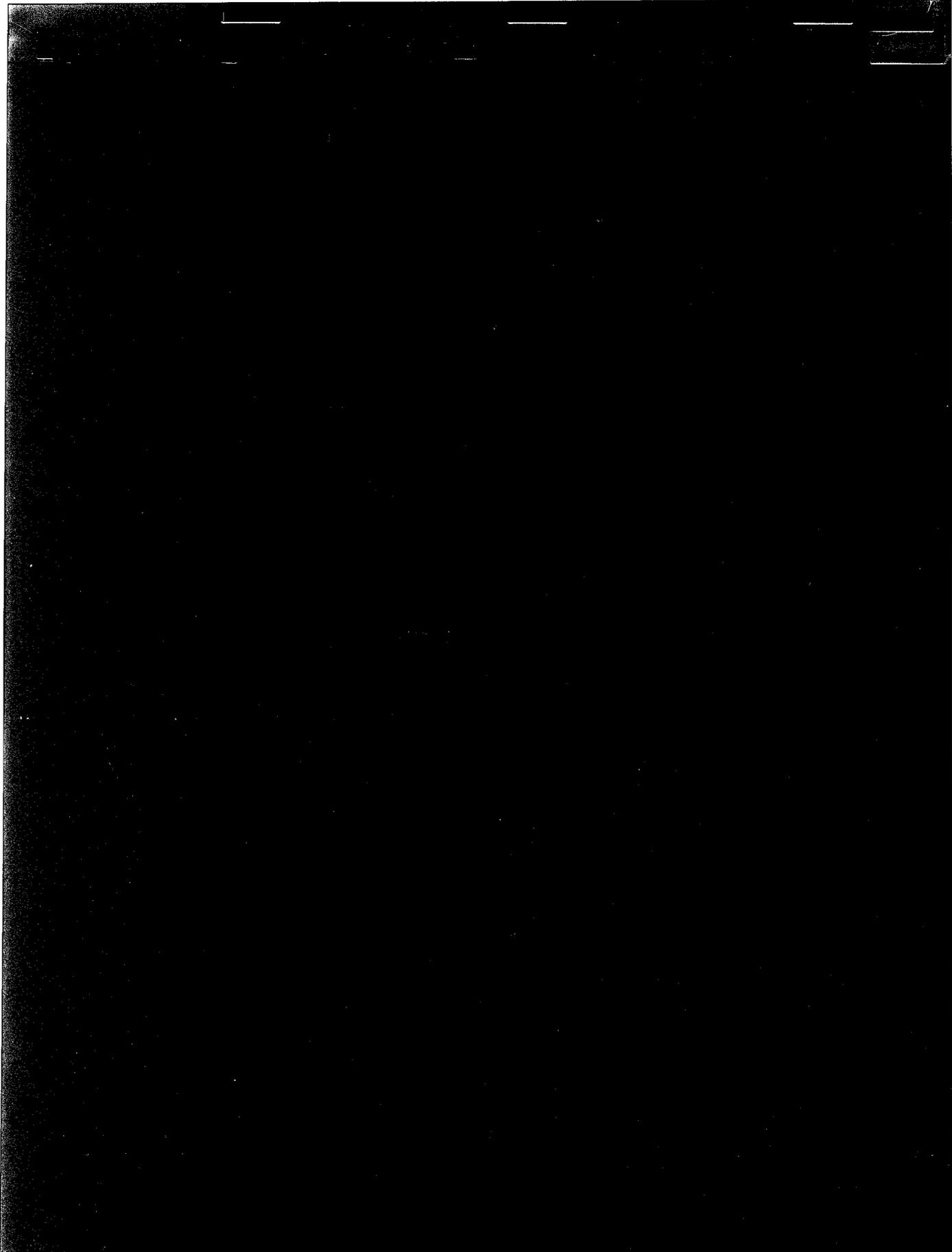
Kurz nach der Uebernahme erschien schon die erste Neuigkeit: Das 3.5"-Floppy-System mit oder ohne CPM 2.2, 80Z-Karte und NewWord.

Anfang dieses Jahres wurde dann der neue MTX von MCL vorgestellt. Dieser MTX512 S2 hat als einzige Hardware-Neuerung einen 256K RAM Speicher. Im Informationsblatt wird darauf hingewiesen, dass das MTXBasic alle 256K RAM voll unterstützt. Als Erweiterung (oder im Paket) gibt es ein Basic- und ein CPM-Disk-System. Im Basic-Disk-System ist ein 3.5" 1MB-Laufwerk + I/F + MTXBasic Disk Filing System enthalten. Im CPM-System kommt noch CPM 2.2 + 512K-SiDisc + 80Z-Karte + NewWord dazu. Ein Televideo 920 Emulator, der sich auf der System-Diskette befindet, soll mögliche Format-Probleme lösen!

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

- Uwe Grass, Wachholtzstr. 8, 3300 Braunschweig, 0531-343167:  
MTX 500 mit 512k, R5232, 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.
- Uwe Beythien, Lorenzengasse 1, 2000 Hamburg 60, 040-275263:  
MTX, 512k, FDX, 2 Lw. (1x 40, 1x 80), Panasonic KXP 1091 (NLQ), TP 200, 100 Disc: VB DM 2100.-
- Frank Bueschler, Am Ochsenzoll 3, 2000 Norderstedt, 040-5277581  
 verkauft MTX 500, 96k, FDX, 2 Lw., TP 200, RS 232: DM 1500.-  
 Optional im Gerät 512k-Karte DM 250.-, TEAC GFV-Lw. 300.-
- Holger Petersen, Händelstr. 22, 2950 Leer-Loga, 0491-7865:  
 Neuwertiger STAR SG-15 Matrixdrucker (Din A3) VB DM 900.-  
Hisoft-Pascal-ROM für MTX VB DM 120.-
- Horst Holzappel, Krähenhoop 9, 3180 Wolfsburg, 05361-61235 (ab 17h):  
MTX 500, FDX, 2 Lw. TP 200, DMX 80, ca. 35 Discs, Reinigungsdisc, Infos: DM 1500.-
- Hauke Ahrensfeld, St. Georg-Str. 6, 3100 Celle, 0514-123490:  
MTX 500: DM 110.-, folg. Spiele je DM 7.-: Dennis, JetSetWilly, AstroPac, MissionAlphatron, 3D-TachyonFighter, Turbo, Quogo. Cass-Rec von Sony m. Counter: DM 30.-
- Eberhard Grisebach, Silcherstr. 32, 7307 Aichwald, 0711-363174:  
Farbmonitor Thomson CM 36512 AR (nur für 80-Zeichen) DM 450.- (Bei Versand zzgl. P&V).
- Peter Michalke, Yorckstr. 2a, 4670 Lünen 6, 0231-877407:  
MTX 500 incl Staubschuthaube, FDX, 2 Lw., TS 200, RS 232, Centronics-Kabel, dBASE II (deutsch), mehrere Public-Domain dazu, Chess, 6 Spiel-Cassetten: VB DM 1700.- (wenig genutzte Anlage)
- Andreas Viebke, Thurgauer Str. 9, 1000 Berlin 51, 030-4955689:  
Mephisto exclusive mit Modul MM2, makelloser, 5 Monate alter Super-Schachcomputer im Holzbrett mit Leuchtdioden und unsichtbaren Magnetsensoren (elegant, ein Plastik-Zeug). Je 10 Spiel- und Problemstufen, Lern- und Infomodus, vor- u. rückwärtsspielen innerhalb einer Partie, viele, viele andere Extras. DM 750.- (neu DM 1000.-)
- Alfred Popellak, Weilersweg 5, 6966 Seckach-Gr. 0931-59263:  
MTX 500, FDX, 2 Lw., DMX 80, TP 200, Turbo 3.0 VB DM 1900.-
- Jörg Heße, Haputstr. 86, 5650 Solingen 1, 0212-201338:  
MTX500, +128k, FDX, 2 Lw (1x40, 1x77/80), DMX 80, TP 200, RAM4.1, Info's, umfangreiche Programme incl. Literatur: VB DM 1800.-
- Jens Trzyputz, Harburger Ch. 109, 2000 Hamburg 28, 040-753855:  
MTX 512, +64k, FDX, 2 Lw, Monitor, Epson RX 80: VB DM 1300.-
- Amr Diab, Handschusheimer Landstr. 54a, 6900 Heidelberg, 06221-400153:  
MTX 500, FDX, 2 Lw. , Farbmonitor, Info 1-17, dBASE II, CLUB-PD's: VB DM 1500.-
- Dr. Jürgen Jacob, Böninggweg 35, 3040 Soltau, 05191-14402:  
MTX 512, FDX, DMX 80, TP 200, s/w-Ferseher für Ton, Dia-Show, 7 BASIC-Discs, 9 Spiele. Voll funktionsfähig. DM 1000.-

C L U B: Memotech gibt's noch



C L U B: Betrug / d B A S E

**Macke in dBASE II ----> dSA**

(Herbert Herberg, 2000)

dBASE II macht mir so richtig FREUDE! Alles funktioniert völlig problemlos, und fehlerfrei bis auf eine Winzigkeit: Wenn ich eine Datenbank (.DBF) habe, diese Indizieren lasse (INDEX ON .....), den neuen Index aktiviere (SET INDEX TO ....) und dann mit einer WHILE-Schleife ausdrucken will, dann werden einige Sätze bei der Ausgabe übergangen. Das ist bei der **Mitgliederliste** im Info 18 passiert. Alle, die dort fehlen habe das 'Nicht-Gelistet' Flag gesetzt, was ich für die Update-Liste (wie in diesem Info) verwende. Fazit: dBASE hat die Angewohnheit bei Verwendung eines neu erstellten Indexes etwas nachlässig mit der Indizierung zu sein. Der Index erwischt anscheinend nicht alle. Mein Gott bin ich froh, daß bei meiner Club-Verwaltung für die meisten Sachen der Index garnicht benutzt wird. Bislang habe ich noch nie erlebt, daß ich Daten verlor! Den brauche ich nur, um bestimmte Mitglieder direkt zu suchen - und die waren auch immer da. HILFE

**Patch OSBORNE-dBASE**

(Herbert Herberg, 2000)

Damit das Osborne-dBASE Version 2.3B auf unserem MTX läuft müssen folgende zwei Patches gemacht werden:

Datei	Adresse	Alter Inhalt HEX	Neuer Inhalt HEX
DBASE.COM	3964	C6	C9
DBASEMSG.COM	18C2	1A	0C

C L U B: Diverses von Thomas Wulf aus Nürnberg

Nürnb. Roritzerstraße - ram4 - sorry - treffen

info 19 seite

# hallo leute!

mögen die maschinen für uns arbeiten.  
 obwohl die edv mit als 1. aus dem umzug auftauchte  
 hat sie nicht mehr zu tun gekriegt  
 als eine julia-menge zu rechnen (ca 1 qm),  
 in zcpr und ram4 hineinzuschmecken (mmmh)  
 und kundzutun von einigen veänderungen..  
 und was nicht alles in den letzten monaten passiert ist!  
 so bin ich umgezogen, gatte und vater ( ja herbert,  
 du hattest es richtig: aus pflaum ist wulf geworden).  
 Aber es heißt Roritzerstraße, nicht RÖ. wie in der Liste  
 und ich hab auch ein modem.

...  
 und schließlich also auch besitzer von ram4/zcpr2.  
 und das nimmt einen auch ganz schön in anspruch, bis die  
 letzten feinheiten erkannt sind. wovon noch keine rede sein  
 kann..  
 nicht alle features von zcpr2 sind für mich von besonderem  
 interesse; so sind benannte directories und menue-programme  
 wohl nur dann von bedeutung, wenn man die maschine mit  
 unerfahrenen benutzern teilt. und felix ist gerade 17 tage alt..  
 immerhin sind es möglichkeiten auch für sich selber die  
 organisation des disketten und file-systems zu überdenken.  
 gold wert sind dagegen die uhrzeit und datums-funktion, sowie die  
 erweiterten fähigkeiten der ramdisk: 4-mal mehr funktionstasten,  
 software-mäßig einstellbare f-tabellen, verbesserte bedienung  
 und: die klick-erweiterung. das ist schon etwas ganz besonderes:  
 mit gewissen spezialprogrammen ist eine art pseudo-multitasking  
 möglich: jedes normale programm kann unterbrochen werden und  
 eine klickerweiterung eingeschoben! D.h. z.B. mitten in dbase  
 kann man das klick-programm hardcopy aufrufen und den bildschirm  
 drucken; danach läuft wieder dbase oder was auch immer!!  
 wer sidekick kennt, weiß dieses 'schnell mal zwischendurch' zu  
 schätzen. Was hatte MS-DOS außer der hardcopy-taste doch gleich noch  
 zu bieten ??! (..ach ja: die letzte eingabezeile steht zum editieren  
 bereit.. könnten wir eigentlich auch können oder?)  
 und 25 zeilen am schirm! und..und..  
 keine angst, ich hör schon auf zu schwärmen..

...  
 eine entschuldigung muß ich loswerden:  
 da zeige ich euch immer wieder von meinen fractals,  
 aber mit einem verbreiteten programm, das bei euch allen für alle rechnet  
 bin ich noch nicht rübergekommen - sorry, tut mir am leidesten!

C L U B: Diverses von Thomas Wulf aus Nürnberg

- treffen - sub und enter - info19 seite .

im dezember hatten wir in nürnberg wieder ein treffen  
 bekannte und neue gesichter und zwei müde fdxe;  
 ohne heiße neue aufrüstungen konnten wir wieder nicht mehr als 1-2 dutzend  
 anlocken, die sich angeregt unterhielten über alles und sich.  
 vielleicht im frühjahr oder herbst.

...  
 was mir schon lange auf der zunge lag ist durch ZEX plötzlich  
 obsolet geworden: nämlich die tolle kombination aus SUB und ENTER.  
 damit füttert man auch eigene programme mit input  
 statt selber einzutippen hatte ich im letzten frühling  
 zwei mtxe mit sub-files voller daten 2 wochen allein gelassen  
 worauf das bild aus info 16 erschien. alle 10-20 stunden wurden wieder  
 eine rechen- oder sicherungsbefehle abgerufen, während wir in paris waren.  
 einige spielregeln wollte ich mitteilen:  
 : der enter-puffer, der bei jedem aufruf angelegt wird darf nicht  
 überschritten werden, sonst abbruch. die gröÙe hängt von der systemgröße ab.  
 : der enter-puffer, so vorhanden, wird als 1. abgearbeitet!  
 also wird ein sub-file demo1.sub:

```
dir a:
enter ; wo kommt der kommentar?
dir b:
```

die frage zuerst und dann erst dir a: und b: zeigen! folgt: wenn eine enter  
 eingabe nötig wurde, müssen alle logisch folgenden auch dahinter in den enter-puffer!  
 : sub-files können verkettet werden aber nicht (?) als subroutinen aufgerufen werden.  
 also ist ein sub-file demo2.sub:

```
; rufe dich!
sub demo2
```

eine endlosschleife. (abhilfe meist <DEL>)  
 folgt: ein sub aufruf in einem sub-file ist nur am ende sinnvoll.  
 : es können 9 parameter übergeben werden (\$1 bis \$9) die beim aufruf  
 von sub zur erstellung des ablauf-files (\$\$\$) eingefügt werden  
 sub demo3 pas ruft also nacheinander 3 dirs ab  
 ; demo3.sub: abc-directories filter:\$1  
 dir A:\*\$1  
 dir B:\*\$1  
 dir C:\*\$1

: ein geschachtelter enter befehl wird ausgeführt (das heißt sein  
 text ein 2. mal (weiter hinten) in den puffer geschrieben), wenn er  
 ausgeführt wird, nicht wenn er (mitsamt eigenem text) als nutztext  
 des äußeren enter das 1. mal in den Puffer kam.  
 : parameter werden aber sofort aufgelöst. (da aber dabei aus \$\$ ein  
 einfacher \$ wird kann man die auflösung verzögern.)

```
; demo4.sub: enter ↑ 3
enter dir a:\\enter dir c:
enter dir b:
```

liefert also das gleiche wie demo3.  
 : wo man keine zwei backslashes (\) für zwei <RET>s hintereinander  
 setzen kann, was als einer interpretiert würde und nichts dazwischen  
 (FORMxx!) hilft ein einzeln auf eine zeile gesetztes enter.

C L U B: Diverses von Thomas Wulf aus Nürnberg

- sex - make - bye info 19 seite .

tja. nun, einige monate später ist das alles für mich nicht mehr von bedeutung unter ram4 und zcpr2 läuft enter nicht mehr und es ist viel interessanter, zu testen, ob und wie der nachfolger ZEX seine aufgaben erfüllt. wird er controllcodes weitergeben und tipptasten- und edit-eingaben durch- und hintereinander abarbeiten und wenn ja wie?

die ersten erfahrungen sind noch keineswegs erschöpfend, aber im großen und ganzen ja.

und noch viel mehr!

allerdings: die alten sub-files müssen neu, enterfrei geschrieben werden. vielleicht kann man ja ein transformationsprogramm schreiben; aber nicht ohne die details der ZEX-regeln..

doch davon ein anderes mal..

übrigens: claudios fractalaufgabe rechnet bei mir 2h:38min bei einem radius von 5. das ist sicher nicht so schnell, wie eine flotte assembleroutine, - (her damit) aber es ist in turbo und rechnet also auf 10 stellen genau.

programme, die blß auf 4 stellen rechnen sind für tiefe zooms nicht genug! und die tollen beschleunigungstricks ( von 200min auf 20 sec) aus der c't 3/87 sind natürlich (leider) nur bluff: die grundfigur ist ja nur einmal zu rechnen, wenn überhaupt, für die interessanten ausschnitte ist der erforderliche test sogar ein gehöriger ballast!

ach ja: make macht files.

in einer steuerdatei werden, die zur erzeugung nötigen abhängigkeiten festgehalten, dann bringt ein aufruf: make oder make datei einen batchlauf in gang der mit der meldung up to date endet.

die steuerdatei enthält gruppen der form:

vergleichszeile

TAB TAB befehlszeile(n)

die vergleichszeilen haben die form:

zieldatei TAB liste von vergleichsdateien

das kriterium ist das alter von ziel gegen die liste: ist ziel älter als eines der files in der liste, so werden die befehle ausgeführt, bis ziel 'up to date' ist. (folglich muß ziel von wenigstens einem befehl getoucht werden, sonst loopt's!)

zweck: komplexe assemblierungen effektiv und sicher zu steuern:

genau das veraltete neu zu übersetzten bzw zu binden.

ich kenne auch jemanden, der make zum zweck von tagessicherungen einsetzt.

-v schalte den verbose-mode an (mitteilsam)

-z scheint eine endlosschleife zu bewirken (zweck unbekannt)

solche unix-tools sind natürlich ideal in festplatten/multiuser-umgebungen.

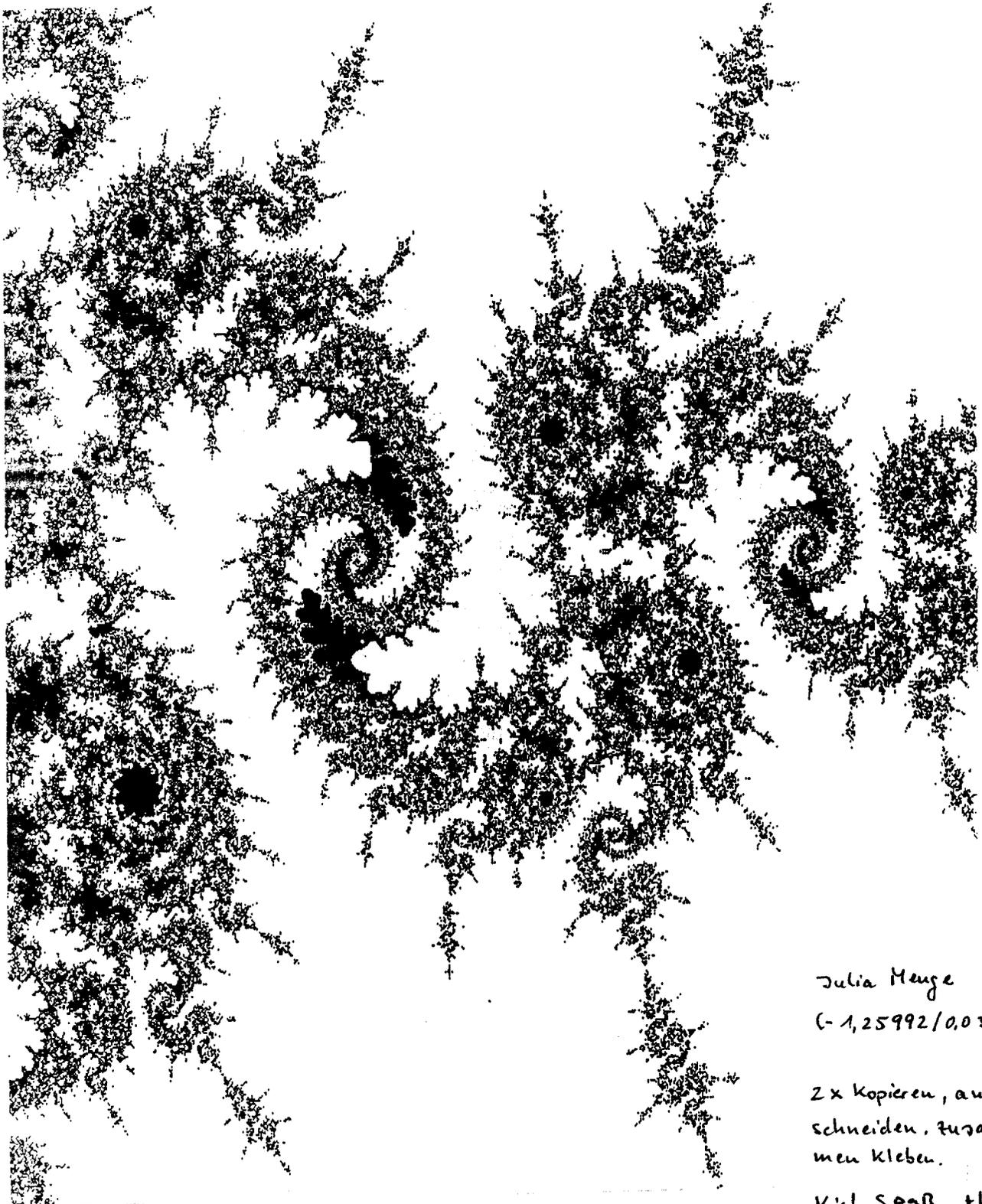
zuguterletzt: neue arbeit hab auch: in cics-cobol für ne druckerei.

