

CLUB 80

Clubinfo
der

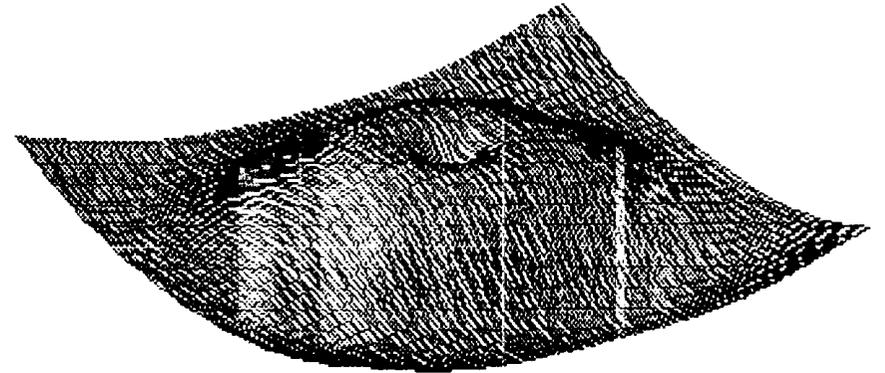
WANDY -
GENTE -
und KOMMEDIEN -
ANWENDER

10. AUSGABE

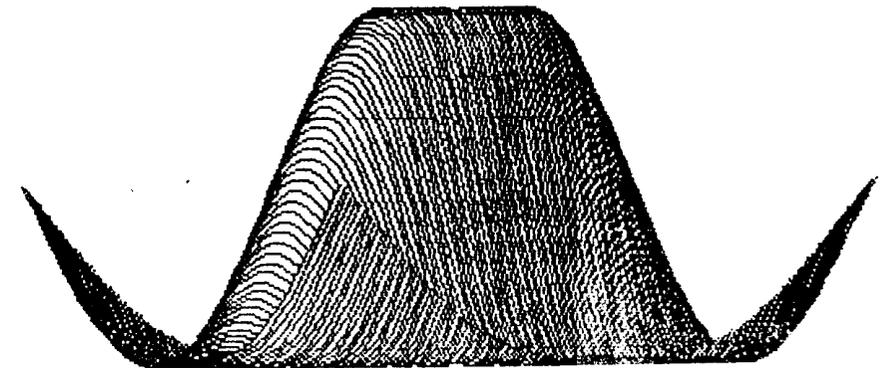
Kontaktadresse : CLUB 80 / Günther WAGNER / Gartenstraße 4 / 8281 Neubuam

Tel.: 089 3/3351 (18 - 20 Uhr)

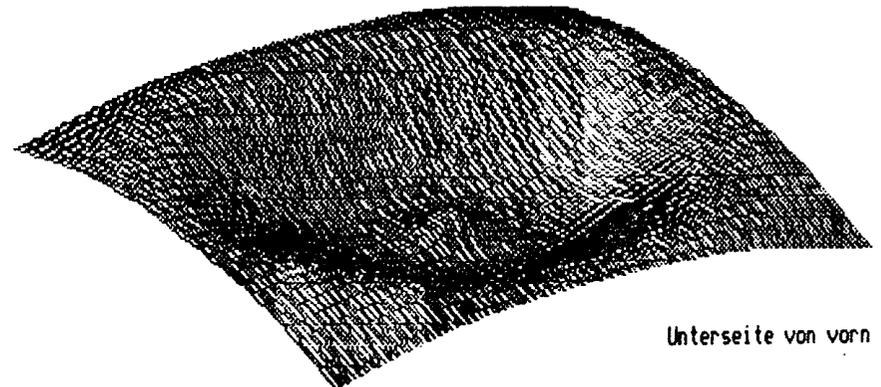
3D-HRG-Modell in verschiedenen Ansichten



Draufsicht von vorn



Seitenansicht von vorn



Unterseite von vorn

Inhaltsverzeichnis

Seite:

Clubinternes

Der Vorstand informiert 81 - 82

Software

nochmals LHOFFSET 83 - 84
Extended NEWDOS 88 85 - 10
Platz in SYS 1 11 - 14
Auflistung NEWDOS 88 Befehle 15
Eröffnen von Systemfiles 16
PDrive-Identifikation 17
DDE und LMT 19 - 20
Ein Byte und seine Folgen 21 - 22
HRG-Pixel's 23 - 32
Cassetten-Verzeichnis 33 - 34
nochmals EDITOR 35 - 36
Minni-RAM-Floppy 37 - 39
Vergleichsliste G-Dos/NewDos 40
Umlaute im NEWDOS 41 - 42
Filevergleich 43 - 44

Hardware

TRS 88 Umbau (16k auf 48k) 45 - 48
nochmals Taktumschaltung 49

Seite:

Börse

Wer hat was ???--Wer sucht was ??? . 51 - 52

Sonstiges

Computer anders genutzt 53 - 54
Urheberrecht 55 - 56
Ein alter Bekannter ?? 57
Kurzes 58

Programmbibliothek

Exklusiv 59
Neue Programme 59 - 61

Club -Bücherei

Bücherliste 63 - 64
Buchtest 65

Die letzten Seiten

Impressum 66
Schluß 67 - 68
Adressenliste letzte Seite
Anmeldung am INFO-Ende

Liebe Clubmitglieder,

dieses ist jetzt das letzte Clubinfo, das Euch noch 1985 erreicht - Nr. 11 kommt ja schon Januar 1986. Keine Angst - der Rückblick auf das Jahr 85 kommt erst im nächsten Info.

In diesem Info darf ich auf andere Punkte hinweisen - wie heißt es so schön: "Ohne Moos nicht's los!"
Ja - es ist bald wieder soweit - der Jahresbeitrag für 1986 wird fällig. Um es Euch so einfach wie möglich zu machen, liegen vorgedruckte Zahlscheine bei. Bitte denkt daran, daß Ihr möglichst schon Mitte Dezember den Jahresbeitrag für 1986 in Höhe von 50 DM überweist (bzw. anderweitig bezahlt).

Da wir gerade beim Geld sind; folgende Beiträge sind noch immer ausständig (berücksichtigt sind Zahlungen bis einschließlich 30. Oktober):

Jeweils 10 DM Nachzahlung von: Troesch, May, Wies, Hummel,
Baltes, Smerling, Fröhlich

Ich bitte um baldige Bezahlung der ausstehenden Beträge!

Clubtreffen 1986

Unser Clubtreffen 1986 nimmt schön langsam Gestalt an. Sowohl Termin als auch Unterkunft stehen fest.

Termin: Freitag 11.04. bzw. Samstag 12.04. bis Sonntag 13.04.

Das heißt folgendes: Wer will, kann schon am Freitag anreisen. Er hat dann die Möglichkeit, sich länger und intensiver mit anderen Mitgliedern zu unterhalten bzw. über die Computerei zu reden. Das eigentliche Treffen aber dauert so in etwa von Samstag Nachmittag bis Sonntag Mittag.

Unterkunft: Hotel Taunusblick, 5429 Holzhausen, Tel 06772/8343

Der Erholungsort Holzhausen liegt am Kreuzungspunkt der Bundesstraße 260 (Wiesbaden-Nassau, auch Bäderstraße genannt) mit der Bundesstraße 274 (St. Goarshausen-Limburg) und ist über Autobahnen von jeder Himmelsrichtung gut zu erreichen (Landkarte anbei). Das Hotel hat der Hartmut gefunden und für unser Clubtreffen für geeignet befunden. Neben schönem Gastraum und gemütlichen Nebenraum steht auch ein hoteleigenes Hallenbad (3* Bm) zur Verfügung. Besonders letzteres dürfte für mitkommende Familienangehörige und sonstigem Anhang interessant sein - ich hoffe ja auf alle Fälle, daß einige Mitglieder Ihre Frau bzw. Freundin mitnehmen. Für diese nicht an der Computerei interessierten Anhängsel unserer Mitglieder besteht so z.B. die Möglichkeit zu einem Stadtbummel durch Wiesbaden, Koblenz (beides ca. 30 Autominuten), Limburg a.d.Lahn oder Dietz a.d.Lahn (ca. 20 Autominuten) und zu einem Spaziergang durch den schönen Taunus.

Die Übernachtung mit Frühstück kostet 28,-- DM. Das Essen ist gut und preiswert. Der Hartmut glaubt, daß das Hotel mit seiner Atmosphäre und Umgebung zu einem gelungen Clubtreffen beitragen wird.

Ihr findet als Beilage das Anmeldeformular für unser Clubtreffen. Bitte füllt alle dieses Formular aus; auch diejenigen, die am Clubtreffen nicht teilnehmen. Wir brauchen unbedingt einen Überblick über die Teilnehmerzahl. Bitte sendet mir alle das Formular nach Möglichkeit bis Jahresende zu (Je früher, desto besser).

Denkt daran, daß die Anreise auch in Fahrgemeinschaften möglich ist. Damit werden die Fahrtkosten billiger und auch Schülern ist die Anreise erleichtert. Dies müßt Ihr aber selber unter Euch organisieren. Bitte macht Euch auch Gedanken über das Treffen; wie sollte es ablaufen, was sollten wir im Gegensatz zum ersten Clubtreffen ändern, über was wollt Ihr sprechen etc.

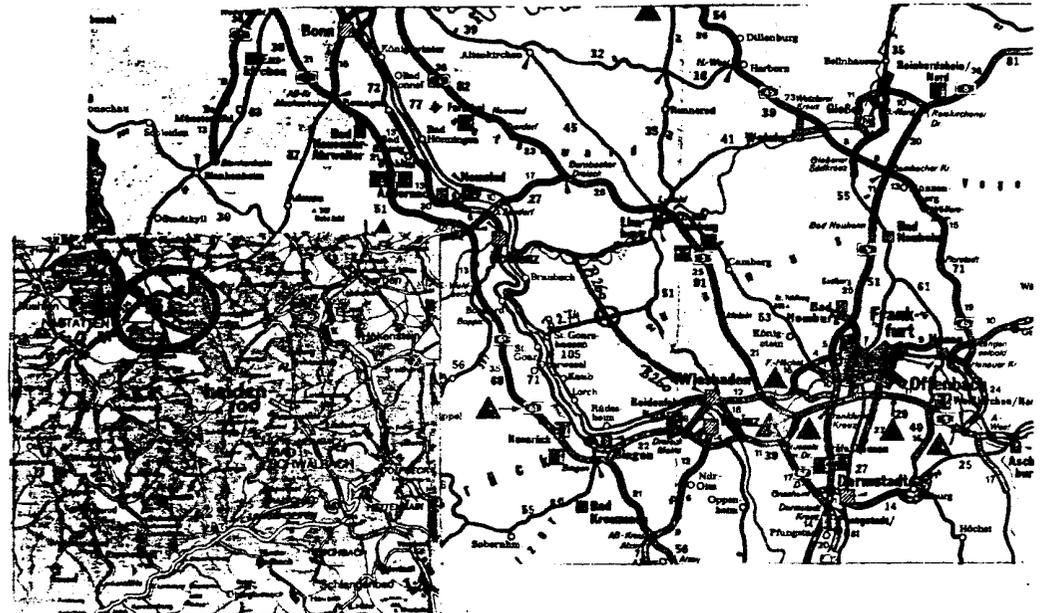
Nicht vergessen:

- alle den Jahresbeitrag 1986 zahlen !
- alle das Anmeldeformular zurücksenden !

D A N K E

Ihr hört wieder im Januar von mir (kommt gut rüber in's neue Jahr)

bis dann
Euer *Gunther Wagner*



Zu Gerald's Artikel über LMOFFSET

Lieber Gerald, in Deinem interessanten Beitrag im Info 9/85 forderst Du Hartmut und mich auf, mehr zur Record-Organisation zu sagen. Das will ich später gerne tun. Zunächst möchte ich aber mit der Paral-Spritze einer Wanze zulaube rücken, die sich dort eingeschlichen hat:

42E9 ist nicht etwa eine feste Adresse im DOS, sondern der erste frei verfügbare Speicherplatz unter Level 2. Hier beginnen z. B. BASIC-Programme oder Variable, wenn noch kein Programm im Speicher steht. Auch Maschinenprogramme stören ab hier nicht mehr, wenn BASIC nicht gebraucht wird. Und hier hast Du die Autoren von LMOFFSET bei einem Irrtum erwischt: Bereits ab 42E8 ist der Speicher frei. Dort steht eine 00, die normalerweise den Anfang einer neuen (hier also der allerersten) BASIC-Zeile markiert.

Hinter der letzten Programmzeile folgt wieder eine 00. Würden weitere Zeilen folgen, so enthielten die beiden nächsten Bytes den Zeiger auf die übernächste Zeile. Wenn dort aber zwei weitere Nullen stehen, weiß der Interpreter, daß das Programm zuende ist und ab hier die Variablen abgelagert werden können. Steht aber kein Programm im Speicher, dann folgen diese beiden Nullen gleich der ersten, so daß wir ab 42E8 die Bytefolge 00-00-00 vorfinden. Genau das bedeuten die drei NOPs, die Du als Tabelle interpretierst. Sie signalisieren dem Interpreter lediglich, "sorry, kein BASIC", so daß er z. B. nach RUN oder LIST unverrichteter Dinge sofort wieder zu "READY" zurückkehrt.

So, das wurde jetzt ein bißchen ausführlicher, denn ich meine, daß der BASIC-User wissen sollte, was intern passiert, wenn er programmiert. Aber zu Deinem Record-Thema:

Jeder DOS-Record beginnt mit einem Header-Byte, das dem DOS sagt, welcher Art der folgende Code ist. Wie Du richtig feststellst, bedeuten 01 den Start von Maschinencode und 02 die Ankündigung der Startadresse an Programmende. Da wäre noch 05 zu erwähnen, das dieselbe Funktion hat wie REM in BASIC: Hier folgt ein Kommentar, den das DOS nicht weiter beachtet. Nach diesem Header kommt ein Byte, das die Länge des folgenden Records angibt. Zwischen 03 und FF bedeutet dieses Byte eine Länge von 3-255 Bytes. 00 meint 256. Viele SYS-Files haben eine Recordlänge von 256 Bytes. Was es mit den ebenfalls möglichen Zählbytes 01 und 02 auf sich hat, folgt gleich.

Bis hierher wird kaum ein Leser "aha" gesagt haben, denn das ist das Übliche. Jetzt kommt's aber: Es sind auch die Recordlängen 257 und 258 möglich. NEWDOS und seine Enkel G-DOS und H-DOS sind in der Lage, diese Recordlänge aus den Zählcodes 01 und 02 herauszulesen. Weshalb das zumindest für 02 höchst erstaunlich ist, wird nach der folgenden Erläuterung klar:

Nach dem Header (01, 02, 05 ...) und dem Zählbyte folgen zwei weitere Bytes, die noch nicht zum eigentlichen Programmcode gehören. Sie bedeuten in der Reihenfolge LSB-MSB die Ladeadresse des Records. Diese beiden Bytes werden im Zählbyte mitgezählt. So erklärt es sich, daß das Programmende mit 02-02-yy-xx angekündigt wird: 02 (finito) - 02 (zwei Bytes folgen) - yy (LSB der Startadresse) - xx (MSB). Der Zählcode 02 bedeutet demzufolge 2 Bytes und nicht 258. Oder eben doch. Also beides. Wie das DOS das auseinanderhält, weiß ich auch nicht. Es tut es, finden wir uns damit ab, und freuen wir uns darüber!

Mit ASM/CMD (und wohl auch mit seinem Großvater EDTASM/CMD) entstehen immer Records von maximal 252 Bytes Länge. - 252 plus die beiden mitgezählten Adressbytes ergeben 254 (FS), plus Header plus Zählbyte sind das genau 256 Bytes, also ein kompletter Sektor. So fängt jeder Record in einem zusammenhängenden Maschinensprache-File links oben mit FS an,

wonach die Ladeadresse folgt. Das ist optisch sehr angenehm und macht einen "aufgeräumten" Eindruck.

ZEUS/CMD erzeugt Records von 256 Bytes Länge. Sie beginnen mit 01-02 (01 für Maschinencode, 02 für 258 Bytes, also 2 Adress- plus 256 Programmbytes). Das wirkt "unordentlicher", hat aber seinerseits wieder einen anderen Vorteil: In einem Sektor lassen sich Speicheradressen sehr leicht auszählen, denn wenn z. B. ein Record bei 8000 beginnt, so startet der nächste mit 8100. Bei EDTASM wäre es stattdessen 80FC.

Ein weiterer Vorteil ist noch wesentlich wichtiger: Wenn in einem Sektor ganz unten rechts der Header, das Zählbyte und die Ladeadresse des nächsten Records stehen, so ist der komplette nächste Sektor mit reinem Maschinencode beschrieben, es erscheinen keine Bytes, die zur Recordorganisation gehören. Dann haben wir es mit einem sog. "core image code" zu tun, also Maschinensprache "netto", wie z. B. in PC97/SYS. Wenn ein solcher Sektor diskrelativ oder dateirelativ eingelesen wird, kann er sofort angesprochen werden. Seine Ladeadresse bestimmt der User. Mit diesem Trick arbeitet z. B. das alte H-DOS, um die Funktionentabelle bei Bedarf reinzuschlüpfen. Auch die beiden FORM-Tabellen in SYS28/SYS von G- (H-) DOS sind als core image code angelegt.

Einen solchen Sektor zu erzeugen, ist kein Problem. Man schreibt z. B. mit DEFM ein Stück Dummy-Code von genau 248 Bytes Länge. Seine Ladeadresse liegt irgendwo, z. B. im ROM. Wenn die Adresse des nächsten, ernst gemeinten Records nicht unmittelbar anschließt, wird der alte Record nach dem 248. Byte geschlossen. Die nächsten Bytes zur Definition des nächsten Records werden dann direkt dahinter geschrieben. Zwei Header, zwei Zählbytes und zwei Adressen addieren sich zu 8 Bytes. 8 + 248 = 256 Bytes, also ein Sektor. So beginnt der reine Maschinencode des nächsten Records tatsächlich ganz links oben im nächsten Sektor und endet ganz rechts unten.

Diese Erklärung ist nun ebenfalls wesentlich ausführlicher geworden, als es Deine Anregung erfordert hätte. Es sollen aber alle etwas von dem profitieren, was ich im Laufe der Zeit (und der Entwicklung von H-DOS) herausgefunden habe.

Du möchtest die BREAK-RESET-Akrobatik umgehen und schlägst dazu vor, Teile des ROM-Inhalts ins eigene Programm zu übertragen, um BASIC zu initialisieren. Hierzu genügen die paar Bytes von 0676-0680 und von 0685-069A. Anschließend hüpfst man einfach nach 0075, den Rest macht der Computer alleine. Dann steht gleich READY auf dem Bildschirm. Mit dem SYSTEM-Befehl (oder USR) kommt man in sein eigenes Programm. Es geht noch komfortabler, aber da wird es allmählich kompliziert:

Die Routine ab 0075 landet bald bei 1A1C. Dort steht CALL 41AC. Wenn nun in 41AC der Befehl JP PROGRAM steht, wird das eigene Programm direkt angesprochen. Den alten Code an 41AC (RET) kann man anschließend restaurieren, um Level 2 komplett im alten Zustand zu haben. Auf diese Weise werden letzte Initialisierungen übersprungen, die aber ziemlich unwichtig sind: Der Cursor kommt nicht an den nächsten Zeilenanfang und der Cassettmotor wird nicht ausgeschaltet. Da der Recorder aber nicht lief, ist das egal. Daß READY nicht ausgegeben wird, ist ebenso wurscht, weil das eigene Maschinenprogramm vielleicht andere Sorgen hat, als ausge-rechnet auf eine BASIC-Eingabe zu warten.

Wenn das Maschinenprogramm das Level-2-BASIC nicht benötigt (keine RSTs, keine CALLs ins ROM usw.), kann man auf jegliche Initialisierung verzichten. Man darf dann sogar nach 4000 laden, der untersten RAM-Speicherstelle.

Arnulf Epp

Extended NewDOS 80

Erweitern des NewDOS80 mit GDOS-, HDOS- und selbstgestrickten Funktionen

Wenn man sich einmal die Mühe macht zu ermitteln, welches Betriebssystem von TRS 80- und VideoGenie- Benutzern bevorzugt wird, kommt man zu folgendem Ergebnis:

NEWDOS 80
GDOS (HDOS)
TRSDOS
DOSPLUS
MULTIDOS
sonst. (VTOS,LDOS usw)

Wie man feststellen kann, ist NEWDOS das meistverwendete Diskettenbetriebssystem. Platz zwei belegt GDOS, die restlichen Systeme rangieren nur unter "ferner liefern"!

Den Grund, warum NEWDOS beliebter ist als GDOS, sehe ich keineswegs in der Leistungsfähigkeit (GDOS hat mehr Befehle als NEWDOS), sondern in der etwas ungeschickten, eingedeutschten Syntax der GDOS-Befehle.

Wenn man jedoch den sehr komfortablen Befehlssatz von GDOS (noch besser HDOS, z.B. OUT, ID usw.) benutzen will, muß man eben in den sauren Apfel beißen!
Besser gesagt, man musste es!!!

In der letzten Woche meines Urlaubs habe ich mich endlich an ein Projekt gemacht, das mir schon lange vorschwebte. Ich wollte das NEWDOS erweitern. Anfänglich dachte ich nur an die HDOS-Befehle ID und OUT. Am Ende wurden daraus 10 neue Befehle!

Inzwischen ist das Projekt praktisch abgeschlossen und erprobt. Ich finde es deshalb an der Zeit, euch meine Vorgehensweise darzulegen, damit sie jedermann nachvollziehen und sich seine eigenen DOS-Erweiterungen basteln kann (oder meine übernehmen).

Die Grundlagen, die ich zur Erweiterung von NewDOS benötigte und die im Rahmen dieser Artikelserie noch genauer erklärt werden, habe ich einer Reihe von Artikeln unseres Clubmitglieds und Oberzappers Arnulf Sopp entnommen (erschienen im Info des Computerclub Bremerhaven). Ich möchte ihm daher an dieser Stelle besonders für seine vielen Anregungen danken!

Die Anregungen und Informationen wurden durch die Lektüre des neuen Röckrath-Buches "Das DOS Buch" von H. Grosser, erheblich vertieft. Dieses Buch enthält sehr viele tiefgründige und genaue Informationen über das Betriebssystem NewDOS80 und seine Abkömmlinge. Ich kann das Buch nur wärmstens empfehlen!

Nun genug der langen Vorrede. Los Werk!!!

Grundlagen:

Der Aufruf eines NewDOS-Befehls:

Die Befehle, die unser NewDOS (und auch GDOS und HDOS) versteht, sind im sog. Library verzeichnet. Das Library von NEWDOS, GDOS und HDOS ist im Systemfile 1 (SYS1/SYS) untergebracht.

Wird ein Befehl aufgerufen, durchsucht das System das Library.

APPEND	ATTRIB	AUTO	BASIC2	BLINK	BOOT	BREAK	CHAIN
CHNON	CLEAR	CLOCK	CLS	COPY	CREATE	DATE	DEBUG
DIR	DO	DUMP	ERROR	FORMAT	FREE	HIMEM	JKL
KILL	LC	LCDVR	LIB	LIST	LOAD	MDBORT	MDCOPY
MDRET	PAUSE	PDRIVE	PRINT	PROT	PURGE	R	RENAME
ROUTE	STMT	SYSTEM	TIME	VERIFY	WRDIRP		

Hat es den eingegebenen Befehl gefunden, weist es dem A- und dem C-Register der Z80-CPU bestimmte, für den Befehl spezifische Werte, zu.

Beispiele:

	A-Reg.	C-Reg.
DIR	2A	80
ID	F8	80
ATTRIB	E9	85
AUTO	E9	84

Danach wird mit dem Z80-Befehl RST 28 ein Teil des DOS angesprochen, das das Systemfile lädt und startet, welches die Routine für den entsprechenden Befehl enthält. Welches Systemfile dies ist, wird durch den Wert bestimmt, der zuvor im A-Register (ACCU) der CPU abgelegt wurde.

Der im C-Register gespeicherte Wert dient nach dem Einsprung in das entsprechende Sysfile zum Durchsteigen einer Entscheidungstabelle, die zur richtigen (fast alle Systemfiles enthalten mehrere) Unterroutine führt.

Verteilung der Systemfiles im Directory und Zusammensetzung des "Aufrufbytes":

Das Directory von NewDOS-, GDOS- und HDOS-Disketten ist eine relativ flexible Sache. Es kann, bestimmt durch die FDrive-Parameter, praktisch auf fast allen Spuren einer Diskette liegen (außer auf der Spur 0) und die unterschiedlichsten Größen annehmen. Etwas am Directory ist jedoch starr und immer gleich, die Verteilung der Systemfiles!

Das Directory kann, wie schon gesagt, unterschiedlich groß sein, seine Mindestlänge beträgt jedoch zehn Sektoren (entspricht bei 55/80-Disketten zwei Granulen). Die ersten beiden Sektoren eines Directorys werden immer vom Granule Allocation Table und vom Hashcode Index Table belegt. Aus diesem Grund sind die max. 32 Systemfiles auf die acht verbleibenden, mindestens verfügbaren Sektoren verteilt! Die eigentliche Aufteilung ist aus dem nachfolgenden Schaubild ersichtlich:

Aufbau des Direktory und Verteilung der Systemfiles

DIR/SYS relative Sektoren:

	0	1	2	3	4	5	6	7	8	9	10	11 usw.	
relative Bytes im Sektor	Granulare Allokationstabelle	Hashcode Indikationstabelle	BOOT	DIR	SYS0	SYS1	SYS2	SYS3	SYS4	SYS5	weitere	File-	
00													
10													
20			SYS6	SYS7	SYS8	SYS9	SYS10	SYS11	SYS12	SYS13	Einträge		
30													
40			SYS14	SYS15	SYS16	SYS17	SYS18	SYS19	SYS20	SYS21			
50													
60			SYS22	SYS23	SYS24	SYS25	SYS26	SYS27	SYS28	SYS29			
70													
80			weitere	File-	Einträge								
90													
A0													
B0													
C0													
D0													
E0													
F0													

Der Unwissende hat Mut, der Wissende hat Angst.

Weibliche Logik

Im Stadtpark trägt der Oberprimaner seiner Angeboteten begeistert Goethes Ballade vom Zauberlehrling vor. Sie, nachdem sie ihm schweigend gelauscht hat, meint schließlich skeptisch, daß es im Zeitalter der Elektronengehirne wohl auch so sei und man sich der gerufenen Geister nicht erwehren könne.

»Aber das ist doch unmöglich«, ereifert er sich, »ein Computer kann sich nicht selbständig machen! Wir geben ihm Informationen, und er zieht daraus logische Schlüsse. Nur geht es sehr viel schneller und genauer als es das menschliche Gehirn vermag.«

Da sie ihn zweifelnd ansieht, versucht er zu erklären: »Von Hannover bis Hamburg sind es 160 Kilometer. Wie weit ist es also von Hamburg nach Hannover?« »Ebenfalls 160 Kilometer«, antwortet sie.

»Gut«, lobt er, »das ist eine logische Folgerung. Der – « »Moment«, unterbricht sie ihn. »Ich möchte dich auch etwas fragen: Wie viele Tage sind es von Weihnachten bis Neujahr?«

»Natürlich 7 Tage«, entgegnete er irritiert, »aber das hat doch damit nichts zu tun!«

»Und ob!« trumpft sie auf, »denn von Neujahr bis Weihnachten sind es genau 358 Tage! Nun sage mir, wo deine Logik bleibt!«

Der Spezialist ist in seinem winzigen Weltwinkel vortrefflich zu Hause, aber er hat keine Ahnung von dem Rest.

Ärgern Sie sich nicht, wenn Ihr Haar ausfällt. Stellen Sie sich vor, es täte weh und müßte gezogen werden wie Zähne.

Da die Systemfiles relativ häufig benötigt werden, werden sie nicht über ihren Namen aufgerufen (wie z.B. jedes Maschinenprogramm), sondern von einem speziellen Teil des DOS unter Zuhilfenahme eines "Aufrufbytes" direkt geladen. Das verkürzt die Ladezeit erheblich und bringt somit eine Beschleunigung bei der Ausführung von DOS-Kommandos.

Das Aufrufbyte setzt sich folgendermaßen zusammen:

Die Bits 0 bis 2 stellen den relativen Sektor minus 2 dar, in dem der Fileeintrag des gewünschten Systemfiles steht.
z.B.: SYS8/SYS steht im relativen Sektor 4 des Direktory
4 - 2 = 2 bzw. als Bitmuster 0 1 0
Bits im Aufrufbyte: 2 1 0

Die Bits 3 und 4 stellen den relativen Fileeintrag in dem Sektor dar, in dem das Systemfile steht.
z.B.: SYS8/SYS steht an zweiter Stelle im rel. Sektor 4, d.h. es stellt den ersten relativen Fileeintrag in diesem Direktorysektor dar. Bitmuster 0 1
Bits im Aufrufbyte: 4 3

Die verbleibenden Bits 5 bis 7 können zur Übergabe von Parametern benutzt werden. Sie sollten auf alle Fälle größer 0 sein (am besten Bit 7 gesetzt).
z.B.: 1 0 0
Bits im Aufrufbyte: 7 6 5

Daraus ergibt sich folgendes Bitmuster zum Aufruf des Systemfiles 8: Bit 7 6 5 4 3 2 1 0
1 0 0 0 1 0 1 0
und dies entspricht der Hexzahl 8A

Das sieht im ersten Moment sehr kompliziert aus, ist es aber nicht! Wenn man die Prozedur ein paarmal durchgerechnet hat, geht sie einem relativ schnell von der Hand. Zur Übung hier noch ein Beispiel:

Aufruf des SYS21/SYS
SYS21/SYS steht im rel. Sektor 9 des Direktory und ist dort als drittes File eingetragen. Für die Bits 5 bis 7 wollen wir das Bitmuster 1 0 1 einsetzen. Daraus ergibt sich folgende Bitfolge: 1 0 1 1 0 1 1 1
Bits im Aufrufbyte: 7 6 5 4 3 2 1 0
und dies entspricht der Hexzahl B7

Ich glaube, damit wäre die Sache wohl klar und wir können uns dem im C-Register gespeicherten "Sprungbyte" widmen.

Das "Sprungbyte":

Die Systemfiles haben im Schnitt eine Größe von fünf Sektoren. Das entspricht ca. einem kByte Programm. Darin läßt sich natürlich mehr als nur eine kurze Routine, z.B. für das Anzeigen der Library oder des Direktorys, unterbringen. Ein Systemfile kann jedoch, genau wie jedes Maschinenprogramm auch, nur eine Startadresse haben. Man benötigt deshalb ein Argument, mit dessen Hilfe man in die gewünschte Unterroutine gelangt.

Beim Aufruf der Systemfiles wird dieses Argument im C-Register der CPU zwischengespeichert und am Beginn des Systemprogramms ausgewertet. Eine solche Auswertung zeigt das folgende disassemblierte Listing von SYS28/SYS:

Disassembliertes Listing des SYS28/SYS in der von mir geänderten Form. Das Systemfile enthält die Befehle F(orm), DR(uck), LF(eed) und OUT!

Die Befehle erhalten im SYS1/SYS folgende Werte zugewiesen:

Befehl:	Accu	C-Register
LF	FE	81
DR	FE	82
OUT	FE	87
F	FE	88

ORG	4D00H	Programmbeginn und gleichzeitig Startadresse!
START	CP	OFEH "Aufrufbyte" richtig?
	JR	NZ,ERROR wenn nein, dann Fehler!
	DEC	C Vom Inhalt des C-Registers 1 abziehen!
	JP	Z,LFeeD C-Register jetzt = 0?
		Ja, dann zur Routine, Nein, dann weiter!
		Das gleiche Spielchen wie oben!
	DEC	C
	JP	Z,DR
	DEC	C
	JP	Z,ERROR
	DEC	C
	JP	Z,ERROR
	DEC	C
	JP	Z,ERROR
	DEC	C
	JP	Z,OUT
	DEC	C

Hier folgt die Bearbeitungsroutine für den Befehl F(orm)!

Dem aufmerksamen Leser (und das seid Ihr ja hoffentlich alle?!) wird sofort auffallen, daß dem C-Register jeweils ein Wert von 8x zugewiesen wird (Bitmuster: 10xx xxxx), während in der Sprungtabelle im Systemprogramm davon ausgegangen wird, daß die ersten vier Bit des Wertes auf 0 stehen! Wie man sonst könnte nach einmaligem Dekrementieren des C-Registers (DEC C) die LF-Routine erreichen, wenn zuvor 81 in "C" geladen wurde???

Die Lösung dieses scheinbaren Widerspruchs ist relativ einfach. Das System benutzt die Bits 5 und 6 des Sprungbytes selbst und setzt sie vor dem Aufruf des Systemprogramms auf 0.

Die Bedeutung der Bits:

Bit	Inhalt	Bedeutung
7	1	ist immer gesetzt und bildet die Endmarkierung für den davorstehenden Befehltext
6	1	der Befehl darf nicht vom MiniDOS aufger. werden
6	0	der Befehl darf vom MiniDOS aufgerufen werden
5-0	x	Entry-Zähler

Die erste Aktion nach dem Start des Systemprogramms ist die Abfrage des Inhalts des Accu. Stimmt dieser nicht, so wird angenommen, das DOS hat sich "verlaufen" und es wird ein "Illegal DOS-Funktion"-Error angezeigt.

Die Stufen der Sprungtabelle, die zur ERROR-Routine führen, können übrigens sehr leicht zum Einbau bzw. Aufruf eigener Routinen benutzt werden (wenn noch Platz ist im Systemfile).

Nun aber weg von der grauen Theorie und hin zur Praxis!

Platz im SYS 1

Um zusätzliche Lib-Befehle installieren zu können, muß man sich zuerst Platz im Sys 1 schaffen. Das könnte man prinzipiell dadurch erreichen, daß man die bestehenden Befehlswörter kürzt. So könnte man z.B. aus BASIC2 B2 machen, aus ERROR E usw.. Meiner Meinung nach ist das jedoch ein ungeschickter Weg (wie er z.B. auch bei GDOS beschränkt wurde) und ich war mir sicher, daß es auch einen anderen geben mußte.

Wer sich SYS1/SYS einmal mit Superzap oder DDE anschaut, stellt fest, daß am Ende des Files (Sektor 4 relativ) 32 Bytes mit 00 gefüllt sind. Ein bißchen Platz ist also schon da, jedoch an einer total falschen Stelle.

Das Library endet, wenn man sich das Systemfile einmal disassembliert stellt man dies schnell fest, bei der Speicherstelle 50B6H. Als Endzeichen dient die Byte 00. Gleich anschließend folgt dann ein Teil des Programms zur Anzeige des Libraries (LIB).

```

5094 53595354454D  DEFM  'SYSTEM'  'LIBBefehl SYSTEM
509A 81            DEFB  81H       'Wert für C-Register
509B E9          DEFB  0E9H      'Wert für A-Register
509C 00          DEFB  00H       'Schlußbyte
509D 54494D45    DEFM  'TIME'    'LIBBefehl TIME
50A1 8A          DEFB  8AH       'weiter wie oben!
50A2 E9          DEFB  0E9H
50A3 00          DEFB  00H
50A4 564552494659 DEFM  'VERIFY'
50AA 84          DEFB  84H
50AB E5          DEFB  0E5H
50AC 00          DEFB  00H
50AD 575244495250 DEFM  'WRDIRP'
50B3 80          DEFB  80H
50B4 53          DEFB  53H
50B5 00          DEFB  00H       'Schlußbyte WRDIRP-Befehl
50B6 00          DEFB  00H       'Schlußbyte Library

```

'Hier beginnt die Routine zur Anzeige des Librarys (LIB-Befehl)'

'man beachte: für die nächsten 36 Bytes nur relative Sprünge!!!'

```

50B7 215B4F      M50B7  LD      HL,M4F5B 'Anfang des Library in HL
50BA 0E0B        M50BA  LD      C,0BH
50BC 060B        M50BC  LD      B,0BH
50BE 7E          M50BE  LD      A,(HL)
50BF CB7F        BIT    7,A
50C1 23          INC    HL
50C2 2005        JR     NZ,M50C9
50C4 CD9F51      CALL  M519F
50C7 10F5        DJNZ  M50BE
50C9 23          M50C9  INC    HL
50CA 23          INC    HL
50CB 7E          LD      A,(HL)
50CC B7          OR     A
50CD CA9D51      JP     Z,M519D
50DD 0D          DEC    C
50E1 CC9D51      CALL  Z,M519D
50E4 28E4        JR     Z,M50BA
50E6 CD9551      CALL  M5195
50E9 18E1        JR     M50BC

```

'hier ist es mit den relativen Sprüngen zuende!!!'

Schaut man sich die dort aufgeführten Programmschritte einmal genauer an, erkennt man, daß innerhalb eines Bereichs von 36 Bytes nur relative Sprünge auftreten. Dieser Umstand ließ in mir die Idee reifen, dieses verschiebbare Programmteil auf den freien Platz am Ende von SYS1/SYS zu verlegen.

Der Platz dort ist jedoch nur 32 Bytes lang. Aus diesem Grund mußte die DOS-Readymeldung etwas gekürzt werden. Als Ergebnis der Aktion hat man 36 Bytes freien Raum im Library, in dem man eine ganze Menge neue Befehle einfügen kann (bei günstiger Ausnutzung max. 9 Befehle). Ich glaube aber, dafür lohnt sich die Arbeit!

Die Änderungen

Zunächst muß man die sehr lange Meldung "Mini-NewDOS/80 Ready" erheblich kürzen. Sie lautet dann "M-DOS >". Während sich der Beginn für die MiniDOS-Meldung nicht verschiebt, wird der Anfang der normalen DOS-Meldung auf die Speicherstelle 51B2 vorverlegt. Man muß dies der Routine, die diese Meldung benutzt, mitteilen. Dazu werden im Sektor 0 rel. des SYS1/SYS das Byte E1 rel. von B5 auf B2 ändern.

Nach der normalen Ready-Meldung des DOS folgen noch drei Bytes, die zur CLS-Routine gehören. Diese werden wieder direkt an die neue Meldung angehängt. Im Zusammenhang damit muß im Sektor 0 rel. das Byte 4D rel. von C5 auf B8 geändert werden!

Direkt im Anschluß daran kommen dann die 36 Bytes der LIB-Routine aus dem oben angegebenen Listing. Natürlich muß auch hier die aufrufende Routine über die Veränderung informiert werden. Dazu muß man im Sektor 0 rel. die Bytes 25/26 rel. von B750 in BB51 ändern.

Ob ihr's glaubt oder nicht, das war schon alles! Mehr Änderungen sind nicht zu machen, um sich einen ziemlich großen Platz im Library zu verschaffen, in den man dann einiges einbauen kann. Dazu aber später. Hier erst nochmal ein bißchen von der unabdingbar nötigen Theorie.

Das Schlußbyte

Wenn ihr euch nochmal den ersten Teil des Disassemblerlistings anschaut, werdet ihr feststellen, daß nach dem Befehlsword und den beiden Werten für das C- und das A-Register noch ein Byte steht. Dieses Byte, ich bezeichne es als Schlußbyte, enthält folgende, für die Befehlsausführung wichtige Informationen in seinem Bitmuster versteckt:

Bit	Inhalt	Bedeutung
0	1	besitzt der anzugebende Dateiname kein Extension, so ist /CMD als Default anzunehmen
1	1	wie oben, jedoch /JCL (bzw. /JOB bei GDOS)
2	x	nicht benutzt
3	0	das Registerpaar HL zeit auf die zur Ausführung benötigten Parameter
3	1	der Zeiger auf die benötigten Parameter ist auf dem Stack abgelegt
4	1	dem Befehl folgen weitere Parameter
4	0	dem Befehl dürfen keine weiteren Param. folgen
5	1	der Befehl erwartet zwei Dateinamen als Param.
6	1	bei der Ausführung des Befehls ist eine neue Datei zu eröffnen
6	0	bei der Ausführung des Befehls darf nur eine schon bestehende Datei geöffnet werden
7	1	SYS1 (net eine neue oder bestehende Datei zur Bearbeitung durch das aufgerufene Systemprogramm

Quelle: Das DOS P... , Grosser

000000: 0100 004D FE23 CABA 4DFE 432B 73FE 63CA ...M.#. M.C(s.c.
000010: 304E FEB3 CA3D 51FE A3CA 2A4F FEC3 2B3F ON. =Q. *D..(?)
000020: 0D2B 3A0D CAB7 500D CA32 4D0D CADB 500D (. ...F..2M...P.
000030: 2B52 0DCA FA50 0D2B 430D CA34 4E0D 2B0C (R...F.(C..4N.(.
000040: 0D0D CA50 510D 2B3C 3E2A B7C9 21C5 51C3 ...PD.(<)*C.!D.
000050: 6744 D901 01E3 11D3 49C5 D5D9 EFF1 C9CD gD.....I.....
000060: 2B51 0100 00EB 216A 43CB 76CB F62B 04ED +D....!jC.v....
000070: 4B9D 43C5 ED73 9D43 EBC3 354E F1F1 180F KFC...sFC..5N....
000080: AF37 180B 216A 437E E62F 772B CBAE AFF3 #7...!jCB./w+...#
000090: 216B 4336 002B 462B 4E1E 0BF5 CB50 2023 !kC6.+F+N....F #
0000A0: F1F5 3B02 2B0A FE3B 2B06 1E04 CB69 2013 ..B.(..B(...i .
0000B0: CBB6 CB70 205E 3A6C 43CB 772B 0BCB 6920 .p ^:1C.w(...i
0000C0: 071E 0C16 EB7A 4BEF CB7B 2014 31E0 41CBzK..x .1.A.
0000D0: 6F21 B045 2213 433E C32B 023E C932 1243 o! E".C>.(.>.2.C
0000E0: 21B5 51CB 782B 07ED 789B 4321 B051 FB3E !Q.x(...aC!Q.>
0000F0: 0DCD 3300 CB69 CC67 4421 6A43 CBEE 010B ...3..i.gD!jC....

000000: 0100 004D FE23 CABA 4DFE 432B 73FE 63CA ...M.#. M.C(s.c.
000010: 304E FEB3 CA3D 51FE A3CA 2A4F FEC3 2B3F ON. =Q. *D..(?)
000020: 0D2B 3A0D CABE 510D CA32 4D0D CADB 500D (. ...D..2M...P.
000030: 2B52 0DCA FA50 0D2B 430D CA34 4E0D 2B0C (R...F.(C..4N.(.
000040: 0D0D CA50 510D 2B3C 3E2A B7C9 21BB 51C3 ...PD.(<)*C.!D.
000050: 6744 D901 01E3 11D3 49C5 D5D9 EFF1 C9CD gD.....I.....
000060: 2B51 0100 00EB 216A 43CB 76CB F62B 04ED +D....!jC.v....
000070: 4B9D 43C5 ED73 9D43 EBC3 354E F1F1 180F KFC...sFC..5N....
000080: AF37 180B 216A 437E E62F 772B CBAE AFF3 #7...!jCB./w+...#
000090: 216B 4336 002B 462B 4E1E 0BF5 CB50 2023 !kC6.+F+N....F #
0000A0: F1F5 3B02 2B0A FE3B 2B06 1E04 CB69 2013 ..B.(..B(...i .
0000B0: CBB6 CB70 205E 3A6C 43CB 772B 0BCB 6920 .p ^:1C.w(...i
0000C0: 071E 0C16 EB7A 4BEF CB7B 2014 31E0 41CBzK..x .1.A.
0000D0: 6F21 B045 2213 433E C32B 023E C932 1243 o! E".C>.(.>.2.C
0000E0: 21B2 51CB 782B 07ED 789B 4321 B051 FB3E !Q.x(...aC!Q.>
0000F0: 0DCD 3300 CB69 CC67 4421 6A43 CBEE 010B ...3..i.gD!jC....

000300: 4154 C02B 0046 0100 FA4F 5245 45B0 4A00 AT.(.F...DREE J.
000310: 4B49 4D45 4D82 E900 4A4B 4CB0 A510 4B49 HIMEM...JKL I.KI
000320: 4C4C 8045 904C 43B9 E500 4C43 4456 52B8 LL E LC...LCDVR
000330: E500 4C49 42B2 E300 4C49 5354 85F0 8B4C ...LIB...LISTI .L
000340: 4F41 4480 A450 4D44 424F 5254 85E3 004D OAD .PMDORTI...M
000350: 4443 4F50 59B2 EBB0 4D44 5245 54B6 E300 DCOFY .MDRET...
000360: 5041 5553 458B EB00 5044 5249 5645 B3E9 PAUSE .PDRIVE...
000370: 0050 5249 4E54 86F0 8B50 524F 54B6 E900 .PRINT...PRDT...
000380: 5055 5247 45B9 E900 52B0 2300 5245 4E41 PURGE...R #.RENA
000390: 4D45 81E4 B052 4F55 5445 81F0 0053 544D ME...ROUTE...STM
0003A0: 5489 EB00 5359 5354 454D 81E9 0054 494D T...SYSTEM...TIM
0003B0: 458A E900 5645 5249 4659 84E5 0057 5244 E...VERIFY...WRD
0003C0: 4952 50B0 5300 0021 5B4F 0E0B 060B 7ECB IRP S...'XD...B.
0003D0: 7F23 2005 CD9F 5110 F523 237E B7CA 9D51# ...QF.##B .Q
0003E0: 0DCC 9D51 2BE4 CD95 511B E1F3 CD2B 5121 ...Q(...!Q....+Q!
0003F0: 6A43 7EE6 C020 333A 2240 F5ED 739B 43CB jCB.. 3:"5..s.C.

000300: 4154 C02B 0046 0100 FA4F 5245 45B0 4A00 AT.(.F...DREE J.
000310: 4B49 4D45 4D82 E900 4A4B 4CB0 A510 4B49 HIMEM...JKL I.KI
000320: 4C4C 8045 904C 43B9 E500 4C43 4456 52B8 LL E LC...LCDVR
000330: E500 4C49 42B2 E300 4C49 5354 85F0 8B4C ...LIB...LISTI .L
000340: 4F41 4480 A450 4D44 424F 5254 85E3 004D OAD .PMDORTI...M
000350: 4443 4F50 59B2 EBB0 4D44 5245 54B6 E300 DCOFY .MDRET...
000360: 5041 5553 458B EB00 5044 5249 5645 B3E9 PAUSE .PDRIVE...
000370: 0050 5249 4E54 86F0 8B50 524F 54B6 E900 .PRINT...PRDT...
000380: 5055 5247 45B9 E900 52B0 2300 5245 4E41 PURGE...R #.RENA
000390: 4D45 81E4 B052 4F55 5445 81F0 0053 544D ME...ROUTE...STM
0003A0: 5489 EB00 5359 5354 454D 81E9 0054 494D T...SYSTEM...TIM
0003B0: 458A E900 5645 5249 4659 84E5 0057 5244 E...VERIFY...WRD
0003C0: 4952 50B0 5300 0000 0000 0000 0000 0000 IRP S.....
0003D0: 0000 0000 0000 0000 0000 0000 0000 0000
0003E0: 0000 0000 0000 0000 0000 00F3 CD2B 5121+Q!
0003F0: 6A43 7EE6 C020 333A 2240 F5ED 739B 43CB jCB.. 3:"5..s.C.

000400: FEFB 3E0D CD33 00C3 01EA F850 8A4D 216A ...>..3....P M!j
000410: 43CB 7ECA 444D ED7B 9B43 3E0E CD33 00F1 C.B.DM.aC>>..3..
000420: B747 3E0F CC33 007B 3222 40F3 CBBE AF0B #>..3.x2"5...#
000430: FDE1 DDE1 F1C1 D1E1 D9C1 D1E1 0BFB C9F1
000440: E5D5 C508 D9E5 D5C5 F5DD E5FD E5D9 0BF5
000450: C9CD 5051 F57E D603 2B02 D60A 2B01 23F1 ..PD.B..(...(.#.
000460: C911 8044 D506 20CD 5A51 D106 00C9 7EFE .. D...ZQ....B.
000470: 2A20 0412 1323 05E5 7ED6 30FE 0ACD 8951 * ...#...B.O...Q
000480: 3016 7ED6 2EFE 0DCD 8951 3B06 3E03 12F1 0.B.....QB.>...
000490: AFC9 7E12 1323 10EA F601 E17E C9DB 7ED6 #.B.#.....B..B.
0004A0: 41FE 1ADB D620 FE1A C93E 20CD 9F51 10F9 A....> .Q..
0004B0: C93E 0DD5 F5CD 3300 F1D1 C943 4D44 4A43 .>...3....CMDJC
0004C0: 4C54 4F00 4D69 6E69 2D4E 6577 444F 532F LTO.Mini-NewDOS/
0004D0: 3B30 2052 6561 6479 0D1C 1F03 0000 0000 BU Ready.....
0004E0: 0000 0000 0000 0000 0000 0000 0000 0000
0004F0: 0000 0000 0202 004D 0000 0000 0000 0000M.....