(IchBinMüde)

(GONG)

na denn, bis bald! thomas wulf

G R A F I K: Ein Fractal / C L U B: Kritik



Julia Menge  
(-1,25992/0,035)

2x Kopieren, aus-  
schneiden, zusam-  
men kleben.

Viel Spaß Thomas

Was ich (Klaus Muerling, 8702) in den Info's vermisse, sind (BASIC-) Routinen zur Erzeugung von schönen Grafiken auf dem VS 4. Weißt Du, so nicht allzu lange Programme mit mathematischem Background, die man so zum Entspannen mal eben reinklopft und sich dann freut, was da auf dem Bildschirm so dabei entsteht. Natürlich nicht so Riesenberechnungsprogramme wie Apfelmännchen, wo sich stundenlang nichts tut.

BASIC: Kamikaze / SAVE, LOAD, VERIFY in Assembler

ABSTURZ MAL ANDERS: KAMIKAZE?

Peter Würfel hat im Info 18, Seite 16 behauptet, bei der Ausführung des folgenden BASIC-Programms stürze der Rechner ab.

```
10 REM Kamikaze
20 INPUT H
30 LET H$="( "+RIGHT$(STR$(H)),LEN(STR$(H))-1)+"")"
```

Das stimmt. Neugierig wie ich nun mal bin, wollte ich wissen, wieso. Erstmal ein Blick auf die absturzverursachende Zeile 30: Wieso ist der erste STR\$-Ausdruck vollständig geklammert?

```
30 ... .....(STR$(H)).....
```

Entfernt man diese überflüssigen Klammern, tut der Rechner (bzw. BASIC-Interpreter) genau das, was er soll!

```
30 LET H$="( "+RIGHT$(STR$(H),LEN(STR$(H))-1)+"")"
```

Glück gehabt. Stocherei überflüssig. Merke: BASIC bzw. FDXB ist so dumm, daß es schon weh tut. Bevor Du bei der Programmeingabe auf die RET-Taste drückst, entferne alle Klammern um Ausdrücke, die nichts zur Sache tun. Schärfe Dir ein, daß ein noch so gutes Programm aus heiterem Himmel verschluckt werden kann. Bevor Du RUN tippst und auf RET drückst: DISC SAVE "...".

AV

**SAVE, LOAD und VERIFY im Assembler**  
von Rafael Künzli

Um einen MERGE-Befehl zu programmieren ist das Call #AAE am besten geeignet. Folgende Registerpaare müssen definiert werden:

- HL - Startadresse des (z.B. zu ladenden) Speicherblocks
- DE - Länge des Blocks in Bytes.

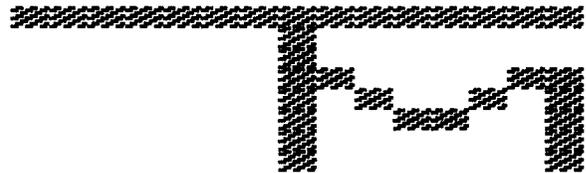
Die Systemvariablen #FD67 und #FD68 bestimmen die Funktion des Calls, und zwar wie folgt:

	#FD67	#FD68
SAVE	0	0
LOAD	0	1
VERIFY	1	0

Diese Variablen müssen also je nach Funktionswunsch und nach obigem Schema definiert werden. Sie behalten ihre Aufgabe auch beim FDX/SDX-System!

```
Beispiel: LD A,0
          LD (#FD68),A
          LD (#FD67),A
          LD HL,#8000
          LD DE,500
          CALL #AAE
```

Diese Beispiel-Programm speichert den Speicher-Bereich von #8000 bis #8000+500 auf Cassette.

B A S I C: Micro-Assembler / Interrupt-Hardcopy

**Thomas Mäurer**  
 Duisburger Str.296 - 4200 Oberhausen I

Tel. 0208/20883 - Konto Nr. 10.086.668  
 bei Sparkasse Oberhausen (Blz. 365.500.00)

Firma / Herrn / Frau / Frä.

Ihre Nachricht vom

Oberhausen, den

MTX User Club  
 Herbert Herberg  
 Sonnenau 2  
 2000 Hamburg 76

28.2.1987

Lieber Herbert, liebe MTX-ler!

Wie Ihr seht, habe ich eine kreative Phase gehabt. Naja, wenn man nach einigen nützlichen Dingen in den Infos leider vergeblich sucht, dann erfindet man das Rad halt zum millionstenmal neu (womit ich die kleine RELOC-Routine meine. Sowas ähnliches hat mit Sicherheit schon jemand aus dem Club entwickelt, aber wenn man keine FDX hat, braucht man sowas anscheinend nicht (!????)).

### Edasm

Bevor ich nun anderen evtl. Hilfe anbiete, komme ich erstmal zu meinen eigenen Problemen. Vor einigen Wochen habe ich bei Christian Wöhlbier einige Programme bestellt, von denen ich einen Teil besprechen werde. Darunter war u.a. ein sogenannter "Macro-Assembler". Dieses Programm (genauer Name: "Edasm") vom englischen "GENPAT"-MTX-Club entpuppte sich alsbald als Micro-Assembler, mir ist es bis jetzt jedenfalls noch nicht gelungen, so elementare Z 80 - Befehle wie "JP", "JR" und "CALL" anzuwenden. Wer mit diesem Assembler schon gearbeitet hat und weiß wie es geht, nehme bitte mit mir Kontakt auf.

(Thomas Mäurer)

### Interruptbetriebene Hardcopy

Diese Routine, die Ihr hier im Info findet, wird mit (RUN) gestartet in den Arbeitsbereich verschoben und dann mit *RAND USR(61423)* initialisiert (Initialisierung auch notwendig, wenn ein weiteres Programm geladen wurde). Aufgerufen wird das Programm mit der ESC-Taste. Gedruckt wird jeweils der augenblicklich vorhandene Bildschirm (VS 4 oder VS 5). Beim VS 4 müssen nach dem Drücken der ESC-Taste zusätzlich noch die Werte für den Vergrößerungsfaktor (1 bis 3, bei DMX nur bis 2) und Dichte (1 bis 2) eingegeben werden. Die eingeklammerte Passage muß bei zu wenig Speicherplatz weggelassen werden, der Computer treibt hier RAM-Banking! Mit diesem Programm kann man ausgezeichnet Hardcopies aus dem PANEL heraus machen!

Wer nicht soviel eintippen möchte, kann diese und die beiden anderen Utilities aus diesem Info auch gratis bei mir bestellen (natürlich + DM 5.- für

D&amp;P(U)

B A S I C: Interrupt-Hardcopy

```

CODE LD DE,#EFEF T3: CALL #79 NEXBYTE:LD A,(HL)
LD HL,ANFANG JR Z,T3 LD B,C
LD BC,#200 SUB #31 LD D,0
LDIR JR C,T3 BIT: SRL A
RET CP 2 DJNZ BIT
ANFANG: LD HL,#FA98 JR NC,T3 DB #DD,#44
LD (HL),#C3 INC A VG: PUSH AF
INC HL DB #DD,#6F RL D
LD (HL),0 XOR A POP AF
INC HL OUT (2),A DJNZ VG
LD (HL),#F0 LD C,24 DB #DD,#45
LD HL,#FD5E OUT (2),A DRUCK: IN A,(4)
LD (HL),79 ( LD A,1 CP 10
OUT (0),A ) JR NZ,DRUCK
LD A,#FD LD DE,#A700 LD A,D
OUT (5),A LOOPA: LD B,0 OUT (4),A
IN A,(5) LOOPB: IN A,(1) IN A,(0)
CP #FE LD (DE),A IN A,(4)
RET NZ INC DE DJNZ DRUCK
DI DJNZ LOOPB DEC HL
LD A,1 DEC C DEC E
LD (#FD75),A JR NZ,LOOPA JR NZ,NEXBYTE
RST 10 DB #DD,#44 LD DE,248
DB #82,#1B,#40 LD HL,0 AND A
LD HL,(#FF5B) LD DE,192 SBC HL,DE
BIT 5,(HL) MUL: ADD HL,DE PUSH HL
JR NZ,T1V DJNZ MUL LD DE,#A700
XOR A EX DE,HL SBC HL,DE
OUT (2),A LD HL,0 POP HL
LD A,#1C DB #DD,#45 JR NC,LOOP1
OUT (2),A LD A,B JR NZ,LOOP1
LD C,24 MUL1: ADD HL,DE DEC C
LD B,40 DJNZ MUL1 JR NZ,NEXLIN
DRUCKA: IN A,(4) ADD A,#4A POP HL
CP 10 LD IY,#F0C9 LD DE,8
JR NZ,DRUCKA LD (IY+0),A ADD HL,DE
IN A,(1) LD (IY+1),L EX DE,HL
OUT (4),A LD (IY+2),H LD HL,#BEFF
IN A,(0) XOR A SBC HL,DE
IN A,(4) DB #DD,#45 EX DE,HL
DJNZ DRUCKA DB #DD,#5C JR NC,NEXSPALT
RST 10 MUL2: ADD A,E XOR A
DB #82,#D,#A DJNZ MUL2 RST 10
DEC C DB #DD,#6F DB #82,#1B,#40
JR NZ,NEWS DB #DD,#7C LD (#FD75),A
XOR A ADD A,A OUT (0),A
LD (#FD75),A LD (#F0BD),A EI
EI RETI
RET DB #83,#1B,#33 RET
T1V: CALL #79 ZEILABS:DB 0
JR NZ,T1V LD HL,48647
T1: CALL #79 NEXSPALT:PUSH HL
JR Z,T1 LD C,8
SUB #31 NEXLIN: RST 10
JR C,T1 DB #86,#D,#A,#1B
CP 3 DICHT: DB 0,0,0
JR NC,T1 POP HL
INC A PUSH HL
DB #DD,#67 LOOP1: LD E,8
T2: CALL #79
JR NZ,T2
    
```

*LD A, (#FAD2)*

*Interrupt-  
betriebe  
Hardcopy*

B A S I C: Relozier-Routine

```

Ø REM *****
1 REM * *
2 REM * ===== RELOC ===== *
3 REM * *
4 REM * 1. Dieses Programm mit (RUN) *
5 REM * starten. *
6 REM * 2. Programm laden, bei dem die *
7 REM * Sprungadressen geändert *
8 REM * werden sollen. *
9 REM * 3. In die Zwei-Byte-Register *
10 REM * #F600 UND #F602 DEN ANFANG *
11 REM * DES EIGENTLICHEN ASSEMBLER- *
12 REM * PROGRAMMES UND DEN BESTIM- *
13 REM * MUNGORT EINTRAGEN. *
14 REM RAND USR(62720) AUFRUFEN. *
15 REM * *
16 REM *****
20 CODE

42A1 LD HL,RELOC
42A4 LD DE,#F500
42A7 LD BC,#100
42AA LDIR
42AC RET
42AD RELOC: LD HL,(#F600)
42B0 PUSH HL
42B1 DEC HL
42B2 LD D,(HL)
42B3 DEC HL
42B4 LD E,(HL)
42B5 POP HL
42B6 ADD HL,DE
42B7 MARKER: LD A,(HL)
42B8 INC HL
42B9 CP Ø
42BB JR Z,NULL
42BD CP 2
42BF JR Z,ZWEI
42C1 CP 8
42C3 JR Z,ACHT
42C5 CP #FF
42C7 RET Z
42C8 M3BIS5: LD DE,3
42CB ADDHL: ADD HL,DE
42CC LD E,(HL)
42CD ADD HL,DE
42CE INC HL
42CF JR MARKER
42D1 NULL: BIT 7,(HL)
42D3 INC HL
42D4 JR Z,NULL
42D6 LD B,3
42D8 ANZADR: LD C,(HL)
42D9 INC C
42DA NICHTS: DEC C
42DB JR Z,LOOP
42DD INC HL
42DE INC HL
42DF JR NICHTS

42E1 LOOP: INC HL
42E2 DJNZ ANZADR
42E4 JR MARKER
42E6 ACHT: LD DE,2
42E9 JR ADDHL
42EB ZWEI: RST 8
42EC MSB: BIT 7,(HL)
42EE INC HL
42EF JR Z,MSB
42F1 INC HL
42F2 PUSH HL
42F3 LD HL,(#F602)
42F6 ADD HL,DE
42F7 EX DE,HL
42F8 POP HL
42F9 LD C,(HL)
42FA INC HL
42FB INC C
42FC NOCHMAL: DEC C
42FD JR Z,ANZREL
42FF PUSH DE
4300 RST 8
4301 LD IX,(#F600)
4305 ADD IX,DE
4307 POP DE
4308 LD (IX+0),E
430B LD (IX+1),D
430E JR NOCHMAL
4310 ANZREL: LD C,(HL)
4311 INC HL
4312 INC C
4313 NOCH: DEC C
4314 JR Z,MARKER
4316 INC HL
4317 INC HL
4318 JR NOCH
431A RET

Symbols:
RELOC 42AD MARKER 42B7
NULL 42D1 ZWEI 42EB
ACHT 42E6 M3BIS5 42C8
ADDHL 42CB ANZADR 42D8
NICHTS 42DA LOOP 42E1
MSB 42EC NOCHMAL 42FC
ANZREL 4310 NOCH 4313

```

*T. Mänter*

B A S I C: REM-Entferner

```

0 REM *****
1 REM *
2 REM * 1. Dieses Programm mit "RUN" *
3 REM * starten. *
4 REM * 2. Programm laden, aus dem die*
5 REM * REMs entfernt werden sollen. *
6 REM * 3. RAND USR(54000) ausführen. *
7 REM *
9 REM *****
10 CODE
    
```

*T. Mauerer*

416F	LD HL,ANFANG	41C5	LD HL, (#FAA7)
4172	LD DE, #F736	41C8	SBC HL, DE
4175	LD BC, #100	41CA	LD (#FAA7), HL
4178	LDIR	41CD	LD HL, (#FAAC)
417A	RET	41D0	SBC HL, DE
417B ANFANG:	DI	41D2	LD (#FAAC), HL
417C	CALL #27FB	41D5	LD HL, (#FACC)
417F	LD IY, (#FAAA)	41D8	SBC HL, DE
4183 ANF:	PUSH IY	41DA	LD (#FACC), HL
4185	POP IX	41DD	LD HL, (#FAD6)
4187 REM:	LD E, (IX+0)	41E0	SBC HL, DE
418A	LD D, (IX+1)	41E2	LD (#FAD6), HL
418D	AND A	41E5	PUSH IY
418E	PUSH IX	41E7	POP HL
4190	POP BC	41E8	PUSH IX
4191	LD HL, (#FAA7)	41EA	POP DE
4194	SBC HL, BC	41EB	LDIR
4196	JR Z, ENDE	41ED	PUSH IX
4198	AND A	41EF	POP IY
4199	ADD IY, DE	41F1	JR REM
419B	LD A, #80	41F3 ENDE:	EI
419D	CP (IX+4)	41F4	RET
41A0	JR NZ, ANF		
41A2	LD HL, (#FACC)		
41A5	LD BC, (#FAAA)		
41A9	INC BC		
41AA	ADD HL, BC		
41AB	PUSH IY		
41AD	POP BC		
41AE	SBC HL, BC		
41B0	JR Z, ENDE		
41B2	PUSH HL		
41B3	POP BC		
41B4	LD HL, 0		
41B7	AND A		
41B8	SBC HL, BC		
41BA	JR Z, ENDE		
41BC	AND A		
41BD	LD HL, (#FAA4)		
41C0	SBC HL, DE		
41C2	LD (#FAA4), HL		

Symbols:			
ANF	4183	REM	4187
ANFANG	417B	ENDE	41F3

F O R T H: auf Cassette

(Thomas Mäurer)

Fig-Forth auf Cassette / Patches

Wer bei Christian Wöhlbier dieses Programm bestellt hat und nicht unbedingt dazu noch eine englische Tastatur kaufen möchte (deren Preis möglicherweise den Zeitwert des MTX übersteigt) bzw. das Gedächtnis eines Blindschachspielers besitzt, um sich die sehr veränderte Tastaturbelegung einzuprägen, kommt um einige Patches nicht umhin. Unter XBASIC läuft die Sache noch einigermaßen angenehm:

1. XBASIC laden.
2. Das Forth-Programm befindet sich bekanntlich in zwei Teilen auf der Cassette, wovon der erste Teil lediglich die Aufgabe hat, den zweiten zu laden. Wir spulen also die Cassette zum zweiten Teil.
3. Damit das Maschinenprogramm nachher auch mit (RUN) gestartet werden kann, schreiben wir folgende BASIC-Zeile:  
`10 RAND USR(16641) : RAND USR(16645)`
4. Mit `USER CREATE,20,"2100"` bilden wir nun eine Assemblerzeile 20 mit der Länge #2100.
5. Mit dem Befehl `USER BLOCK,1,"4100","2000"` laden wir nun das Maschinenprogramm von der Cassette in unsere Assemblerzeile.
6. Nun ins PANEL:  
Geändert werden müssen folgende Passagen:  
von #5466 bis #5473 (das ist das nicht funktionierende  
?TERMINAL )  
von #547B bis #5484 (das ist die Forth-Routine KEY )  
von #5F43 bis #5F4B (das ist INKEY\$ )  
Die Änderungen gehen aus dem folgenden HEX-DUMP der korrigierten Version hervor:

```

5466 : 0 . 0 . 0 . CD .
546A : 79 y 0 . 0 . 0 .
546E : 0 . 21 ! 0 . 0 .
5472 : 0 . 0 . 28 ( 1 .

547B : 0 . 0 . 0 . CD .
547F : 79 y 0 . 0 . 0 .
5483 : 0 . 0 . 28 ( F3 .

5F43 : 0 . 0 . 0 . CD .
5F47 : 79 y 0 . 0 . 0 .
5F4B : 0 . 6F o 26 & 0 .

```

Das gepatchte Programm wird nun ganz normal abgespeichert. Wer sich diesen recht erheblichen Aufwand sparen möchte, kann mir auch einfach seine Original-Forth-Cassette zusenden. Er erhält dann die reparierte Fassung mit noch einigen zusätzlichen Features, z.B. dem neuen Befehl "PR", der es erlaubt, wahlweise die Ausgaben auf den Bildschirm und (!) / oder auf den Drucker zu lenken, möglicherweise einen anderen Editor (in Arbeit) bzw. eine "VERIFT"-Funktion, etc. , gegen eine Aufwandsentschädigung von DM 5.-

So, jetzt kann mich Herbert in die Rubrik "Wer tut was" aufnehmen.

S P I E L E: KritikSoftware: Spiele / Kritik

(Thomas Mäurer)

Wie ich schon an anderer Stelle erwähnte, habe ich bei Christian W. einige Programme bestellt. Da ich selbst schon in einem früheren Info angeregt hatte, mehr Software-Besprechungen (insbesondere Cassetten) ins Info zu bringen und ich mit dem Wunsch nicht allein stehe, bietet sich mir nun die Gelegenheit zu einem ordentlichen Verriß. Vorweg sei klargestellt, daß die Erwartungen an die Software bei den kleinen Preisen sicherlich nicht zu hoch sein sollten, jedoch muß das ja nicht bedeuten, daß billige Programme automatisch schlecht sind, im Gegenteil: "Murder at the Manor" z.B., eines der teuersten Programme, entpuppte sich als Niete.

Das eben genannte Programm darf man wohl mit "The man from Granny" und "Caves of Orb" in einem Atemzug nennen.

Allen drei Spielen gemeinsam ist, daß es sich hier (sieht man von den Titelbildern beim Laden ab) um reine Text-Adventure, die ich als dem Medium Computer nicht adäquat empfinde, handelt. Sie mögen vielleicht für Kinder, die sich in der englischen Sprache üben sollen, ganz nett sein, sind aber auch zu diesem Zweck nur bedingt brauchbar, denn die deutsche Tastatur wurde mal wieder nicht berücksichtigt. Alle drei Spiele sind - wegen der fehlenden Graphik - wenig ansprechend.

Nun zum positiven Teil: den Spielen "Escape from Zarcos", "Goldmine", und "Firehouse-Freddie".

Alle drei Spiele gehören der Kategorie "Arcade-Spiele" an, es sind keine Kriegsspiele. Es geht also nicht darum, die Erde zu "verteidigen", Haie zu ermorden, feindliche Stellungen zu erobern. (Da fällt mir gerade ein äußerst abschreckendes Beispiel zu ein: Für den ZX 81 oder den Spectrum von Sinclair gibt es doch tatsächlich ein BASIC-Spielebuch mit einem Spiel, bei dem die Aufgabe darin besteht, möglichst viele Fußgänger mit dem Auto zu überfahren, und das in einer Stadtbücherei!)

In den oben genannten Spielen hat der Spieler stattdessen die (sinnvollere) Aufgabe, Teile eines Raumschiffes, das bei einer Bruchlandung auf einem fremden Planeten auseinandergefallen ist, einzusammeln ("Escape from Zarcos"), Gold aus einer Mine zu holen ("Goldmine") oder mit Hilfe eines Schlüssels aus einer Falle mit gefährlichen Monstern zu entkommen. All' diese Spiele finden auf mehreren Bildern statt, Graphik und Animation reizen die technischen Möglichkeiten, die der MTX bietet, voll aus. "Escape from Z." bietet obendrein noch einen recht guten Sound, der sich aber auch abstellen läßt (!). "Esc. f. Z." und "Firehouse-Freddie" ähneln sich in einigen Details, z.B. der Idee mit den Fließbändern oder dem Springen mit der HOME-Taste, sodaß sich einem der Verdacht aufdrängt, daß sie vom selben Autor stammen, was aber nicht bedeutet, daß man sich nicht beide kaufen soll.

Alle drei Spiele sind abwechslungsreich und ihr Geld voll und ganz wert, für Vielspieler (zu denen ich mich allerdings nicht zähle) besteht bei "E. v. Z." - wenn dieses Spiel mit dem Joystick betrieben wird - die Gefahr einer Bänderdehnung im Handgelenk (die Sprünge haben es in sich!). Zum gleichen Spiel sollte schließlich noch erwähnt werden, daß es (auch wenn auf der Hülle Gegenteiliges vermerkt ist) auf dem MTX 512 nur mit vorherigem *POKE 64122,0: NEW* läuft. Das ändert aber nichts daran, daß es nach meiner Meinung das beste ist.

NewWord: Adreßverwaltung**Adreßdateien für Briefe in NewWord nutzen** (Herbert Herberg, 2000)

Kürzlich sah ich, wie Olaf Krumnow Briefe in NewWord erzeugt. Ich habe immer den Briefkopf in den Brief kopiert, den Namen und das Datum eingetragen - und so viel Platz verschenkt. Es geht wesentlich eleganter, und vor allem einfacher!!!