000400: FEFB 3E0D CD33 00C3 01EA F850 8A4D 216A ...>..3....P M!j
000410: 43CB 7ECA 444D ED7B 9B43 3E0E CD33 00F1 C.B.DM.aC>>..3..
000420: B747 3E0F CC33 007B 3222 40F3 CBBE AF0B #>..3.x2"5...#
000430: FDE1 DDE1 F1C1 D1E1 D9C1 D1E1 0BFB C9F1
000440: E5D5 C508 D9E5 D5C5 F5DD E5FD E5D9 0BF5
000450: C9CD 5051 F57E D603 2B02 D60A 2B01 23F1 ..PD.B..(...(.#.
000460: C911 8044 D506 20CD 5A51 D106 00C9 7EFE .. D...ZQ....B.
000470: 2A20 0412 1323 05E5 7ED6 30FE 0ACD 8951 * ...#...B.O...Q
000480: 3016 7ED6 2EFE 0DCD 8951 3B06 3E03 12F1 0.B.....QB.>...
000490: AFC9 7E12 1323 10EA F601 E17E C9DB 7ED6 #.B.#.....B..B.
0004A0: 41FE 1ADB D620 FE1A C93E 20CD 9F51 10F9 A....> .Q..
0004B0: C93E 0DD5 F5CD 3300 F1D1 C943 4D44 4A43 .>...3....CMDJC
0004C0: 4C54 4F00 4D2D 444F 5320 3E0D 1C1F 0321 LTO.M-DOS >...!
0004D0: 5B4F 0E0B 060B 7ECB 7F23 2005 CD9F 5110 XO...B# .. QF
0004E0: F523 237E B7CA 9D51 0DCC 9D51 2BE4 CD95 .##B.#.Q(...!
0004F0: 511B E100 0202 004D 0000 0000 0000 0000 Q.....M.....

Hier nun eine Auflistung aller NewDOS80-Befehle mit ihren Sprung-, Aufruf- und Schlußbytes

Befehl	Sprung-	Aufruf-	Schlußbyte	Systemfile
APPEND	C0	68	00	6
ATTRIB	85	E9	88	7
AUTO	84	E9	00	7
BASIC2	86	EB	00	9
BLINK	81	E5	00	3
BOOT	8A	EB	10	9
BREAK	85	E5	00	3
CHAIN	C3	EB	8A	9
CHNON	C5	EB	00	9
CLEAR	84	F0	00	14
CLOCK	82	E5	00	3
CLS	89	E3	10	1
COPY	C0	48	00	6
CREATE	82	F0	40	14
DATE	8B	E9	00	7
DEBUG	83	E5	00	3
DIR	80	2A	00	8
DO	C3	EB	8A	9
DUMP	87	E9	CB	7
ERROR	87	F0	00	1
FORMAT	C0	28	00	6
FREE	80	4A	00	8
HIMEM	82	E9	00	7
JKL	80	A5	10	3
KILL	80	45	10	3
LC	89	E5	00	3
LCDVR	88	E5	00	3
LIB	82	E3	00	1
LIST	85	F0	B0	1
LOAD	80	A4	50	2
MDBORT	85	E3	00	1
MDCOPY	82	EB	B0	9
MDRET	86	E3	00	1
PAUSE	88	EB	00	9
PDRIVE	83	E9	00	7
PRINT	86	F0	88	1
PROT	86	E9	00	7
PURGE	89	E0	00	7
R	80	23	00	1
RENAME	81	E4	B0	2
ROUTE	81	F0	00	14
STMT	89	EB	00	9
SYSTEM	81	E9	00	7
TIME	8A	E9	00	7
VERIFY	84	E5	00	3
WRDIRP	80	53	00	17
und die neue Funktion "ID":				
ID	80	FB	00	22

Das war eine geballte Ladung Information, jetzt kommt wieder ein Praxisteil! Es geht weiter mit der "Eröffnung" der Systemfiles, die normalerweise nicht auf einer NewDOS80-Diskette vorhanden sind (SYS22-SYS29), dort aber für unsere beabsichtigten Erweiterungen unbedingt gebraucht werden.

Eröffnen von Systemfiles

Wie wir schon gehört haben, werden die Systemfiles nicht über ihren Namen, sondern über ihren genau bestimmten Platz im Directory aufgerufen (siehe "Grundlagen"). Auf einer "normale" NewDOS-Systemdiskette fehlen die Systemfiles 22-29. Aber um neue Funktionen implementieren zu können, brauchen wir diese Files. Hier nun ein Trick, wie man die fehlenden Systemfiles ins Directory einträgt.

1. Man nehme eine NewDOS-Systemdiskette und erstelle mit dem Kommando COPY,0,1,,FMT,CBF,/SYS eine "Nursystemdiskette". Wenn man zuvor die PDRIVE-Parameter der Zieldiskette auf SS/SD/40 Tr. gesetzt hat, spart man später eine Menge Zeit.
2. Man fülle die Diskette mit Dummyfiles. Dabei hilft folgendes kurze Programm:

```
10 CLEAR1000 : CLS
20 CMD"FREE
30 INPUT"Wieviele Fileeinträge sind im Directory der Diskette im Laufwerk 1 noch frei";FE
40 FOR X=1 TO FE
50 X$ = STR$(X) : Y = LEN(X$)-1
60 FE$ = RIGHT$(X$,Y)
70 CM$ = "CREATE,FILE"+FE$+"/DUM:1,REC=5"
75 PRINT$256,CM$
80 CMD"CM$
90 NEXT X
100 CMD"S=DIR 1
```
3. Mit SUPERZAP kann man nun feststellen, welche der eröffneten Files an der Stelle steht, an der normalerweise ein Systemfile erscheinen sollte.
4. Die Dummyfiles haben durch den Zusatz zum CREATE-Befehl REC=5 schon die richtige Länge. Den richtigen Inhalt bekommen sie durch das Überkopieren des leeren SYS15/SYS (enthält im Model3-NewDOS die Befehle FORMS und SETCOM).
5. Die Dummyfiles bekommen nun durch den RENAME-Befehl noch ihren richtigen Systemfilenamen.
6. Zu guter Letzt kann man die nicht benötigten Dummyfiles weglöschen.

Achtung: wer das Betriebssystem GDOS (oder HDOS) sein eigen nennt, kann sich diese umständliche Prozedur sparen. GDOS/HDOS beinhaltet die kompletten Systemfiles, so daß man mit folgender, erheblich kürzerer Methode auskommt:

1. eine G/H-DOS-Systemdiskette mit dem oben schon einmal genannten Befehl COPY,0,1,,FMT,CBF,/SYS kopieren.
2. Die GDOS-Originaldisk durch ein NewDOS ersetzen und mit dem gleichen Befehl, die unter NewDOS vorhandenen Systemfiles auf das Backup des G/H-DOS aufkopieren.

Man erhält dadurch ein NewDOS mit allen Systemfiles, das zudem den Vorteil hat, daß man die interessanten GDOS-(noch besser HDOS) Funktionen schon auf der Diskette hat und nur noch in das vergrößerte Library des NewDOS einzubauen braucht.

PDrive-Identifikation für NewDOS80

Nachdem wir nun alle Bedingungen erfüllt haben, die vor der Erweiterung des NewDOS-Befehlssatzes zu erfüllen waren, können wir mit dem Einbau der verschiedenen Funktionen beginnen. Wir fangen mit der am schnellsten und einfachsten zu implementierenden Funktion an, der Identifizierungsroutine für PDriveangaben ID.

Die Funktion stammt aus dem H-DOS unseres Hacktory Arnulf Sopp und ihre Fähigkeit PDriveangaben zu erkennen geht so weit, daß sie sogar die Option TI=xxL erkennt. Diese Option wird benötigt, wenn man 40 Track-Disketten auf 80 Track-Laufwerken lesen und schreiben will und bereitet den sonstigen im Umlauf befindlichen Einstellroutinen unlösbare Schwierigkeiten. Der Aufruf erfolgt mit dem Syntax ID,n<,A>, wobei "n" die Nummer des Laufwerks darstellt, welche die zu testende Diskette enthält. Wird die Option ",A" verwendet, schreibt die Routine die richtigen PDriveangaben gleich auf die Systemdiskette und macht sie wirksam.

Der Einbau dieser sehr wertvollen Funktion ist deshalb so einfach, weil sie praktisch aus dem HDOS ohne jede Änderung übernommen werden kann. ID liegt im HDOS im SYS22/SYS und wird folgendermaßen im Library eingetragen: 49 44 80 FB 00 hex

I D . . . ASCII

Vorgehensweise:

1. SYS22/SYS von HDOS auf SYS22/SYS von NewDOS80 (vorher eröffnen!!!) kopieren.
2. Mit SUPERZAP im SYS1/SYS Sektor 3 rel. die oben aufgeführten Bytes ab dem Byte C6 rel. eintragen.

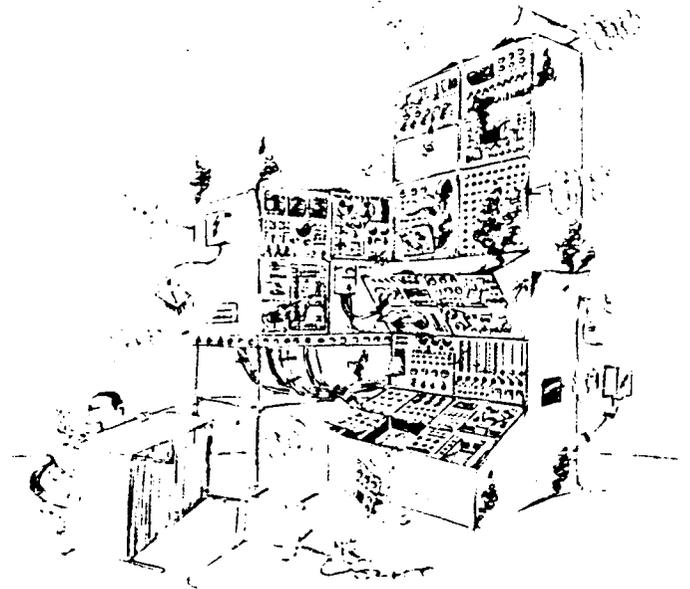
Achtung: Nach dem Befehl WRDIRP und dem zugehörigen Sprung- und Aufrufbyte folgt noch das Schlußbyte (00) des Befehls. Dieses Bytes darf nicht überschrieben werden! Die Bytefolge in der Reihe "C0 - CF" des Sektordumps muß lauten:

C0 49 52 50 80 53 00 49 44 80 FB 00 00 00 00 00
I R P S . I D

Nachdem diese Änderungen durchgeführt sind, hat man zum erstenmal das Gefühl, die Arbeit, die man sich mit dem "Platz machen" im SYS1/SYS und der "Eröffnung" der Systemfiles gemacht hat, hat sich gelohnt. Und ich kann euch versichern, es werden noch eine ganze Menge solcher Erfolgserlebnisse folgen, wenn ihr euch selbst einmal daran macht, interessante und nützliche Funktionen (auch wenn sie durch andere Programmierer entwickelt wurden und praktisch "geklaut" sind*) ins NewDOS einzubauen. Viel Spaß dabei wünscht euch, euer

Karl-Heinz Obermann

* Es wird sicher niemand etwas dagegen haben, wenn man Programme und Routinen, die man mit einem DOS zusammen erworben hat (z.B. GDOS + NewDOS -> an mich = HDOS von Arnulf Sopp!!!) in ein anderes DOS einbaut, solange man dann damit nicht als Eigenentwicklung hausieren geht oder sogar Geld zu machen versucht. Aber wie ich die Clubmitglieder kenne, wird so etwas sicher nicht der Fall sein!



»Hundertmal habe ich dir gesagt: Keine widersprüchlichen Fragen stellen!«

☹☹ Hast du dir erst Wissen erworben, so weißt du, was dir fehlt. ☹☹

Nachdem man das SYS1 geändert und sich im Library Platz für neue Befehle geschaffen hat, ist man bestrebt möglichst viele Funktionen ins DOS einzubauen. Dabei sollte man jedoch bedenken, daß die 36 freien Libraybytes sehr schnell gefüllt sind und man deshalb bei der Wahl der einzubauenden Befehle ein paar Dinge beachten sollte.

Die meisten Funktionen, die das DOS zur Verfügung stellt, könnte man auch durch "normale" Maschinenprogramme erledigen lassen. So ist z.B. die FORMAT-Routine durchaus als /CMD-File denkbar (siehe das alte TRSDOS 2.3). Sinn macht der Einbau ins Betriebssystem eigentlich nur, wenn man die Funktion von einem Programm oder von MiniDOS aus aufrufen will. Bei den GDOS-Funktionen DDE (DiskDaten Editor) und LWT (Laufwerktest) ist das fraglich!

Wer will schon mitten in einem BASIC-Programm einen Laufwerktest machen!? Ich glaube nicht, daß es dafür eine Anwendung gibt. Es genügt also, den Laufwerktest als CMD-File zu übernehmen und nicht ins DOS einzubauen.

Beim DiskDatenEditor verhält sich die Sache etwas anders. Hier gäbe es manchmal durchaus sinnvolle Anwendungen, die einen Einbau ins Library rechtfertigen würden. Leider wird dies durch den Aufbau von DDE verhindert. Das Programm benutzt nämlich den Bereich um 5500h als Sectorbuffer. Dieser Bereich wird aber auch von vielen Programmen (so auch von BASIC) benutzt. Dadurch ist ein Aufruf von DDE aus dem BASIC zwar möglich, die Rückkehr endet jedoch immer in einem Systemabsturz! Also ist auch die DDE-Funktion nicht als DOS-Befehl, sondern als CMD-File zu übernehmen.

Der Einbau von DDE
DDE ist in seinem Systemfile (HDOS SYS15/SYS) ganz allein und ist aus diesem Grunde wie ein ganz normales Maschinenprogramm aufgebaut. Um es ins NewDOS zu übernehmen, muß man es nur von einer HDOS-Diskette mit dem Befehl:

```
COPY SYS15/SYS:1 TO DDE/CMD:0
```

herüberziehen. Es läuft auf Anhieb und benötigt keine weiteren Zap's.

Der Einbau von LWT
LWT macht da schon etwas mehr Probleme. Es muß nämlich, um als CMD-File lauffähig zu sein, von seinem Lade-/Laufbereich 4D00h-51E7h nach 5D00h-61E7h (Startaddr. 5D7Ch) verlegt werden. Außerdem muß man, wenn man ein TRS80-Modell besitzt, zwei Bytes im Sektor 0 rel. ändern. Sonst stimmt die angezeigte Geschwindigkeit nämlich nicht, die Laufwerke laufen immer zu langsam. Zusätzlich sollte man die Maßeinheiten von der Anzeigeskala entfernen, da sie nicht mehr stimmen.

Die Verlegung in den höheren Bereich erreicht man am leichtesten, indem man sich das SYS23/SYS des HDOS von einem Disassembler als Sourcecode auf Diskette schreiben läßt, die DRG-Adresse ändert und wieder, jetzt als LWT/CMD, assembliert. Die richtige Startadresse kann man nachträglich mit DDE ändern (letzter Sector von LWT/CMD, Bytes FE und FF von 005Dh in 7C5Dh ändern).

Damit die Anzeige für den TRS80 passt, müssen im rel. Sektor 0 die Bytes C6 und C7 von C85B in 2B59 geändert werden. Die Löschung der Maßeinheiten der Anzeigeskala erfolgt im relativen Sektor 1 im Bereich 20h bis 100h.

HDOS SYS23/SYS SECTOR 0 REL.

Table with 16 columns and 16 rows of hex data and ASCII characters, representing Sector 0 of HDOS SYS23/SYS.

NEWDOS80 LWT/CMD SECTOR 0 REL.

Table with 16 columns and 16 rows of hex data and ASCII characters, representing Sector 0 of NEWDOS80 LWT/CMD.

HDOS SYS23/SYS SECTOR 1 REL.

Table with 16 columns and 5 rows of hex data and ASCII characters, representing Sector 1 of HDOS SYS23/SYS.

NEWDOS80 LWT/CMD SECTOR 1 REL.

Table with 16 columns and 5 rows of hex data and ASCII characters, representing Sector 1 of NEWDOS80 LWT/CMD.

Viel Spaß beim Zapen und Arbeiten mit DDE und LWT, wünscht euch, Ralf Rühl Obermann

Ein Byte und seine Folgen

Welch seltsame Symptome bei der Änderung eines einzelnen Bytes im SYS1/SYS auftreten können, mußte ich vor kurzem am Beispiel des Programmes HELP/CMD erfahren.

Nachdem ich im System 1 das Abschlußbyte der DOS-Ready-Meldung von OD in O3 geändert hatte (der Cursor steht dann hinter der Meldung, statt eine Zeile tiefer), bekam ich beim Aufruf von HELP,Befehl immer nur den freundlichen Hinweis, daß HELP die NEWDOS-Befehle erklärt und mit dem Syntax HELP,Befehl aufgerufen wird. Alle Bemühungen, dem Programm etwas anderes zu entlocken, scheiterten!

Eine kurze Durchsicht des disassemblierten Listings deckte den Grund für das seltsame Verhalten des Programms auf. Das hinter dem HELP eingegebene Suchargument für den Erklärungstext wird auf folgende Art gerettet:

```
LD      HL,(4020H)      'Cursoradresse -> HL
LD      DE,003BH      'Zeilenlänge (64)-"HELP,"(5)=59
SBC     HL,DE          'HL-DE=Beginn des Arguments
```

Durch die Änderung im Sys1 verschiebt sich der abziehende Wert erheblich. Er beträgt dann:
Zeilenlänge(64)-"HELP,"(5)-DOS-Meldung(unterschiedlich).
Beispiel: DOS-Ready: HELP,AUTO abziehender Wert= 31H!

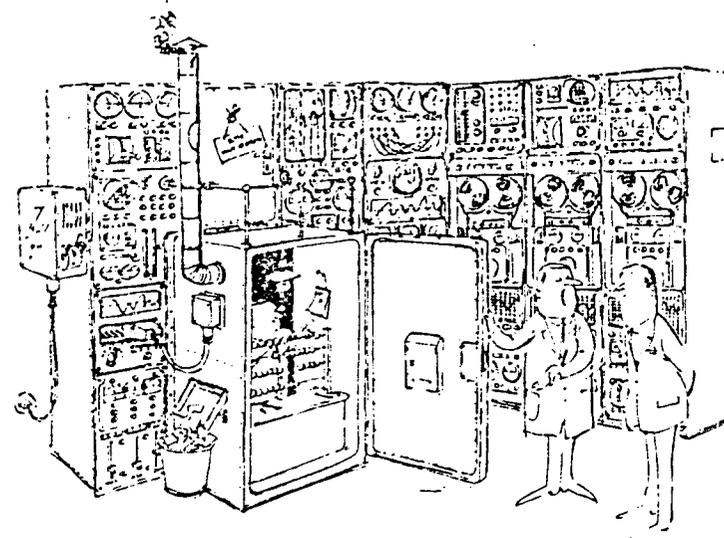
Das betreffende Byte findet man im Sector 0 (relativ) als Byte 18 (relativ) in HELP/CMD. Dieses Byte muß auch angepaßt werden, wenn der Name des Programms (HILF/CMD) so geändert wird, daß er mehr (oder weniger) als vier Buchstaben hat!

Wie man sieht, können kleinste Änderungen am DOS oder in Programmen unerwartete, teilweise fatale Folgen nach sich ziehen. Ich bitte also jeden, der in dem von mir geänderten NewDOS einen Fehler findet, mir diesen mitzuteilen, damit dieser schnellstmöglich behoben werden kann!
Im voraus vielen Dank!!!

Karlheinz Obermann

DOS> HELP,ATTRIB

```
000000: 0102 006C 0000 0000 0000 00DD 2100 00DD ...1.....!...
000010: 3931 F0DF 2A20 4011 3600 AFED 5211 006C 91...* 5.6. R..1
000020: 0606 7EFE 2028 07CD 9B6C 2313 10F4 AF12 ..B. (...1#...
000030: 21B0 6C11 F497 EBAF ED52 EBD5 C13E 25ED !...R...>%
000040: B1C2 4F6C 1100 6C1A B7CA 646C BE20 E423 .01..1..dl |.#
000050: 131B F42A 2040 11D4 FF19 EB21 A76C 010B ...* 5.....!|..
000060: 00ED B0DD F9C3 2D40 010A 003E 25ED B923 ...-5...>%.#
000070: 23CD C901 0140 0011 003C D5CD B16C 23D1 #...5...<..1#
000080: EB09 EB18 F57E FE0D CBFE 2528 0512 1323 .....B....%(...#
000090: 18F3 D1D1 EB09 0922 2040 DDF9 C32D 40CB ..... " 5...-5.
0000A0: F7CB AF12 C900 0000 0000 0020 2020 4552 ::... .. ER
0000B0: 524F 5220 2020 2020 2548 454C 500D 4865 ROR %HELP.He
0000C0: 6C70 2069 7374 2065 696E 2048 696C 6673 lp ist ein Hilfe
0000D0: 7072 6F67 7261 6D6D 2066 7565 7220 4E45 programm fuer NE
0000E0: 5744 4F53 2D38 3020 322E 3020 2E0D 416C WDOS-B0 2.0 ..Al
0000F0: 6C65 2042 6566 656B 6C65 2064 6965 2062 1e Befehle die b
```



»... und das ist das eigentliche Gehirn der Anlage!«

HRG-Pixel's aus dem BASIC !!

Vor kurzem befasste ich mich mit dem Problem der HRG-Ansteuerung vom Basic aus, um ein paar Linien für eine Eingabemaske setzen, ohne einen HRG-Treiber zu laden.

Nachdem mir zu diesem Thema noch Verschiedenes unklar war, habe ich mich dazu im Club umgehört, und fand beim Arnulf schon einige ausgearbeitete Werke, die ich Euch nach dieser kleinen Einleitung vorstellen möchte.

Zuerst aber noch folgendes:

Das Bearbeiten einer ganzen Grafikseite (immerhin 12K) vom Basic aus, ist eine doch recht langwierige Angelegenheit. So dauert zum Beispiel das Löschen der HRG mit dem schnellsten Basicprogramm zirka 2 Minuten. Wer aber nur schnell eine Linie ziehen- oder ein Kästchen zeichnen will, dem seien folgende Programmlistings, am Ende dieses Artikels, an's Herze gelegt.

Nachdem die Hauptprogrammteile schon vom Arnulf geschrieben waren, blieb mir eigentlich nur noch übrig einige Features dazu zu setzen und das Ganze im Clubinfo zu veröffentlichen.

Ich hoffe, damit noch einigen von Euch gedient zu haben, zumal die HRG doch schon weit verbreitetes Zubehör unseres Computers ist.

JENS NEUEDER

Es folgt nun ein Auszug eines Briefes von Arnulf an mich. Danach kommen drei Beiträge zu der "HRG-Pixel", die den Brief und sich gegenseitig noch ergänzen. Zum Abschluß des Ganzen die versprochenen Listings, die als Programme unter Pix1/BAS und Pix2/BAS in der Clubbibliothek zu finden sind.

Das Ganze ist nicht besonders schwierig. Verstehen muß Du eigentlich nur, wie man ein ganz bestimmtes Byte innerhalb des externen HRG-Speichers adressiert. Der Speicher hat 12 kB, daher gehen seine Adressen von 0000-2FFF. Da man auf einen Port nur höchstens FF ausgeben kann, nicht aber z. B. 1AB3 oder so, muß die HRG-Adresse in zwei Bytes aufgeteilt werden. Das MSB ist Adresse/256, das LSB AdresseAND255. Das LSB wird auf Port 2 ausgegeben, das MSB auf Port 3. Jetzt erst kann man ein Byte über Port 5 in die HRG einschreiben (OUT5,byte) oder über Port 4 aus der HRG lesen (byte=INP(4)).

Jedes Byte steht für 6 Punkte auf dem Bildschirm. Die beiden höchstwertigen Bits (also der 7. und 8. Punkt) sind auf dem Bildschirm nicht sichtbar. Wenn Du nun an einer bestimmten HRG-Adresse einen bestimmten von 6 Punkten setzen willst, muß Du wissen, welchen Bit im

ausgegebenen Byte dieser Punkt entspricht. Ganz links ist der 1. Punkt, der mit dem Bit 0 gesetzt wird. Das bedeutet, daß die binäre Zahl (Dein Byte) umgekehrt gedacht werden muß: Die niedrigste Stelle ist links, die höchste rechts. In einem Byte entsprechen die Bits 0-5 (nur diese 6 sind wichtig) den Zahlen 1, 2, 4, 8, 16, 32. Also: Wenn Du über Port 5 das Byte 16 aus gibst, nachdem Du über die Ports 2 und 3 die Adresse definiert hast, dann wird der 5. Punkt von links gesetzt. Um nun nicht bei der Ausgabe eines Punktes die übrigen Punkte an dieser HRG-Adresse zu löschen, ist es wichtig, daß Du so vorgehst: X=INP(4):Y=16:Z=XORY:OUT5,Z oder: OUT5,INP(4)OR16.

Für das Rücksetzen eines Punktes gilt das Entsprechende: Zunächst muß Du das Komplement des Bytes finden (das zu löschende Bit steht auf 0, alle anderen auf 1). Bevor ich Dir jetzt lange Opern über das binäre Einerkomplement vorsinge, gebe ich Dir eine Tabelle wie die obige, aber mit 0 im angesprochenen Bit und 1 in den übrigen (wieder für Bit 0-5): 254, 253, 251, 247, 239, 223. Jetzt nehmen wir wieder das obige Beispiel. Der 5. Punkt von links wird diesmal aber gelöscht: X=INP(4):Y=239:Z=XANDY:OUT5,Z oder: OUT5,INP(4)AND239. Statt 16 steht nun 239, statt OR steht AND, weil ein Bit ausmaskiert werden muß.

Oben sprach ich das Adressieren an. Hier haben wir zufällig eine Erleichterung. Die LSB der HRG-Adressen stimmen mit denen des normalen Bildschirms genau überein. Bei den ersten und allen weiteren vier (normalen und HRG-) Bildschirmzeilen geht das LSB von 0 bis FF, also in der 1. Zeile von 0-63, in der 2. von 64-127, in der 3. von 128-191 und in der 4. von 192-255. So kannst Du Dir das LSB aus der Video-Display-Map im TRG-90-Handbuch (Appendix E) herausfiedeln. Mit dem MSB ist es auch nicht viel komplizierter: Die Bildschirmadressen gehen von 3C00-3FFF, die HRG-Adressen von 0000-2FFF. Das ist ein Unterschied von 3C00 (15360 dez).

Ich fange einen neuen Absatz an, aber nur aus optischen Gründen. Nun kannst Du beim Setzen oder Rücksetzen eines Punktes in der Video-Display-Map die Adresse heraussuchen und 15360 subtrahieren. Das ist dann die Adresse in der HRG, wenn der Punkt in der obersten Punktzeile dieser Anzeigestelle gemeint ist. Meinst Du aber eine tiefere Punktzeile, dann addiere pro Punktzeile 1 kB (= 0400 hex oder 1024 dez) hinzu. Das ist dann die korrekte HRG-Adresse. Dazu paßt dann der Schmus aus meinem Artikel "Die HRG 1b programmieren".

Himmel, gerade las ich mir das Vorige noch einmal durch und verstand kaum eine Silbe! Es ist schon ein bißchen vigeliensch, es wi hier op platt seggt. Lies Dir bitte die drei Artikel je dreieinhalbmal durch, jeweils durch ein Fils oder Alt unterbrochen, und lies anschließend diesen Brief laut Deiner Katze vor. Wenn dann noch Fragen offen sind, bastele einen Molotow-Cocktail, löse eine Rückfahrkarte nach Eitorf/Sieg und statte der Fa. RE-Elektronik einen Besuch ab. Grüße von mir!

Die HRG 1b programmiert

Die Zusatzplatine für hochauflösende Graphik hat innerhalb des Clubs bereits etliche Freunde gefunden. Vielleicht sind einige von Euch von der chaotischen Anleitung immer noch so gefrustet wie ich es war, als ich mir gestern nach dem Einlöten die Früchte des soeben abgewischten Schweißes gönnen wollte. Mit der gegen Aufpreis erhältlichen Software geht es zwar ganz gut, aber die blockiert mal wieder das Hirn. Außerdem bietet sie überwiegend zusätzliche BASIC-Befehle, so daß der Assembler-Programmierer letztenendes mit seiner HRG alleingelassen ist. Nicht ganz, siehe Fotokopie der einzigen Seite der Anleitung, die überhaupt (obgleich in BASIC) auf die direkte Programmierung der Karte eingeht. Wer aber nach Lektüre dieser Seite schlauer ist als zuvor, verdient Bewunderung.

So schrieb ich eine Stelle dieser Seite zunächst von hinten nach vorne, um mit der sonderbaren Bitphilosophie mehr anfangen zu können:

```

>          Port 3 (HOB)      < >      Port 2 (LOB)      <
Bit:  7  6  5  4  3  2  1  0  7  6  5  4  3  2  1  0
      > n.b.< > Bereich C < > Bereich B < > Bereich A <
  
```

Jetzt liest sich das Ganze wie eine 16-Bit-Zahl und beginnt, einen Sinn zu bekommen. Diese Zahl kann man sich sehr wohl als Adresse eines Bytes im Graphikspeicher vorstellen. Sie ist es auch, wie die selbstquälische wiederholte Lektüre dieser Anleitung (besser: Verschleierung) und ein paar Stunden an der Tastatur ergaben. Das Resultat war eine Tabelle, die auf der nächsten Seite oben wiedergegeben ist.

Das Video-RAM kann man sich wie eine Perlenschnur vorstellen, wo sich Byte an Byte fügt. Mit dem RAM der Graphikkarte ist es nicht anders. Allerdings belegen die Bytes des Video- und des Graphik-RAMs auf dem Bildschirm physikalisch leider nicht denselben Platz. Das ist ja auch logisch, wenn 12 kB in den Raum passen sollen, den normalerweise 1 kB belegt. Außerdem ist die Reihenfolge nicht die gewohnte, in der der Inhalt des Graphikspeichers ausgegeben wird. Das erste kB wird genau da abgelegt, wo es auch das Video-RAM tut, aber natürlich nur die oberste Punktreihe einer jeden Bildschirmstelle. Dann wiederholt sich dieser Vorgang, wobei nach jedem Kilobyte die nächste Punktreihe angesteuert wird, bis alle 12 Reihen geladen und die 12 kB des Graphikspeichers abgearbeitet sind.

Deshalb ist für die wiedergegebene Tabelle folgende Erläuterung erforderlich: Je nach Punktreihe muß der angegebenen Zahl ein Vielfaches von 0400 (1024 dez.) hinzugezählt werden. Oder einfacher: Das MSB der Adresse wird nach jeder Punktreihe um 4 erhöht.

Die gewohnten Videoadressen gehen von 3C00-3FFF, die Graphikadressen intern von 0000-2FFF. Betrachtet man nur die oberste Dotreihe, so gehen sie von 0000-03FF! Das ist gleich Videoadresse minus 3C00, weiter nichts! Da das LSB (auf Port 2) exakt mit dem des Video-RAMs übereinstimmt, haben wir es mit simpelster Mathematik zu tun, auch wenn die Anleitung eher an eine schöngeistige Disziplin denken läßt. Freilich darf man nicht vergessen, dem MSB je nach seiner Dotreihe 00, 04, 08, 0C usw. hinzuzuaddieren.

Die zweite Tabelle auf der nächsten Seite gibt ein Beispiel für eine einzelne Bildschirmstelle. Es ist die Stelle 3D00, also das erste Byte der 5. Zeile. Die zweite Spalte der Tabelle zeigt die Nachbarstelle zur Rechten (3D01). Daraus erkennt man, daß das LSB wie beim normalen Video-RAM behandelt werden kann. Ganz rechts ist die letzte Stelle der Zeile wiedergegeben. Der Anwender braucht nun nur noch zu wissen, welche der sechs möglichen Punkte er in der betreffenden Dotreihe setzen möchte. Jedem Punkt ist ein Bit von 0-5 zugeordnet. Dieser Wert geht über Port 2 ins Graphik-RAM, bzw. ... ausgelesen werden.

Zugegeben, diese Erläuterung ist auch nur denjenigen Lesern auf Anhieb verständlich, die mit Binär- und Sedezimalzahlen vertraut sind. Das trägt mir voraussichtlich wieder Senge von Seiten der BASIC-Lobby ein. Aber der BASIC-User hat mit der lieferbaren Software brauchbare Werkzeuge, so daß die zitierte Seite der HRG-Anleitung für ihn ohnehin von geringem Interesse ist.

An der HRG-Anleitung ist weitere Kritik angebracht! Auf der abgebildeten Seite wird im ersten Absatz gesagt, wie die Ports anzusteuern sind (Input bzw. Output). Tatsächlich aber reagieren alle 6 Ports bei einem Input so, als hätte ein Output von FF (255 dez.) stattgefunden. Man kann demnach die Graphik auch durch Lesen der Ports 0 und 1 ein- oder ausschalten. Das ist bei den G-DOS-Befehlen PORT und INFO leider fatal, denn sie lesen alle Ports. Ergebnis: Die prompt eingeschaltete Graphik übermalt die Anzeige, derentwegen wir ja schließlich den Befehl eingegeben haben.

So ist denn zwar die HRG-Platine nicht ganz ohne Tücken (die Anleitung ist sogar die Tücke selber), aber letztenendes kocht sie auch nur mit Wasser und kann gehandhabt werden.

Und wie sie gehandhabt werden kann! Seit den obigen Zeilen sind zwei Tage vergangen, und vor einer Stunde war eine Modifikation meiner Graphik-Hardcopy-Routine fertig, die bei <JKL> auf Wunsch auch noch die hochauflösende Graphik mit ausdrückt, nach Lust und Laune in positiver oder negativer Darstellung, in jedem Falle gemischt mit Text und der normalen Pixelgraphik.

Das Video-RAM und die zugeordneten HRG-Adressen:

1. Bildschirmzeile:	0000	0001	0002	003D	003E	003F
2. Bildschirmzeile:	0040	0041	0042	007D	007E	007F
3. Bildschirmzeile:	0080	0081	0082	00BD	00BE	00BF
4. Bildschirmzeile:	00C0	00C1	00C2	00FD	00FE	00FF
5. Bildschirmzeile:	0100	0101	0103
12. Bildschirmzeile:	02FD	02FE	02FF
13. Bildschirmzeile:	0300	0301	0302	033D	033E	033F
14. Bildschirmzeile:	0340	0341	0342	037D	037E	037F
15. Bildschirmzeile:	0380	0381	0382	03BD	03BE	03BF
16. Bildschirmzeile:	03C0	03C1	03C2	03FD	03FE	03FF

Die 5. Bildschirmzeile (3D00 - 3D3F) mit HRG:

1. Dotreihe:	0100	0101	0102	013D	013E	013F
2. Dotreihe:	0500	0501	0502	053D	053E	053F
3. Dotreihe:	0900	0901	0902	093D	093E	093F
4. Dotreihe:	0D00	0D01	0D02
9. Dotreihe:	213D	213E	213F
10. Dotreihe:	2500	2501	2502	253D	253E	253F
11. Dotreihe:	2900	2901	2902	293D	293E	293F
12. Dotreihe:	2D00	2D01	2D02	2D3D	2D3E	2D3F

Für die hochauflösende Graphikplatine HRG 1b gibt es etliche Treiber, die eine große Menge neuer Befehle zu ihrer Ansteuerung anbieten. Viele dieser Features sind ebenso leistungsfähig wie überflüssig. Im rauhen Alltag geht es darum, z. B. irgendeinen Funktionsgraphen zu zeichnen oder eine Balken- oder Tortengraphik. Das kann der Microsoft-Interpreter schon lange; es brauchen nicht große RAM-Bereiche, meist im kostbaren Hinem, durch Treibersoftware verschlissen zu werden.

Das hier vorgestellte BASIC-Programm (jawoll!) ist ein vollständiger Treiber für solche HRG-Funktionen, die den Befehlen SET, RESET und POINT entsprechen. Die Koordinaten X und Y (für 384*192 Punkte) werden im Target-Programm (hier Zeilen 10-40) geladen. Die Auswahl einer dieser drei Funktionen erfolgt durch Ansprung eines der drei Unterprogramme ab Zeile 10000. Das UP an 40000 ist allen drei gemeinsam und wird später ausführlich erklärt werden. Es ist quasi die Seele des Ganzen.

Zu den Zeilen 10-40 möchte ich weiter nichts sagen. Da wird halt eine Sinuskurve gemalt (Abb.). Diese Anwendung ist vollkommen willkürlich, damit die Wirkung des Treibers ab Zeile 10000 demonstriert werden kann. Erwähnenswert wäre nur der DOS-Befehl CLS,6 in Zeile 10: Er ist eine Spezialität von H-DOS. Mit NEWDOS oder G-DOS wäre es auch mit BASIC-pur gegangen, aber wozu Zeit vergeuden?

Die Zeile 40000 errechnet aus den Koordinaten X und Y die physikalische Lage eines Punktes innerhalb des HRG-Speichers. Dazu muß zunächst X durch 6 dividiert werden, weil je sechs benachbarte Punkte in Wirklichkeit zu einem Byte gehören. Erst jeweils sechs Punkte später wird wieder ein neues Byte adressiert. Der Rest der Division (B) bestimmt das Bit und damit den Punkt in diesem Byte. Aber das Integer des Quotienten ist der Offset der Adresse nach rechts, weil die HRG-Adressen in waagerechter Richtung mit denen des normalen Bildschirms parallel laufen (die LSB der Adressen sind identisch).

Für die senkrechte Richtung wird zunächst Y durch 12 dividiert. Zwei übereinanderliegende Punkte sind physikalisch nämlich gar nicht benachbart, sondern liegen 1 kB voneinander entfernt: 12 kB hat der HRG-Screen, es gibt pro normaler Bildschirmstelle 12 Punktzeilen. Der Rest der Division Y2 (ähnlich B für die X-Richtung) zählt die Punktzeilen innerhalb einer normalen Bildschirmstelle nach unten ab, bedeutet also je ein volles kB. Die physikalische Stelle des anzusprechenden Bytes im HRG-RAM (Z) errechnet sich daher aus Y2 Kilobytes plus Y1 Bildschirmzeilen plus Offset nach rechts (X1). Diese Adresse muß nun in ein Habyte (ZM) und ein Lobyte (ZL) unterteilt werden, um sie über die Ports 2 und 3 anzusteuern. Anschließend wird noch der binäre Wert des Bits (des Punktes) errechnet (C gleich 2 hoch B). Schließlich wird für die spätere logische Verknüpfung noch A mit dem bisherigen Wert dieses Bytes geladen (über Port 4). Dieses ziemlich komplizierte Unterprogramm ist allen drei Funktionen gemeinsam.

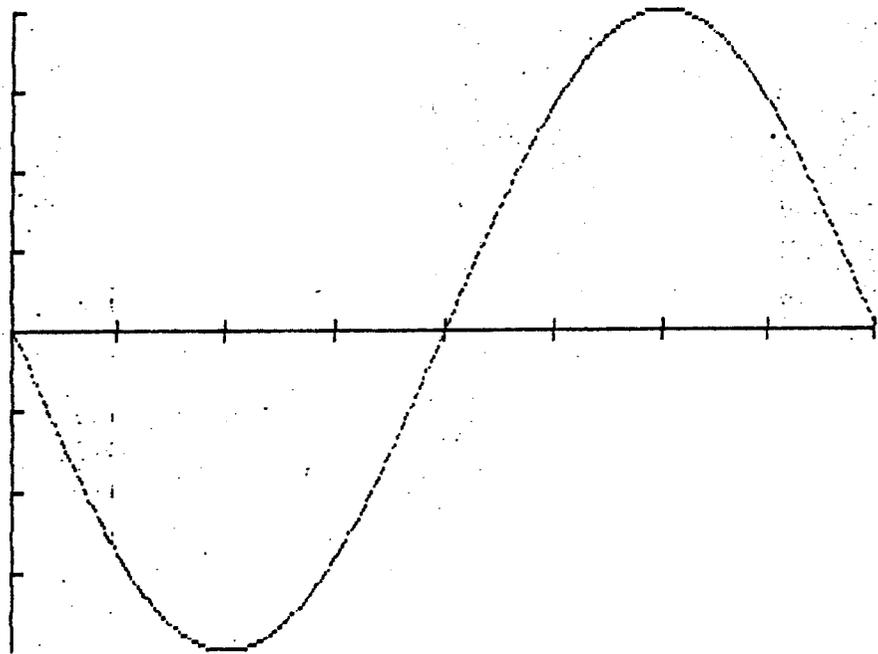
Die SET-Routine in Zeile 10000 oderiert das neue Bit mit dem bisherigen Wert und gibt das Resultat als neues Bitmuster auf den Port 5 aus. Die RESET-Routine (20000) undiert den alten mit dem Komplement des neuen Wertes und gibt das Resultat aus. So verschwindet dieses Bit. Die POINT-Routine (30000) ordnet der Variablen D das Wahrheitsflag 0 (falsch, Punkt nicht gesetzt) oder -1 zu (wahr, Punkt gesetzt), genau wie bei der Klötzchengraphik.

Dieser Treiber ist so komplex, daß ich ihn selbst kaum kapiere. Wie man weiß, ist BASIC weder meine Leidenschaft noch meine Stärke. Deshalb wäre ich dankbar für Anregungen, wie man das Programm einfacher und vielleicht auch schneller machen könnte. Ein Bedarf ist sicherlich vorhanden, denn wer möchte schon z. B. ein Ungetüm wie GRAPE n, um ir-

gendwo einen lausigen Punkt zu setzen oder eine simple Kurve zu zeichnen wie die abgebildete?

Arnulf Sopp

```
10 CLS: CMD"CLS,6": OUT1,255%: DEFINT A-Z'
    Bildsch. u. Graph. löschen, Gr. einschalten, nur Integer
20 Y=96%: FOR X=0 TO 383%: GOSUB10000: NEXT: FOR X=47 TO 383 STEP 48: FOR Y=93 TO 99%:
    GOSUB10000: NEXT: NEXT'
    Abszisse mit Teilungen ziehen
30 X=0: FOR Y=0 TO 191%: GOSUB10000: NEXT: FOR Y=0 TO 191 STEP 24: FOR X=1 TO 5%: GOSUB
    B 10000: NEXT: NEXT'
    Ordinate mit Teilungen ziehen
40 FOR X=0 TO 383%: Y=SIN(X/61%)*95.5+95.5: GOSUB10000: NEXT: END' Sinusgraphen zie
    hen
10000 GOSUB40000: OUT5,A OR C: RETURN'
    SET-Unterprogramm
20000 GOSUB40000: OUT5,A AND (NOT C): RETURN'
    RESET-Unterprogramm
30000 GOSUB40000: D=((A AND C)<>0%):RETURN'
    POINT-Unterprogramm
40000 X1=INT(X/6%): B=X-X1*6%: Y1=INT(Y/12%): Y2=Y-Y1*12%: Z=Y2*1024%+Y1*64%+X1: ZM=Z/
    256%: ZL=Z AND 255%: OUT2,ZL: OUT3,ZM: C=2^Z*B: A=INP(4): RETURN'
    gemeins. Unterpr., hauptsächlich Koordinaten bestimmen
```



Den HRG-Speicher löschen

In meinem Artikel "Die HRG 1b programmieren" versuchte ich aufzuzeigen, daß diese Karte nahezu ebenso einfach adressiert wird wie irgendein RAM-Bereich, denn sie hat intern ebenfalls gewöhnliche 16-Bit-Adressen. Jener Beitrag schwebte ein wenig im luftleeren Raum, denn es blieb bei der Theorie. Hier ein wenig Alltagspraxis. Das im Anschluß an diesen Text gelistete Assemblerprogramm löscht den HRG-Speicher, eine Funktion, die man zu Beginn einer jeden neuen Graphik braucht.

Die internen Adressen der HRG reichen von 0000-2FFF. Das bedeutet folgendes: Das höchste MSB einer Adresse beträgt 2F. Da in der Schleife LOOP1 der Akku dekrementiert wird, noch bevor er auf den Port geht, muß der Anfangswert 1 höher, also auf 30 gesetzt werden. Das geschieht im ersten Befehl LD BC,3002. Das MSB+1 steht nun im Register B, gleichzeitig lädt dieser Befehl C mit der Portadresse 02.

Im nächsten Schritt wird der Akku mit dem MSB geladen, um die Ausgabe auf den Port 03 vorzubereiten. Sie erfolgt nach dem oben erwähnten Dekrementieren. Mit dem LSB, das auf Port 02 ausgegeben werden muß, ist es etwas komplizierter. Genauer gesagt ist es genauso simpel, um jedoch Speicherplatz zu sparen, geht das Programm einen etwas gewundenen Weg:

Das LSB muß für jedes MSB, also 30h-mal, die Werte FF-00 durchlaufen, um alle 256 LSBs zu erreichen. In diesem Falle wird mit 00 begonnen und über FF bis 01 gegangen. Der Anfangswert steht nach XOR A im Akku. Mit diesem Wert wird B geladen, weil dieses Register den Befehl DJNZ ermöglicht. Auf diese Weise werden ebenso rückwärts fortlaufend alle Werte erzeugt wie beim MSB. Sie gehen nach dem Eintritt in die Schleife LOOP2 auf den Port 02.

Der Akku bleibt in dieser Schleife immer auf 00. Dieser Code bedeutet für die HRG eine leere Anzeigestelle, also ein Blank. Es wird über den Port 05 ausgegeben. Die jeweilige Speicherstelle der HRG ist damit gelöscht. Nach 256 Durchläufen dieser Schleife wird das alte MSB vom Stack geholt, wo es seit dem Befehl PUSH BC lag. Ebenfalls mit DJNZ nimmt es nacheinander alle Werte von 30-01 an. Um auf die tatsächlich nötigen Werte 2F-00 zu kommen, ist der Befehl DEC A notwendig, der in Zeile 130 steht. Das war es bereits. Mit RET geht es ab ins Betriebssystem, sei es DOS oder was auch immer.

Das ist wieder einmal so ein Feature, das gleich bei Booten laufen sollte, um den Müll von Zufallscodes zu beseitigen, die sich nach dem Einschalten im Graphikspeicher befinden. Bei H-DOS ist das der Fall. Um aber eine Graphik, die im Speicher steht, nicht nach jedem BOOT neu erzeugen zu müssen, kann das Löschen mit der gedrückten Shifttaste auch verhindert werden. Das gilt nota bene für H-DOS, nicht für das hier vorgestellte Programm.

Arnulf Sopp

```

B000          00100      DRG      B000H      ;Adresse beliebig
B000 010230   00110 START LD      BC,3002H   ;B= MSB 30, C= Port 02
B003 78      00120 LOOP1 LD      A,B      ;A <- MSB der HRG-Adresse
B004 3D      00130      DEC      A          ;30 -> 2F usw.
B005 D303    00140      OUT      (03H),A    ;MSB der HRG-Adr. ausgeb.
B007 C5      00150      PUSH     BC        ;aktuelles MSB retten
B008 AF      00160      XOR      A          ;A <- 00 = 256
B009 47      00170      LD       B,A      ;LSB der HRG-Adresse
B00A ED41    00180 LOOP2 OUT      (C),B    ;ausgeben
B00C D305    00190      OUT      (05H),A    ;00 = Graphik-Blank
B00E 10FA    00200      DJNZ    LOOP2    ;LSB 100 -> 01
B010 C1      00210      POP      BC        ;altes MSB und Port
B011 10F0    00220      DJNZ    LOOP1    ;nächste 256 Adressen
B013 C9      00230      RET          ;z. B. ins DOS
B000          00240      END      START
00000 mal gepennt
34572 Zeichen verfügbar
    
```

Programmieren der HRG1b (a) mit Hilfe der Ports

Zur Ansteuerung der HRG1b werden folgende Ports benutzt:

Port 0:	Ausschalten des Grafikbildschirms	Output
Port 1:	Einschalten des Grafikbildschirms	Output
Port 2:	Untere Ansteueradresse (LOB)	Output
Port 3:	Oberer Ansteueradresse (HOB)	Output
Port 4:	Einlesen eines Grafikpunktes	Input
Port 5:	Setzen eines Grafikpunktes	Output

Mit Port 0 und 1 schalten Sie die Grafik ein bzw. aus, wie mit #OPEN und #CLOSE.

Beispiel: OUT 0,0 = #CLOSE OUT 1,0 = #OPEN

Mit Port 2 und 3 wird der Bildschirmpunkt adressiert, den Sie ansprechen wollen. Die Bits der beiden Ports sind wie folgt aufgeteilt:

	Port 2 (LOB)								Port 3 (HOB)							
Bit:	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
	> Bereich A								<> Bereich B <> Bereich C <> n.b.<							

Bereich A:	Bit 0-5	Port 2	Position der 64 normalen Spalten
Bereich B:	Bit 6+7 & 0+1	Port 2+3	Position der 16 normalen Zeilen
Bereich C:	Bit 2-5	Port 3	Position der 12 Zeilen pro Char.
n.b.	: Bit 6+7	Port 3	nicht benutzt

Mit Port 4 koennen Sie dann 6 nebeneinanderliegende Bildschirmpunkte einlesen. Benutzt sind Bit 0-5. Beispiel: X=INP(4) oder PRINT INP(4)

Mit Port 5 werden bis zu 6 nebeneinanderliegende Punkte gesetzt. Gebraucht werden nur Bit 0-5. Beispiel: OUT 5,63 ==> 6 Punkte

Beispiel: Sie wollen den Punkt 21,31 setzen.