Die Chose besteht aus vier Teilen:

1. Dem STARTER, der Datum, Briefdatei und Nachname des Adressaten erfragt. Nach Eingabe dieser DREI Werte erscheint der Brief automatisch - vorausgesetzt, die Briefdatei existiert, und der Name ist in der Adreßdatei.
2. Die Adreßdatei PERS.DAT, in der die Adreßdaten drin stehen. Je Adressat eine Zeile mit diversen Angaben, die durch Kommata getrennt sind.
3. Dem Briefkopf BRFKOPF, der vom STARTER aufgerufen wird, die Adreßdatei absucht, und wenn der Adressat gefunden wurde loslegt. Am Ende des Briefkopfes steht ein NewWord-Kommando, welches dann den vom STARTER erfragten Brieftext anhängt.
4. Der eigentliche Brieftext, dessen Dateiname von STARTER erfragt wird.

Ich habe die Dateien von Olaf leicht gekürzt, und in *kursiv* Erläuterungen eingefügt, die nicht Teil der Dateien sind!

STARTER

```
.av "Aktuelles Datum ? (TT.MM.JJJJ)           ", DATUM
.av "Zu druckende Briefdatei ?               ", BRIEFDATEI
.av "Adressat ? (Nur den Nachnamen!!)        ", EMPF
.fi BRFKOPF
```

*In die NewWord-Variablen DATUM, BRIEFDATEI und EMPF werden die drei notwendigen Werte eingelesen, und dann wird der Briefkopf aufgerufen mittels dem NewWord-Kommando .FI (File-Insert = Datei Einfuegen)*

PERS.DAT

```
Herr,Herbert,Herberg,,Sonnenau 2,2000,Hamburg 76
Herr,Uwe,Beythien,,Lorenzengasse 1,2000,Hamburg 60
Herr,Kurt-Bernd,Rohloff,,Kafkastraße 14,8000,München 83
Herr,Ulrich,Hönisch,,Wachholtzstraße 8,3300,Braunschweig
An den,Verlag Heinz,Heise,Redaktion c't,Postf. 610407,3000,Hannover 61
An den,,DMV-Verlag,Redaktion "PASCAL int.",Postfach 250,3440,Eschwege
```

*Der Aufbau der Daten spricht wohl fuer sich. Ich habe nur den Text "PASCAL international" aus Platzgruenden gekuerzt. Die Zeilen koennen natuerlich laenger sein. Wichtig ist, dass alle Zeilen gleich viele Eintraege und Kommata enthalten.*

New Words: AdreßverwaltungBRFOKPF

```
.df pers.dat
.rv ADRESSZU, VORNAME, NAME, ZUSATZ, STRASSE, PLZ, WOHNORT
.dm "&NAME%"
.if &NAME% = &EMPF%
.pfon
.rm80
.mt0
.po0
.lm2
.lh14
```

```
.cw23
.lh2
```

Olaf Krumnow

```
.po0
.cw10
.lm0
.lh8
^S_
.lm1
.pfoff
.mt3
```

\_ ^S

Wiesnerring 19c  
2050 Hamburg 80  
( 040 ) 7249566

den &amp;DATUM%

```
.cw12
.lh9
.cw6
```

^S^Olaf Krumnow Wiesnerring 19c 2050 Hamburg 80^S

```
.cw10
.if &ADRESSZU% = Herr
    Herrn
.ei
    &ADRESSZU%
.ei
    &VORNAME% &NAME%
    &ZUSATZ%
    &STRASSE%
    &PLZ% &WOHNORT%
```

```
-
.cw12
.lh8
.po6
.rm75
.pfdis
.fi &BRIEFDATEI%
.ei
```

New Word: Adreßverwaltung / Tips

*Ich habe auch hier einige der Zeilen verkuerzt, und die Unterstreichungen herausgenommen, damit erkennbar ist, wie der Briefkopf aussieht.*

WIE PASSIERTS?

Wenn NewWord mit Merge Print arbeitet, und eine Daten-Datei mit .DF angegeben bekommt, dann wird der Text so oft verarbeitet, bis alle Daten der Daten-Datei eingelesen sind. Also wird der o.g. Brief für jede Zeile der Datei PERS.DAT abgearbeitet.

Nachdem eine Zeile der PERS.DAT eingelesen wurde, wird mit .DM der Name auf dem Bildschirm ausgegeben - damit man sieht, NewWord tut was. Nun wird der Name der vom STARTER eingelesen wurde (NewWord-Variable EMPF) mit dem eingelesenen aus PERS.DAT (NewWord-Variable NAME) verglichen. Sind sie nicht gleich, so werden alle Zeilen bis zum .EI übergangen - also der ganze Rest des BRFKOPF. Fazit: Ist EMPF nicht gleich NAME, so geht NewWord ans Ende des Textes (wegen des dortigen .EI), und fängt mit der Bearbeitung von BRFKOPF wieder oben an. Ist jedoch der Name gefunden (EMPF = NAME), dann wird direkt hinter dem .IF weitergemacht, d.h. der Briefkopf gedruckt, und mit dem NewWord-Befehl .FI &BRIEFDATEI& der eigentliche Briefftext angehängt. (STARTER hat den Namen der Datei mit dem Briefftext ja in die NewWord-Variable BRIEFTEXT eingelesen. Wenn das fertig ist, fängt NewWord wieder oben an - so lange, bis PERS.DAT abgearbeitet ist.

Dieses Verfahren ist sicherlich nicht übermäßig schnell, da für jeden Brief die gesamte Datei PERS.DAT abgearbeitet wird, aber dafür wesentlich bequemer - und einmal eingegebene Adressen müssen nicht wieder herausgesucht werden.

Übrigens könnt Ihr in Olaf's BRFKOPF noch einige Ideen finden, wie man mit den NewWord-Variablen die Ausgabe gestalten kann.

Tips

(Herbert Herberg, 2000)

**F:** Wie kann ich bewirken, daß ein Punkt ('.') in der ersten Spalte ausgedruckt wird, und nicht als NewWord-Kommando interpretiert wird?

**A:** Vor den Punkt zwei mal ^P^B (oder ähnliches) setzen.

**F:** Wie kann ich mit NewWord in einem Ersetz-String einen Zeilenumbruch einbauen?

(Anwendung: .PRN-Dateien von MSO enthalten ^L-Codes für den Anfang einer neuen Seite, hinter denen der Text direkt weiter geht. In NewWord wird das durch eine Zeile der Form

```
^L-----F
angezeigt. Diese sollen durch '.PA' Zeilen ersetzt werden.)
```

**A:** Such-String: ^P^L

Ersetz-String: .PA^P^M^P^J

**F:** Wie schalte ich den Paper-End Detektor von NewWord aus aus?

**A:** Oben im DOC das NewWord-Kommando .xE1B38

Damit bewirkt ein ^P^E, daß der Drucker ESC '8' erhält, d.h. auf Paper-Ende-Ignorieren gestellt wird. Nun muß noch sichergestellt sein, daß das BIOS nur das BUSY-Signal untersucht. Unter RAM 4 ist das immer der Fall. Wenn nicht, muß in CBIOS.ASM der Drucker-Status geändert werden. Original nach dem IN-Befehl: ANI OFH/ CPI OAH. Das muß nun geändert werden in ANI 01H/ CPI 00H. (BOBO-Assembler). Danach mit SUB CPMGEN die Änderungen auf die Systemspuren. Wie das geht stand schon mehrfach im Info!

Nun noch im DOC ein ^P^E eingeben, und der Drucker druckt sogar, wenn das Papier zu Ende ist.

New Word: NWPRINTN.OVR-PrinterOverlay von CLUB.009

### Wie verwenden, anpassen, einbauen ?

(Herbert Herberg, 2000)

Dieses Overlay zu aktivieren ist garnicht so schwer - wenn man weiß wie. Leider haben wir das auf der PD nicht deutlich gesagt. Falls Du NewWord Version 2.02 hast, ist alles kein Problem. Altes (d.h. Original-) NWPRINT.OVR von der Arbeitsdiskette löschen, und das NWPRINTN.OVR von CLUB.009 auf die Diskette kopieren. Nun noch ein

```
A>REN NWPRINT.OVR=NWPRINTN.OVR
```

damit NewWord das neue Overlay unter dem bekannten Namen findet.

Hast Du aber NewWord 2.16 oder 2.17, dann muß die Versionsnummer im NWPRINTN.OVR geändert werden, da NewWord sonst das Overlay verweigert. Diese Arbeit läßt sich problemlos mit dem guten alten DDT.COM (von der Original-Systemdisc) erledigen. Es geht auch mit DDTZ, DU, DU2, MONI, ner DDT-Session!

**Fett** gedruckte Zeichen muß Du eingeben.

Dabei ist / = RET, und ^C = CTRL-C.

```
A>ddt nwprintn.ovr/
```

```
DDT VERS 2.2
```

```
NEXT PC
```

```
AE00 0100
```

```
-d100/
```

```
0100 00 03 4E 65 77 77 6F 72 64 20 32 2E 30 32 20 20 ..Newword 2.02
0110 4F 76 65 72 6C 61 79 20 52 65 6C 65 61 73 65 64 Overlay Released
0120 20 6F 6E 20 33 31 20 4D 61 79 20 38 34 20 20 OD on 31 May 84 .
0130 0A 43 6F 70 79 72 69 67 68 74 20 28 43 29 20 31 .Copyright (C) 1
0140 39 38 33 20 4E 65 77 73 74 61 72 20 53 6F 66 74 983 Newstar Soft
0150 77 61 72 65 20 49 6E 63 6F 72 70 6F 72 61 74 65 ware Incorporate
0160 64 2E OD OA 41 6C 6C 20 72 69 67 68 74 73 20 72 d...All rights r
0170 65 73 65 72 76 65 64 2E OD OA OA 00 1A 00 00 00 eserved.....
0180 00 00 00 54 59 50 45 57 52 FE 00 OF 47 44 52 41 ...TYPEWR...GDRA
0190 46 54 FE 00 1F OE 44 31 36 31 30 FE 00 1F OE 44 FT....D1610....D
01A0 36 33 30 FE 00 1F OE 51 53 50 52 31 31 FE 00 3A 630....QSPR11...:
01B0 27 53 52 35 35 30 FE 00 55 2D 54 49 38 35 35 FE 'SR550..U-T1855.
```

```
-s10c/
```

```
010C 30 31/
```

```
010D 32 36/
```

```
010E 20 ./
```

```
-d100/
```

```
0100 00 03 4E 65 77 77 6F 72 64 20 32 2E 31 36 20 20 ..Newword 2.16
0110 4F 76 65 72 6C 61 79 20 52 65 6C 65 61 73 65 64 Overlay Released
0120 20 6F 6E 20 33 31 20 4D 61 79 20 38 34 20 20 OD on 31 May 84 .
0130 0A 43 6F 70 79 72 69 67 68 74 20 28 43 29 20 31 .Copyright (C) 1
0140 39 38 33 20 4E 65 77 73 74 61 72 20 53 6F 66 74 983 Newstar Soft
0150 77 61 72 65 20 49 6E 63 6F 72 70 6F 72 61 74 65 ware Incorporate
0160 64 2E OD OA 41 6C 6C 20 72 69 67 68 74 73 20 72 d...All rights r
0170 65 73 65 72 76 65 64 2E OD OA OA 00 1A 00 00 00 eserved.....
0180 00 00 00 54 59 50 45 57 52 FE 00 OF 47 44 52 41 ...TYPEWR...GDRA
0190 46 54 FE 00 1F OE 44 31 36 31 30 FE 00 1F OE 44 FT....D1610....D
01A0 36 33 30 FE 00 1F OE 51 53 50 52 31 31 FE 00 3A 630....QSPR11...:
01B0 27 53 52 35 35 30 FE 00 55 2D 54 49 38 35 35 FE 'SR550..U-T1855.
```

```
-^C
```

```
A>save 173 nwprintn.ovr/
```

Nun muß dieses NWPRINTN.OVR genauso wie oben beschrieben auf die New-Word-Arbeitsdiskette kopiert und umbenannt werden.

New Word: NWPRINTN.OVR-PrinterOverlay von CLUB.009

Willst Du nun drucken, tust Du es einfach - aber nicht die ESC-Taste drücken, sondern mit RET, RET, ... solange im Print-Menü vorgehen, bis Dir unten die zur Auswahl stehenden Drucker angezeigt werden. Für den DMX80 gibt es 2 Treiber:

P1090: Kann Microspacing, keine Grafik

DMX80: Kann kein Microspacing, aber Grafik und durchgehendes Untersteichen (wenn im Document ein .ul on steht).

Damit nun der gewünschte Drucker voreingestellt ist, muß NWINSTAL her! Aber leider kennt NWINSTAL nicht die neuen Treiber, also geht's ins Menü H - Special Patches, und dort in das, bei dem unter Auswahl A der Printer Signon Text genannt ist. Dann: s.u.

(Beachte bitte, daß ich Version 2.16 habe, also die Menüs evtl. etwas anders aussehen. Wo ... steht, habe ich Ausgaben von NWINSTAL nicht wiedergegeben - wozu auch die Menüs alle vollständig wiedergeben?)

Die Session unten geht davon aus, daß der eingestellte Drucker DRAFT ist, und der Drucker DMX80 eingebaut werden soll.

A>nwinstal nw/

NWINSTAL 28 May 85  
Copyright 1983,1984 Newstar Software Incorporated.  
All rights reserved.

Memotech Version

## MAIN MENU

A	Terminals	E	Document related items
B	Printers	F	File related items
C	Computer related items	G	Fresh user area
D	Newword related items	H	Special patches
J	Help with this menu	X	End of installation

What is your choice? H

## PATCH MENU #1

What is your choice? 3

New Word: NWPRINTN.OVR-PrinterOverlay von CLUB.009

PATCH MENU #3

- A PRNID - Printer signon text
- M ULUNI - Printer uniniti...
- ...
- 1 Go to Patch Menu #1
- 2 Go to Patch Menu #2
- J Help with this menu
- 4 Go to Patch Menu #4
- X Return to Main Menu

What is your choice? **A**

While patching:

- X - Return to Patch Menu
- . - End of changes, re-display
- RETURN - Leave current byte unchanged; advance to next location
- ' - Enter the next character in ASCII

0-9, A-F - Enter the hex digit

```

0534 44 72 61 66 74 20 50 72 69 6E 74 65 72 0D 0A 00 Draft Printer...
      44 72 61 66 74 20 50 72 69 6E 74 65 72 0D 0A 00 Draft Printer...

0544 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
      00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....

0554 00 00 00 00 00 00 00 00 44 52 41 46 54 00 00 20 .....DRAFT..
      00 00 00 00 00 00 00 00 00 'D 'M 'X 'B 'O 00 X

```

PATCH MENU #3

...

What is your choice? **X**

MAIN MENU

What is your choice? **X**

All changes have been made.

Newword is now installed for...

...

Do you want to change this? Y/N **N**

Es erscheint auch weiterhin der Text 'Draft Printer', den Ihr sicherlich oben schon entdeckt habt, aber der Treiber DMX80 wird genommen. Wer will kann auch gerne die Text ändern.

Leserbrief: Uwe Groos, 6393

HALLO HERBERT  
-----

ALSO...

Seit ich dir das letzte (und zugleich auch erste) mal geschrieben habe, ist ja wohl einige Zeit vergangen..

Aber das ist ja wohl kein Wunder !

Du glaubst nicht, was ich für Augen gekriegt habe, als der Stapel Infos vor meinem Gesichtsfeld lag. Oh mann, das erste was ich gedacht habe war "????????".

Also, was macht man nun mit sowas ?

Man nehme einen großen Ordner, loche alle Seiten und hefte sie sorgfältig ein ! Und dann ?? Dann hab' ich Schussel einen Monat nicht reingeguckt (Das ärgert mich heute noch !).

Aber dann....

Dann fängt man ja an und denkt....

Dann denkt man so..."Warum hast du das Dingens überhaupt..?"

Also reingucken...zumal ich das dumme Problem hatte, das ich ein Spiel, genannt "IRON" (noch in Basic), unbedingt für zwei Joysticks lauffähig haben wollte. Und, siehe da, gleich auf den ersten Seiten fand ich schon ein entsprechendes PROGRAMM ! Ei, wie find' ich denn das ?

Dann wurde es auch irgendwie nötig, das ich mal NEWWORD benutzen mußte...wobei ich festgestellt habe: Feines Programm !

Übrigens...für SUPERCALC hab' Ich noch keine Verwendung gefunden.

Tja und dann kam für mich die Erleuchtung. ---> \* PASCAL \* <---  
Und da bin ich jetzt am Basteln...

Ich habe für das Pascal-Programm leider nur das englische Manual, durch das ich mich mühsam kämpfe. Noch etwas zur Installierung von Turbo-Pascal: Ich versuche, wenn es nicht mehr allzulange dauert, alle installierungscodes (CTRL-SEQUENCES) in einen Ausdruck zu bringen, da dies im INFO (NR.?) etwas ungünstig beschrieben ist.

Außerdem habe ich dort teilweise Änderrungen durchgeführt.

Apropos Änderungen...

- Geändert habe ich:
1. Den Kompletten Aufbau des Floppy-Laufwerks
    - 1.1 Netzteil für Terminal auf Rückwand
    - 1.2 Netzschalter auf Rückwand
    - 1.3 Anzeige für 0V, +5V und -5V anstelle des Netzschalters
    - 1.4 Entfernen der selbstsichernden Muttern und ersetzen durch Einschlagmutter  
(Mutter ist ins Material eingeschlagen)
    - 1.5 Trennen der Auflage für Floppy-Laufwerke von der Frontplatte und befestigen am Gehäuseboden
    - 1.6 Ändern der Rückwand:
      - 1.6.1 Möglichkeit für 3 Centronics-Ports, (Momentan 1 belegt)
      - 1.6.2 Bus Expansion Port (Kontron ECB-Bus)
      - 1.6.3 8-Zoll Laufwerk (Nicht belegt)
      - 1.6.4 5 1/4-Zoll Laufwerk (Nicht belegt)  
(Alles auf neuer Alu-Platte.)
  2. Einbau von Zweitlaufwerk
  3. Einbau eines Druckerspooles (aus c't 6/85)  
(Centronics von Terminal an Spooler gelegt)
  4. Einbau eines Piezzo-Piepsers (FDX-So.Out)

Leserbrief: Uwe Groos, 6393

Des weiteren habe ich mir die Mühe gemacht und  
 1.auf altbewährte Weise meine Joy-Sticks umgebaut, da diese, wie wohl bekannt sein dürfte, nach einiger Zeit nur noch schalten, wenn es ihnen ein unwiderstehlicher Drang gebietet.  
 2.haben mir meine beiden Monitore in ihrem alten Kleid nicht mehr gefallen, also neues Gehäuse im Memotech-Look entwickelt und gebaut !  
 3.habe ich einen nagelneuen Prommer, fast c't, Platine etwas geändert und somit nur noch einseitig mit Leiterbahnen versehen.  
 4.habe ich mir eine I/O-Karte gebaut (wie Prommer mit ECB-Bus), von dem ich Schaltplan, Steckerbelegung u.s.w. im nächsten Brief zum Ausdruck bringe. (sieht momentan etwas wild aus!).

Das sind momentan ausgeführte Anderrungen...

Geplant ist in der nächsten Zeit dann ein zweites Gehäuse im Floppy-Look, als Träger für alle weiteren Bus-Platinen wie z.B. den Prommer...Wer kann mir einen Tip geben wo ich diese Alu-Gehäuse beziehen kann?

Am Basteln bin ich auch an einer Bildabtastung, wie in einem eurer INFOS beschrieben (Druckermäßig mit Fototransistor und so), habe eine andere Analog-Auswertung gebaut, experimentiere jetzt mit verschiedenen Fototransistoren.

Irgendwann will ich mir dann einen Plotter selbst bauen, Stabiler als die im Handel Erhältlichen, dann mit Möglichkeiten der Bildabtastung und auch zum Platinenbohren. Natürlich sollte er dann auch noch Normale Plottertätigkeiten ausführen können. Beabsichtigt ist eine Absolute Verfahrwegmessung (Maschinenlesbarer Maßstab auf der X und Y-Achse).Momentan experimentiere ich mit der Ansteuerung für die Schrittmotoren.

Wer zu den bis jetzt genannten Tätigkeiten noch irgendwelche Fragen hat, soll mir doch bitte bescheid geben, da die Anleitungen zu den einzelnen Umbauaktionen doch etwas komplizierter sind (z.B. die zur Floppy). Für Informationen oder Anregungen im Bezug auf die geplanten Aktionen hab' Ich natürlich immer ein offenes Ohr.

Wer dann auch noch Interesse am TRON-Spiel hat, dem sei gesagt, es ist noch nicht vollendet... näheres, wenn jemand nach dem Listing fragt.(habe sowieso vor es in Pascal umzuschreiben !..und dann auch auf internationalen Standard (80 Z-Monitor))

Das also zu dem !

So...jetzt muß ich auch mal was Fragen....

Und zwar...

Wer kann mir helfen ein BIOS, für Memotech, CP/M 3.0 zu bekommen (wenn möglich mit erklärtem Listing). Weiterhin gab's bei CP/M 3 schon ein Problem besonderer Art. Alle Programme von CP/M 2 sollen auf dem 3er lauffähig sein. Denkste ! NEWWORD's Installer läuft nur auf CP/M 2 ! Mei Bruder hat das nämlich auf seiner Mühle (ein GRIP von Conitec) versucht, mit dem Resultat, daß alles wunderbar lief, aber später nicht übernommen wurde. Kann mir in der Beziehung einer einen Tip geben, an was das ligt ?

Problem Zusatzspeicher: Wie immer, wenn man bei VOBIS kauft, Speicher nur für 500er, passt nicht an 512er ! Also Decodierung drangehängt (selbstgebastelt)...nur...Alles OK, laut Anleitung erscheint im Basic ohne Floppy eine 2,3 oder 5...doch mit Floppy kommt se nich ! Was kalappert da wieder nich oder was muß man ändern (ROM in Floppy oder im MIX) ??

Und dann gibt's da noch was...Teilweise spinnt meine Tastatur ! Es sind so ca. zwei Tastenreihen längs, die manchmal nicht gereinigt sind Buchstaben zu liefern. Erst wenn man relativ fest gedrückt hat, erscheint ein Buchstabe!

Leserbrief: Uwe Groos, 6393

Außerdem ist die Tastatur manchmal Pausenlos, was heißt, daß ich Zeichen doppelt und dreifach vorgezaubert bekomme, was durch den gleichen, enormen Tastendruck behoben werden kann.

Ich schätze Platinenbruch oder Wackelkontakt.

Wer kennt sich da aus ??