#OPEN	= OUT 1,0	
#SET(21,31)	= OUT 2,131	Bit 0,1,7 gesetzt
	OUT 3,32	Bit 5 gesetzt
	OUT 5,4	Bit 2 gesetzt

Beispiel: Sie wollen wissen ob der Punkt 100,100 gesetzt ist

#POINT(100,100),X	= OUT 2,16	Bit 4 gesetzt
	OUT 3,22	Bit 1,2,4 gesetzt
	X=INP(4)	wenn in der Variablen X jetzt das Bit 4 gesetzt ist, dann ist der Punkt 100,100 vorhanden.

```

1 ' PIX1 /BAS          == DEMOVERSION ==
2   OUT1,0:           'Grafik anschalten
3   OUT0,0:           'Grafik ausschalten
4   OUT2,X:           'LSB-Adresse (0-255)
5   OUT3,X:           'MSB-Adresse (0-47)
6   OUT5,X:           'Ausgabewert (0-63) für 6 Pixel
7   X=INP(4):         'lesen des Ausgabewertes
10 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
11 'XX   Ändern der HRG vom Basic aus   XX   von
12 'XX   im zwölfzeilenstepp          XX   Arnulf SOPP
13 'XX   schnellste Möglichkeit        XX
14 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
15
CLS:OUT1,0:FORA=0TO47:OUT3,A:FORB=0TO255:OUT2,B:OUT5,0:NEXTB,A:OUT0,0
20 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
21 'XX   Ändern der HRG vom Basic aus   XX   von
22 'XX   der Reihe nach                XX   Jens NEUEDER
23 'XX   zeilenweise                  XX
24 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
25 CLS:A=0:OUT1,0
26 FORB=AT047STEP4:OUT3,B:FORC=0TO63:OUT2,C+D:OUT5,1:NEXTC,B
27 D=D+64:IFD=256THEND=0:A=A+1
28 IFA<4THEN2ELSEOUT0,0
30 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
31 'XX   Ändern der HRG vom Basic aus   XX   von
32 'XX   der Reihe nach                XX   Jens NEUEDER
33 'XX   spaltenweise                  XX
34 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
35 CLS:OUT1,0:FORC=0TO63:A=0:D=0
36 OUT2,C+D:FORB=AT047STEP4:OUT3,B:OUT5,8:NEXTB
37 D=D+64:IFD=256THEND=0:A=A+1
38 IFA<4THEN3ELSENEXTC:OUT0,0
40 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
41 'XX   Ändern der HRG vom Basic aus   XX   von
42 'XX   nach Bildschirmadressen       XX   Jens NEUEDER
43 'XX   (15360 - 16383)                XX
44 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
45 CLS:OUT1,0:P=-1:FORA=15360TO16383:FORC=0TO11:IFC=11THENP=63ELSEP=1
46 B=(CX1024)+A-15360:OUT2,BAND255:OUT3,B/256:OUT5,P:NEXTC,A:OUT0,0

```

Wo PRESSEFREIHEIT herrscht und jedermann lesen kann, da ist Sicherheit.
Thomas Jefferson

```

1 'HRG - Grafik vom Basic aus == DEMOPROGRAMM ==
2 '                               Anwendungen von Jens NEUEDER
3 'CMD*c1h/cmd"
5 CLS:CLEAR1000:DEFINT A-Z:OUT1,0
9 '===== Grafik festlegen =====
10 '----- Horizontale Linie -----
11 Y=100:FORX=0TO383:GOSUB101:NEXT
12 '----- Bewegte horizontale Linie -----
13 Y=110:FORL=1TO5:FORX=0TO20:GOSUB101:NEXTX:FORX=0TO20STEP2:
   GOSUB102:NEXTX:FORX=0TO20:GOSUB102:NEXTX,L
14 '----- Laufende horizontale Linie -----
15 Y=120:FORL=0TO20:FORX=LTOL+10:GOSUB101:NEXTX:X=L:GOSUB102:
   NEXTL
80 '----- Rahmen -----
81 Y=10:FORX=10TO60:GOSUB101:NEXT
82 Y=20:FORX=10TO60:GOSUB101:NEXT
83 X=10:FORY=10TO20:GOSUB101:NEXT
84 X=60:FORY=10TO20:GOSUB101:NEXT
90 '----- Fläche -----
91 FORX=150TO175:FORY=50TO80:GOSUB101:NEXTY,X
97 GOTO122:' Sprung zur Löschroutine
98 '=====
99 '           Grundprogramm von Arnulf SOPP
100 '===== Grafikmodus =====
101 GOSUB111:OUT5,AORC:RETURN           'setzen
102 GOSUB111:OUT5,AAND(NOTC):RETURN    'löschen
103 GOSUB111:D=((AANDC)<0%):RETURN      'invertieren
110 '===== Grafikpunkt errechnen =====
111 X1=INT(X/6%):B=X-X1*6%:Y1=INT(Y/12%):Y2=Y-Y1*12%:
   Z=Y2*1024%+Y1*64%+X1:ZM=Z/256%:ZL=ZAND255%:OUT2,ZL:OUT3,ZM:
   C=Z/8B:A=INP(4):RETURN
120 '===== Grafik löschen =====
121 CLS:FORI=0TO255:OUT2,IAND255:OUT3,I/256:OUT5,63:NEXT
122 Z%=INKEY$:IFZ%=""THEN122ELSECMD"c1h/CMD":END
123 'CLH/CMD ist ein File aus HRG-PACK.
   Es kann auch ein eigenes File erstellt werden, wie im INFO
   unter "Den HRG-Speicher löschen" beschrieben ist.

```

```

100 CLEAR 5000: DEFINT U: DIM EE(15), EA(15), E$(15), F$(20): GOSUB 2100
0
500 REM -----
510 REM *** HAUPTPROGRAMM
520 REM -----
530 CLS: PRINT TAB(11) "C A S S E T T E N - V E R Z E I C H N I S"
540 PRINT STRING$(64,"<")
550 PRINT @ 267, "W a h l m g l i c h k e i t e n : "
560 PRINT @ 395, "<1>   Eingeben bzw. editieren
570 PRINT @ 459, "<2>   Abspeichern"
580 PRINT @ 523, "<3>   Laden"
590 PRINT @ 587, "<4>   Drucken"
595 PRINT @ 651, "<5>   Programmende"
600 A$=INKEY$: IF A$<"1" OR A$>"5" THEN 600
610 IF A$="1" THEN GOSUB 1000: GOSUB 2000: GOTO 500
620 IF A$="2" THEN PRINT @ 832, "Dateiname :   ": INPUT D$: GOSUB 3000:
GOTO 500
630 IF A$="3" THEN PRINT @ 832, "Dateiname :   ": INPUT D$: GOSUB 4000:
GOTO 500
640 IF A$="4" THEN GOSUB 5000: GOTO 500
650 CLS: PRINT "P r o g r a m m e n d e": PRINT @ 832, "": END
1000 REM -----
1010 REM *** EINGABEMASKE 1 AUSGEBEN
1020 REM -----
1025 EE=50030: GOSUB 23000
1030 CLS: FOR I=0 TO 14: PRINT USING "##   ":I: IF F$(I)<>" THEN PRINT
F$(I) ELSE PRINT STRING$(57,"#")
1035 NEXT I
1040 GOSUB 20000
1050 FOR I=0 TO 14: F$(I)=E$(I): NEXT I
1060 RETURN
2000 REM -----
2010 REM *** EINGABEMASKE 2 AUSGEBEN
2020 REM -----
2025 EE=50040: GOSUB 23000
2030 CLS: PRINT TAB(12) "E I N G A B E   R   C K E N T I T E L"
2040 PRINT @ 256, "1)   ": IF F$(15)="" THEN PRINT STRING$(57,"#") ELSE
PRINT F$(15)
2050 PRINT "2)   ": IF F$(16)="" THEN PRINT STRING$(57,"#") ELSE PRINT
F$(16)
2055 PRINT: PRINT
2060 PRINT "Datum       :   ": IF F$(17)="" THEN PRINT "##.##.1985" ELS
E PRINT F$(17)
2070 PRINT "Laufzeit    :   ": IF F$(18)="" THEN PRINT "C 90" ELSE PRIN
T F$(18)
2080 PRINT "Mono/Stereo :   ": IF F$(19)="" THEN PRINT "S" ELSE PRINT F
$(19)
2090 PRINT "Dolby (J/N) :   ": IF F$(20)="" THEN PRINT "J" ELSE PRINT F
$(20)
2100 GOSUB 20000
2110 FOR I=15 TO 20: F$(I)=E$(I-15): NEXT I
2120 RETURN
3000 REM -----
3010 REM *** CASSETTEN-VERZEICHNIS ABSPEICHERN
3020 REM -----
3030 OPEN "O", 1, D$
3040 FOR I=0 TO 20: PRINT #1, F$(I): NEXT I
3050 CLOSE 1
3060 RETURN

```

```

4000 REM -----
4010 REM *** CASSETTEN-VE ICHNIS LADEN
4020 REM -----
4030 OPEN "I", 1, D$
4040 FOR I=0 TO 20: INPUT #1, F$(I): NEXT I
4050 CLOSE 1
4060 RETURN
5000 REM -----
5010 REM *** CASSETTEN-VERZEICHNIS DRUCKEN
5020 REM -----
5030 LPRINT CHR$(27); CHR$(14); CHR$(27); CHR$(28)
5040 LPRINT CHR$(240); STRING$(59,241); CHR$(242)
5050 FOR I=0 TO 14: LPRINT CHR$(245); TAB(60) CHR$(245): LPRINT CHR$(245
); "   ": F$(I); TAB(60) CHR$(245): NEXT I
5060 LPRINT CHR$(245); TAB(60) CHR$(245): LPRINT CHR$(244); STRING$(59,2
41); CHR$(249); LPRINT CHR$(245); TAB(60) CHR$(245)
5070 LPRINT CHR$(245); "   ": F$(15); TAB(60) CHR$(245): LPRINT CHR$(245);
TAB(60) CHR$(245)
5080 LPRINT CHR$(245); "   ": F$(16); TAB(60) CHR$(245): LPRINT CHR$(245);
TAB(60) CHR$(245)
5090 LPRINT CHR$(244); STRING$(59,241); CHR$(249): LPRINT CHR$(245); TAB
(60) CHR$(245)
5100 LPRINT CHR$(245); " Datum :   ": F$(17); TAB(30) "Cassette   ": F$(
18); TAB(60) CHR$(245): LPRINT CHR$(245); TAB(60) CHR$(245): LPRINT CHR$
(245); TAB(60); CHR$(245)
5110 LPRINT CHR$(245); IF F$(19)="M" THEN PRINT " Mono"; ELSE LPRINT "
Stereo"
5120 LPRINT TAB(30) "Dolby :   ": F$(20); TAB(60) CHR$(245): LPRINT CH
R$(245); TAB(60) CHR$(245): LPRINT CHR$(246); STRING$(59,241); CHR$(247)
5130 RETURN

```

23030 REST EE: F=0

```

50030 DATA 4,61,68,125,132,189,196,253,260,317,324,381,388,445,452,509,5
16,573,580,637,644,701,708,765,772,829,836,893,900,957,0,0
50040 DATA 261,318,325,382,528,538,592,596,656,657,720,721,0,0

```

CASSETTEN - VERZEICHNIS

MANFRED MANN'S EARTH BAND

1) Somewhere In Africa

Tribal Statistics / Eyes Of Nostradamus / Third World Service / Demolition Man/ Brothers And Sisters Of Azania/ Africa Suite (Part a-d: Brothers And Sisters Of Africa) To Bantustan?; Koze Kobenini?; Lalala / Redemption Song (No Kwazulu) / Somewhere In Africa

2) Glorified Magnified

Meat/ Look Around/ One Way Glass/ I'm Gonna Have You All/ Down Home / Our Friend George / Ashes To The Wind / Wind/ It's All Over Now Baby Blue/ Glorified Magnified

MANFRED MANN'S EARTH BAND

Somewhere In Africa --- Glorified Magnified

Date: 30.10.1985 Cassette C 98

Dolby: J

Das Schreiben derselben habe ich mir heute durch den Comp. vereinfacht (Ergebnis siehe links). Ich habe den im letzten Info vorgest. Editor geladen und dann in ca. 3 Stunden dieses Programm geschrieben. Im Bereich des Editors gab es dabei folgende Änderung: Ich benötige 2 Eingabemasken und daher 2 verschiedene DATA-Werte für den Editor, deshalb in Zeile 23030 der REST-Befehl (= RESTORE) u. a. gibt es in der Bibliothek ein Programm dafür. Die Zeilen 20000-50020 siehe Info 9, Seite 7). Die Druckerroutine ist auf meinen LP VI angepasst und muß ev. abgeändert werden.

Günther Wagner

ACHTUNG: EDITOR wurde verbessert;
Bericht in diesem Info

(von Günther Wagner)

Der im letzten Info vorgestellte Editor wurde jetzt nochmals verbessert - hier der aktuelle Bericht darüber.

Das größte Manko des Editors war bisher die lange Einlesezeit vom Bildschirm in die Variable (siehe Punkt J. Zeile 20260-20320 des letzten Artikels). Diese Einleseroutine habe ich nun umgeschrieben und dabei ganz einfache Techniken verwendet. Auf die Idee brachte mich übrigens der Hartmut; die Lösung wurde mir mit dem Buch Basic Faster & Better erleichtert.

Zunächst machen wir einen Ausflug in den Speicher unseres Computers - speziell in den String-Speicher. Wenn wir VARPTR (String) ausführen, so erhalten wir einen sehr wertvollen Adresswert. Die Adresse gibt uns die entsprechende String-Länge an. Die Adresse um 1 erhöht ergibt das LSB der Startadresse des Stringwertes, die Adresse um 2 erhöht das MSB. Aus LSB und MSB können wir uns die Adresse errechnen, an der die Abspeicherung des Strings beginnt:

$$\text{Speicherplatz} = \text{MSB} * 256 + \text{LSB}$$

Zum Beispiel:	VARPTR (AN\$)	ergibt	29000
	PEEK (29000)	ergibt	113
	PEEK (29001)	ergibt	205
	PEEK (29002)	ergibt	48

Wir wissen also jetzt, das der String AN\$ eine Länge von 113 hat und die Abspeicherung bei Adresse $48 * 256 + 205 = 12493$ beginnt.

Nun können wir natürlich auch in die 3 Speicherstellen POKE'n - genau das machen wir auch. Wir wollen z.B. daß AN\$ den Inhalt der 1. Bildschirmzeile erhält. In Anlehnung an obiges Beispiel genügen folgende Anweisungen:

VARPTR (AN\$)	ergibt	29000
POKE 29000,64		
POKE 29001,0		
POKE 29001,60		

Die 1. Bildschirmposition ist der Adressenwert 15360.

$$15360 : 256 = 60 \text{ Rest } 0$$

d.h. unser neues MSB ist 60, LSB ist 0 und die String-Länge ist 64 (1 Bildschirm-Zeile).

Ich hoffe, daß ich mich bisher verständlich genug ausgedrückt habe. Nun - genau diese Methode wende ich nun auch für den EDITOR an.

In der Zeile 21270 wird der Anfang der Variable festgelegt und in PO% abgespeichert. Der Anfang der Variablen wird auf den Anfang des jeweiligen Eingabefeldes gelegt (EE()). In der Zeile 21270 wird auch die Länge der Variablen festgelegt. Diese ist genauso lang wie die Länge des Eingabefeldes und wird in A1% abgespeichert.

Nun folgt in der Zeile 21280 das VARPTR-Kommando und die beiden POKE-Kommandos. Beim 1. POKE-Kommando (MSB) wird 60 addiert, da in PO% die Bildschirmposition steht und diese ja um 15360 (= $60 * 256$) erhöht werden muß, damit man die Adressposition erhält. Nach der Zeile 21280 ist also der Inhalt von AN\$ gleich dem Inhalt des jeweiligen Eingabefeldes (wir habe AN\$ auf den Bildschirm "gelegt").

In der Zeile 21300 übergeben wir dann nur noch den Inhalt von AN\$ in E\$(), wobei die Zeile 21290 und der LEFT\$-Befehl dafür sorgen, daß beim ersten '#' das Ende der Eingabe erkannt wird.

So - Ihr braucht jetzt also nur beim Listing des 9. Infos die Zeilen 20260-20300 wie folgt abändern:

```
20260 FOR G=0 TO F
20270 PO% = EA(G) : A1% = EE(G) - EA(G)
20280 AN$ = " " : POKE VARPTR(AN$), A1% : POKE VARPTR(AN$)+2, INT(PO%/256)+60 :
      POKE VARPTR(AN$)+1, PO%-INT(PO%/256)*256
20290 GG = INSTR(AN$, "#") - 1 : IF GG < 0 THEN GG = LEN(AN$)
20300 E$(G) = LEFT$(AN$, GG)
```

Yes

In der kritischen Phase des Vietnam-Krieges befragen die Generale im Pentagon ihren großen Computer, was sie jetzt tun sollten:

»Weiter vorgehen?«

»Zurückgehen?«

»Die Stellungen halten?«

Der Computer klickt eine Weile nachdenklich, dann antwortet er: »YES!«

Konsterniertes Schweigen der Generale. Dann tippt einer ärgerlich auf der Eingabe-Tastatur: »Yes what?«

Wieder klickt der Computer - erheblich länger diesmal - und dann schreibt er: »YES, SIR!«

Mini-RAM-Floppy im "sicheren Plätzchen"

Helmut Bernhardt stellt in c't 5/85 unter dem Titel "Ein sicheres Plätzchen" eine Schaltung vor, die im Adreßbereich 3900-39FF RAM verfügbar macht (diese wie auch fast alle folgenden Zahlenangaben in Hex). Dort liegen ursprünglich die oberen nicht genutzten Adressen der Tastatur. Der Autor schlägt vor, dort Maschinenprogramme unterzubringen. Da alle gängigen Anwenderprogramme im Adreßraum ab 4300 (Level 2) bzw. 4200 (DOS) residieren, liegt es nahe, im neu gewonnenen RAM allgemeine Systemerweiterungen unterzubringen. Hier soll eine Methode vorgestellt werden, nach der NEWDOS-80 und seine Abkömmlinge E-DOS und H-DOS mit wertvollen zusätzlichen Möglichkeiten ausgestattet werden können.

Eine DOS-Anforderung, d. h. das Laden und Anspringen eines SYS-Files, wird über die RST-28-Routine abgewickelt. Dazu muß ein Code im Akku stehen, dessen binäre Bitkonfiguration darüber entscheidet, welche Systemdatei geladen und welche Routine innerhalb dieses Files angesprungen wird. Dieser Code muß mindestens 20 (hex, wohlgernekt) betragen, andernfalls kehrt RST 28 unverrichteter Dinge zurück. In der Praxis kommt aber nur ein einziger Fall vor (abgesehen von Programmierfehlern), in dem der Requestcode kleiner als 20 ist: Wird im ROM-Tastaturtreiber die BREAK-Taste erkannt, dann wird RST 28 mit 01 im Akku angesprungen (und ohne Wirkung sofort wieder verlassen, wie gesagt).

Bei diesem Requestcode entscheiden die drei unteren Bits 0-2 darüber, in welchem Sektor des Inhaltsverzeichnisses der Systemdiskette die SYS-Datei zu suchen ist. Die Bits 3 und 4 geben an, welches der vier dort eingetragenen Systemfiles gemeint ist. BOOT/SYS (oder GDOS/SYS bzw. HDOS/SYS) wird im Prinzip nur nach dem Einschalten aufgerufen. Sein Requestcode entspräche dem Bitmuster xxx-00-000. Das bedeutet, daß im "nullten" Dateieintrags-Sektor die "nullte" Datei gemeint ist, also im Sektor 02 des Inhaltsverzeichnisses der 1. Eintrag. Ein Aufruf des Bootfiles mit RST 28 kommt nicht vor. Die acht möglichen Requestcodes mit dem Muster xxx00000 stehen deshalb für unsere Zwecke zur Verfügung.

Es wird noch wesentlich mehr: Die Bedingung, daß der Code im Akku mindestens 20 betragen muß, verringert die theoretisch möglichen 256 (diesmal dez) Codes glatt auf die Hälfte, was wir nicht hinnehmen müssen. Stattdessen kann man auf den BREAK-Code 01 testen und bei Übereinstimmung zurückspringen, um nicht bei jedem BREAK die Floppy in Gang zu setzen. Anschließend kann geprüft werden, ob der Requestcode höchstens 1F beträgt. Falls nein, handelt es sich um eine Anforderung an die Apparatur, Inc. bzw. TCS. Dann geht es eben in der alten Routine im DOS-Kern weiter. Andernfalls ist es ein ehemals wirkungsloser Code bis 1F, mit dem der Anwender nun etwas anstellen kann.

Außerdem ist die Tatsache interessant, daß bei einem RST 28 alle Register zunächst unverändert in der Bearbeitungsroutine ankommen. So können beliebige Parameter an eine selbstgeschriebene DOS-Erweiterung übergeben werden. NEWDOS-80 (G-DOS, H-DOS) macht sich das zunutze, indem es dem Register C bei den meisten Library-Befehlen eine Zeigerfunktion zuordnet.

Um die oben skizzierten neuen Möglichkeiten auszunutzen, muß man wissen, wie RST 28 arbeitet. Der Einsprung ist natürlich bei 0028 in der "zero-page", der "Seite 0", also im Bereich der ersten 256 Bytes des ROM. Dort steht ein Vektor nach 400C, wo wiederum nach 4BC2 weiterverzweigt wird. An der Adresse 4BC2 wird der Stapelzeiger SP (stack pointer) zweimal inkrementiert. Die Wirkung ist, daß die RET-Adresse sozusagen vom Stack verschwindet. Dadurch verliert der RST 28 im Gegensatz zu den anderen RSTs seinen CALL-Charakter. Es wird ein gewöhnlicher JP daraus. Anders als bei einem Unterprogrammaufruf geht deshalb die Kontrolle endgültig an die angesprungene Routine über. Unter welchen Umständen sie dennoch mit einem RET verlassen werden kann, soll später erläutert werden.

den.

Nach dem Quasi-Lischen der RET-Adresse folgt die oben angesprochene Prüfung auf 20. Wenn der Requestcode kleiner ist, geht das Carry-Flag auf 1 und es erfolgt ein Sprung nach 4312, von dort nach 4E80, wo nur der Akku auf 00 gesetzt und aus der RST-28-Routine zurückgekehrt wird. Dieser bedingte Sprungbefehl JP C,4312h kann nun leicht durch einen Sprung in die eigene Routine im Bereich 3900-39FF ersetzt werden (wer das "sichere Plätzchen" nicht hat, kann natürlich auch sonstwohin springen). Ein Teil dessen, was dort bei H-DOS passiert, geht aus dem Listing am Ende dieses Artikels hervor:

Im residenten Teil von SYS0 ist an der Adresse 4BC6 der bedingte Sprungbefehl durch einen JP rst28 ersetzt. Dieses Label steht für die Adresse 3A5B. Dort wird geprüft, ob die BREAK-Taste mit 01 im Akku den RST verursacht (s. c.). Bei Übereinstimmung wird der Sprung nach 4312 nachgeholt. Sonst wird getestet, ob eines der beiden Bitmuster 000xxxxx oder xxx00000 vorliegt. Falls ja, ist unser neues RAM-SYS-File zuständig und wird angesprungen. Andernfalls geht im DOS-Kern bei 4BC9 die Bearbeitung wie gewohnt weiter. Auf diese Weise sind 39 zusätzliche Requestcodes möglich. Wie aus dem Listing hervorgeht, wird davon bisher nur ein Teil genutzt. Zukünftige Erweiterungen werden diesen Vorrat nach und nach verkleinern.

Die Tatsache, daß der RST 28 wie ein JP behandelt wird, hat gute Gründe: Bei vielen Systemdateien ist ein Rücksprung zum Caller nicht sinnvoll, manchmal, etwa beim Auftreten eines I/O-Fehlers, u. U. sogar fatal. Gleichwohl kann eine SYS-Datei wie ein Unterprogramm aufgerufen werden. Dazu ist es lediglich nötig, den RST 28 nicht direkt zu programmieren, sondern stattdessen einen CALL an eine Adresse, wo ein RST 28 steht. Das ist z. B. bei 4402 der Fall oder auch im ROM bei 0454, wo BREAK den RST aufruft. Ein RST ist für den Z80 ein CALL. Wenn man nach dem angegebenen Muster einen CALL ruft, befindet man sich deshalb bereits in der zweiten Unterprogrammebene. Das zweimalige Inkrementieren des Stackpointers am Beginn der RET-28-Routine geht eine Ebene höher, so daß bei einem RET nun die richtige Rückkehradresse gefunden wird.