Dann hab' Ich noch ein größeres Problem, was sich heißt: MS/DOS  
Wer kann mir Infos über MS/DOS besorgen ? Wer weiß, wie es funktioniert ? Wer weiß die Unterschiede zwischen CP/M und MS/DOS. Wie kann CP/M mit MS/DOS zusammenarbeiten (Dateizugriff, READ, WRITE u.s.w.). Wer kennt entsprechende Emulationsprogramme (Möglichst in MS/DOS-format geschrieben), Wo kann ich sowas erstehen (wenn Möglich für SONY-PC's oder IBM-Compatibel) Also wenn da jemand Ahnung hätte und mir schreiben würde, das wäre wirklich GREAT !!!!

Übrigens hab' auch ich mein EPROM (80 Zeichen-Karte) geändert...  
In NEWWORD lade ich darum gleich den Zweiten Zeichensatz ein, da ich im Normalen Zeichensatz ä,ö und ü durch die Normalen ASCII-Codes ersetzt habe. Auf Wunsch kann ich das ROM-Listing rausrücken, auf ganz besonderen Wunsch Sogar mit den entsprechenden Buchstaben, Pixelmäßig angegeben.

Noch ein kleines Ärgernis. Beim einschalten von die Computer gibts komisches Ton aus die Lautsprecher, der nicht geht aus ! Also fragt man sich ja...was tun ? Ich löste das bis vor kurzem immer so, das ich FDXB lud (Ton geht aus) und dann die beiden hübschen Tasten neben Space betätigte ..!.. oder ich zog das Kabel aus dem Hi-Fi-Ausgang am MIX...! Das war natürlich nicht die feine englische Art. Also habe ich ein Programm erstellt, das die Soundkanäle des Grundgerätes auf 0 setzt. Das Listing habe ich hinten angehängt

Eigentlich wollte ich diesem Brief auch noch die Einstellung für PASCAL (Installer) mit den Dazugehörigen Fragen (da ich mit dem Listing im INFO Schwierigkeiten hatte....man hat sich vertippt) beilegen, doch ich weiß nicht, ob das noch klappt.

Was wollte ich eigentlich sonst noch ??

Ja ! eines wollte ich auf jeden Fall noch mal loswerden: Ich finde es erstmal Klasse, das der USERCLUB überhaupt existiert und daß das alles so super läuft.

Dank euch hab' ich beim MIX erstmal durchgecheckt. Übrigens haben wir ja ein doofes Disketten-Format..vergl. wo Directory steht.. Mits Computer von mein Bruder kann ich nur über ECMA-70 Format fahren. Aber ansonsten bin ich mit dem Hobel doch relativ zufrieden. Dann noch eins: Bei euch gibts doch so einen VDP-Prozessor Experten. Kann man da mal Unterlagen über den TEXAS-Prozessor bekommen ? Will mal sehen, was der so kann.

Und nochwas: Wer weiß, wie ich die Buchstaben (ö,ä,ü und ß) passend auf meinen Drucker bekomme ? Ich habe leider nicht so einen Luxusdrucker, auf dem man den Deutschen Zeichensatz einstellen kann (großes Problem in NEWWORD) !

Auf jeden Fall hoffe ich auf Gute Zusammenarbeit und mindestens einer Antwort auf meine Fragen..

Also...

Bis demnächst...

Uwe Groos  
zur Burg 3  
6393 Wehrheim 1

TURBO: Installation

(Uwe Groos, 6393)

HIER DAS LISTING FUER TINST  
(Installer für Turbo-Pascal)

=====

TURBO Pascal installation menu.

Choose installation item from the following:

(S)creen installation            (C)ommand installation            e(X)it

Enter S, C, or X:

Eingabe nach Bedarf...

SCREEN INSTALLATION : sind Informationen für die  
Bildschirmbearbeitung wie zum Beispiel der Code für Clear Screen,  
für Grundeinstellung (bei mir 1.Zeichensatz mit vollem ASC-II-  
Code), Taktfrequenz des Prozessors und so weiter...

COMMAND INSTALLATION : sind Informationen für die Bearbeitung des  
Programms über die CONTROL- und ESCAPE-SEQUENCES des Computers  
sowie der Tastaturbelegung (Eingabe der Codes von den Tasten INS  
und DEL zum Beispiel)...

EXIT : reguläres verlassen des Programms !

-----  
ERROR MESSAGES im Modus COMMAND INSTALLATION :  
-----

Duplicate definition. Error occurred between question (No.) (No.)  
- Doppelte Definition. Fehler zwischen Frage

Zwei verschiedene Funktionen mit gleichem  
Control-Character belegt. 1.Belegung auf  
erster Nummer, 2. auf zweiter Nummer.

Commands starting with the same letter must have the same length.  
Error occurred between question (No.) (No.)

- Kommandos, die mit dem selben Buchstaben beginnen, müssen  
gleich lang sein. Fehler zwischen Frage  
tritt meist bei mehrgliedrigen Kommandos  
auf wie z.B. Ctrl-B Ctrl-B für Begin Block  
Diese Funktion ist Korrekt, doch Ctrl-B  
darf nur noch in doppelten Kommandos ver-  
wendet werden. Angabe der Fehlerstelle wie  
Oben.

The total maximum Length of commands are exceeded

- Die größtmögliche Länge der Kommandos ist überschritten.  
Es können keine Kommandos mehr aufgenommen  
werden. Abhilfe kann durch Abkürzen der  
mehrgliedrigen Kommandos geschaffen werden.

->

Neue Eingabe hinter diesem Pfeil,  
<RET> übernimmt das alte Kommando

T U R B O: Installation

## COMMAND INSTALLATION

## CURSOR MOVEMENTS:

1: Character left	.....	->	Ctrl-H
2: Alternative	Nothing	->	
3: Character right	.....	->	Ctrl-Y
4: Word left	.....	->	Ctrl-F
5: Word right	.....	->	Ctrl-G
6: Line up	.....	->	Ctrl-K
7: Line down	.....	->	<LF>
8: Scroll down	.....	->	Ctrl-C
9: Scroll up	.....	->	Ctrl-R
10: Page up	.....	->	Ctrl-I
11: Page down	.....	->	Ctrl-V
12: To left on line	.....	->	Ctrl-P Ctrl-L
13: To right on line	.....	->	Ctrl-P Ctrl-R
14: To top of page	.....	->	Ctrl-P Ctrl-I
15: To bottom of page	.....	->	Ctrl-P Ctrl-X
16: To top of File	.....	->	Ctrl-Z
17: To end of file	.....	->	Ctrl-X
18: To beginning of block	.....	->	Ctrl-P Ctrl-B
19: To end of block	.....	->	Ctrl-P Ctrl-E
20: To last position	.....	->	Ctrl-P Ctrl-C

## INSERT &amp; DELETE:

21: Insert mode on/off	.....	->	Ctrl-U
22: Insert line	.....	->	Ctrl-S
23: Delete line	.....	->	Ctrl-I
24: Delete to end of line	.....	->	Ctrl-E
25: Delete right word	.....	->	Ctrl-U Ctrl-W
26: Delete Character under cursor	.....	->	Ctrl-D Ctrl-U
27: Delete left Character	.....	->	<DEL>
28: Alternative	Nothing	->	

## BLOCK COMMANDS:

29: Mark block begin	.....	->	Ctrl-B Ctrl-B
30: Mark block end	.....	->	Ctrl-B Ctrl-E
31: Mark single word	.....	->	Ctrl-B Ctrl-S
32: Hide/display block	.....	->	Ctrl-B Ctrl-H
33: Copy block	.....	->	Ctrl-B Ctrl-C
34: Move block	.....	->	Ctrl-B Ctrl-N
35: Delete block	.....	->	Ctrl-B Ctrl-D
36: Read block from disc	.....	->	Ctrl-B Ctrl-R
37: Write block to disc	.....	->	Ctrl-B Ctrl-W

## MISC. EDITING COMMANDS:

38: End edit	.....	->	Ctrl-B Ctrl-O
39: Tab	.....	->	Ctrl-I
40: Auto tab on/off	.....	->	Ctrl-Q Ctrl-I
41: Restore Line	.....	->	Ctrl-Q Ctrl-L
42: Find	.....	->	Ctrl-Q Ctrl-F
43: Find & replace	.....	->	Ctrl-Q Ctrl-R
44: Repeat last find	.....	->	Ctrl-A
45: Control character prefix	.....	->	^

T U R B O: Installation

ERROR MESSAGES im Modus SCREEN INSTALATION :

=====

Numeric entry expected  
 Erwarte Numerische Eingabe - Es wurde keine Zahl sondern ein Buchstabe eingegeben.

Legal Range is (min) (max) , please re-enter:  
 Legaler Bereich ist (von) (bis) , Bitte neu eingeben:  
 - Es wurde eine Zahl auerhalb des zulssigen Bereiches eingegeben. Das Programm fordert eine Wiederholung der letzten Eingabe.

-----

SCREEN INSTALATION

Choose one of the following terminals:

- |                   |        |                         |
|-------------------|--------|-------------------------|
| 1) ADDS 20/25/30  | 13) .. | 25) ..                  |
| 2) ADDS 40/60     | 14) .. | 26) ..                  |
| 3) ADDS Viewpoint | 15) .. | 27) ..                  |
| 4) ADM 3A         | 16) .. |                         |
| 5) Ampex D 80     | 17) .. |                         |
| 6) ANS1           | 18) .. | 35) None of the above   |
| 7) ..             | 19) .. | 36) Delete a definition |

Wich Terminal? (Enter no. or ^Q to exit): 35  
 Terminal type: Memotech\_FDX\_System\_  
 Send an initialisation string to the terminal? Y  
 Command: Ctrl-Z  
 Send a reset string to the terminal? Y  
 Command: Ctrl-L  
 CURSOR LEAD-IN command: Ctrl-C  
 CURSOR POSITIONING COMMAND to send between line and column:  
 CURSOR POSITIONING COMMAND to send after both line and column:  
 Column first (Y/N)? Y  
 OFFSET to add to LINE: 32  
 OFFSET to add to COLUMN:  
 Binary address (Y/N)? Y  
 CLEAR SCREEN command: Ctrl-L  
 Does CLEAR SCREEN also HOME cursor (Y/N)? Y  
 DELETE LINE command: <ESC> J  
 INSERT LINE command: <ESC> I  
 ERASE TO END OF LINE command: Ctrl-E  
 START HIGHLIGHTING command: Ctrl-F Ctrl-D  
 END HIGHLIGHTING command : Ctrl-F ctrl-B  
 Number of rows (lines) on your screen: 24  
 Number of columns on your screen : 80  
 Delay after CURSOR ADRESS (0-255 ms) : 0  
 Delay after CLEAR, DELETE and INSERT (0-255 ms) : 0  
 Delay after ERASE TO END OF LINE and HIGHLIGHT (0-255): 0  
 Is this definition correct? (Y/N)? Y

Hardware dependent information

TURBO: Installation / Interrupts**Betrifft TURBO-Installation**

(Herbert Herberg, 2000)

Die von Uwe vorgeschlagenen Kommando-Belegungen (COMMAND INSTALLATION) ist nur dann so sinnvoll, wenn der MTX ohne Tastatur-Overlay von Bernd Preusing (sein es nun ein TAST.COM, oder ein RAM3, RAM4), da dann auf den Pfeiltasten die beknackten Memotech-Codes liegen. Bei Bernnds Treibern liefern die Pfeiltasten ja zum Glück die CP/M-üblichen Codes CTRL-X, -S, -D, -E für down, left, right, up.

Wenn die Tastaturbelegung nicht Memotech, sondern Preusing ist, solltet Ihr die Original-TURBO-Belegung lassen, die dann den NewWord (also auch WordStar)-Standards entsprechen, und auch das einzig vernünftige, weil merkbare sind!

**TURBO-Pascal und der Interrupt unter RAM3/RAM4**

(Olaf Krumnow, 2000)

Herbert rief mich an und erzählte mir, jemand hätte Probleme mit der Interruptprogrammierung in TURBO-Pascal. Also habe ich mich mal range-setzt und was ausprobiert.

Die Interruptprogrammierung von TURBO-Pascal ist an sich recht einfach, da der von TURBO erzeugte Assemblercode voll interruptfähig ist. Probleme treten ab RAM3 auf, weil dort einige Betriebssystemroutinen nicht mehr auf Bank 0 sondern auf Bank 1 des MTX liegen. Daraus folgt, daß die anzuspringende Interruptroutine im Common-Bereich des MTX, also bei Adressen >\$D000 liegen muß, da der Interrupt prinzipiell von jeder Bank kommen kann. Das ist aber bei den meisten TURBO-Pascal-Programmen nicht der Fall. Wie wird das Problem gelöst? Es muß eine kurze Routine (in Zukunft 'Aufrufer' genannt) in diesem Common-Bereich liegen, Bank 0 anwählen und dann die eigentliche Interruptroutine in Pascal aufrufen. Platz kann man schaffen, indem in TURBO beim Compilieren die Obergrenze z.B. auf \$D000 festgelegt wird und der Raum ab \$D000 für den Aufrufer genutzt wird. So geschehen im unten abgedruckten Programm. 'IntRout' ist die Routine, die mit jedem Interrupt ausgeführt werden soll. Sie muß immer diesen Namen haben. Als erstes in dieser Routine müssen alle benötigten Register gerettet werden, mit Ausnahme des AF-Registers (wurde schon vom Aufrufer gerettet). Welche Register das sind, hängt davon ab, was die Interrupt-Routine ausführt. Im Zweifelsfall also alle Register retten (Zweit- und Indexregister nicht vergessen!). Das Beispielprogramm kommt mit dem Erstregistersatz aus. Was darf die Interrupt-Routine NICHT ?? Sie darf keine Funktion ausführen, bei der eine Bankumschaltung durchgeführt werden muß. Dazu gehören auch und insbesondere die Bildschirmausgabe und die Diskettenzugriffe, sowie demnächst vermutlich auch die Tastaturabfrage (Funktions-tastendecodierung). Da diese Funktionen aber ohnehin viel Zeit benötigen und ein Interrupt so kurz wie möglich sein sollte, werden sie wohl ohnehin nicht benutzt. Wer doch eine Bank umschalten muß, muß Änderungen beim Aufrufer durchführen. Notfalls bei mir anrufen!

Was macht das Beispielprogramm unten? Es zählt bei jedem Interruptaufruf den Zähler Cnt um eins hoch. Auf Tastendruck wird der gerade aktuelle Wert ausgegeben. Mit ^C wird der Interrupt gelöscht und das Programm beendet. Die Routine 'Setze\_Interrupt' lädt den Aufrufer an die definierte Stelle (beim Übersetzen des Programms nicht vergessen, die 'End address' auf höchstens \$D000 zu setzen !!!), und initialisiert den CTC-Kanal 2 mit den im Aufruf angegebenen Werten. Die Bedeutung dieser Werte entnimmt man bitte geeigneten Artikeln (z.B. Info 5-25). Danach läuft der Interrupt, bis er wieder gelöscht wird. Die abgedruckte Version ist unter RAM 4.2 getestet und läuft auch mit den KLICK-Overlays bei mir problemlos.

TURBO: Interrupts

Das Programm und die Beschreibung sind nur schnell zusammengeschustert, weil das Info in Kürze erscheint und der Artikel noch rein soll. Wenn jemand irgendetwas nicht versteht oder sonstige Probleme auftauchen, so kann er gerne anrufen.

Das Programm:

```

const ctc2 = 10 { Port fuer CTC2-Kanal };
var Cnt : integer;

procedure IntRout; { auf keinen Fall Parameter !! }
begin
  InLine ( %c5/%d5/%e5 ); { Register retten, je nach Bedarf }
  Cnt := succ(Cnt);
  InLine ( %e1/%d1/%c1/%c9 ) { Register zurueck }
end; { IntRout }

procedure Setze_Interrupt (steuer,wert:byte);
var Version : array[1..5] of char absolute %ffc0;
begin
  if Version < '@RAM4' then begin { ggf. auf RAM3 (ndern )
    writeln('Falsche RAM-Version...');
    halt
  end;

InLine ( (* ; Laderoutine Interrupts Turbo-Pascal *)
(* ; Olaf Krumnow 23.03.87 *)
(*INTROUT EXT ; Name der Interruptroutine *)
(*ZIEL EQU ODOOOH ; Laufadresse des Aufrufers *)
(*START ; Start der Initialisierung *)
%21/*+%001D (* LD HL,ROUTINE *)
/%11/%00/%D0 (* LD DE,ZIEL *)
/%01/%13/%00 (* LD BC,LENROUT *)
/%ED/%B0 (* LDIR ; Int-Routine verschieben *)
/%F3 (* DI *)
/%21/%94/%F0 (* LD HL,OF094H ; unter RAM4 ist hier
der INT-Vektor *)
/%36/%00 (* LD (HL),OOH ; Lo-Byte ZIEL *)
/%23 (* INC HL *)
/%36/%D0 (* LD (HL),ODOH ; Hi-Byte ZIEL *)
/%3E/%03 (* LD A,3 *)
/%D3/%0A (* OUT (10),A *)
/%D3/%0A (* OUT (10),A ; Reset CTC2-Kanal *)
/%FB (* EI *)
/%C3/*+%0015 (* JP ENDE ; Initialisierung des Kanals *)
(* ; Aufrufer, der im Common-Bereich liegen mu~ *)
/%F5 (*ROUTINE PUSH AF ; AF retten *)
/%3E/%B0 (* LD A,BOH ; Bank $B0 setzen *)
/%D3/%00 (* OUT (0),A *)
/%CD/INTROUT (* CALL INTROUT ; Interruptroutine aufrufen *)
/%3A/%DF/%FF (* LD A,(OFFDFH) ; alte Bank laden *)
/%F6/%B0 (* OR BOH ; sicherheitshalber Bit 7 setzen *)
/%D3/%00 (* OUT (0),A *)
/%F1 (* POP AF ; Akku zurueck *)
/%FB (* EI ; Interrupts freigeben *)
/%ED/%4D (* RETI ; RETURN from Interrupt *)
(*LENROUT EQU $-ROUTINE ; Laenge der Routine *)
(*ENDE *)
(* END *)
);

```

T U R B O: Interrupts

```
( jetzt initialisiere CTC2-Kanal )

port[ctc2] := steuer;  ( Steuerwort )
port[ctc2] := wert;   ( Zeitkonstante )

end; ( Setze_Interrupt )

procedure Loesche_Interrupt;
begin
  port[ctc2] := 3;
  port[ctc2] := 3;  ( CTC2-Kanal deaktivieren )
end;

var ch : char;

begin ( main )

  Cnt := 0;
  writeln('Dr)cke <RET> zum Starten des Interrupts');
  readln;

  writeln('^j'Taste dr)cken, um Z(hler zu zeigen');
  writeln('^C f)r Ende');

  Setze_Interrupt($85,255); ( Parameter an eigenen Bedarf anpassen )

  repeat
    read(kbd,ch);
    writeln(Cnt)
  until ch=^c;

  Loesche_Interrupt;

end.
```

H A R D W A R E: Ton Aus

(Uwe Groos, 6393)

## ASSEMBLER-PROGRAMM

SOOFF.....SOUND OFF

Dieses Programm setzt alle Sound-Kanäle des MIX zurück auf Null. Umgewandelt in einen COM-File kann es jederzeit vom CP/M aufgerufen werden.

Einen automatischen Aufruf bei Systemstart erreicht man durch

STARTUP &lt;Programmname&gt;

STARTUP SOOFF

Einen automatischen Aufruf mehrerer Programme erreicht man durch

STARTUP &lt;Programmname&gt;&lt;ASCII-Char 92&gt;&lt;Programmname&gt;&lt;ASCII..&gt;

STARTUP SOOFFOCONFIG6FDXB

Nachfolgend ist das Assembler-Listing sowie das Listing in Maschinensprache für SOOFF zu finden.

## Programm SOOFF

```

0E 06      LD C,06          ; Lade in C die Portnummer
3E 9F      LD A,9F          ; Lade in A die Adresse
ED 79      OUT C,A         ; Ausgabe von A zum Tongenerator
DB 03      IN A,03         ; Übernahmebefehl für den Tongenerator
3E BF      LD A,BF          ; Lade in A die Adresse
ED 79      OUT C,A         ; Ausgabe von A zum Tongenerator
DB 03      IN A,03         ; Übernahmebefehl für den Tongenerator
3E DF      LD A,DF          ; Lade in A die Adresse
ED 79      OUT C,A         ; Ausgabe von A zum Tongenerator
DB 03      IN A,03         ; Übernahmebefehl für den Tongenerator
3E FF      LD A,FF          ; Lade in A die Adresse
ED 79      OUT C,A         ; Ausgabe von A zum Tongenerator
DB 03      IN A,03         ; Übernahmebefehl für den Tongenerator
C3 00 00   JP 0000        ; Programm Ende

```

Ende des Programms

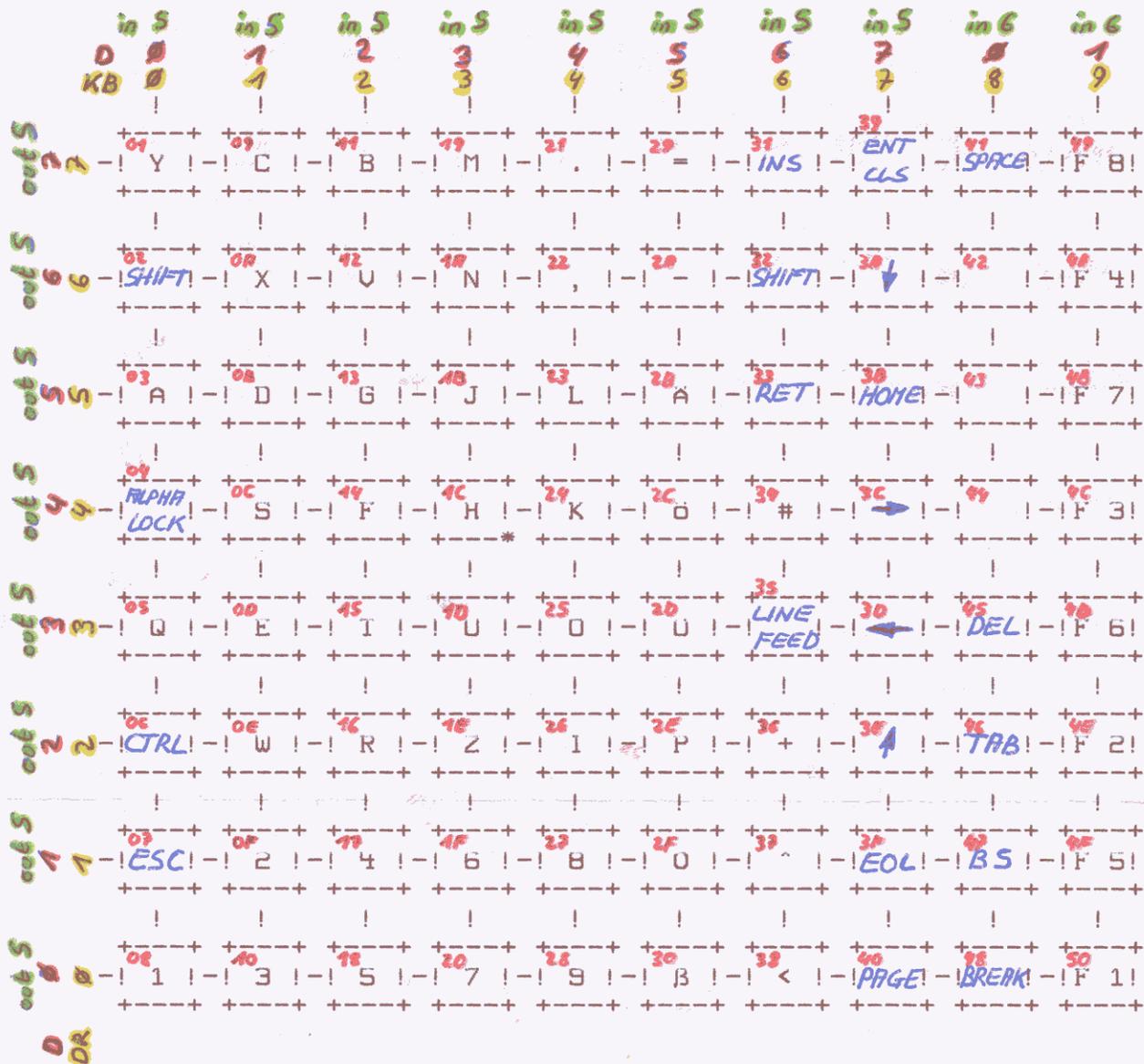
Dieses Programm bearbeitet nur den Sound-Generator des MIX-Grundgerätes, nicht aber den Baustein zur Sound-Generierung in der FDX-Einheit (dieser Chip schafft sowieso nur einen bescheidenen BEEP).