So erklärt sich das RET in Zeile 164 des Listings. Mehr möchte ich zum gelisteten Teil des Programms nicht sagen, denn es geht hier nur um die Methode, RST 28 für eigene Anwendungen nutzbar zu machen. Die hier nicht interessierenden Teile des Programms sind durch LIST OFF ausgespart. Wer Interesse daran hat, kann einen großen Teil davon aus dem darüberstehenden Sektordump rekonstruieren. Das hier Gelistete ist darin unterstrichen. Es handelt sich um den relativen Sektor 10h von SYS0/SYS. Ursprünglich hat SYS0 nur 15 Sektoren, wurde aber für die Erweiterungen mit APPEND um weitere 5 Sektoren verlängert.

```

001000: 0100 003A FF10 FE41 3CDE FF10 FE0D 4110 .....AK.....A.
001010: F0F1 C93A 5038 FE01 200A 3A40 38FE 1000 .....S...:S8..
001020: D77A 1038 E6FE 28BC 0310 3E0D 5444 090C .....S...:TD(.
001030: 7EE6 DFFE 4ACA E73A FE4E 2804 3E2F 185E 8...J...N...>/.^
001040: 2157 4822 0845 3E7A 32D7 4521 1038 22D4 !SH".E>:2.E!."
001050: 453E DA32 C64E 2112 4322 C74E AF18 40FE E>.2.M!."K...S.
001060: 01CA 1243 F5E6 1F28 0AF1 F5E6 E028 0AF1 ...C...:.....(..
001070: C3C9 4BF1 FEE0 2007 3E01 D3F0 3EFD EFEF ...K...>.....>
001080: DECE FE60 CA06 2E73 FE40 2838 FE90 295E ...:.....S(B..(A
001090: FE40 2897 FE00 28A8 FE0F 3008 3E37 5701 ..(....:0.>7..
0010A0: D1E1 FBC9 CB67 2004 0EFO 1807 0EDF 3E0A .....g.....>.
0010B0: ED78 3CED 7921 0030 7E2F 775E 2F77 ED78 .y.k.v!."0B/w./w.x
0010C0: 3E08 18DB 3E08 D3DF 3CDE DF3C 3CDE DF21 >...>...<...<<..!
0010D0: 0000 545D 7E2F 775E 2804 3E08 18C0 AFD3 ..TUB/w..(.).....
0010E0: DF01 E037 EDE0 DEDF AF18 B421 F439 2208 ...7.....!."9".
0010F0: 453E CD32 D345 210F 3A22 D445 3E03 32C6 E>.2.E!:"E>.2.

```

```

4BC6          00001      OFE      4bc6h      :RST-28h-Routine
4BC6 C3E93A   00002      JF        rst28      :umleiten
                00003
                00114 ;Ansprung DOS-Request (RST 28h):
3A6B FE01     00115 rst28  CP        01h      :RST 28h nach BREAK?
3A6D CA1243   00116          JP        Z,4312h   :falls ja
3A60 FE       00117          PUSH     AF         :Requestcode retten
3A61 E61F     00118          AND      1fh       :Requestbits isolieren
3A63 2B0A     00119          JF       Z,ramsys  :falls Code xxx00000b
3A65 F1       00120          POP      AF         :sonst Requestcode rest.
3A66 FE      00121          PUSH     AF         :und wieder retten
3A67 EAE0     00122          AND      0e0h     :Requestbits ausmaskieren
3A69 2B04     00123          JF       Z,ramsys  :falls Code 000xxxxb
3A6B F1       00124          POP      AF         :sonst Code restaurieren
3A6D C3C94B   00125          JF       4bc9h    :und RST 28h à la E-DOS
                00126
                00127 ;Ansprung hier, falls RAM-SYS-File zuständig:
3A6F F1       00128 ramsys  POP      AF         :Requestcode restaurieren
3A70 FE00     00129          CP       0e0h     :V24?
3A72 2007     00130          JF       N2.nov24  :falls nein
                00131
3A73 E5       00140 nov24   PUSH     HL         :Register retten
3A7C DE       00141          FUSH    DE
3A7D CE       00142          FUSH    BC
3A7E FE60     00143          CP       6Ch      :HRE-Speicher löschen?
3A80 CA063E   00144          JP      Z,hrgrcls  :falls ja
3A83 F3       00145          DI       :bist keine Störungen!
3A84 FE40     00146          CP       40h      :RCM -> RAM kopieren?
3A86 2B7E     00147          JF      Z,copy    :falls ja
3A8B FE00     00148          CP       80h      :INI,J?
3A8A 2B5E     00149          JF      Z,ini;    :falls ja
3A8C FE40     00150          CP       0e0h     :INI?
3A8E 2897     00151          JR      Z,ini     :falls ja
3A90 FE00     00152          CP       0c0h     :INI,N?
3A92 28AE     00153          JR      Z,inin   :falls ja
3A94 FE0F     00154          CP       04h      :auf Bank-RAM testen?
3A96 300E     00155          JR      N2.ramtest :falls ja
                00156
                00157 ;raus mit oder ohne Fehlercode:
3A98 3E37     00158          LD      A,77h    :Code für DOS-Fehler
3A9A E7       00159 error   DR       A        :Fehlerflag NZ
3A95 C1       00160 exit   POP     BC       :Register restaurieren
3A9C D1       00161          POP     DE
3A9D E1       00162          POP     HL
3A9E FE      00163          EI           :INTs ( ) per zulassen
3A9F C5       00164          FF      :und falls

```

Mini-Befehlseingabe:

MINI-NEWDOS/80 READY

M>
;
/
!

MDCOPY
MDRET
MDBORT

Befehlseingabe:

NEWDOS/80 READY

```

& (J,N)
? bzw. LIB
I bzw. DIR (I,S,A,B,/typ)
> bzw. COPY (QPDN=,ZPDN=,KDWA,
  QKW=,NZKW=,ZZND,AZN=,QN=,
  BZN,NZN=,BZD,SQD,IVU,SBIV=,
  AEIV=,EDK,AZKW=,FRD,BEA,NVD,
  IDL=,XDL=,FRAG,---
0 (VON=,BIS=,HIMEM=)
@ (KEINE,MO,TA,DR,ST=,NL)
AIK
ATTRIB (BKW=,HKW=,BEA=,PROT=,
  KEIN,START,LESEN,ÄNDERN,
  NAME,KILL,NULL)
B2
BL (J,N)
CREATE (ANZ=,ADE=,ADF=,LOG=)
CONT (J,N,D)
DATUM (tt.mm.jj)
E
LC
---
N
NDF (SPUR=,STOP=,MAG)
PD (SP=,SEK=,SWZ=,EIB=,
  SBIV=,AEIV=)
PROT (DATUM=,BKL,ZU,AUF)
S
UHR
V+ (J,N)
ZEIT (hh:mm:ss)

```

```

DEBUG (Y,N)
LIB
DIR (I,S,A,U,/ext)
COPY (SPDN=,DPDN=,NDMW,
  SPW=,NDPW=,DDND, ODN=,SN=,
  KDN,NDN=,KDD,USD,BDU,DDSL=,
  DDGA=,CBF,ODPW=,USR,UPD,DFO,
  ILF=,XLF=,CFWO,UBB
CLEAR (START=,END=,MEM=)
ROUTE (CLEAR,DO,KI,PR,MM=,NL)
WRDIRP
ATTRIB (ACC=,UPD=,UDF=,PROT=,
  LOCK,EXEC,READ,WRITE,
  (RE)NAME,KILL,FULL)
BASIC2
BLINK (Y,N)
CREATE (REC=,ASE=,ASC=,LRL=)
CHNON (Y,N,D)
DATE (MM/DD/YY)
ERROR
LCDVR
LC
RENAME
FORMAT (PFST=,PFTC=,RWF)
PDRIVE (TC=,SPT=,TSR=,GPL=,
  DDSL=,DDGA=)
PROT (DATE=,RUF,LOCK,UNLOCK)
SYSTEM
CLOCK
VERIFY (Y,N)
TIME (hh:mm:ss)

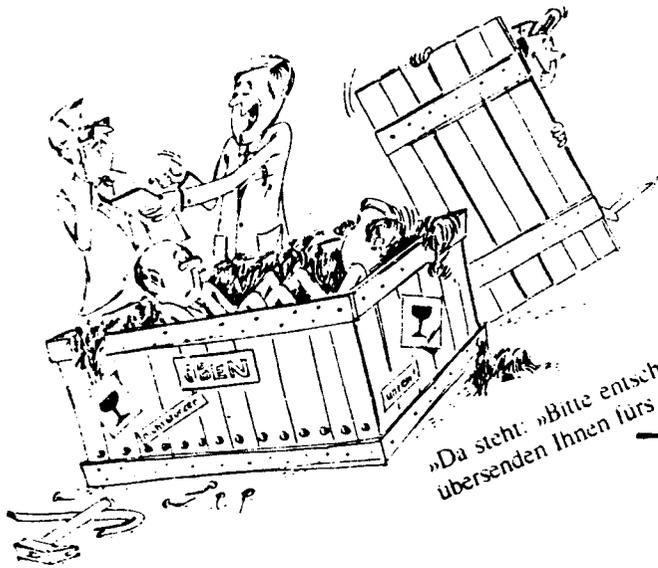
```

nur bei Genie-DOS:

nur bei NEWDOS80.2:

DISK
DR
FORM
INFO
LF
PORT
V24
Z
F#
##
80
64
DDE

FORMS
SETCOM



„Da steht: „Bitte entschuldigen Sie die lange Lieferfrist. Wir übersenden Ihnen fürs erste zwei erstklassige Rechner!«“

Umlaute im NEWDOS ?? - aber natürlich !!

Ihr könnt Euch sicher noch erinnern, wie ich im letzten Clubinfo nach einer Antwort für mein "UMLAUTPROBLEM MIT NEWDOS" -dies war die richtige Groß-/Kleinschreibung der Umlaute wie bei den Normalbuchstaben und die Akzeptanz von Umlauten in den Filenamen- suchte.

Nun, nach einigen kleinen Tips über die Tastatureingaberoutine, habe ich die Antwort darauf selbst gefunden. Aber erst einmal etwas "Vorgeschichtliches" zu diesem Thema.

Nach der Erkennung meines Problems probierte ich dann verschiedene DOS-Versionen aus. Dabei bemerkte ich, daß diese "FEHLER" im Genie-DOS nicht auftauchten, wohl aber im NEWDOS. So stellte ich fest, daß im NEWDOS die Umlautschreibung nicht berücksichtigt ist. Was wiederum verständlich ist, da NEWDOS, ein amerikanisches Programm ist, und die englisch Schreibweise keine Umlaute kennt.

Nun ist es leicht zu sagen, man arbeite mit dem Genie-DOS. Auch schon deshalb, weil G-DOS eine "(DEUTSCHE) NEWDOS-VERSION" ist. Andererseits aber, wäre es dann nötig, das DOS auf die entsprechenden Disketten umzukopieren, sowie auch die "DEUTSCHEN BEFEHLE" für das DOS sich anzueignen, wo man doch gerade die englischen erst richtig anwenden kann. Ich bin deshalb einen anderen Weg gegangen, den ich hier beschreiben möchte.

Da die beiden DOS -wie vorher schon erwähnt- bis auf die Deutschschreibung der Meldungen und Befehle ziemlich gleich sein müssen, liegt es nahe, beide zu vergleichen. Das sich nun dabei der Computer in's Zeug legen darf ist ja klar. Dazu habe ich ein kleines "FAULHEITSPROGRAMM" -wie ich es nenne, wenn der Computer für mich arbeitet- geschrieben.

Das Listing dazu ist am Ende dieses Berichtes angehängt und das Programm gibt es natürlich in der Clubbibliothek. Mein Programm fragt nun nach dem zu vergleichenden File, das in beiden Laufwerken zu Verfügung stehen sollte. Danach werden die entsprechenden File miteinander verglichen, wobei die darstellbaren Zeichen, auf dem Bildschirm erscheinen. Bei Differenzen während des Vergleichs wird dann, wenn gewünscht, der Platz, das Zeichen sowie dessen Wert in Dez und Hex ausgedruckt.

Nun hatte ich es recht einfach. Ich ließ meinen Computer vergleichen und suchte dann in den Ausdrucken nach dem/der entsprechenden Zeichen/Zeichenfolge. Da ich die SYS-File der Reihe nach verglich, fand ich schon in SYS0/SYS der ersten Zap, welcher zu ändern wäre :

RELATIVER SEKTOR 2 BEI 8A : VON 1A AUF 1F

1F bedeutet das höchste abzuarbeitende Zeichen (). Vorher war hier das (Z) durch 1A implementiert.

Der Fehler mit der unrichtigen Abarbeitung der Umlaute war behoben. Jetzt steht noch die Umlautverwendung bei Filenamen aus. Dieses Hemmnis müsste in SYS2/SYS liegen, da hier die Filenamen auf Richtigkeit überprüft werden. Beim Vergleich fand ich nun Punkt 2 unser Änderung :

RELATIVER SEKTOR 4 BEI 63 : VON 5B AUF 5F

Auch hier wird der Wert für den maximalen Zahlenwert höher gesetzt. Das Abspeichern von Umlautfilenamen ist nun möglich. Es werden aber nur Großbuchstaben der Umlaute anerkannt. Dies nun zu beseitigen ist der nächste Schritt. Wieder zurück zu -wir hatten es ja schon- SYS0/SYS, wo die Eingaberoutine abgelegt ist. Es folgt Zap 3 :

RELATIVER SEKTOR 2 BEI AA : VON 7B AUF 7F

Jetzt werden auch "KLEINUMLAUTE" bei der Filenameneingabe akzeptiert. In der Zwischenzeit sind aber im DOS auch einige Fehler aufgetaucht, die sich folgender Art auswirken. Befehle wie "KILL ...", "RENAME ..." oder andere, die auf die Programmnameneingabe angewiesen sind erzeugen bei der Umlauteingabe eine Fehlermeldung. Wir können also Umlautfilenamen im Moment nicht weiter bearbeiten bzw. mit Umlauten benannte Programme können nur durch "PURGE .." wieder gelöscht werden. Es werden also noch weitere Änderungen nötig sein.

Es wird weitergesucht und eine Fortsetzung dieses Artikels folgt.

Jens NEUEDER

☹☹ **Der menschliche Verstand ist in der Praxis nicht verlässlich, am wenigsten in größter Not.** ☹☹

☹☹ **Der Kopf ist jener Teil unseres Körpers, der uns am häufigsten im Wege steht.** ☹☹

```

10 CLEAR1000:DEFINT A-Z:D=0:Y$="#####"
20 CLS:PRINT$320,CHR$(23);"Filevergleich"
30 PRINT$448,"von Jens NEUEDER  Version 1.2"
40 PRINT$554,"18.10.1985"
50 PRINT$576,"Panoramastraße 21":PRINT"7178 Michelbach /Bilz":PRINT"Tel.: 0791/42877"
60 PRINT$832,STRING$(32,CHR$(127)):PRINT$896,"Bitte Vergleichfilename eingeben":INPUTA$
70 CLS:PRINTA$:PRINT"Eingabe richtig ?      (Ja/Nein)"
80 Z$=INKEY$:IFZ$=""THEN80ELSEIFZ$="N"THEN60ELSEIFZ$="n"THEN60
90 PRINT:PRINT"Ausdruck erwünscht ?      (Ja/Nein)"
100 Z$=INKEY$:IFZ$=""THEN100ELSEIFZ$="N"THEN110ELSEIFZ$="n"THEN110ELSE=1:
    LPRINTCHR$(14);:LPRINTCHR$(27);"!";:LPRINTA$;" -File werden verglichen :":
    LPRINTCHR$(27);CHR$(34);:LPRINTCHR$(15)
110 PRINT:PRINT"Bitte die File in beiden Laufwerken bereithalten !":PRINT"    ---> < ENTER >"
120 Z$=INKEY$:IFZ$(<)CHR$(13)THEN120ELSE140
130 (1)B$;
140 CLS:PRINT"Die File => ";A$;" <=> werden miteinander verglichen"
150 OPEN"R",1,A$+"":0,"FF",1:PRINT"File in 0 ";
160 OPEN"R",2,A$+":1,"FF",1:PRINT"File in 1 eröffnet."
170 PRINT$200,". laufendes Byte":PRINT$264,". Sektor":PRINT$328,". relatives Byte";:
    PRINT$448,"File in 0 :";PRINT$576,"File in 1 :";
180 A=1:Q=-1
190 GET1,A:GET1,,130:V1=ASC(B$)
200 GET2,A:GET2,,130:V2=ASC(B$)
210 S=INT(A/256):R=INT(A-(S*256))
220 PRINT$192,USINGY$;A;:PRINT$256,USINGY$;S;:PRINT$320,USINGY$;R;
230 IFV1>31THEN240ELSEI$=I$+" ":GOTO250
240 IFV1<192THENI$=I$+CHR$(V1)ELSEI$=I$+" "
250 IFV2>31THEN260ELSEJ$=J$+" ":GOTO270
260 IFV2<192THENJ$=J$+CHR$(V2)ELSEJ$=J$+" "
270 I$=RIGHT$(I$,64):J$=RIGHT$(J$,64)
280 PRINT$512,I$;:PRINT$640,J$;
290 IFV1=V2THEN360ELSEIFD=0THEN360

```

☺ Reklame ist die Kunst, auf den Kopf zu zielen und die Brieftasche zu treffen. ☺

FILEVERGLEICHSPROGRAMM

☺ Weiß ich nicht. Ich beschwere mein Gedächtnis nicht mit Tatsachen, die ich in einem Konversationslexikon finden kann. ☺

```

300 IFQ(<)STHENLPRINT:LPRINT"Sektor :";S;" ab relativem Byte :";R
310 Q=S:H=R:GOSUB430:R$=H$:H=V1:GOSUB430:V1$=H$:H=V2:GOSUB430:V2$=H$
320 LPRINTR$;"-> 1: (";V1$;)"--(";V1$;".hex)--(";
330 IFV1>32THENLPRINTCHR$(V1);)"",ELSELPRINT)"",
340 LPRINT"in -> 2: (";V2$;)"--(";V2$;".hex)--(";
350 IFV2>32THENLPRINTCHR$(V2);)"",ELSELPRINT)"",
360 A=A+1
370 IFNOTEOF(1)THEN380ELSEPRINT$448,"File 1 zu Ende.":CLOSE1:IFD=1THENLPRINT:LPRINT"Ende File 1";
380 IFNOTEOF(2)THEN190ELSEPRINT$576,"File 2 zu Ende.":CLOSE2:IFD=1THENLPRINT" File 2."
390 PRINT$768,"Vergleich abgeschlossen.":A=A-1:IFD=1THENLPRINT:
    LPRINT"Fileende bei :";A;" Bytes -> ";S;" Sektoren"
400 PRINT"<P>rogrammende, <w>eiteres File ? (P/N)"
410 Z$=INKEY$:
    IFZ$="W"THENRUNELSEIFZ$="w"THENRUNELSEIFZ$="P"THEN420ELSEIFZ$="p"THEN420ELSE410
420 END:'cmd"s"
430 H$="":X=H-INT(H/256):GOSUB440:RETURN
440 I=XAND15:J=(XAND240)/16:IFJ>9THENJ=J+55:GOTO450ELSEJ=J+48
450 IFI>9THENI=I+55:GOTO460ELSEI=I+48
460 H$=CHR$(J)+CHR$(I):RETURN

```

Umbau eines TRS-80 Modell 1 16K auf 48k (64k)

Nachdem ich im Sommer des letzten Jahres einen gebrauchten TRS-80 Modell 1 gekauft hatte, der mit 16 k RAM bestueckt war, stand ich vor dem Problem diesen Rechner auf 48 k RAM aufzuruesten.

Zu diesem Zeitpunkt hatte ich schon einen Genie-I mit 48-k (64-k) RAM, und ich dachte das es kein groesseres Problem sein sollte auch den Tandy auf die gewuenschte Speicherkapazitaet zu bringen. Aber erstens kommt es anders, und zweitens als man denkt.

Da ich die technischen Handbuecher fuer beide Geraete besitze verglich ich erst einmal die Schaltunterlagen von beiden Rechnern. Auf den ersten Blick sahen auch beide Schaltplaeue sehr aehnlich aus, aber gerade beim interessanten Teil, naemlich dem Adressdekoder fuer die RAM's und ROM's unterschieden sich beide Geraete erhaeblich. So musste ich feststellen das der Genie-Computer von seiner Grundauslegung her schon fuer den Umbau von 16-k auf 48-k mit den neueren 64-k-RAM Speicherchips ausgelegt war, waehrend der TRS-80 einen voellig anderen Adressdekoder hatte, der sich auf den ersten und zweiten Blick ueberhaupt nicht fuer diesen Umbau eignete.

Es musste also ein mehr oder minder neuer Adressdekoder eingebaut werden. Ich hatte zuerst vor, den Dekoder des Genie-I in den TRS-80 einzubauen. Aber auch dies erwies sich als nicht so einfach machbar, da sich beide Rechner in der Ansteuerung der ROM's und des Bildspeichers stark unterschieden.

Also was tun ?

Ich entschied mich dazu die ROM und Bildspeicheradressierung so zu lassen wie sie war und nur die Ansteuerung der RAM's zu aendern.

Zuerst zum Adressdekoder Z21. Dieser Baustein zerlegt die unteren 32K, die die ROM's, Bildspeicher, Tastatur und die alten 16 K RAM enthalten in Bloecke zu je 4K. Die Ansteuerung der alten RAM's konnte ja nun wegfallen, also machte ich folgende Ueberlegung, die neue RAM's sollen nicht angesprochen werden, wenn auf die unteren 16K zugegriffen wird, bei einer Adresse oberhalb von 3FFFH muss der RAM Bereich angesprochen werden.

Dies wird nun so erreicht: Wenn einer der ersten 4 Ausgaenge von Z21 LOW wird, so bedeutet dies, das auf die unteren 16K zugegriffen wird, also darf in diesem Fall der RAM Bereich nicht angesprochen werden.

Ich nahm also ein 4-fach-NAND und legte die Eingaeuge dieses Chips auf die 4 Ausgaenge des Z21 (Pin 9 - 12), den Ausgang des NAND legte ich auf X3 Pin 15 von wo aus die RAM-Select Leitung abgeht (RAM *). Ist diese Leitung LOW, so werden die RAM-Bausteine selectiert, und zwar geht das Signal ueber X71 Pin 12 - Pin 5 auf Z67 Pin 15, den Select-Eingang eines Buffers ueber den MCAS geschaltet wird.

Somit war das Ziel der Adressierung schon erreicht. Wird eine Adresse der ROM's des Bildspeichers oder der Tastatur (0000H - 3FFFH) angesprochen, so wird der RAM-Bereich nicht selectiert, wird aber eine Adresse ueber 3FFFH (4000H - FFFFH) angesprochen so wird der RAM-Bereich selectiert.

Nun aber zum praktjischen Teil des Umbau's. Als erstes mussten die alten RAM's (4116) gegen die neuen RAM's (4164) ausgetauscht werden.

Zuerst musste die Spannungsversorgung der RAM's geaendert werden.

Tabelle 1:

4116 -> :	Pin 1	—	-5 V
	Pin 8	—	+12 V
	Pin 9	—	+5 V
	Pin 16	—	GND
4164 -> :	Pin 1	—	N.C.
	Pin 8	—	+5 V
	Pin 9	—	A 7
	Pin 16	—	GND

Wie aus dieser Tabelle zu ersehen muss der Pin 1 freigemacht werden. Auf der Platine sind die einzelnen Pins der 8 Speicher IC's durch Leiterbahnen miteinander verbunden. Also wurde festgestellt von welchem IC aus die Leiterbahn als naechstes musste aus dem Stromversorgungsanschluss Pin 9 der Adresseingang A 7 gemacht werden. Dies geschah aehnlich dem Umbau von Pin 1. Zuerst stellte ich fest, von wo aus die Versorgungsspannung aus dem Netzteil kam, dann trennte ich diese Leiterbahn auf, entfernte noch die Siebkondensatoren (100nF) die auf dieser Leiterbahn gegen Masse lagen und dann wurde der Pin 9 der Speicher IC's mit dem Adressmultiplexer Z51 Pin 2 verbunden. Damit der Adressmultiplexer auch die Adresse A 7 erzeugen kann muss er auch etwas modifiziert werden. Es muessen die Adressleitungen A 14 und A 15 auf die Pins 14 und 15 gelegt werden. Diese drei Anschlusse sind normalerweise nicht belegt.

Als letztes musste noch der Pin 8 der Speicher-IC's von der +12 V Versorgung getrennt, und mit der +5 V Versorgung verbunden werden. Dies war fast der schlimmste Teil des ganzen Umbaus, da an jedem einzelnen der 8 Speicher-IC's der Pin 8 von den Leiterbahnen getrennt werden musste und die neuen Anschlusse mit Schalllitze auf der Platine nachverdrahtet werden musste.

Als letztes wurden dann die neuen Speicher-Bausteine in die Fassungen eingesetzt. Allerdings ist zu empfehlen, vor dem Einbau der IC's den Rechner schon einmal einzuschalten und die Spannungen an den IC-Sockeln zu pruefen, ob auch alle Aenderungen korrekt sind.

Dann koennen die neuen Speicherbausteine getestet werden. Also Rechner einschalten. Es sollte jetzt wie gewohnt 'MEM-SIZE' auf dem Bildschirm erscheinen. Nach dem Druecken der 'ENTER'-Taste dauert der Speicherausbauteil jetzt etwa dreimal solange wie bisher. Also keine Panik wenn es jetzt etwas laenger dauert bis 'READY' erscheint. Nachdem der Rechner sich mit 'READY' gemeldet hat, sollte man 'PRINT MEM' eingeben und die 'ENTER'-Taste druecken. Wenn alles richtig geklappt hat, sollte jetzt die Zahl '48340' auf dem Bildschirm erscheinen.

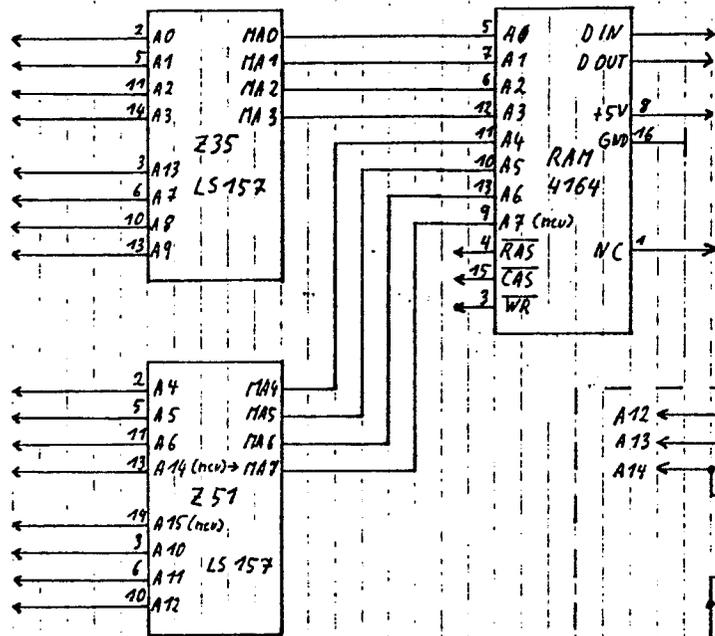
Jetzt noch ein paar Tips zu den Speicher-Bausteinen selbst :

Bei meinem ersten Umbau klappte alles wunderbar, aber nach ein paar Tagen als ich ein ziemlich grosses Programm im Speicher hatte fing mein Rechner ploetzlich an zu Spinnen. Einer der auftretenden Effekte war ein sporadisch auftretender Kaltstart ab Adresse 0000H ('MEM-SIZE'), ein weiterer Fehler war, dass sich der Rechner irgendwo im Programm aufhaengte und nur durch ausschalten und wiedereinschalten erneut zum Arbeiten zu bewegen war. Diese und aehnliche Effekte traten aber nur auf, wenn die 'MEM-SIZE' Eingabe ueber 32767 lagen, also im neu dazugewonnenen Speicherbereich lag.

Der Verstand ist unser größtes Vermögen, aber Armut schändet nicht.

Bildung ist die Fähigkeit, sich alles anzuhören und dabei weder die Selbstbeherrschung zu verlieren noch das Selbstvertrauen.

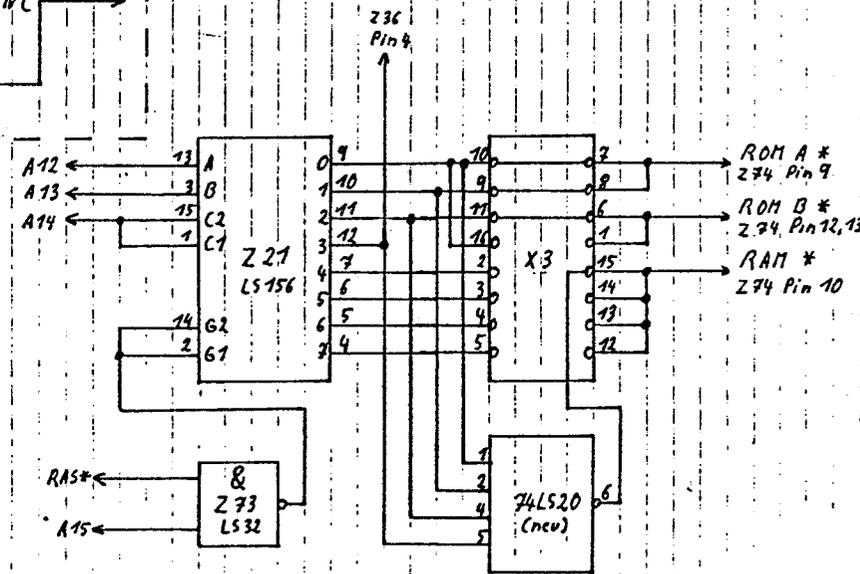
Address multiplexer



Also fiel der Verdacht auf die neuen Speicher IC's. Da ich diese IC's mit einigen Arbeitskollegen in einer Sammelbestellung bezogen hatte, erkundigte ich mich bei diesen, ob auch sie Probleme mit den Speichern hatten. Und siehe da, einige hatten Probleme und andere nicht. Ein Vergleich der Hardware der einzelnen Rechner brachte uns dann auf den REFRESH. Und zwar wurden in einigen Rechnern die Speicher von der CPU aus REFRESH'd und in einigen anderen Rechner wurde dies von speziellen REFRESH-Controllern erledigt. Wir fanden also heraus, dass alle Rechner, die den CPU-REFRESH benutzen Probleme mit den Speicher IC's hatten. Nach dem Durchsehen von diversen Datenblättern fanden wir schließlich heraus wo der Wurm zu suchen war. Die Z80-CPU erzeugt einen 7 Bit REFRESH. Nun gibt es allerdings einige Hersteller die einen 8 Bit REFRESH fuer ihre 64K dynamischen RAM's brauchen. Wenn man nun diese 8 Bit REFRESH IC's direkt an den Z80 anschliesst dann werden die Adressen 8000H -> FFFFH nicht refresh't, das bedeutet das die Informationen, die auf diese Speicherplaetze geschrieben werden nach einer gewissen Zeit verschwinden. Also Leute, aufpassen beim Kauf von 4164 IC's, fuer den TRS-80, und den Genie I muessen es IC's mit 7 Bit REFRESH sein.

Bernd Retzlaff

Address decoder



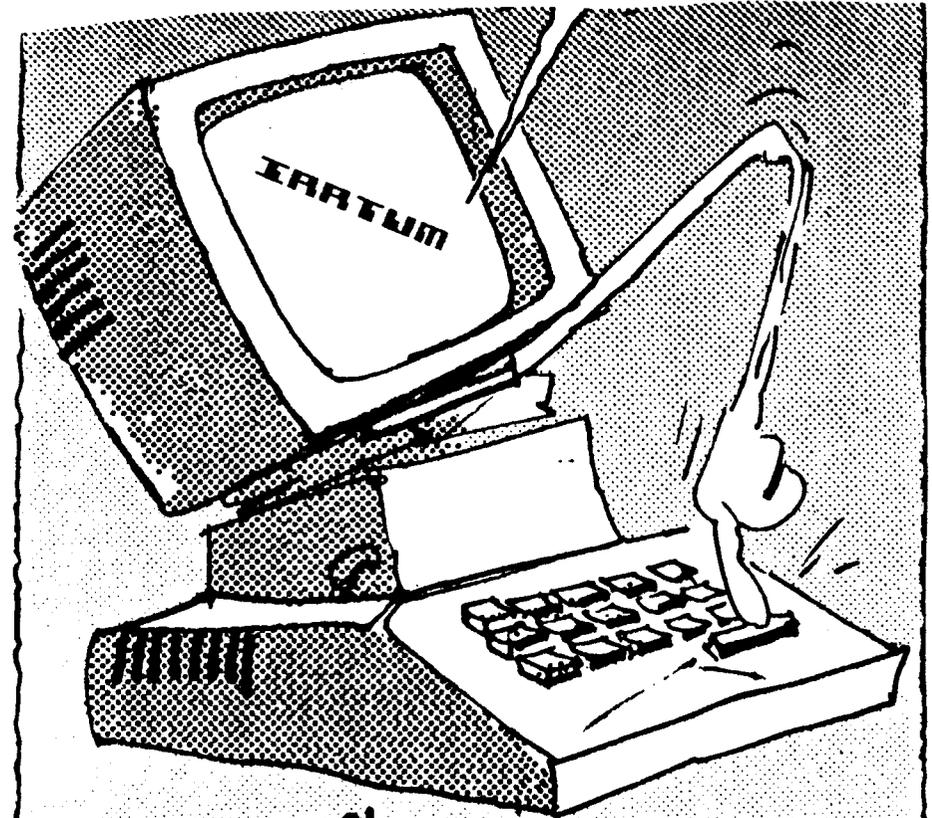
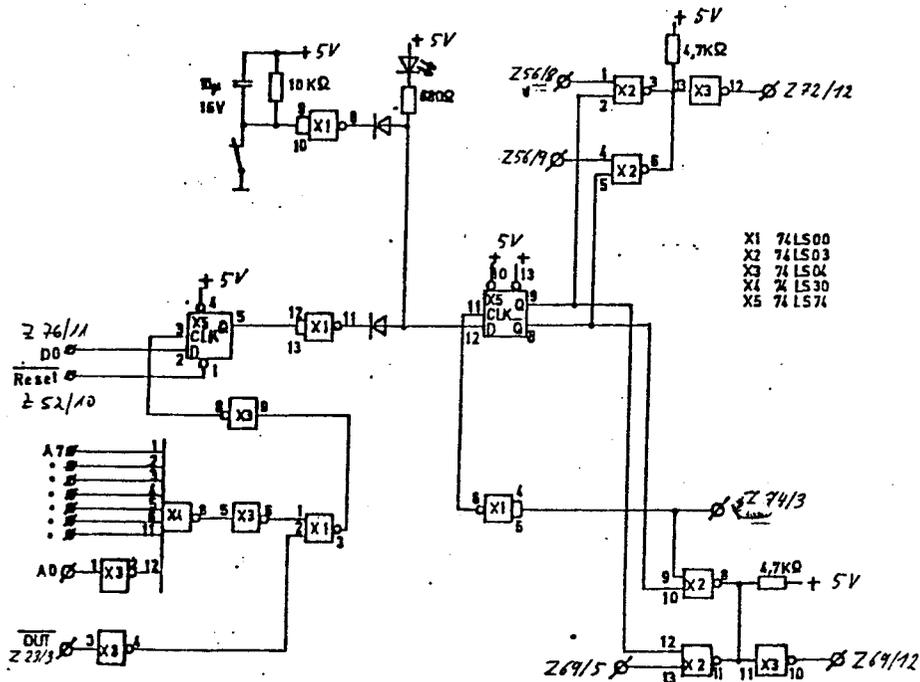
Computerurteil

Ein Mann besitzt zwei Uhren, die beide defekt sind. Die eine geht grundsätzlich 10 Minuten vor, das Laufwerk der anderen ist kaputt. Der Mann beschließt, das Elektrohirn zu befragen.
 Frage an den Computer:
 »Welche Uhr soll ich behalten?«
 Antwort des Computers: »DIE KAPUTTE UHR!«
 Rückfrage: »Weshalb?«
 Antwort des Computers: »DIE ERSTE UHR GEHT STÄNDIG FALSCH. DIE KAPUTTE UHR ZEIGT ZWEIMAL TÄGLICH DIE RICHTIGE ZEIT AN.«

 * Ergänzung zur synchronisierten Taktumschaltung *

Ergänzend zu den Artikeln ueber die synchronisierte Taktumschaltung in den Clubinfo's 8 und 9 moechte ich hier eine Aenderung vorstellen in der die Anschlusspunkte fuer den TRS-80 Modell 1 angegeben sind. Weiterhin moechte ich noch hinzufuegen das die Leiterbahnen zwischen Z 56/8 und Z 72/12. und zwischen Z 69/5 und Z 69/12 zu trennen sind.

Bernd Retzlaff



sagte
 der
 Computer
 und
 knipste
 sich aus.

BÖRSE

WER HAT WAS -- WER SUCHT WAS --

Suche Handbuch für den
GRAPE - Grafiktreiber

Dieter Weiß

ZU VERKAUFEN !!

Genie Modell 1 im TRS-80 M III Gehäuse
mit folgenden Einbauten:
Monitor, HRG, RS 232, 80-Zeichenkarte,
2 Laufwerke 40 Track DS/DD, CPM-fähig
mit externem Netzteil

Preis auf Anfrage

Gerald DREYER

ZU VERKAUFEN !!

Exatron Stringy-Floppy für TRS-80 M I

Mit 45 Waver in verschiedener Länge
(teilweise bespielt) mit 3 Registern
und zugehörigen Taschen

Preis 350,--DM

Jens NEUEDER

Mein EG 64 MBA ist zu haben! Noch immer schwärme ich von ihm, es fehlt ihm auch nicht das geringste, aber ein Kumpel hat einen entwickelt, der noch mehr kann. Wer so gerne wie ich im Computer herumläßt, mag auf die Veröffentlichung der Schaltung in unserer Clubpostille warten. Wer aber lieber die Finger davon läßt, kann meinen für DM 100,- (VB) kriegen (Neupreis DM 195,-, neuerdings soll er wohl DM 150,- kosten). An meinem hängt eine zusätzliche I/O-Platine dran, die alle Leitungen (inkl. IORD und IQWR) enthält, die man für portgesteuerte Peripherie braucht. Es lassen sich damit gleichzeitig ca. 30 Geräte über lauter verschiedene Ports ansteuern. Die Platine ist fest mit dem MBA verbunden und natürlich im Preis inbegriffen.

Ein Freund bietet einen Typenraddrucker TP-II von Smith-Corona für DM 600,- an. Die Daten:

12 Z/s, 10 Z/inch, 105 Z/Zeile

unidirektional, max. Papierbreite 33 cm

Schnittstellen parallel Centronics, seriell RS232C, Puffer 256 Bytes

Wer Interesse hat, wende sich bitte an

Frank Helferich, Schneidemühler Str. 20b, 7500 Karlsruhe 1 (0721-688828).

Arnulf Sopp

*** 80-Zeichen-Karte für unsere Computer zum Sonderpreis.
Bei einer Mindestbestellmenge von 6 Stück bekommen
wir die Karte zum Clubpreis von 215,-DM + Porto.

- Bildschirmformat: 80 * 25 Zeichen/Zeilen
- 8 Zeichensätze à 128 Zeichen (deutsch, amerik., Grafik, invers)
- Bankinglogik für CP/M 2.2 eingebaut. Angepasstes CP/M erhältlich
- Treiber für NEWDOS und GDOS (HDOS) und Colour Genie + Handbuch im Lieferumfang. Ebenso Schalt- und Bestückungsplan, Listings der Treibersoftware + Diskette
- Ausführliche Informationen von Reter Spieß anfordern.

Der Stolz ist das Rückgrat des Menschen, und die Eitelkeit ist das Schwänzchen daran.

Der Computer einmal anders genutzt
 Für viele mag es neu sein, aber mit einem Computer (selbst einen TRS 80) kann man nicht nur Daten verwalten, Briefe schreiben u.ä., sondern sogar rechnen. Das war scheinbar von den Erbauern nicht so vorgesehen, denn die Rechenarten werden in Handbüchern nur nebenbei erwähnt und außerdem rechnet das Ding verdammt ungenau (mit Single und Double), aber mit etwas Glück bekommt man doch ganz brauchbare Ergebnisse. Ich möchte hier etwas zu Sinus und Cosinus sagen. Diese Funktionen braucht man eigentlich oft, wenn irgendwelche Rikme oder Grafiken ins Programm mit hinein-spielen. Also dort, wo es Punkte gibt, die Koordinaten haben und wo Winkel zu finden sind.



»FRAU MÜLLER, IHR SEID DIE SCHÖNSTE HIER!«

Diese Winkel stellt der normale Mensch nun in Grad dar. Wer allerdings glaubt, der Computer tut das auch, der wird enttäuscht: der TRS 80 arbeitet mit Radiant. Um einen Winkel von Grad in Radiant umzuwandeln, nehme ich den Winkel mal Pi und teile das ganze durch 180. Da der TRS 80 Pi nicht kennt, nehmen wir den Winkel einfach mal ".0174533". Jetzt können wir diesen neuen Wert für cos, sin und tan benutzen. Lassen wir uns aber durch atn (arcustangens, Texas Instruments; invtan) einen Winkel ausrechnen, so steht der auch in Radiant. Um die Einheit Grad zu erreichen, nehmen wir den Winkel also mal "57.29578" (bzw. mal 180 durch Pi).

Was machen wir aber, wenn wir arcsin oder arccos benutzen wollen? An diesem Punkt verläßt uns der Computer. Wir müssen diese Funktion also mit atn ausdrücken. Dafür gibt eine Formel:

$$\arccos(x) := \text{ATN}(\text{SQR}(1-x^2)/x)$$

$$\arcsin(x) := \text{ATN}(x/\text{SQR}(1-x^2))$$

Bei x=0 versagt die erste Formel, bei x=1 die zweite. Dann kommt jeweils Pi/2 bzw. 90 Grad heraus.

Nun sind diese Ergebnisse mit Vorsicht zu genießen, denn sie sind nicht eindeutig. Wenn Ihr eine Tabelle von -180 bis +180 Grad mit cos(x) und arccos(cos(x)) (oder sin(x) und arcsin(sin(x))) erstellt, werden sich einige verwunderliche Ergebnisse zeigen, die in der Natur dieser Funktionen begründet liegen. Vor allem liegen die Ergebnisse alle zwischen -90 und +90 Grad. Um also das wahre x-raus zu bekommen, nehmt Ihr folgende Unterscheidungen vor (y = Ergebnis durch atn in Grad!):

arccos:

-90 < y < 0 ;	x = 180 + y	oder	x = -180 - y
0 < y < 90:	x = y	oder	x = -y

arcsin:

-90 < y < 0 ;	x = y	oder	x = -180 - y
0 < y < 90:	x = y	oder	x = 180 - y

Es gibt dafür viele Anwendungen. Eine kleine sei hier genannt: Ihr habt zwei Orte mit den Breiten-/Längengraden (b1/l1) und (b2/l2). Für den Winkel W zwischen den beiden Orten gilt:

$$\cos(W) = \sin(b1) \sin(b2) + \cos(b1) \cos(b2) \cos(l1 - l2)$$

$$\langle \Rightarrow \quad W = \arccos(\dots)$$

Dieser Winkel (in Radiant) mal dem Erdradius genommen, ergibt die Entfernung zwischen den beiden Orten (Luftlinie über der Erdoberfläche, nicht direkt durch den Erdball bohrt!!!).

Also Entfernung KM = 6368 * W (in Kilometer).

Wenn ihr jetzt mal die Entfernung von Hamburg (10 O, 53 N) nach Hamburg (!) berechnet, werdet Ihr überrascht sein, wie unser TRS 80 es schafft, diesen Ort von sich selbst weg zu verschieben.

Für den arccos und arcsin gibt es sicherlich noch mehr Anwendungen (stellt doch mal welche vor!) und die Entfernungsformel läßt sich auch gut einsetzen (z.B.: wie weit hat es jedes Clubmitglied zum nächsten Treffpunkt; natürlich gilt das Ergebnis nur für Leute mit einem Hubschrauber). Ich selbst habe versucht, ein kleines Luftfahrt-Spiel daraus zu machen, bin aber später an einigen mathematischen Problemen gescheitert. Man kann trotzdem tolle Sachen berechnen, vor allem, wenn man sich auf Ausschnitte der Erdkugel beschränkt (Beispiel: ein Flugzeug kann mit vollem Tank 100 km fliegen; wo zwischen Hamburg und München muß es zum Nachtanken landen?)

Viel Spaß mit den Winkeln!

Gerald Schröder

(atn-Formel aus "Basic-Kurs", herausgegeben vom Gymnasium Hittfeld
 Entfernungsformel von Elessar)

Urheberrechtschutz für Programme

Am 1. 7. 1985 ist das Gesetz zur Änderung von Vorschriften auf dem Gebiet des Urheberrechts in Kraft getreten. Primärer Zweck der Neuregelung war es, ein neues Abgabesystem für Fotokopierer sowie für Bild- und Tonaufzeichnungen einzuführen. Gleichzeitig wurde der Urheberschutz auf Computer-Software ausgedehnt.

Hieß es früher in § 2 Abs. 1 des Urheberrechtsgesetzes: „Zu den geschützten Werken der Literatur, Wissenschaft und Kunst gehören insbesondere: 1. Sprachwerke, wie Schriftwerke und Reden“, so lautet der Gesetzestext heute: „1. ... Sprachwerke, wie Schriftwerke und Reden, sowie Programme für die Datenverarbeitung.“

Abwägung im Einzelfall

Damit sind Programme nunmehr ausdrücklich urheberrechtlich geschützt. Zwar hat dies nur klarstellende Bedeutung: Der Bundesgerichtshof hatte nämlich mit Urteil vom 9. 5. 1985 (Aktenzeichen I ZR 52/83) ein Urteil des Oberlandesgerichts Karlsruhe aufgehoben und zur weiteren Sachaufklärung zurückverwiesen. Er bejahte dabei grundsätzlich die Urheberrechtsfähigkeit von Computerprogrammen. Der Gesetzgeber hat diese Entwicklung nunmehr klar gebilligt. Zu beachten ist jedoch, daß § 2 Abs. 2 des Urheberrechtsgesetzes nicht geändert worden ist. Danach gilt: Werke im Sinne dieses Gesetzes sind nur persönliche geistige Schöpfungen. Das heißt, nicht die mathematische oder technische Idee, sondern ihre Verkörperung im fertigen Programm verdient und erhält Schutz. Die rein „handwerksmäßige“ Handhabung allgemeinen Wissens ist somit wie bisher ungeschützt. Der Streit der Experten wird sich also zukünftig nicht mehr um die Frage drehen, ob Software als solche geschützt ist oder nicht; vielmehr wird man sich darüber auseinanderzusetzen haben, welche

Software eine „persönliche geistige Schöpfung“ im Sinne des Urheberrechtsgesetzes ist und welche bloße Handwerks-„Kunst“ ist.

Freiheitsstrafe bis zu fünf Jahren

Geschützt wird Software zukünftig durch die Strafvorschriften des Urheberrechtsgesetzes. § 53 Abs. 4 Satz 2 bestimmt dazu: „Ebenso ist die Vervielfältigung eines Programms für die Datenverarbeitung (§ 2 Abs. 1 Nr. 1) oder wesentlicher Teile davon stets nur mit Einwilligung des Berechtigten zulässig.“ Verstöße gegen dieses Verbot werden gemäß § 106 mit Freiheitsstrafe bis zu einem Jahr oder Geldstrafe bestraft. Handelt es sich um gewerbsmäßige unerlaubte Verwertung, greift § 108 a ein. Danach ist eine Freiheitsstrafe bis zu fünf Jahren oder eine Geldstrafe vorgesehen und auch schon der Versuch mit Strafe bedroht.

Zitat des Monats

Im Amiga wird der Mikroprozessor sehr viel effektiver genutzt, da er in diesem Computer nur die Aufgabe, die er am besten beherrscht, nämlich das Addieren und Vergleichen von Zahlen, auszuführen hat.

Aus einem Test in der Zeitschrift „Computer Persönlich“, 1985, Heft 19.

Das genannte Einwilligungs-Erfordernis ist nur dann gegeben, wenn entweder das ganze Programm oder wesentliche Teile davon verwertet werden sollen. Die Verwertung unwesentlicher Teile wird nicht vom Verbot – und somit auch nicht von den Strafvorschriften – erfaßt. Somit taucht das Problem auf, abzuklären, was wesentlich oder unwesentlich ist. Dies wird sich nur von Fall zu Fall klären lassen. Maßgebend dürfte die Relation zum Ganzen sein.

Interessen-Wahrnehmung durch VG Wort

Es empfiehlt sich, eventuelle Vergütungsansprüche durch eine Verwertungsgesellschaft wahrnehmen zu lassen. Insofern bietet sich die VG Wort an. Sie macht Ansprüche eines Software-Erstellers aus dem Urheberrechtsgesetz für diesen geltend. Voraussetzung ist, daß der jeweilige Urheber der VG Wort die von ihm geschaffenen Werke mitteilt. Nähere Einzelheiten sind bei der VG Wort, Goethestraße 49, 8000 München 2, zu erfragen (siehe auch mc-Impressum).

Anmerkung der Redaktion: Auf Anfrage teilte die VG Wort uns mit, daß es bisher noch keine konkreten Fälle der Interessenwahrnehmung durch sie im Bereich der Computer-Software gebe. Die VG Wort nehme lediglich „Sekundärinteressen“ wahr, während die primäre Lizenzvergabe beim Software-Entwickler bzw. bei dem von ihm lizenzierten Unternehmer liege.

Literatur

- [1] Kahlen, H.: Urheberrechtlicher Schutz für Software. mc 1984, Heft 11, Seite 52.
- [2] Kahlen, H.: Verbessertes Urheberrechtsschutz. mc 1985, Heft 2, Seite 96.

den an rechtmäßig erworbenen (diese werden nämlich auch mitgenommen, da die Ermittlungsbeamten hier nicht unterscheiden können) Disketten erst gar nicht entsteht.