H A R D W A R E: Tastaturbelegung

(Uwe Groos, 6393)

TASTATURBELEGUNG DES MIX-COMPUTERSYSTEMS

Die Tastatur des MIX besteht aus 79 Tasten (von 80 möglichen), die beiden RESET-Tasten (die hier nicht aufgeführt sind) eingeschlossen. Die Tastatur belegt 2 Ports. Port 5 komplett (IN und OUT), Port 6 nur auf den beiden niederwertigen Bits (nur IN). Die freien Bits von Port 6 werden für den Soundgenerator verwendet.



Die Zahlen in den Tastenfeldern geben die Wertigkeit der gedrückten Taste an. DR sind die OUT-Leitungen von Port 5 an die Tastatur, die vom Z80 angesteuert werden. KB sind die IN-Leitungen von der Tastatur auf Port 5 und Port 6, wobei Port 6 in der Tastaturabfrageroutine aufaddiert wird und damit eine Wertigkeit von 8 oder 9 bekommt (Port 5: = von 0 - 7)

H A R D W A R E: Tastaturbelegung

ANHANG ZUR TASTATURBELEGUNG  
 =====

In der Tastaturabfrageroutine des MTX-Computersystems ist ein Fehler!

Die Tastaturabfrage verläuft im Groben folgendermaßen:

1. Anfangswerte setzen.
2. Erste Tastenreihe (Port 5) auf "low" setzen.
3. Alle Spalten (Port 5) einlesen.
4. Wenn keine Taste gedrückt ist, dann alle Spalten (Port 6) einlesen.
5. Ctrl- und Shift-Tasten ausblenden.
6. abspeichern der letzten gedrückten Taste.
7. Berechnung des ASC-II-Codes.

Der Fehler taucht beim einlesen von Port 6 auf. Sind dort zwei Tasten in der selben Reihe gedrückt, so wird durch die Berechnung der Taste eine Taste von Port 5 angewählt.

BEISPIEL:

-----	gedrückt wird gleichzeitig	Ergebnis
	Taste DEL & Taste F 6	CSR n.Links
	Taste TAB & Taste F 2	CSR n.oben
	Taste BS & Taste F 5	wie EOL
	Taste BREAK & Taste F 1	wie PAGE
	Taste SPACE & Taste F 8	wie ENT/CLS

Diese Erscheinung kann in Newword wunderbar getestet werden. Wie gesagt, die beiden Erstgenannten Tasten müssen gleichzeitig gedrückt werden. Es gibt auch nur 5 Möglichkeiten, da drei Tasten von Port 6 nicht belegt sind.

Ich versuche im nächsten Brief eine Rom-Änderung bekannt zu geben, doch ich möchte es erstmal gern ganz auseinandernehmen (ich bin ja erst bei der Tastaturabfrage).

Falls schon jemand diese Heldentat vollbracht hat (das Rom dort zu ändern, bzw. das Gesamte Rom assembliert und erklärt), der soll mir doch bitte schreiben oder gleich das Listing mit Erklärung zusenden. Vielleicht bekomme ich dann heraus, wie ich meinen Zusatzspeicher ansprechen kann, wie ich Graphik auf die 80-Zeichen-Karte bekommen kann (nicht diese Klotz-Graphik), wie ich einen Light-pen ansteuern kann u.s.w.

Ich tue hier inzwischen mein bestes, und versuche mich durch die 37 Seiten disassemblierter MTX-Lektüre durchzuwühlen!

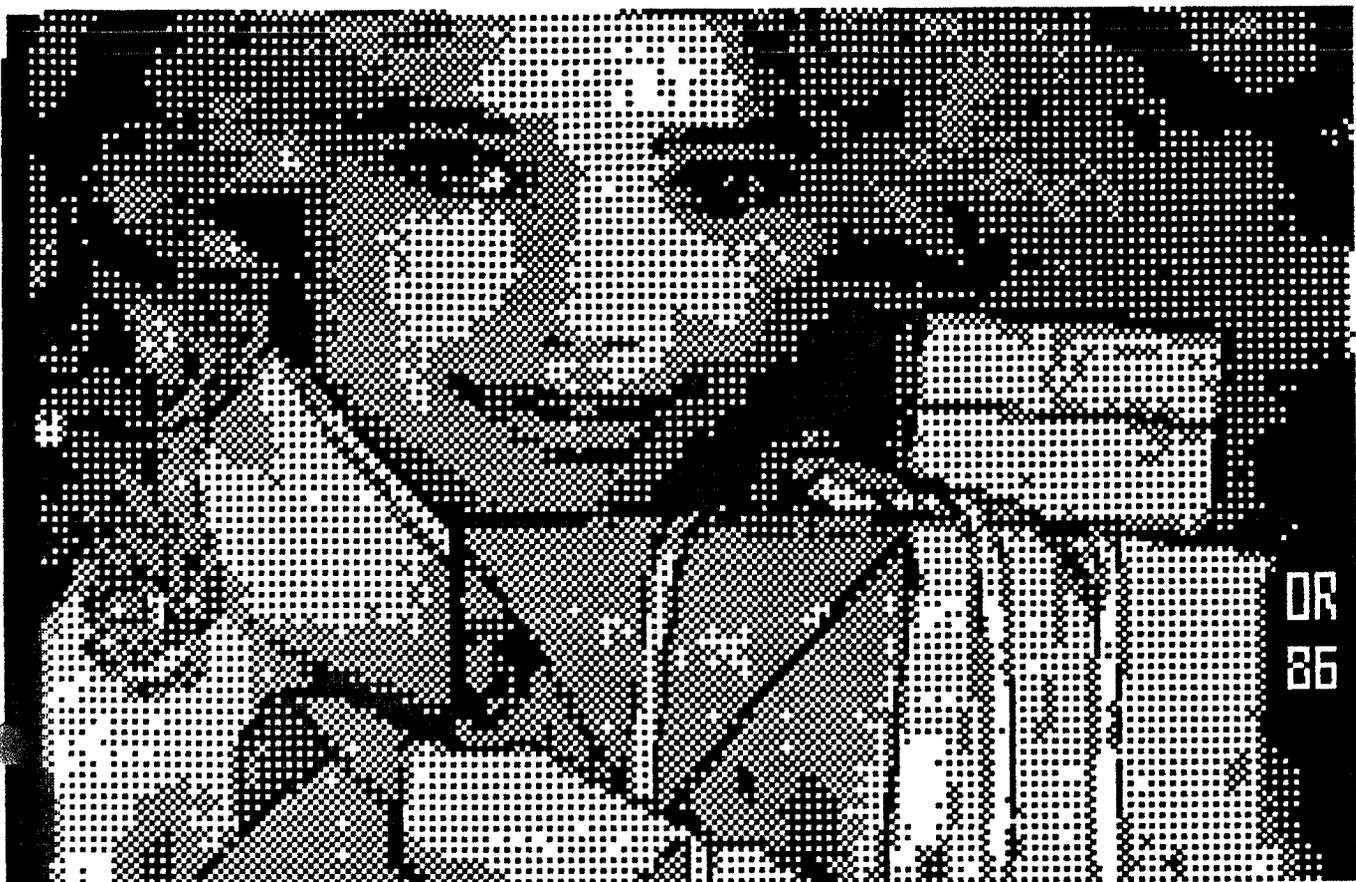
Also, laßt mir noch ein bißchen Zeit...



DIGITIZED by Otmar Rücker

# Bilddigitalisierer für DMX

Dieses Bild besteht aus 6 VS-4 Bildschirmen (je 2 nebeneinander)



DR  
26

**MEMOTECH - COMPUTER  
KALENDER 1987**

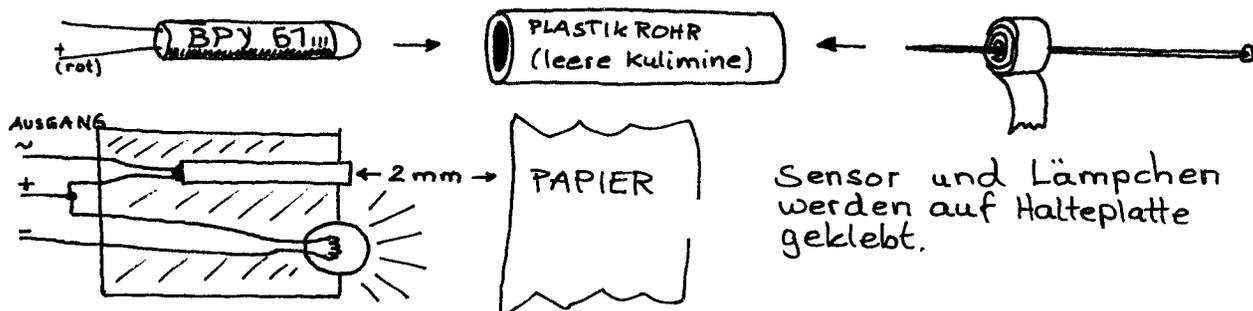
Januar	Februar	März	April	Mai	Juni
MO 5 12 19 26	MO 2 9 16 23	MO 2 9 16 23 30	MO 6 13 20 27	MO 4 11 18 25	MO 1 8 15 22 29
DI 6 13 20 27	DI 3 10 17 24	DI 3 10 17 24 31	DI 7 14 21 28	DI 5 12 19 26	DI 2 9 16 23 30
MI 7 14 21 28	MI 4 11 18 25	MI 4 11 18 25	MI 1 8 15 22 29	MI 6 13 20 27	MI 3 10 17 24
DO 1 8 15 22 29	DO 5 12 19 26	DO 5 12 19 26	DO 2 9 16 23 30	DO 7 14 21 28	DO 4 11 18 25
FR 2 9 16 23 30	FR 6 13 20 27	FR 6 13 20 27	FR 3 10 17 24	FR 1 8 15 22 29	FR 5 12 19 26
SA 3 10 17 24 31	SA 7 14 21 28	SA 7 14 21 28	SA 4 11 18 25	SA 2 9 16 23 30	SA 6 13 20 27
SO 4 11 18 25	SO 1 8 15 22	SO 1 8 15 22 29	SO 5 12 19 26	SO 3 10 17 24 31	SO 7 14 21 28
Juli	August	September	Oktober	November	Dezember
MO 6 13 20 27	MO 3 10 17 24 31	MO 7 14 21 28	MO 5 12 19 26	MO 2 9 16 23 30	MO 7 14 21 28
DI 7 14 21 28	DI 4 11 18 25	DI 1 8 15 22 29	DI 6 13 20 27	DI 3 10 17 24	DI 1 8 15 22 29
MI 1 8 15 22 29	MI 5 12 19 26	MI 2 9 16 23 30	MI 7 14 21 28	MI 4 11 18 25	MI 2 9 16 23 30
DO 2 9 16 23 30	DO 6 13 20 27	DO 3 10 17 24	DO 1 8 15 22 29	DO 5 12 19 26	DO 3 10 17 24 31
FR 3 10 17 24 31	FR 7 14 21 28	FR 4 11 18 25	FR 2 9 16 23 30	FR 6 13 20 27	FR 4 11 18 25
SA 4 11 18 25	SA 1 8 15 22 29	SA 5 12 19 26	SA 3 10 17 24 31	SA 7 14 21 28	SA 5 12 19 26
SO 5 12 19 26	SO 2 9 16 23 30	SO 6 13 20 27	SO 4 11 18 25	SO 1 8 15 22 29	SO 6 13 20 27

H A R D W A R E: Digitizer - On Stage

Ergänzungen zum Beitrag "LOW COST DIGITALISIERER" (Info 10)  
 Otmar Rücker (5102)

Nachdem sich vereinzelt Clubmitglieder mit Fragen über den Drucker-Scanner an mich gewandt haben, möchte ich Euch den neuesten Stand der Dinge kurz mitteilen. Wie das Beispiel wohl zeigt, arbeitet der Bildabtaster (scanner) mittlerweile ganz ordentlich. Der SENSOR sitzt links am Druckkopf des DMX80. Zur Beleuchtung der Papiervorlage hat sich eine Leuchtdiode nicht bewährt. Selbst eine klare Leuchtdiode war nicht in der Lage, alle Farbfrequenzen des Farbspektrums gleichmäßig zu erzeugen. Die Folge war, daß rote Farbvorlagen immer als weiß interpretiert wurden. Stattdessen arbeitet jetzt eine Miniaturglühlampe neben dem Fototransistor, der die unterschiedliche Helligkeit auf dem Papier als elektrische Signale (Stromschwankungen) weitergibt.

Als Fototransistor benutze ich immer noch den teuren BPY 61, weil ich noch keinen besseren gefunden habe. Dieser Sensor steckt in einem kleinen Plastikrohr (leere Kulimine, 1,5 cm). Zur FOKUSSIERUNG wird der stabförmige Sensor in das Röhrchen gesteckt und hinten lichtdicht verklebt. Der Lichteintritt vorne ist jetzt durch das Röhrchen begrenzt auf ca. 5 mm<sup>2</sup>. Das reicht zum experimentieren, bringt aber sehr unscharfe Abbildungen. Besser ist es, wenn man den Lichteintritt des Röhrchens verkleinert, indem man einen 2mm breiten Streifen von schwarzem Klebeband auf eine STECKNADEL wickelt, das ganze in den Lichteintritt zwingt und dann die Stecknadel vorsichtig wieder herauszieht. Jetzt ist der Lichteintritt vor dem Sensor ausreichend klein!



Die elektrischen Signale des Fototransistors leitet man sinnvollerweise über ein flexibles Kabel (3-polig, denn die Glühlampe benötigt ja auch Saft) zur Transistorschaltung. Diese ist bereits in Info 10 beschrieben. Von dort aus ist ein 5-poliges Kabel mit Port 7 (Parallelport auf der Platine des Grundgerätes) verbunden.

PORT 7 habe ich als 25 polige Buchse links am Tastaturgehäuse fest installiert (hinten ist ja kein Platz mehr, wenn man die RS 232 eingebaut hat). Der Port ist so sehr gut zugänglich wenn man mehrere Geräte über ihn betreiben will (z.B. ADC- und DAC-Wandler zur Digitalisierung von Sprache oder Meßwerten).

H A R D W A R E: Digitizer - On Stage

Und nun zur Software: Die Helligkeitswerte des Papierbildes stehen am PORT 7 als Zahlen zur Verfügung und werden in einer Schleife vom Programm abgefragt, während sich der Druckkopf des DMX über das Papier bewegt.

Basic :           FOR I=0 TO 255:LET P(I)=INP(7):NEXT I  
ASSEMBLER :       siehe Listing IN A,(7)  
                  LD (DE),A ...

Zahlenwerte:

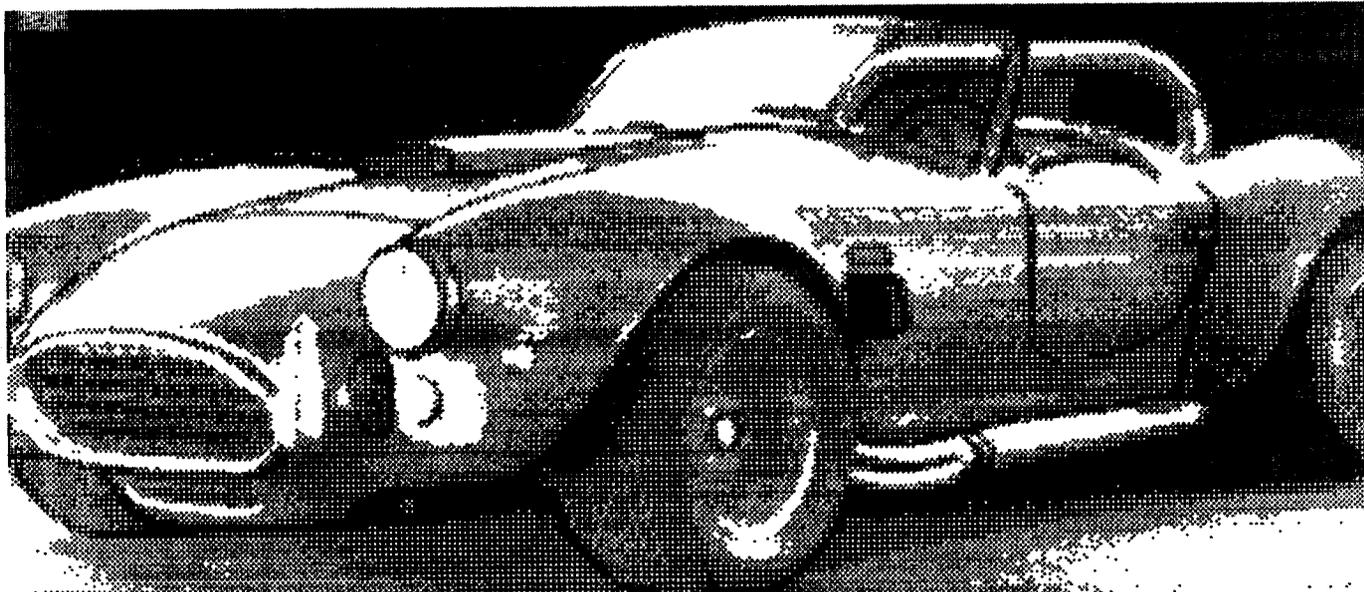
schwarz 255, dunkelgrau 254, grau 252, hellgrau 248, weiß ist uninteressant, denn dann braucht man auch nicht zeichnen.

Das Programm tastet Bilder im DIN-A 4 Format ab und baut diese als Bildschirmgrafik (VS 4) auf. Da die Auflösung des VS 4 jedoch ziemlich dürftig ist (255\*192 Punkte), lege ich für meine Bildabtastung 2 Bildschirme nebeneinander. Das heißt: es werden immer 2 Bildschirme gleichzeitig! aufgebaut. Der Aufbau der Bildschirme ist während des Abtastens gut zu verfolgen, da der betreffende linke oder rechte Bildschirm jeweils vom RAM geladen wird und so blitzschnell umgeschaltet werden kann. Sind 2 Doppelbilder fertig, so werden sie vom Programm automatisch auf Disc gespeichert und der nächste Doppelaufbau beginnt, bis daß ein ganzes DIN-A 4 Bild eingelesen ist. Ein kompletter Bildausdruck kann aus 8 Bildschirmseiten bestehen. Theoretisch ist das kaum begrenzt, da beliebig viele Doppelbilder untereinandergehängt werden können.

Die Auflösung des DIGITALISIERERS beträgt im Normalfall ca. 512(Spalten)x 768(Reihen)! wobei die Zahl 768 nicht als absolute Grenze anzusehen ist.

Um dieses Programm zu realisieren, habe ich mich als Basic-Mensch ein wenig mit Assembler befassen müssen, das Programm wird also nicht sehr elegant sein, aber es funktioniert. Ein Doppelbild (2x VS 4) ist immerhin in 15 min. aufgebaut. Aber was solls. Claudio Romanazzi hat schon eine Kopie des Programms und wird es sicher noch verbessern können.

Für Verbesserungsvorschläge bin ich jederzeit zu haben. Sollte jemand mit der Hardware Schwierigkeiten haben, so bin ich unter 02405 - 91822 bis ca. 23.30 Uhr zu erreichen, wer später anruft, der sollte sich schämen.



H A R D W A R E: Digitizer - On Stage

1 GOTO 10  
2 CODE

```

800F      LD HL,512 ; ZEILE IN 512 PUNKTEN ABTASTEN
8012      LD DE,59000
8015 MAIN: IN A,(7)
8017      LD (DE),A
8018      INC DE
8019      DEC HL
801A      LD A,H
801B      OR L
801C      CALL EINS
801F      CALL EINS
8022      CALL EINS
8025      JR NZ,MAIN
8027      RET
8028      RET
8029 EINS: CALL ZWEI
802C      CALL ZWEI
802F      CALL ZWEI
8032      CALL ZWEI
8035      RET
8036 ZWEI: CALL WAIT
8039      CALL WAIT
803C      CALL WAIT
803F      CALL WAIT
8042      CALL WAIT
8045      CALL WAIT
8048      CALL WAIT
804B      CALL WAIT
804E      RET
804F WAIT: NOP
8050      NOP
8051      NOP
8052      NOP
8053      NOP
8054      NOP
8055      NOP
8056      NOP
8057      NOP
8058      NOP
8059      NOP
805A      NOP
805B      NOP
805C      NOP
805D      NOP
805E      NOP
805F      NOP
8060      RET
8061      RET
8062      RET

```

## Symbols:

EINS	8029	ZWEI	8036
WAIT	804F	MAIN	8015

3 RETURN  
4 CODE

Leseprogramm  
für den  
Digitalisierer

Bremsschleifen sind nötig,  
da die Leseschleife auf die  
Rücklaufgeschwindigkeit des  
Druckkopfes angepaßt ist.  
(nur DMX-80)  
Wer NOPs wegläßt, erhält  
Ovale anstatt Kreise!

H A R D W A R E: Digitizer - On Stage

3 RETURN  
4 CODE

```

80F7      DI
80F8      LD HL,0      ;1. BILD EINLESEN
80FB      LD A,L
80FC      OUT (2),A
80FE      LD A,H
80FF      OR 64
8101      AND 127
8103      OUT (2),A
8105      LD C,24
8107      LD DE,35840
810A LOOP1: LD B,0
810C LOOP2: LD A,(DE)
810D      OUT (1),A
810F      INC DE
8110      DJNZ LOOP2
8112      DEC C
8113      JR NZ,LOOP1
8115      RET
8116      RET

```

Diese Programmteile  
tauschen VS 4 und RAM  
Inhalte aus  
35840 und 41984 sind  
Adressen im Ram für  
Bildhälften A und B.

Symbols:

LOOP1 B10A LOOP2 B10C

5 RETURN  
6 CODE

```

8191      LD HL,0      ;      1. BILD ABSPEICHERN
8194      LD A,L
8195      OUT (2),A
8197      LD A,H
8198      AND 63
819A      OUT (2),A
819C      LD C,24
819E      LD DE,35840
81A1 LOOP1: LD B,0
81A3 LOOP2: IN A,(1)
81A5      LD (DE),A
81A6      INC DE
81A7      DJNZ LOOP2
81A9      DEC C
81AA      JR NZ,LOOP1
81AC      LD HL,0      ;      2. BILD LADEN
81AF      LD A,L
81B0      OUT (2),A
81B2      LD A,H
81B3      OR 64
81B5      AND 127
81B7      OUT (2),A
81B9      LD C,24
81BB      LD DE,41984
81BE LOOP3: LD B,0
81C0 LOOP4: LD A,(DE)
81C1      OUT (1),A
81C3      INC DE
81C4      DJNZ LOOP4

```

H A R D W A R E: Digitizer - On Stage

```
81C6      DEC C
81C7      JR NZ,LOOP3
81C9      RET
81CA      RET
81CB      RET
```

Symbols:

```
LOOP1    81A1      LOOP2    81A3
LOOP4    81C0      LOOP3    81BE
```

7 RETURN  
8 CODE

```
8295      LD HL,0      ;2.BILD ABSPEICHERN
8298      LD A,L
8299      OUT (2),A
829B      LD A,H
829C      AND 63
829E      OUT (2),A
82A0      LD C,24
82A2      LD DE,41984
82A5 LOOP1: LD B,0
82A7 LOOP2: IN A,(1)
82A9      LD (DE),A
82AA      INC DE
82AB      DJNZ LOOP2
82AD      DEC C
82AE      JR NZ,LOOP1
82B0      EI
82B1      RETI
82B3      RET
82B4      RET
```

Symbols:

```
LOOP1    82A5      LOOP2    82A7
```

9 RETURN

```
10 REM *****
20 REM *** LESEN .BAS ***
30 REM *****
35 LET G=59000
40 PRINT "WIE SOLL DAS BILD HEISSEN ? "
50 PRINT "BISHER SIND AUF DER DISKETTE FOLGENDE TITEL :": DISC DIR "*.DBD"
60 INPUT "Wählen Sie einen NEUEN Namen      ";NAME$
70 INPUT "Wieviele Seiten hat das NEUE Bild ? ";S
75 FOR N=1 TO S
86 VS 4: COLOUR 2,11: COLOUR 3,1: COLOUR 4,10
90 GOSUB 4
92 VS 4: CLS : GOSUB 6
94 VS 4: CLS : GOSUB 8
99 ATTR 3,0: CLS : REM >>jetzt folgt die Routine für die Lesebewegung<<
100 FOR X=1 TO 192
110 LPRINT CHR$(27);"3";CHR$(3);
140 LPRINT CHR$(27);"D";CHR$(78);CHR$(0);
145 LPRINT CHR$(10);
146 LPRINT CHR$(9);
```

← für DIN-A4  
Bilder hier eine  
>4< eingeben.

Drucker einstellung  
110 - 154

H A R D W A R E: Digitizer - On Stage

```

153 LPRINT CHR$(27); "K"; CHR$(1); CHR$(0); CHR$(0);
154 LPRINT CHR$(27); "<";
155 PAUSE 100
156 GOSUB 2
157 GOSUB 4
159 POKE 58999, X: POKE 58998, 0
160 GOSUB 170
162 GOSUB 6
163 POKE 58999, X: POKE 58998, 255
164 GOSUB 170
165 GOSUB 8
166 NEXT X
167 GOTO 200: REM-----BILDAUFBAU MIT VIER GRAUSTUFEN-----
170 CODE
    
```

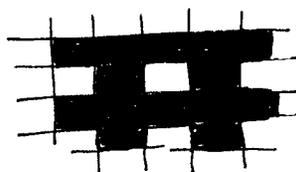
```

85F9      LD A, (58998)
85FC      LD C, A
85FD      LD B, 0
85FF      LD HL, 59000
8602      ADD HL, BC
8603 ANFANG: RST 10
8604      DB #44
8605      JR START
8607 LINE:  RST 10
8608      DB #85, 2
860A VAR1:  DB 0
860B VAR2:  DB 0
860C VAR3:  DB 0
860D VAR4:  DB 0
860E      RET
860F START: LD B, 255
8611 LOOP:  LD A, B
8612      LD (VAR1), A
8615      LD (VAR3), A
8618      LD A, (58999)
861B      LD C, A
861C      LD A, 192
861E      SUB C
861F      LD (VAR2), A
8622      LD (VAR4), A
8625 SCHWARZ: LD A, (HL)
8626      CP 255
8628      JP NZ, DUNKEL
862B      CALL LINE
862E      JP NOT
8631 DUNKEL: LD A, (HL)
8632      CP 254
8634      JP NZ, GRAU
8637      LD A, (58999)
863A      RRA
863B      JP NC, STEP
863E      LD A, B
863F      RRA
8640      JP C, GRAU
8643      CALL LINE
8646      JP NOT
8649 GRAU:  LD A, (HL)
864A      CP 252
    
```

Vorbereitung für spätere Effekte: anstatt mit LINE (4 koordinaten) kann auch der PLOT-RST10 verwendet werden.

Pixelverteilung: jedes Pixel wird gedruckt.

Musteralgorithmus für dunkelgraue Bildpartien



Pixelverteilung:

H A R D W A R E: Digitizer - On Stage

```

864C      JP NZ,HELL
864F      LD A,(58999)
8652      ADD A,B
8653      RRA
8654      JP NC,HELL
8657      CALL LINE
865A      JP NOT
865D HELL: LD A,(HL)
865E      CP 24B
8660      JP NZ,NOT
8663      LD A,B
8664      RRA
8665      JP C,NOT
8668      LD A,(58999)
866B      ADD A,B
866C      RRA
866D      JP NC,NOT
8670 STEP: CALL LINE
8673 NOT:  INC HL
8674      LD A,B
8675      CP 0
8677      JP Z,ENDE
867A      DJNZ LOOP
867C      JP LOOP
867F ENDE: RET
8680      RET
8681      RET
    
```

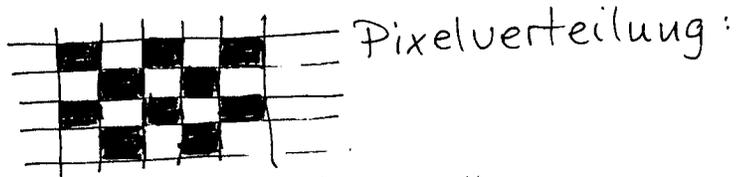
Symbols:

ANFANG	8603	START	860F
LINE	8607	VAR1	860A
VAR2	860B	VAR3	860C
VAR4	860D	SCHWARZ	8625
DUNKEL	8631	NOT	8673
GRAU	8649	STEP	8670
HELL	865D	LOOP	8611
ENDE	867F		

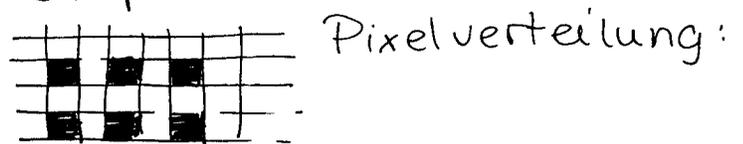
```

199 RETURN
200 REM -----ABSPEICHERN FERTIGER DOPPELBILDER-----
205 REM FALLS EINMAL UNFERTIGE BILDSCHIRME ABGESPEICHERT WERDEN SOLLN, DANN
    FOLGENDE ZEILE AB >LET N=1 BENUTZEN UND SPÄTER UMBENENNEN.
210 REM IM NOTFALL PROGRAMM AB HIER STARTEN.....> LET N=1: LET NAME$="ENDE"
220 LET CPMA$=NAME$+"A"+MID$(STR$(N),2,1)+".DBD"
222 LET CPMB$=NAME$+"B"+MID$(STR$(N),2,1)+".DBD"
223 VS 5
225 PRINT "DAS BILD WIRD ALS ";CPMA$;" UND ";CPMB$;" ABGESPEICHERT "
230 DISC OPEN #2,CPMA$,"0"
235 DISC ERA CPMA$
240 DISC WRITE CPMA$,41984,6144
245 DISC CLOSE #2
250 DISC OPEN #2,CPMB$,"0"
260 DISC ERA CPMB$
270 DISC WRITE CPMB$,35840,6144
280 DISC CLOSE #2
290 VS 4
300 NEXT N
990 STOP
    
```

Algorithmus für graue  
Bildpartien



Algorithmus für helle  
Bildpartien



Die Datenfiles der  
Bildschirminhalte  
erhalten zum Namen  
die Bildschirmhälfte,  
die Anzahl der Doppelbilder  
und den Extend .DBD

(z.B. NAME A1.DBD)

H A R D W A R E: Digitizer - On Stage



Der Digitalisierer läuft und läuft  
ft und läuft

scanned by O.Rücker 1987

3 Doppelbilder, 4 Graustufen

H A R D W A R E: Digitizer - On Stage

```

100 PRINT "DOPPELBILDDRUCK"
110 PRINT : DISC DIR "*.DBD"
115 VS 4: COLOUR 2,3: COLOUR 3,1: CIRCLE 100,100,20: ATTR 3,0: CLS : VS 5
116 PRINT "Eingabe des Namens OHNE Extend und OHNE Seitenbezeichnung und Nummer
      Beispiel: Bild Nr. 2F erscheint im Directory als 2FA1.DBD und so weiter"
120 INPUT "WELCHES BILD ? ";NAME$
121 INPUT "START BEI NR. ? ";R
122 INPUT "ENDE BEI NR. ? ";S
123 REM HAUPTSCHLEIFE-----
124 FOR N=R TO S
130 LET CPMA$=NAME$+"A"+MID$(STR$(N),2,1)+".DBD"
140 LET CPMB$=NAME$+"B"+MID$(STR$(N),2,1)+".DBD"
145 VS 5: PRINT "ES WIRD DAS DOPPELBILD ";CPMA$;" UND ";CPMB$;" AUSGEDRUCKT ."
147 VS 4
148 DISC OPEN #2,CPMA$,"I"
150 DISC READ CPMA$,41984
152 DISC CLOSE #2
155 DISC OPEN #3,CPMB$,"I"
160 DISC READ CPMB$,35840
162 DISC CLOSE #3
165 LPRINT CHR$(27);"A";CHR$(8);
167 LPRINT CHR$(27);"P";CHR$(0);
168 REM SCHLEIFE FÜR JE 2 BILDER -----
170 FOR Y=24 TO 1 STEP -1
180 GOSUB 500
190 GOSUB 300
200 GOSUB 400
210 GOSUB 300
215 LPRINT
220 NEXT Y
230 NEXT N
270 REM PROGRAMMENDE-----
280 STOP
290 REM DATENAUSGABE AN DEN DRUCKER-----
300 LPRINT CHR$(27);"K";CHR$(0);CHR$(1);
305 VS 4
310 FOR X=0 TO 255
315 LET A$=GR$(X,8*Y-1,8)
320 LPRINT A$;
330 NEXT X
340 RETURN
400 CODE

```

```

83E8      LD HL,0
83EB      LD A,L
83EC      OUT (2),A
83EE      LD A,H
83EF      OR 64
83F1      AND 127
83F3      OUT (2),A
83F5      LD C,24
83F7      LD DE,#8C00
83FA LOOP1: LD B,0
83FC LOOP2: LD A,(DE)
83FD      OUT (1),A
83FF      INC DE
8400      DJNZ LOOP2
8402      DEC C

```

PROGRAMM  
ZUM AUSDRUCKEN  
FERTIGER  
BILDER.

H A R D W A R E: Digitizer - On Stage

```
8403      JR NZ,LOOP1
8405      EI
8406      RETI
8408      RET
8409      RET
```

Symbols:

```
LOOP1    83FA    LOOP2    83FC
```

```
490 RETURN
500 CODE
```

```
846F      DI
8470      LD HL,0
8473      LD A,L
8474      OUT (2),A
8476      LD A,H
8477      OR 64
8479      AND 127
847B      OUT (2),A
847D      LD C,24
847F      LD DE,#A400
8482 LOOP1: LD B,0
8484 LOOP2: LD A,(DE)
8485      OUT (1),A
8487      INC DE
8488      DJNZ LOOP2
848A      DEC C
848B      JR NZ,LOOP1
848D      RET
848E      RET
```

Symbols:

```
LOOP1    8482    LOOP2    8484
```

```
590 RETURN
1000 DISC SAVE "DRUCKEN.BAS": RUN
```

viel Spaß  
Ever Otmar

H A R D W A R E: Datenerfassung mit BASIC-EMUF**MTX und BASIC-EMUF**

(K.Muerling,8702)

Wenn man extern Daten sammeln will, ist es ungünstig, hierfür den MTX zu verwenden. Zum einen kann man ihn schlecht vor Ort einsetzen, zum anderen blockiert man sich für die Zeit der Datenerfassung den Rechner. Besser ist es, für die Datenerfassung einen Einplatinen-Computer einzusetzen und die Daten später zur Weiterverarbeitung zum MTX zu übertragen.

Aus der Vielzahl der Einplatinencomputer erschien mir der **BASIC-EMUF** der Zeitschrift MC am geeignetsten. Der Bauvorschlag war in der MC 12/86 erschienen. (EMUF = Einplatinen-Mikrocomputer für universelle Festprogrammanwendung)

Der BASIC-EMUF ist auf einer Europakarte aufgebaut. Er besteht aus dem CPU- und dem Peripherie-Modul. Die CPU ist ein 8052AH mit integriertem BASIC-Interpreter sowie seriellen Schnittstellen für Terminal und Drucker! Dadurch ist der EMUF in Verbindung mit dem MTX ein ideales Entwicklungssystem. Um mit dem EMUF Verbindung aufzunehmen, muß man den MTX als Terminal betreiben, z.B. mit dem Programm CONTACT oder M1 (Modem). Der 8052AH hat eine Auto-Baud-Funktion, d. h. nach dem Einschalten betätigt man die Spacetaste des MTX und die Startupmeldung des EMUF erscheint auf dem Schirm. Der EMUF hat die Baudrate des MTX festgestellt und arbeitet dann mit dieser. Man kann jetzt auf dem MTX in 8052AH-Basic programmieren.

Eine wichtige Funktion ist die Möglichkeit, erstellte BASIC-Programme per Basicbefehl ins EPROM zu brennen. Und zwar nicht nur ein Programm, sondern soviele, wie eben ins EPROM passen. Einfach PROG eingeben, und schon wird das Programm ins EPROM geschrieben. Das nächste Programm wird einfach hintendran gehängt. Die Verwaltung der Programme erfolgt automatisch, der Aufruf eines einzelnen Programmes einfach mit ROM1, ROM2 usw. Programme lassen sich im EPROM betreiben oder mit XFER ins RAM kopieren.

Die CPU verfügt über mehrere (Basic-)Interruptmöglichkeiten, z.B. über den Anschluß INT1 oder zeitlich gesteuert über die interne Uhr. Autostart der Programme nach Reset ist möglich. Der 32k-RAM ist akkugepuffert, d. h. Programme im RAM bleiben auch nach dem Ausschalten erhalten.

Was kann man nun mit so einem EMUF machen? Ich benutze ihn z.B. als vollautomatische Wetterstation. Alle 6 min werden Außentemperatur (im weiteren Ausbau auch Barometerdruck und Feuchte) erfaßt und gespeichert. Am Ende der Woche hänge ich den MTX über RS232 dran, löse einen Interrupt aus und spiele die gesammelten Daten rüber. Dort kann ich sie dann grafisch weiterverarbeiten.

Hier noch eine Kurzbeschreibung des EMUF: CPU 8052AH (integrierter BASIC-Interpreter), Schnittstellentreiber MAX232, RAM 8-32k, EPROM 8-16k, Dekodierung über GAL, Uhr RTC58321 akkugepuffert, 12bit-AD-Wandler uPD7002C, PIO 8255, 2 8bit DA-Wandler ZN428, 4 entprellte CMOS-Eingänge, 2 Relaisausgänge. Eine Versorgungsspannung 5V, Stromaufnahme ca. 250 mA. EPROM-Programmierung "on board". RS232-Schnittstellen für Terminal und Printer.

H A R D W A R E: Temperaturwächter für Netzteil

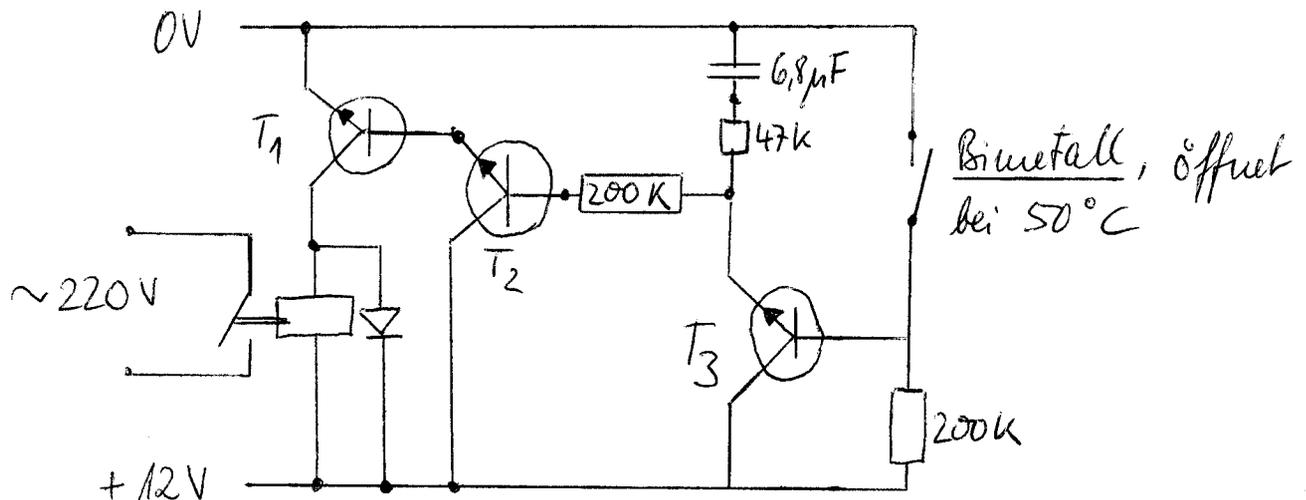
(Lutz Roland, 2000)

Hardware: Temperaturwächter für extern aufgebautes Netzgerät

Ich habe das FDX-Netzteil, den Trafo aus der FDX, der den MTX versorgte, sowie die "Bräter" aus dem MTX in einem separaten Gehäuse vereint. Für die Kühlung der ganzen heißen Versammlung sorgt weiterhin der Lüfter aus der FDX, den ich an dem neuen Gehäuse angebracht habe. Damit mich sein Klimaanlage-sound nicht durchgehend nervt habe ich einen Temperaturwächter eingebaut, der die Temperatur im Gehäuse auf 50 Grad Celsius begrenzt. Jetzt läuft der Lüfter jeweils ca 1.5 Minuten und hält anschließend 7-8 Minuten Ruhe.

Daten: geschaltete Spannung 220 Volt  
 Ruhestrom ca. 60 microamp.  
 Arbeitsstrom (=Lüfterlaufzeit) ca. 40 mA

Bauteile: Arbeitsstromrelais : 12V ca. 300 Ohm Erregerwicklung  
 Transistor T1,T3 : BC 407 A  
 T2 : TUN  
 Diode D1 : 1 N 4148 oder vergleichbare



H A R D W A R E: Immer noch nicht schnell ?

**MTX-Tuning Teil 2, oder 3...** (Rudolf Gmeinwieser, 8443)

Zu diesem Thema ist ja eigentlich schon genug geschrieben worden, aber ich möchte doch noch mal meinen Senf zum Kommentar von Hagen im Info 13 (Blatt 5) geben:

Diesen komischen Quarz (4.4336 MHz) gibt es wirklich, sogar bei 'HW'. Dieser ist aber sowiso überholt, da ich inzwischen sogar einen 5.12 MHz-Quarz aufgetrieben habe (ich sehe Hagen schon die Hände über den Kopf zusammenschlagen...).

**Nun aber zum 5 MHz-Tuning.** (Rudolf Gmeinwieser, 8443)

So unproblematisch, wie Hagen es geschafft hat, wollte mein MTX 512 diese Geschwindigkeitssteigerung nicht hinnehmen. Ich besorgte mir also den Z80B-Satz, und wechselte diese Bauteile aus. Nach dem sich beim ersten Einschalten nichts tat, wechselte ich auch noch den IC 2B (74LS04 geg. 74LS14) aus. Nach dem nächsten Einschalten bootete der MTX ohne Schwierigkeiten, nur stieg er mir beim Laden von RAM3 jedesmal aus. Arbeitete ich aber ohne RAM3 gab es keine Schwierigkeiten, also mußte es an der Speichererweiterung (bis jetzt 128K) liegen.

Beim Untersuchen der Bauteile auf der Speicher-Platine entpuppte sich der IC oberhalb des PAL als 74S37. Diesen wechselte ich gegen einen LS-Typen aus und siehe da jetzt arbeitete RAM3 normal. Da mich aber nun mal das Lötfever gepackt hatte und ich mit dem RESET-Verhalten des MTX bei 5 MHz nicht einverstanden war (den Kondensator C 11 habe ich vorher schon auf der Platine ausgewechselt) nahm ich mir auch gleich die Hauptplatine vor. Auch auf dieser fand ich unter der Bezeichnung 9D einen S-Typ (74S04). Nach diesem Auswechseln gab sich zwar der MTX immer noch ein wenig störisch aber welch ein Wunder, es lief auf einmal FDXB.COM trotz IC 74LS14 (2B) wieder.

Nun tauschte ich diese beiden IC einmal gegeneinander aus (also 74LS04 als 2B und 74LS14 als 9D) und vom nächsten Einschalten an lief FDXB.COM auch unter 5 MHz.

Es kann bei jedem MTX verschieden sein, welcher IC von beiden auf welchem Platz der Richtige ist. Dies kann man aber getrost ausprobieren. Voraussetzung:

Beide IC-Plätze werden mit einem Sockel ausgestattet. Auch können Qualitätsunterschiede der beiden IC schon viel ausmachen, deswegen ruhig zwei oder drei der einzelnen Bausteine ausprobieren.

Um nun nochmal auf diesen 5.12 MHz Quarz zu kommen:  
Auch mit diesem Quarz benimmt sich mein MTX normal und somit werde ich mich schön langsam auf die 6 MHz-Schallmauer vortasten....

Hagen behauptete in Info 13 auch, daß dem TMS9129 über 4 MHz die Luft ausgeht. Ich suchte die ganze Platine nach diesem Baustein ohne Erfolg ab. Sollte es etwa der Grafik-Prozessor TMS9929 A sein, den er meinte? Ich kann es mir aber nicht vorstellen. Der TMS9929 A zeichnet bei mir nämlich auch bei 5 MHz unter FDXB.COM wie ein Wilder. Bis 4.5 MHz (diesen Quarz habe ich auch getestet) hält er ohne weiteres unter TURBO PASCAL mit den VS4-Routinen mit und bei 5 MHz muß nur bei VS4.2 (Plot-Prozedur) bei den schon vorhandenen Verzögerungen (NOP's) nur je eine Null hinzugefügt werden und schon rasen die Linien und Kreise nur so über den Bildschirm.

H A R D W A R E: Mut zum Löten ...

An alle Bastler, (Uwe Grass, 3300)  
der MTX, bzw. die FDX, ist kein geeignetes Gerät um das Löten zu lernen! Wer einen Staubsauger reparieren kann, kann noch lange keinen Computer umbauen.

Ich habe in den letzten Tagen zwei Rechner zur Reparatur auf dem Tisch gehabt. Was ich in diesen Geräten für einen Schrott vorgefunden habe konnte ich beinahe nicht glauben. Da schimpfen die Leute auf ihren Rechner, verlieren aber bei irgendwelchen "Reparaturen" schon mal eine Unterlegscheibe, kleckern mit Lötzinn usw. Ja, da ist es doch kein Wunder, wenn das Gerät nach einem Rempler am Tisch plötzlich nicht mehr will!

Ein ganz kritischer Punkt sind die Lötarbeiten selbst. Da braten die Leute mit einem RiesenlötKolben, beim Verlöten von Wasserrohren hat er aber immer gut funktioniert, auf der Platine rum, daß sich die Leiterbahnen ablösen. Aus der Klemptnerkiste ist auch das Lötfett. Es wird dort eingesetzt, um die Oberfläche von Rohren oder Flächen anzuätzen, damit sich das Lötzinn besser damit verbindet. Wenn jetzt in der Elektronik Lötfett verwendet wird, werden alle Stellen, die damit in Berührung kommen, verätzt und korrodieren sehr schnell. Also, Finger weg von solchen Klamotten!

Genauso problematisch sind kalte Lötstellen. Mir ist es passiert, daß ich ein Bauteil nur angefaßt habe, da hatte ich es schon in der Hand. Wenn man sich jetzt die Übergangswiderstände berechnet, die in so einer Lötstelle auftreten, und die daraus resultierenden Spannungsverluste, dann wundert man sich nur noch, daß die Kiste überhaupt noch ab und an geht. Wenn man dann noch weiß, daß der Ausfall einer Spannung die Zerstörung der Videorams zur Folge haben kann, dann kann ich solche Umbaukünstler zu ihrem Mut nur noch beglückwünschen. Aber oft genug führt so eine Bastelei nur dazu, daß der Bastler den Rechner verkauft, der Käufer ist dann der Angeschmierte (ich habe meinen Rechner aus so einer Aktion, W.-D. Zimmermann oder Jochen Meyer können dazu auch eine ganze Menge erzählen).

Ein ganz großes Problem ist für einige Leute auch die FDX. Da sind zuerst mal die Laufwerke. Der Hersteller hat sich schon einiges dabei gedacht, daß er Pappkarten in die Laufwerke schiebt, wenn diese transportiert werden sollen. Schließlich sind die Köpfe sehr genau justiert. Wenn nun beim Transport die Köpfe nicht gesichert sind, dann verstellen sie sich sehr leicht. Die Folge davon sind dann Lese- und Schreibfehler, die dem Nutzer des Rechners das Leben schwer machen.

Immer wieder erlebt man auch, daß das FDX-Gehäuse nicht richtig verschraubt wird. Solange das Gerät fest auf einer Unterlage steht, macht das nicht viel aus, aber wenn die Post das Gerät transportiert, dann fliegt so ein Karton schon mal ein paar Meter. Da das Gehäuse aber nur dann stabil ist, wenn es vollständig verschraubt ist, kommt es dabei dann häufig zu Schäden.

Der letzte Punkt soll nun noch der Einbau von zusätzlichen Laufwerken sein. Daß dabei nach Möglichkeit keine Späne in der FDX verbleiben sollen, ist wohl eigentlich klar. Aber auch die Befestigung der Laufwerke im Gehäuse sollte sehr stabil vorgenommen werden. Eine Schraube pro Laufwerk reicht dazu sicher nicht aus.

Der Anschluß der Laufwerke macht vielen Leuten ebenfalls einige Probleme. Obwohl schon häufiger darauf hingewiesen worden ist, findet man in vielen Geräten auf mehr als einem Laufwerk Widerstandsarrays zur Terminierung des Busses. Diese erhöhte Belastung der Ausgangstreiber hat schon zur Zerstörung der Floppycontrollerkarte geführt. Eine Reparatur wird dann entsprechend teuer.

H A R D W A R E: Mut zum Löten ... / L e s e r b r i e f

So, nun habe ich meinem Ärger genug Luft gemacht. Als konstruktiven Beitrag werde ich von nun an ebenfalls den Umbau von Rechnern anbieten. Natürlich übernehme ich solche Aufträge nur nach telefonischer Rücksprache (wie alle anderen auch). Ich hoffe damit einige Hobbybastler mit wenig Übung von Umbauversuchen abhalten zu können.

In diesem Sinne, nur Mut  
Uwe

**Leserbrief** von Hartmut Traber, 5270

1. Ehrenrettung für die Aachener
2. Allg. Bemerkungen

Zu 1.: Ich bestellte unmittelbar nach Erscheinen von Info 17 das neue Boot-Eprom und erhielt es (allerdings nach fmdl. Rückfragen) Anfang Januar.

Der Umbausatz hielt, was er versprach. Der Rechner läuft seitdem mit 6 MHz, ohne sonstigen Umbau (Treiber o. ä.). Allerdings hat H.H. anschließend bei anderer Gelegenheit die Verdrahtung noch verbessert.

**Anm.d.HH.:** Aber leider kann Hartmut nun dank der fehlenden BASIC-ROM's einige BASIC-Anwendungen und -Spiele nicht mehr nutzen! Aber Ulrich Hönisch hat 6 MHz ohne solche Tricks am laufen ....

Zu 2.: Nachdem uns Memotech die schöne FDX verpaßt hat, finde ich, daß mögliche Erweiterungen auch (vorrangig) in der FDX getätigt werden sollten, was nicht ausschließt, daß mit dem ECB-Bus sonstige Erweiterungen gemacht werden (universell).

Und den Port 7 wie auch den Bus-Anschluß des MTX sollte man nicht vergessen.

Memotech hat(te) offensichtlich in der FDX nur Speichererweiterungen vorgesehen.

Was aber spricht dagegen, über eine kleine (aber breite) Interface-Platine Einfach- oder Doppel-Europakarten anzuschließen? Insofern finde ich die Bemühungen der "Aachener" löblich, weil sie versuchen, **intern** zu erweitern.

4 Steckplätze für Doppel-Europakarten haben wir!

**Anm.d.HH.:** Eine Speichererweiterung in der FDX ist nur als RAM-Floppy möglich - siehe Memotech Silicon-Discs!!! CPU-Hauptspeicher in die FDX zu packen bringt nichts, da die Übertragung von MTX zu FDX (und umgekehrt) viel zu langsam ist, um dort Hauptspeicher sicher laufen zu lassen. Das BOOT-EPROM wird schließlich auch erst einmal in den MTX kopiert!

Noch ein Tip:

Ich hatte in der Vergangenheit außerordentliche Schwierigkeiten mit zusätzlichen Laufwerken, sei es intern oder extern.

Ich habe Epson wildgemacht, zweimal ein neues Laufwerk bekommen, aber was war es?

**Anm.d.HH.:** EPSON SD 580 - Laufwerke haben eine Platinenkennung. Diese muß für den MTX 1A, 1B sein, nicht 2BV, 3C. TAUSCH!!

Bei mir sind unter dem ausgesägten Laufwerkträger zwei Schalt-  
netzteile, 1 x MTX, 1 x FDX, darüber neuerdings ein **Eisen-**  
blech. Und das hat es gebracht !

Es muß allerdings auch weit nach hinten reichen, und wegen Kurzschlußgefahr isoliert sein !

Ich vermute, daß die Wandlertrafos stark magnetisch streuen.

H A R D W A R E: SRAM-Floppy / EDICTA-Grafik**Preissenkung! Preissenkung! Preissenkung! Preissenkung!**

(Uwe Grass, 3300)

Die Preise für SRAMs sind wieder gesunken! Deshalb ist der neue Preis für die 256k-SRAM-Floppy 450,-- DM. Einzelpreis pro Baustein derzeit bei 26,50 DM.

**Was ist eine SRAM-Floppy**

(Herbert Herberg, 2000)

Eine SRAM-Floppy ist eine der schönsten Einrichtungen in einem Computer. Ihre Wirkung ist, daß mein MTX nach dem Anschalten ohne, daß ein Laufwerk anläuft bootet - ja, daß tut jeder! - , und dann RAM 4, IOLOADER, .... startet. Und das alles schnell - und wie gesagt ohne Laufwerk. Alles was ich brauchte war eine SRAM-Floppy und ein neues BOOT-EPROM. (Beides gibt's bei Uwe Grass). Eine SRAM-Floppy ist eine RAM-Floppy, die aus statischen RAM's aufgebaut ist, und sinnvollerweise (weil dafür gedacht) Akku- oder Batteriegepuffert ist. Also ist diese nicht nur RESET-, sonder auch Ausschaltfest.

**EDICTA-Grafik**

(Herbert Herberg, 2000)

Ich will nicht schon wieder Werbung machen!

Claudio Romanazzi schreibt gerade Software für die EDICTA-Grafik-Karte in Assembler, die dann von TURBO-Pascal aus nutzbar sein werden. Natürlich wird auch eine Programm-Bibliothek dazu erstellt. Hoffentlich schafft er einiges bis zum Clubtreffen! Sowie diese Software fertig ist, schicke ich sie an alle, die die Karte von mir erhalten haben!

ASSEMBLER: Kurs Teil 1**ASSEMBLERKURS****1 - Allgemeine Einführung (Kurt-Bernd Rohloff, 8000)**

Mittlerweile ist es schon über ein Jahr her, seit im Clubinfo der Wunsch nach einer Einführung in Assembler laut wurde. Nun nimmt dieser Assemblerkurs allmählich konkrete Gestalt an. Ihr erinnert euch vielleicht noch, daß ich mich damals anbot, als Koordinator tätig zu werden, wenn sich genügend Leute bereiterklären, die einzelnen Beiträge zu schreiben. Inzwischen haben sich 8 Kondensatoren, nein, wollte sagen: Kapazitäten bei mir gemeldet (Liste s. u.). Wir werden uns in diesem Kurs auf den Assembler beschränken, der uns im BASIC mit dem ASSEM Befehl zur Verfügung steht. Die Übertragung des hier Gelernten auf andere Assembler, die unter CP/M zur Verfügung stehen, sollte dann keine unüberwindlichen Schwierigkeiten mehr bereiten.

Wir werden an Vorkenntnissen bezüglich Assembler nichts voraussetzen, selbst auf die Gefahr hin, daß die ersten Kapitel für viele von euch ziemlich langweilig sind. Allerdings solltet ihr euch mit dem BASIC schon ein bißchen auskennen, denn irgendwo müssen wir ja aufsetzen können. Obwohl Assembler in dem Ruf steht, nur etwas für Cracks zu sein, werdet ihr, wenn ihr diesen Kurs aufmerksam durcharbeitet, sehen, daß es auch nicht schwieriger ist, als BASIC (als erste Programmiersprache) zu erlernen, nur etwas primitiver. Wenn doch mal Probleme auftreten sollten, werft nicht gleich die Flinte ins Korn, sondern fragt statt dessen eure digitalen Freunde oder ruft den Autor einfach mal an. Da die Mitgliederliste nicht ganz up to date (ja, es werden schon einige englische Begriffe vorkommen, keine Panik) ist, folgt hier eine aktuelle

## LISTE DER AUTOREN

Vorname	Zuname
Jürgen	Freimuth
Volker	Griener
Petra	Jochem
Olaf	Krumnow
Andreas	Nickel
Herbert	Oppmann
Kai-Uwe	Pleban
Kurt-Bernd	Rohloff
Claudio	Romanazzi

Dem Kurs liegt die folgende Gliederung zugrunde:

- 1 Allgemeine Einführung
- 2 Binäre Informationsdarstellung
- 3 Die Hardware aus der Sicht des Programmierers
- 4 Die Z80 Befehle
- 5 Ein komplettes Beispielprogramm

Im Kapitel 1 werden wir euch mit den grundlegenden Begriffen sowie den Vor- und Nachteilen der Assemblerprogrammierung bekanntmachen. Außerdem werden wir einen vereinfachten, hypothetischen Computer vorstellen, der eigens für diesen Kurs "gebaut" wurde. In Kapitel 2 geht es darum, wie der Computer mit den berühmt berüchtigten Nullen und Einsen umgeht und wie man damit die Daten, mit denen wir üblicherweise zu tun haben - Zahlen und Buchstaben - darstellen kann. Kapitel 3 erklärt die Innenmechanik des Z80, die für die Assemblerprogrammierung relevant sind. In Kapitel 4 schließlich geht es dann ans Eingemachte.

## A S S E M B L E R: Kurs Teil 1

Hier werdet ihr die Assemblerbefehle kennenlernen. Allerdings haben wir hier nicht auf Vollständigkeit wert gelegt, um den Charakter einer Einführung zu bewahren. In Kapitel 5 werdet ihr dann sehen, wie sich die bisher erarbeiteten Mosaiksteinchen zu einem sinnvollen Ganzen verbinden lassen. In jedem Kapitel können einige Abschnitte als Vertiefungsstoff vorkommen. Diese sind für das Verständnis des folgenden nicht unbedingt nötig und können von euch zunächst übergangen werden. Sie sind mehr als Hintergrundinformation gedacht.

### 1.1 Grundlegende Begriffe und Konzepte

Ich habe bereits mehrfach den Begriff "Assembler" benutzt, in der stillschweigenden Annahme, daß ihr euch darunter irgendetwas vorstellen könnt. Bevor ich diesen Begriff genauer definiere, sollen kurz einige Fakten über Aufbau und Arbeitsweise eines Computers dargestellt werden, die dir wahrscheinlich nicht völlig neu sind. In Kapitel 3 werden wir das Bild verfeinern.

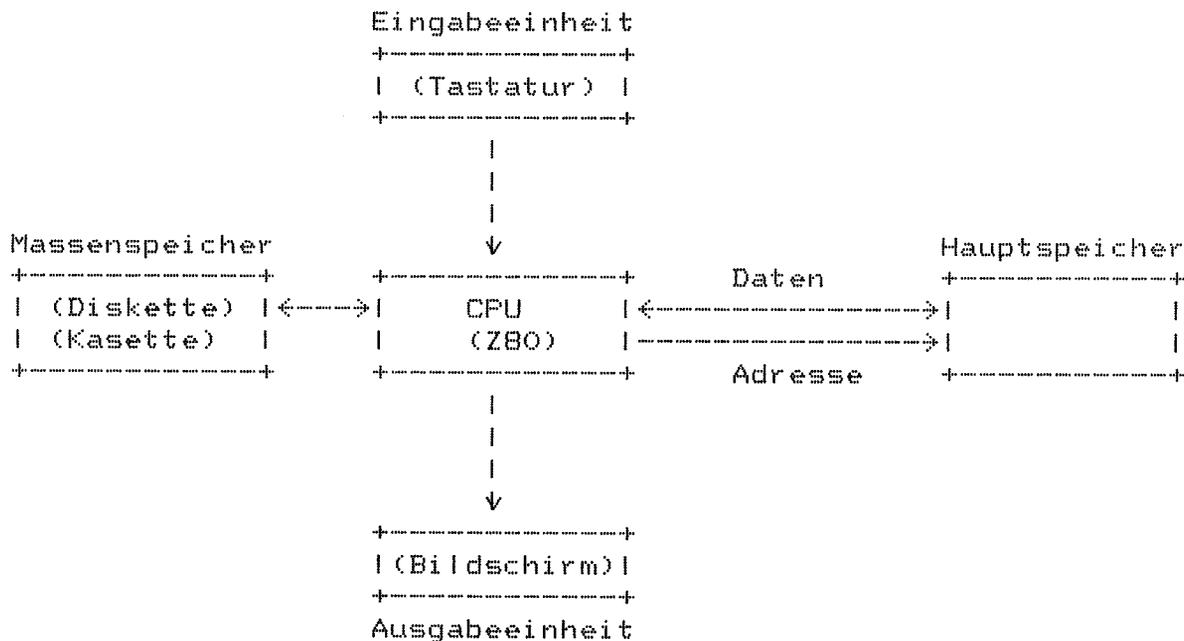
#### 1.1.1 Die von Neumann Architektur

Jeder Computer enthält eine Verarbeitungseinheit, die nach den Anweisungen eines Programms für den Fortgang der Verarbeitung zu sorgen hat. Sie wird meist CPU genannt (von central processing unit, zu deutsch: Zentraleinheit) und kann seit etwa 1970 in Form des Mikroprozessors auf einer integrierten Schaltung hergestellt werden. Im MTX Rechner wird dafür der Z80 Mikroprozessor von Zilog eingesetzt.

Das Programm, daß die Art der Verarbeitung festlegt, und ebenso die zu bearbeitenden Daten werden in einer Speichereinheit abgelegt. Der Speicher (engl.: memory, storage) ist in gleich große Zellen eingeteilt, die jeweils ein Datum aufnehmen können. Dabei wird zwischen den Programmbefehlen und den zu bearbeitenden Daten kein Unterschied gemacht. Wir werden später noch sehen, wie die CPU diese beiden Dinge trotzdem auseinanderhalten kann. Die Speicherzellen werden fortlaufend durchnummeriert, beginnend mit Null. Diese Nummer einer Speicherzelle nennt man ihre Adresse. Neben diesem internen Speicher (Hauptspeicher, Arbeitsspeicher, manchmal findet man auch noch den veralteten Ausdruck Kernspeicher), der aus elektronischen Bauelementen aufgebaut ist, benötigt man für die längerfristige Aufbewahrung der Programme noch einen Massenspeicher. Auf diesem werden die Informationen zumeist magnetisch gespeichert (Diskette, Kasette, Magnetband, Magnetplatten).

Um das Programm und die Daten in den Computer einzugeben, benötigt man ferner eine Eingabeeinheit, in der Regel ist das eine Tastatur. Schließlich möchte man von der CPU auch mal vom Ergebnis eines Programms etwas sehen. Dazu braucht man noch eine Ausgabeeinheit. Hier wäre in erster Linie der Bildschirm zu nennen, es kann sich aber auch um einen Drucker oder eine Datei auf der Diskette handeln. Diese Ein-/Ausgabe-Geräte bezeichnet man zusammenfassend auch als Peripherie. Der Austausch der Informationen zwischen CPU und Speicher/Peripherie geschieht über ein Bündel von Leitungen, die man als Bus bezeichnet. So kann man sich, stark vereinfacht, etwa folgendes Bild von der Architektur eines Computers machen:

A S S E M B L E R: Kurs Teil 1



Intern werden die Informationen grundsätzlich in einer binären Darstellung gehandhabt. Binär bedeutet soviel wie "nur zweier Werte fähig", bisweilen findet man auch den Ausdruck dual dafür. Diese beiden Werte bezeichnet man zumeist, wenn auch willkürlich, mit 0 und 1. Alle diese Buchstaben, die du hier liest, werden von NewWord lediglich als Folge von Nullen und Einsen gesehen. Ebenso liegt das NewWord Programm selbst im Hauptspeicher vor. Die bis hierher dargelegten Konzepte wurden bereits 1946 von John von Neumann vorgeschlagen.

**1.1.2 Die Assemblersprache**

Jede CPU "versteht" einen Satz von elementaren Befehlen (engl.: instruction), der ihr bereits vom Hersteller "eingepflegt" wurde. Auch diese Befehle werden binär verschlüsselt und zur Unterscheidung von BASIC oder CP/M Befehlen Maschinenbefehle genannt. Nur Programme, die in dieser Maschinensprache vorliegen (unter CP/M sind es die .COM Dateien), können von der CPU direkt verarbeitet werden. (Man kann sich streiten, inwiefern man dabei noch von einer Programmiersprache reden kann.) Die ersten Computer mußten tatsächlich in dieser "Sprache" programmiert werden. Ein Programm in Maschinensprache könnte z. B. so aussehen (die Leerzeichen sind nur der Übersichtlichkeit halber eingefügt worden):

```

1000 0000 0010 0110
0100 0000 0000 0101
1001 0001 0001 0000
  
```

Es dürfte anhand dieses kleinen Beispiels schon klar geworden sein, daß die Programmierung in Maschinensprache ein äußerst mühsames Geschäft ist. Die einzelnen Befehle sind für den Menschen ausgesprochen nichtssagend und unübersichtlich, das Eintippen solcher Programme ist sehr fehlerträchtig und Fehler sind schwer zu finden. Es wäre doch weitaus schöner, wenn jeder Maschinenbefehl einen Namen hätte, der an die Funktion dieses Befehls erinnert. Solche symbolischen Befehlsnamen bezeichnet man als Mnemonic (von griech. Mneme: Gedächtnis, Erinnerung). Damit könnte das obige Maschinenprogramm dann so aussehen:

```
LOAD 26
```

```
STD (10)
```

A S S E M B L E R: Kurs Teil 1

Sicher, im Moment kannst du damit noch nicht viel mehr anfangen (wir sind ja auch erst ganz am Anfang), aber auf alle Fälle ist das schon mal etwas übersichtlicher, oder? Du wirst bemerkt haben, daß die Mnemonics der englischen Sprache entlehnt sind. Das ist historisch bedingt und nun nicht mehr zu ändern. Dafür hat es den Vorteil, international einheitlich zu sein (stell dir nur mal vor, Hitachi würde die Mnemonics seiner Mikroprozessoren in japanischen Schriftzeichen angeben). Du siehst an dem obigen Beispiel auch, daß zu dem Befehlsnamen (LOAD: lade (in die CPU); ADD: addiere; STO: speichere (in den Hauptspeicher)) oftmals noch ein Operand gehört, der den Befehl erst vollständig spezifiziert. Zu LOAD gehört die Angabe, was geladen werden soll, nämlich die Zahl 26. Zu ADD gehört die (eine!) Angabe, was addiert werden soll, nämlich die Zahl 5. Warum man hier nicht, wie sonst bei einer Addition, zwei Zahlen angeben muß, werden wir später noch sehen. Zu STO schließlich gehört noch die Angabe, in welche Speicherzelle etwas abgelegt werden soll, im Beispiel die Zelle 10. (Es ist mir durchaus klar, daß dir das Progrämmchen jetzt auch noch nicht klar ist. Fasse dich in Geduld, es kommt alles - zu seiner Zeit.)

Jede Zeile in dem obigen Programm, also Mnemonic plus Operand, stellt einen Assemblerbefehl dar, die drei Zeilen zusammen bilden unser erstes Assemblerprogramm (womit wir endlich beim Thema wären). Die Gesamtheit dieser Assemblerbefehle zusammen mit den syntaktischen Regeln zur Formulierung der Befehle bildet die Assemblersprache, oft auch kurz Assembler genannt. Da zwischen Assemblerbefehl und Maschinenbefehl eine eins-zu-eins Korrespondenz besteht, ist die Assemblersprache immer an die jeweilige CPU gebunden, also maschinenorientiert. Sie wird zumeist vom Hersteller der CPU angegeben. Dies hat jedoch, anders als bei der Maschinensprache, nur empfehlenden Charakter.

Was haben wir nun durch die Einführung der Assemblersprache gewonnen? Für den menschlichen Programmierer ist sie leichter verständlich und besser zu merken (klarer Vorteil), aber die CPU versteht sie nicht (eklatanter Nachteil)! Was tun? Nun, wenn es schon eine Korrespondenz zwischen Assembler- und Maschinenbefehl gibt, dann sollte man doch ein Programm schreiben können, das als Input ein Assemblerprogramm bekommt und als Output das Maschinenprogramm liefert. Genau dies ist die Aufgabe eines Assemblers (to assemble = zusammensetzen, montieren)! Ja, leider wird dieses Wort in mehreren Bedeutungen gebraucht:

- a) für die Assemblersprache einer bestimmten CPU, die oft nicht ausdrücklich genannt wird, weil aus dem Zusammenhang ersichtlich,
- b) als Oberbegriff für die Assemblersprachen aller CPU's und
- c) für das Umsetzprogramm, das Assembler- in Maschinenbefehle überführt.

Man muß dem jeweiligen Zusammenhang entnehmen, was gemeint ist. Unter CP/M steht uns z. B. mit dem Programm ASM.COM ein solcher Assembler zur Verfügung, während er im BASIC, für uns versteckt, eingebaut ist.

S O F T W A R E: MTX-Edit / RAM 4 - Update / KLICK**MTX-Edit und Utilities dazu** (Herbert Herberg)

Die neue Version ist da. Alle mir bekannten Fehler sind jetzt behoben - war beim letzten Update m.E. nicht ganz der Fall. Kleinigkeiten in der Anpassung an RAM 4.2 - also Bernd's Update - verbessert. MTX-Edit unterstützt nun auch verschiedene Bildschirmformate - was natürlich RAM 4.2 voraussetzt, und nur bei aufgerüsteter 80-Zeichen-Karte richtig wsa bringt. Ich kann jetzt bei laufendem Editor zwischen beliebigen Formaten hin- und herschalten. Also mal eben auf 56 Zeilen für den Überblick. Dann zurück auf 28 Zeilen, damit die Buchstaben nicht so klein sind. ...

DocToMac erweitert und dabei verbessert. DocToMac kann aus dem Input drei verschiedene Outputs produzieren:

- Nur den Text (herrlich für HELP-Menüs, wie das HELP in MTX-Edit)
- Vollständige Programme, die für CP/M oder für das KLICK assembliert und gelinkt werden können. Dabei sind schon Olaf Krumnows neueste Entwicklungen auf dem Sektor mit beachtet.

Als Beispiel einer DocToMac-Anwendung ist die Quelle von SYSTAB mit anbei!

Ein .DOC, in dem alles besser beschrieben ist.

Update: DM 4.- Neukauf: 7.-

**Ann.d.HH.:** Die Anpassung der 80-Zeichen-Karte mache ich gerne für Euch

**RAM 4.2 - Update**

(Herbert Herberg, 2000)

Bernd hat's versprochen - also ist es auch fast fertig: Das langersehnte Update zu RAM 4.1. Die ersten Exemplare (d.h. Disketten) gibt's auf dem Clubtreffen in Hannover am 11.04.87.

Einige der Neuigkeiten:

- Editierbare CP/M-Kommandozeile
- Zwischenspeicherung der letzten 6 über BDOS-Funktion 10 eingegebenen Zeilen (also auch CP/M-Kommandozeile)
- KLICK-Overlay NOTIZBLOCK
- Schreiben/Lesen/Formatieren MSDOS-Disks
- Frei definierbares Bildschirmformat, was allerdings für große Formate mehr RAM auf der 80-Zeichenkarte verlangt. Dann können problemlos 96 x 28 Zeichen dargestellt werden. (s.o. Ann.d.HH.)
- Mehr ECB-Floppies werden unterstützt.
- Einige Utilities zu ZCPR2/P2DOS korrigiert/verbessert.
- Die eine oder andere Feinheit ist hinzugekommen

Und Bernd ist auch weiterhin bescheiden:

- Preis für Update DM 5.- incl. D&P&V!

Da ich schon etwas länger mit RAM 4.2 arbeite - wie soll Bernd sonst eventuelle Fehler finden, wenn nicht dadurch, daß er sein Programm vor irgendjemandem testen läßt - bin ich nicht sicher, was ich alles so vergessen habe. Übrigens geht es weiter mit den Programmierereinen zu RAM 4 - oder eigentlich geht's jetzt erst richtig los!! Für Ideen und Beiträge sind wir immer dankbar!

**Betr. KLICK-Programmierung**

(Herbert Herberg, 2000)

Immer wieder wird hierbei der gleiche Fehler gemacht! In KLICK-Overlays ist der BDOS-CALL mit CALL 5 unzulässig!!! Das Programm liegt nicht auf Bank 0, also steht an Adresse 5 nicht der Sprung zum BDOS!

Folgendes geht recht einfach: (RAM4-Handbuch, Seite 51)

Bildschirmausgabe: CALL OFFD3H (Zeichen in C)

Tastatureingabe: CALL OFFD0H (gedrückte Taste in A)

K O M I K**Real Programmers Don't Use Pascal**

Back in the Golden Era of computers, it was easy to separate the men from the boys (sometimes called "Real Men" and "Quiche Eaters," respectively). During this period, the Real Men were the ones who understood computer programming, and the Quiche Eaters were the ones who didn't. A real computer programmer said things like "DO 10 I=1,10" and "ABEND," and the rest of the world said things like "computers are too complicated for me" and "I can't relate to computers - they are so impersonal." A previous work, B. Feirstein's "Real Men Don't Eat Quiche," a 1982 Pocket Books publication, points out that Real Men don't relate to anything and aren't afraid of being impersonal.

But, times change. Today, we are faced with a world in which little old ladies can get computerized microwave ovens, 12-year-old kids can blow Real Men out of the water playing Asteroids and Pac-Man, and anyone can buy and understand his very own personal computer. The Real Programmer is in danger of being replaced by high school students with TRASH-80s!

There are, however, differences between the typical high school junior Pac-Man player and a Real Programmer. Knowing these differences may give kids something to aspire to - a role model, a father figure. It will also help keep Real Programmers employed.

The easiest way to determine who the Real Programmers are is by the programming language they use. Real Programmers use FORTRAN. Quiche Eaters use Pascal. Nicklaus Wirth, the designer of Pascal, was once asked, "How do you pronounce your name?" "You can either call me by name, pronouncing it 'Veert,' or call me by value, 'Worth,'" he replied. One can tell immediately from this comment that Nicklaus Wirth is a Quiche Eater. The only parameter passing mechanism endorsed by Real Programmers is call-by-value-return, as implemented in the IBM/370 FORTRAN G and H compilers. Real Programmers don't need abstract concepts to get their job done; they are perfectly happy with a keypunch, a FORTRAN IV compiler, and a beer. Real Programmers do list processing, string manipulation, accounting (if they do it at all), and artificial intelligence programs in FORTRAN.

If you can't do it in FORTRAN, do it in assembly language. If you can't do it in assembly language, it isn't worth doing.

Computer science academicians have gotten into structured programming rut during the past few years. They claim that programs are more easily understood if some special language constructs and techniques are used. They don't all agree on exactly which constructs, of course, and the examples they use to show their particular point of view invariably fit on a single page of some obscure journal. When I got out of school, I thought I was the best programmer in the world. I could write an unbeatable tic-tac-toe program, use five different computer languages, and create 1,000-line programs that worked. Then I got out into the real world. My first task was to read and understand a 200,000-line FORTRAN program, then speed it up by a factor of two. Any Real Programmer will tell you that all structured coding in the world won't help you to solve a problem like that - it takes actual talent. Some observations on Real Programmers and structured programming:

- Real Programmers aren't afraid to use GOTOs.
- Real Programmers can write five-page-long DO loops without getting confused.
- Real Programmers like arithmetic IF statements because they make the code more interesting.
- Real Programmers write self-modifying code, especially if it saves them 20 nanoseconds in the middle of a tight loop.
- Real Programmers don't need comments: the code is obvious.
- Since FORTRAN doesn't have a structured IF, REPEAT...UNTIL, or CASE statement, Real Programmers don't have to worry about not using them. Besides, they can be simulated when necessary using assigned GOTOs.

Data structures have also been in the press lately. Abstract data types, structures, pointers, lists, and strings have become popular in certain circles. Wirth, the Quiche Eater, actually wrote an entire book ("Algorithms + Data Structures = Programs", Prentice-Hall, 1976) that said you could write a program based on data structures, instead of the other way around. As all Real Programmers know, the only useful data structure is the array. Strings, lists, structures, and sets are all special cases of arrays and can be treated as such without complicating your programming language. The worst thing about fancy data types is that you have to declare them, and real programming languages, as we all know, have implicit typing based on the first letter of the six-character-variable name.

What kind of operating system is used by a Real Programmer? CP/M? God forbid. After all, it is basically a toy operating system. Even little old ladies and grade school students can use and understand CP/M.

K O M I K

Unix is a lot more complicated of course - the typical Unix hacker never can remember what the PRINT command is called this week - but when it gets right down to it, Unix is a glorified video game. People don't do serious work on Unix systems; they send jokes around the world on USENET or write adventure games and research papers.

No, the Real Programmer uses OS/370. A good programmer can find and understand the description of the IJK305I error he just got in his JCL manual. The great programmer can write JCL without referencing to the manual at all. A truly outstanding programmer can find bugs buried in a six-megabyte core dump without using a hex calculator.

OS/370 is a truly remarkable operating system. It's possible to destroy several days' worth of work with a single misplaced space, so alertness in the programming staff is encouraged. The best way to approach the system is through a keypunch. Some people claim there is a timesharing system that runs on OS/370, but after careful study I have come to the conclusion that they are mistaken.

What kind of tools does a Real Programmer use? In theory, a Real Programmer could run his programs by keying them into the front panel of the computer. In the early days, when computers had front panels, this was occasionally done. Your typical Real Programmer knew the entire bootstrap loader by memory in hex, and toggled it in whenever it got destroyed by his program. Back then, memory was memory - it didn't go away when the power went off. Today, memory either forgets things when you don't want it to, or remembers things long after they should be forgotten. Legend has it that Seymour Cray, inventor of the Cray I supercomputer and most of Control Data's computers, toggled in the first operating system for the CDC 7600 on the front panel from memory when it was first powered on. Cray, of course, is a Real Programmer.

One of my favorite Real Programmers was a systems programmer for Texas Instruments. One day, he got a long distance call from a user whose system had crashed in the middle of some important work. Jim repaired the damage over the phone, getting the user to toggle in disk I/O instructions at the front panel, repairing system tables in hex, and reading register contents back over the phone. The moral of this story: while a Real Programmer usually includes a keypunch and a line printer in his tool kit, he can get along with just a front panel and a telephone in emergencies.

In some companies, text editing no longer consists of 10 engineers standing in line to use an 029 keypunch. In fact, the building I work in doesn't contain a single keypunch. The Real Programmer in this situation has to do his work with a text editor program. Most systems supply several text editors to select from, and the Real Programmer must be careful to pick one that reflects his personal style. Many people believe that the best text editors in the world were written at Xerox Palo Alto Research Center for use on Alto and Dorado computers. Unfortunately, no Real Programmer would ever use a computer with an operating system called SmallTalk, and would certainly not talk to the computer with a mouse.

Some of the concepts in these Xerox editors have been incorporated into editors running on more reasonably named operating systems, such as EMACS and VI. The problem with these editors is that Real Programmers consider "what you see is what you get" a bad concept in text editors. The Real Programmer wants a "you asked for it, you got it" text editor; one that is complicated, cryptic, powerful, unforgiving, and dangerous. TECO, to be precise.

It has been observed that a TECO command sequence more closely resembles transmission line noise than readable text. One of the more entertaining games to play with TECO is to type your name in as a command line and try to guess what it does. Just about any possible typing error while communicating with TECO will probably destroy your program, or even worse, introduce subtle and mysterious bugs in a once-working subroutine.

For this reason, Real Programmers are reluctant to actually edit a program that is close to working. They find it much easier to patch the binary object code directly, using a wonderful program called SUPERZAP (or its equivalent on non-IBM machines). This works so well that many programs working on IBM systems bear no relation to the original FORTRAN code. In a number of cases, the original source code is no longer available. When it comes time to fix a program like this, no manager would even think of sending anyone less than a Real Programmer to do the job - no quiche-eating structured programmer would even know where to start. This is called job security.

K O M I K

Some programming tools not used by Real Programmers include:

- FORTRAN preprocessors like MORTRAN and RATFOR. These Cuisinarts of programming are great for making quiche.
- Source language debuggers. Real Programmers can read core dumps.
- Compilers with array bounds checking. They stifle creativity, destroy most of the interesting uses for EQUIVALENCE, and make it impossible to modify the operating system code with negative subscripts. Worst of all, bounds checking is inefficient.
- Source code maintenance systems. A Real Programmer keeps his code locked in a card file, because it implies that the owner cannot leave his important programs unguarded.

Where does the typical Real Programmer work? What kind of programs are worthy of such talented individuals? You can be sure that no Real Programmer would be caught dead writing accounts-receivable programs in COBOL, or sorting mailing lists for "People" magazine. A Real Programmer wants tasks of earth-shaking importance.

Real Programmers work for Los Alamos National Laboratory, writing atomic bomb simulations to run on Cray I supercomputers. They also work for the National Security Agency, decoding Russian transmissions.

It was largely due to the efforts of thousands of Real Programmers working for NASA that our boys got to the moon and back before the cosmonauts. Computers in the Space Shuttle were programmed by Real Programmers, and these true professionals are at work for Boeing, designing operating system for cruise missiles.

Some of the most awesome Real Programmers work at the Jet Propulsion Laboratory in California. Many of them know the entire operating system of the Pioneer and Voyager spacecraft by heart. With a combination of large ground-based FORTRAN programs and small spacecraft-based assembly language programs, they can do incredible feats of navigation and improvisation - such as hitting 10-kilometer-wide windows at Saturn after six years in Space, and repairing or bypassing damaged sensor platforms, radios, and batteries. Allegedly, one Real Programmer managed to tuck a pattern-matching program into a few hundred bytes of unused memory in a Voyager spacecraft that searched for, located, and photographed a new moon of Jupiter.

One plan for the Galileo spacecraft is to use a gravity trajectory past Mars on the way to Jupiter. This trajectory passes within 80 +/- 3 kilometers of the surface of Mars. Nobody is going to trust a Pascal program or programmer for this kind of navigation.

Many of the worlds Real Programmers work for the U.S. government, mainly in the Defense Department. This is as it should be. Recently, however, a black cloud has formed on the Real Programmers horizon. It seems that some highly placed Quiche Eaters at the Defense Department decided that all Defense programs should be written in some grand unified language called Ada. For a while, it seemed that Ada was destined to become a language that went against all the precepts of Real Programming. It is a language with structure, data types, strong typing, and semicolons. In short, it's designed to cripple the creativity of the typical Real Programmer. Fortunately, the language adopted by DoD has enough interesting features to make it approachable - it's incredible complex, includes methods for messing with the operating system and rearranging memory, and Edsger Dijkstra doesn't like it. Dijkstra, as you should know, authored "GOTOs Considered Harmful," a landmark work in programming methodology applauded by Pascal programmers and Quiche Eaters alike. Besides, the determined Real Programmer can write FORTRAN programs in any language.

The Real Programmer might compromise his principles and work on slightly more trivial than the destruction of life, providing there's enough money in it. There are several Real Programmers building video games at Atari, for example. But they don't play the games. A Real Programmer knows how to beat the machine every time and there is no challenge in that. Everyone working at LucasFilm is a Real Programmer because it would be crazy to turn down the money of 50 million "Star Wars" fans. The proportion of Real Programmers in computer graphics is somewhat lower than the norm, mostly because nobody has found a use for computer graphics yet. On the other hand, all computer graphics is done in FORTRAN, so there are some people doing graphics to avoid COBOL programs.

K O M I K

Generally, the Real Programmer plays the same way he works - with computers. He is constantly amazed that his employer actually pays him to do what he would be doing for fun anyway, although he is careful not to express this opinion out loud. Occasionally, the Real Programmer does step out of the office for a breath of fresh air and a beer or two. Here are some tips on recognizing real programmers away from the computer room:

- At a party, the Real Programmers are the ones in the corner talking about operating system security and how to get around it.
- At a football game, the Real Programmer is the one comparing the plays against his simulations printed on 11-by-14 fan-fold paper.
- At the beach, the Real Programmer is the one drawing flowcharts in the sand.
- A Real Programmer goes to a disco to watch the light show.
- At a funeral, the Real Programmer is the one saying "Poor George. And he almost had the sort routine working before the coronary."
- In a grocery store, the Real Programmer is the one who insists on running the cans past the laser checkout scanner himself, because he never could trust keypunch operators to get it right the first time.

What sort of environment does the Real Programmer function best in? This is an important question for the managers of Real Programmers. Considering the amount of money it costs to keep on on the staff, it's best to put him or her in optimal environment.

The typical Real Programmer lives in front of a computer terminal. Surrounding this terminal are the listings of every program he has ever worked on. These are piled in roughly chronological order on every flat surface in the office. You will also find some half-dozen or so partly filled cups of cold coffee. Occasionally, there will be cigarette butts floating in the coffee. In some cases, the cups will contain Orange Crush. And, unless he is very good, there will be copies of the OS JCL manual and the Principles of Operation open to some particularly interesting pages. Taped to the wall is a line-printer Snoopy calendar for the year 1969. Strewn about the floor there will be several wrappers for peanut butterfilled cheese bars (the type that are made stale at the bakery so they can't get any worse while waiting in the vending machine). Finally, in the top left-hand desk drawer, underneath the box of Oreos, is a flowcharting template, left there by the previous occupant. Real Programmers write programs, not documentation, which is left to the maintenance people.

The Real Programmer is capable of working 30, 40, even 50 hours at a stretch, under intense pressure. In fact, he prefers it that way. Bad response time doesn't bother the Real Programmer; it gives him a chance to catch a little sleep between compilations. If there is not enough schedule pressure on the Real Programmer, he tends to make things more challenging by working on some small but interesting part of the problem for the first nine weeks. Then he finishes the task in the last week, in two or three 50-hour marathons. This not only impresses his manager, but creates a convenient excuse for not doing the documentation. In general: no Real Programmer works from 9 to 5, except those on the night shift. Real Programmers don't wear neckties. Real Programmers don't wear high-heeled shoes. Real Programmers arrive at work in time for lunch. A Real Programmer may or may not know his spouse's name. He does, however, know the entire ASCII (or EBCDIC) code table. Real Programmers don't know how to cook. Grocery stores aren't often open at 3 a.m., so they must survive on Twinkies and coffee.

Looking to the future, some Real Programmers are concerned that the latest generation of programmers are not brought up with the same outlook on life as their elders. Many of them have never seen a computer with a front panel. Hardly anyone graduating from school these days can do hex arithmetic without a calculator. Today's college graduates are soft - protected from the realities of programming by source level debuggers, text editors that count parentheses, and user-friendly operating systems. Worst of all, some of these alleged computer scientists manage to get degrees without ever learning FORTRAN! Are we destined to become an industry of Unix hackers and Pascal programmers?

From my experience, I think it's safe to report that the future is bright for Real Programmers. Neither OS/370 nor FORTRAN shows any signs of dying out, despite the efforts of Pascal Programmers. Even more subtle tricks, like adding structured coding constructs to FORTRAN, have failed. Oh sure, some computer vendors have come out with FORTRAN77 compilers, but every one of them has a way of converting itself back into a FORTRAN66 compiler at the drop of an option card - to ompile DO loops as God intended.

K O M I K

Even Unix might not be as bad on Real Programmers as it once was. The latest release of Unix has the potential of an operating system worthy of any Real Programmer. It has two different and subtly incompatible user interfaces, an arcane and complicated teletype driver, and virtual memory. If you ignore the fact that it's structured, even C programming can be appreciated by the Real Programmer. After all, there's no type checking, variable names are seven (10? Eight?) characters long, and the added bonus of the Pointer data type is thrown in. That's like having the best part of FORTRAN and assembly language in one place, not to mention some of the more creative uses for #define.

No, the future isn't all that bad. Why, in the past few years, the popular press has even commented on the bright new crop of computer nerds and hackers leaving places like Stanford and MIT for the real world. From all evidence, the spirit of real programming lives on in these young men and women. As long as there are ill-defined goals, bizarre bugs, and unrealistic schedules, there will be Real Programmers willing to jump in and solve the problem, saving the documentation for later. Long live FORTRAN!

Ed Post  
Wilsonville, Oregon

(from: Datamation 7/83 p.263)