Wie mir der ermittelnde Beamte auf Anfrage erklärte, werden die Disketten dann an die Staatsanwaltschaft weitergeleitet und wahrscheinlich durch einen Verband der Softwareautoren c/o Commodore geprüft. Danach wird dieser Verband die Urheber/Verwertungsberechtigten über diesen Vorgang informieren, die dann ihrerseits drei Monate Zeit haben, Strafantrag gegen den Besitzer/Kopierer der Disketten zu erheben. Ferner ist damit zu rechnen, daß die Steuerfahndung eingeschaltet wird. Auf mich wartet nun die Prüfung der Disketten, die entscheiden wird, ob Strafantrag wegen Verletzung des Urheberrechts gestellt werden kann oder nicht. Da ich in keinem Fall eine entgeltliche Weitergabe von Disketten betrieben habe, wird mir eine Anzeige wegen Betruges erspart bleiben, jedoch könnte ich schlimmstenfalls mit einer Anklage wegen Hehlerei rechnen und so könnte ein Hobby zur Vorstrafe führen.

Urheberrecht

Zur Zeit ermittelt die Kripo gegen (mir bekannt) weit über 100 Softwarefreaks, die 1982/83 Tauschverbindungen zu anderen hatten. Die Adressenliste resultiert aus einer Durchsichtung im süddeutschen Raum. Hier beschlagnahmte die Kripo Tausch- und Adressenlisten sowie Disketten.

Ich glaube, richtig in der Annahme gehen zu können, wenn ich die Behauptung aufstelle, zu der Zeit hat jeder mit jedem irgend etwas getauscht. Ein Unrechtsbewußtsein beschränkte sich eigentlich nur auf den Verkauf kopierter Disketten, davon haben aber die meisten abgesehen.

Ich selbst habe an diesem regen Tauschhandel teilgenommen und seit Mitte Juli 1985 ein Ermittlungsverfahren wegen Verdachts nach § 106, 108 Urheberrechtsgesetz (diese Paragraphen wären vielleicht hier einmal abdruckend) anhängig. Entgegen Ihrer Behauptung in CP 15/85, Seite 75, wird nicht auf Antrag des Urhebers/Verwertungsberechtigten hin dieses Verfahren eingeleitet, sondern es erfolgt zur Zeit eine groß angelegte Ermittlung durch die Staatsanwaltschaft. Die Ermittlung begann morgens um 8 Uhr mit der Durchsichtung gemäß Beschluß des Amtsgerichtes der Wohn- und sonstigen Räume. Die zwei Kripobeamtinnen drohten mit der Mitnahme meiner inzwischen mehreren Computer und sämtlicher Disketten. Da ich nachweisen konnte, diese Geräte zum rechtmäßigen Brotenerwerb zu nutzen und die Suche vorwiegend Spiele betraf, wurden bei mir 80 Disketten mit Spielen (nur Spiele!) sichergestellt, unter denen natürlich auch Kopien waren. Dabei habe ich die Feststellung machen müssen, daß die Ermittlungsbeamten (glücklicherweise) fast gar nicht in der Materie steckten. Ich mußte ihnen erst erklären, wo man Disketten anfaßt und wie man diese lagert. Jedem, dem so etwas passiert, kann ich nur raten, ebenfalls Hinweise zur Behandlung von Disketten den Ermittlungsbeamten auszuhandigen, damit Scha-

Das Unrechtsbewußtsein ist mir bisher noch nicht gekommen und ich glaube, es wird auch nicht kommen. Bei den Softwarepreisen war man überhaupt nicht in der Lage, Disketten zu kaufen. Der Tausch war die einzigste Möglichkeit, andere Softwareprodukte kennenzulernen. Ohne diesen Tauschhandel und das geben sogar Experten zu, wären bestimmte Programme wie zum Beispiel Wordstar oder Multiplan nie bekannt geworden, und die Umsatzzahlen von Spielherstellern hätten sicherlich klägliche Größen. Ich werde es auch nicht einsehen, warum renommierte Firmen auf diese kostenlose Werbung verzichten wollen. Da ich mittlerweile selbst Softwareautor bin, nutze ich darum die Kopiergewohnheiten einiger, um mich bekannt zu machen. Ich sehe es heute noch gerne, wenn ich erfahre, daß einer aus Österreich meine Programme hat und stolz anderen vorführt; mir würde nur der gewerbsmäßige Verkauf dieser Programme sauer aufstoßen.

Ich bin sicher, daß diese über 100 mir bekannten Ermittlungsverfahren nicht die einzigen sind und kann nur jedem, der damals oder heute Disketten tauscht, raten, alle Tausch- und Adressen sofort zu vernichten oder zu verschlüsseln sowie eventuell seine Tauschpartner bitten, ein Gleiches zu tun (Der Verfasser dieses Briefes möchte aus verständlichen Gründen nicht namentlich genannt werden).

H.-P. Schmid hat einen neuen Club!

Vor ein paar Tagen bekam ich einen Fragebogen von Herrn Dr. med. Friedrich Lücke
Deisterallee 14 A
3250 Hameln 1,
der offenbar den Zweck verfolgte, die gemeinsame Hard- und Softbasis der Mitglieder des Hamburger TRS-80- und VG-User-Clubs herauszufinden. Merkwürdig daran war lediglich, daß ich von diesem Club nie gehört habe und höchstens in Abwesenheit und Unkenntnis zur Mitgliedschaft verurteilt worden sein kann. Daß der Club keine Beiträge erhebt, machte mich ebenfalls stutzig, denn wer spendiert das Porto und die Kopierkosten für Infos usw.? Also bat ich Herrn Dr. Lücke um Aufklärung und kriegte einen offensichtlich für viele Empfänger konzipierten Formbrief, der folgendes aufdeckte: Zwei weitere Adressaten (die geantwortet hatten; wer weiß wer noch alles!) kannten den Club überhaupt nicht. Das wirft ein deutliches Licht auf die "Mitglieder"-Verwaltung dieses Clubs. Herr Dr. Lücke weiß nicht, wer für die Mitgliederliste verantwortlich ist!!! Ist der Boß des Clubs der Große Unbekannte?

Nein, es ist ein allzu Bekannter, glaube ich: Mit dem Formbrief kam auch eine Mitgliederliste (mit einigen Streichungen, versteht sich, denn sie enthielt auch Leute wie unsereinen).

Hans-Peter Schmid ist dort Mitglied!

Die vielen Merkwürdigkeiten lassen einen interessanten Schluß zu: Sollte der bekannte Jäger von Raubkopierern, der nur so Geld für die von ihm vertriebenen Programme erlangen kann, wieder mal einen Club gegründet haben? Wenn man dann den Kreis der Mitglieder scheinbar durch ein paar Ahnungslose künstlich erweitert, erweitert man damit auch die Möglichkeiten, Geld zu verdienen. Denn irgendwer wird schon so unvorsichtig sein, Software mit ihm oder einem seiner Helfer zu tauschen.

Herr Lücke hatte vor meiner Antwort an ihn wahrscheinlich keine Ahnung, in welcher prominenter Gesellschaft er sich befindet, sonst hätte er mir nicht ausgerechnet ein solches Intimum wie die Liste der Mitglieder geschickt. Er ist demnach wohl kaum zu zeihen. Ebenso wenig sind es die Mitglieder, die gleichzeitig zu unserem Club gehören. Überhaupt ist der Hamburger Club - von ihm gegründet oder vielleicht auch nicht - wohl nur ein willkommenes Werkzeug für Schmid, ansonsten aber veräutlich ein Haufen von netten Kollegen wie wir, mit Zielen wie den unseren. Zu warnen ist deshalb nicht vor Dr. Lücke, auch nicht vor dem Club schlechthin, aber davor, den Fragebogen auszufüllen, wo es um die Software geht. Und vor allem davor, mit Angehörigen dieses Clubs Programme zu tauschen, die man nicht selber geschrieben hat.

Seien wir mal ehrlich; fast jeder von uns kennt einen, der einen kennt, der schon einmal etwas raubkopiert hat. Wahrscheinlich ist Gervatter Schmid sogar formaljuristisch im Recht. Also haben wir genug Anlaß, auf den Fragebogen gar nicht erst zu reagieren. Schon gar nicht mit einem "geklaften" NEWSSCRIPT, das Schmid vertreibt.

„Die Beleidigungen
sind die Argumente jener,
die über keine Argumente
verfügen.“

Eine gute und eine schlechte Nachricht:
Erst die schlechte: TCS in St. Augustin hat pleite gemacht.
Nun die gute: die Geschäfte der TCS GmbH übernimmt die Phoenix Computer GmbH,
5461 Windhagen, Klarenplatz. Geschäftsführer ist Troaschläger!
(Wie sowas geht, ist mir ein Rätsel; aber seitdem AT&T (amerikanische Tele-
fongesellschaft) in deutschen Zeitschriften für Telefongespräche über'n Teich
wirbt, wundert sich gar nichts mehr.)
Frei wiedergegeben aus: CP 22/85

Wer sich für das ans-Forth interessiert, findet Informationen dazu in:
Ken Knecht: Einführung in Forth; Markt und Technik Verlag AG; 8013 Haar; ISBN
3-922120-73-3; 218 Seiten; 58,- DM
aus: CP

Gerald Schröder

„Keine Zeit ist mit der
Zeit zufrieden; das heißt,
die Jünglinge halten die
künftige für idealer als die
gegenwärtige, die Alten die
vergangene.“

LM 381 Vorverstärker IC

M 0214 A 04 /BAS

Berechnet einen rauscharmen Vorverstärker mit dem Spezial-IC LM381.

Umwelt

D 0215 A 03 /BAS

Eine Regierungssimulation, bei der es vor allem auf die richtige, ökologische Wirtschaftsführung ankommt.

ZAP

G 0216 A 01 /CMD

Ein mittelmäßiges Schießspiel mit Ton.

Kassettenverzeichnis

S 0217 A 02 /BAS

Mit diesem Programm kann man sehr leicht und schnell für seine Musik- (oder Programm-) Kassetten Inhaltsverzeichnisse drucken. Autor: Günther Wagner

Disassembler

S 0219 D 04 /BAS

Ein einfacher Disassembler in BASIC für das Model 4 von Tandy.

MAKEDO

S 0221 D 02 /BAS

Programm zur Erstellung von DD-Files (ähnlich CHAINBLD) für das Model 4 von Tandy

Scrollprotection

S 0222 D 03 /BAS

Mit diesem Programm kann man 1-7 Zeilen des Bildschirms vor dem scrollen schützen. Nur für Model 4 von Tandy!

SETRESET

S 0223 D 03 /BAS

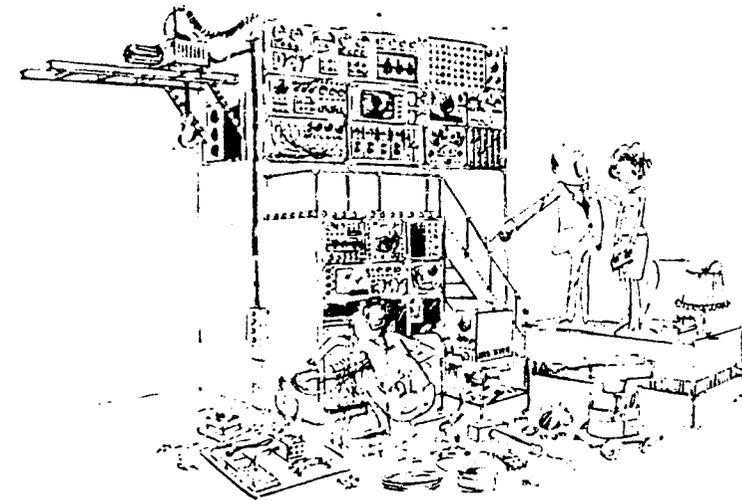
Dieses Programm simuliert die BASIC-Befehle SET, RESET und POINT auf dem Model 4 von Tandy.

MENUE4

S 0224 D 04 /BAS --

Ein Programm zum Aufruf von BASIC- und JCL-Files auf Tastendruck. Nur für Model 4 von Tandy

„Es ist nett, wichtig zu sein. Aber es ist wichtiger, nett zu sein.“



„So, Fraulein Knickdübel, passen Sie gut auf, wie er's macht – dann brauchen wir nicht jedesmal auf die teuren Fachleute zu warten, nur weil das Ding ein bißchen bockt!“

Nr. 0001: BASIC-Brevier: Eine Einf. in d. Prog. v. Heimcomp.
Siegmar Wittig --- Heinz Heise GmbH
Ein BASIC-Kurs fuer Nicht-Mathematiker und echte Amateure. Neben Grund- und Aufbaukurs gibt es zahlreiche Beispiele (eine gute Sammlung!).

Nr. 0002: Computerwissen
Michael Scharfenberger --- Markt & Technik
TIPS fuer die Auswahl und Beschreibung von Anwendungsmoeglichkeiten von Hard- und Software sowie Erklaerung von mehr als 500 Begriffen.

Nr. 0003: Computerspiele und Knoeleien programm. in BASIC
Ruedeger Baumann --- Vogel-Buchverlag (CHIP-Wissen)
Von der Spielidee ueber die Spielstrategie kommt es zum Programm selbst. Keine Sammlung von Spielkonserven - keine Programmierkenntn. erford.

Nr. 0004: Mein Home-Computer - Eine Verbraucherfibel
- --- Vogel-Verlag (HC-Leserservice)
Die besten TIPS fuer Kauf und Anwendung von Home-Computern.

Nr. 0005: Programmieren mit dem ZXB1 in Basic u. Masch.-code
E. Floegel --- Hofacker, Holzkirchen
Sammlung von Spiel-, Schul- und anderen Programmen sowie einem Kapitel ueber die Programmierung des Prozessors Z80 (gute Programme dabei)

Nr. 0006: Games For Your TRS-80
Chris Palmer --- Virgin Books (Great Britain)
Sammlung von 20 Basic-Spiel-Programmen und einer Anleitung, wie man bessere Programme schreibt.

Nr. 0007: Introduction to TRS-80 Graphics
Don Inman --- dilithium Press (Portland-USA)
In diesem Buch wird gezeigt, was man mit der TRS-80 Graphik machen kann und vor allem wie. Beispiele und Aufgaben veranlassen zum experiment.

Nr. 0008: More Basic Computer Games
David H. Ahl --- Creative Computing Press, USA
84 Spiele fuer den TRS-80, wobei einige sehr interessante dabei sind. Das Buch ist fuer Freunde von Basic-Computer-Spielen nur zu empfehlen.

Nr. 0009: BASIC: Dateien, Listen und Verzeichnisse
Busch Rudolf --- Franzis-Verlag GmbH, M nchen
Eine Software-Sammlung mit vielen nuetzlichen Programmen in Kursform (also mit Lern-Effekt).

Nr. 0010: BASIC: Matrix-Operationen
Busch Rudolf --- Franzis-Verlag GmbH, M nchen
Eine Software-Sammlung mit vielen nuetzlichen Programmen in Kursform (also mit Lern-Effekt).

Nr. 0011: TRS-80 PROGRAMS
Tom Rugg und Phil Feldman --- Dilithium Press, Beaverton, USA
32 BASIC-Programme (Erziehung, Anwendung, Spiele, Graphic, Mathematik und Verschiedenes) fuer Level II.

Nr. 0012: Programme und Tricks fuer Genie I und Genie II
Clemens Becher, Franz Seiger --- ?
Viele Programme, TIPS und Tricks fuer den Genie.

Nr. 0013: BASIC: Alles ueber PEEK und POKE
Heiko Requardt --- Franzis-Verlag
Eine Software-Sammlung in BASIC (mit vielen guten TIPS und Tricks fuer den 'Amateur').

Nr. 0014: TRS-80 und Video Genie ROM-Listing fuer Level II
Luidger Roekrath --- ?
ROM-Listing, RAM-Adressen, I/O-Adressen, Unterprogramme, Basic-Anweisungen und Funktionen, Aufzeichnungsformate auf Cassette, ...

Nr. 0015: Machine Language Disk I/O & Other Mysteries
Mihael J. Wagner --- IJG Inc., 1953 West, USA
Alles rund um Disk und DOS wird erkl rt - mit zahlreichen wertvollen Hinweisen und TIPS.

Nr. 0016: Das Modem-Sonderheft (Daten bertr. mit Mikrocomp.)
verschiedene --- MC, Franzis-Verlag, M nchen
Grundlagen, Schaltungstechnik, DF -Programme

Von einem Text für mehr als 10 Leser, also einer Veröffentlichung im wörtlichen Sinne, erwartet der Käufer zu Recht ein fehlerfreies Deutsch. In "Das DOS Buch" von Hartmut Grosser, bei Rückrath zu haben, ist schon der Titel falsch. Auch im Text ist manche orthographische Eigenwilligkeit zu finden. Aber damit ist auch, was Negativkritik an diesem Buch betrifft, mein ganzes Pulver verschossen. Das ist das Beste, was nach meiner Kenntnis zu diesem Thema auf dem Markt ist! Ein Reader für Einsteiger, die begreifen wollen, wie ihr DOS funktioniert, ein unentbehrliches Vademecum auch noch für den ganz eingefleischten Disk-Jockey, der der alten Tante NEWDOS-80 oder ihren Neffen aus der TCS-Sippe endlich das Laufen beibringen will.

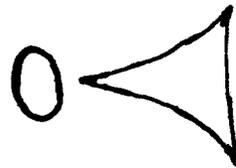
In den ersten beiden Kapiteln wird erläutert, wie der Controller arbeitet und wie sich das ROM sein Lieblings-DOS reinschlurft. Anschließend folgt das Kapitel, das alleine schon den Preis von (leider nicht weniger als) 65 Mark wert ist: Ein ausführlich kommentiertes Listing von SYS0/SYS. Dort werden in zusätzlichen Kommentarzeilen auch Fehler gezeigelt, die in einem späteren Kapitel noch einmal zusammengefaßt sind. Auch alle wichtigen Einsprungsadressen für den Maschinensprache-Anwender sind noch einmal lexikographisch und alphabetisch zusammengestellt.

Wer gerne in der Library fummelt, um eigene DOS-Befehle zu implementieren, wird in Kapitel 5 erschöpfend darüber informiert, was er dazu wissen muß. Solche Befehle landen sinnvollerweise möglichst in einem SYS-File (bei Grosser weiblich), und darüber erfährt der Leser einiges. SYS-Files lassen sich ohne nähere Kenntnisse des Directory (bei Grosser männlich) kaum adressieren. Folgerichtig läßt Kapitel 7 hierüber keine Fragen offen.

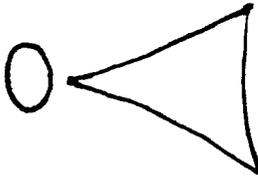
Die Banking-Freaks unter uns, Verliebt in ihren EG 64 MBA oder wen auch immer, staunen in Kapitel 8 nicht schlecht: Das DOS unternimmt bei bestimmten Verrichtungen Dummy-Schreibzugriffe auf das ROM. Am Samstagmorgen kam das Buch mit der Post, am Sonntagnachmittag waren die entsprechenden Adressen (im Buch natürlich haarklein erläutert) auf den Bildschirm-Adreßbereich umgezapt. Wer nämlich gerade mit einer modifizierten ROM-Kopie im Parallel-RAM arbeitet, kann z. B. bei einem Verify-Gang sein blaues Wunder erleben. Daß ich nun etwa Kommentar-Records live auf dem Screen erlebe, ist weniger störend als ein ausgewachsener Systemcrash. Die Tastatur ab 3800 wäre eleganter gewesen, aber seit kurzem ist sie bei mir nur noch 256 Bytes lang. Diese Adresse sei aber allen empfohlen, die das "Sichere Plätzchen" von Helmut Bernhardt (c't 5/85) nicht haben.

Im Anhang gibt es einen Artikel meines langjährigen Freundes Ulrich Heidenreich über seinen neuen Library-Befehl ID. Mit ihm erkennt das DOS automatisch die PDRIVE-Parameter fremder Disketten. Das Gewühle beim Ausprobieren, wenn mal wieder jemand seiner Platte keinen Zettel beigefügt hat, gehört damit der Vergangenheit an. Viele von euch benutzen den Befehl bereits mit Erfolg. Wie ich neidvoll zugeben muß, ist ausgerechnet dieses einzige von H-DOS, das nicht von mir stammt, das interessanteste Feature.

In dieser Rezension habe ich nur eine Auswahl dessen erwähnt, was im "DOS-Buch" (die Freiheit des nachträglichen Bindestrichs sei mir erlaubt) steht. Es ist nur dasjenige, was mich persönlich besonders interessiert. Versteht sich, daß alles für NEWDOS-80 2.x, G-DOS 2.x und Colour-DOS gemünzt ist. Die jeweiligen Unterschiede sind extra aufgeführt. Und das alles für alle Genies und alle Modelle des TRS-80. Selten habe ich beim Kauf eines Buches pro Groschen so viel Information bekommen.



Für das DOS-Buch ist bei Peter Spieß eine Sammelbestellung eingerichtet. Clubpreis: 59,- DM abzgl. 2% (Porto ist schon dabei). Bitte Bestellung baldmöglichst abschicken und die Unterschrift nicht vergessen.



- Impressum -

Vorstand

Kontaktadresse für Clubangelegenheiten
Clubbücherei /Fundgrube
Clubkasse

Günther WAGNER
Gartenstraße 4
8201 Neubuern
Tel.: 08035 /3361
< 18 - 20 Uhr >

Programmbibliothek

Kontaktadresse

Hartmut OBERMANN
Schwalbacher Straße 6
6209 Heidenrod /Kemel
Tel.: 06124 /3913

Redaktion

Kontaktadresse

Jens NEUEDER
Panoramastraße 21
7178 Michelbach /Bilz
Tel.: 0791 /42877
tagsüber 0791 /44-667

Adventure-Ecke

Kontaktadresse

Alexander WAGNER
Therisenstr. 21c
8224 Chieming
Tel.: 08664 /1500

Hardware

Kontaktadresse

Walter ZWICKEL
Lengfelden 123
5101 Bergheim (Austria)
Tel.: 0043662 /51130

Redakteure
dieser Ausgabe

Jens Neueder * Hartmut Obermann
Bernd Retzlaff * Gerald Schröder
Arnulf Sopp * Günther Wagner
sowie Artikel aus: MC und
Computer Persönlich

Bankverbindung des CLUB 80

Sparkasse Rosenheim, BLZ 711 500 00
auf Konto-Nr. 194 712
Postscheckkonto der Sparkasse
Nr. 8077-801

Das INFO erscheint zweimonatlich.

Es erfolgt keine Zensur oder Kontrolle der jeweiligen eingeschickten Infobeiträge durch die Redaktion.

57

Schluß

Hallo Club-80er,

Zum Abschluß des INFO wie immer ein paar Redaktionsworte.

Gerade habt Ihr das 10. und in diesem Jahre auch letzte Clubinfo in der Hand. Bei dieser Gelegenheit möchte sich nun die Redaktion bei Euch für Eure Mühen als Beitragende sowie für Euer Interesse am Clubinfo bedanken. Die Zusammenarbeit mit Euch hat mir sehr viel Freude bereitet. Ich hoffe, Ihr laßt mich auch im nächsten Clubinfojahr nicht im Stich und sendet mir Eure Erfahrungen, Kniffe und anderes Wissenswerte aus dem Computerleben.

Gleichzeitig habe ich aber auch den Wunsch für's nächste Jahr, daß sich mehr Mitglieder an dem Clubinfo mit Beiträgen beteiligen, denn in diesem Heft stammen fast zweidrittel der Artikel -wie Ihr sicher selbst gemerkt habt- aus der Feder von Arnulf und Hartmut. Daß für die Beiden das Clubinfo nun nicht mehr so interessant ist, dürfte klar sein. Aber Ihr könnt Euch ja bei den zwei eifrigen Schreibern mit Eurem Clubinfobeitrag revangieren. Wie schon des öfteren erwähnt, finden auch kurze Beiträge ihren Platz im Clubinfo.

~~Achtung !!!~~
REDAKTIONSSCHLUß
für INFO 11 ist der
31. DEZEMBER 1985

69

Nun noch zwei Neuerungen :

1. In Zukunft wird für die Clubinforubrik Börse eine "Hot-Line" eingeführt. Dies bedeutet für Euch, daß Ihr mich Montags abends zwischen 21Uhr und 22Uhr anrufen, und Eure Börsenaufträge durchgeben könnt. Sie werden dann gesammelt und in der nächstmöglichen Clubinfo veröffentlicht. Ich hoffe, daß dies unsere Börsenspalte etwas wiederbeleben wird.
2. Ab diesem INFO ist die Adressenseite die letzte Infoseite, und zwar so "eingebaut", daß Ihr ohne Aufschlagen der INFO die aktuellen Mitgliederadressen lesen könnt. Die "Rückseite" der Adressenseite bleibt frei. Diese Seite ist für Mitteilungen der Redaktion an den Vorstand, die Redakteure oder andere Informationen an Euch persönlich vorgesehen. Ein weiterer Grund ist, daß ich die Seite bei Probeinfos an Nichtmitglieder abtrennen kann.
Ihr seht, auch wir betreiben Datenschutz !!
Falls nach der Adressenseite noch Seiten angehängt sind, so sind diese zum Abtrennen für Euch oder als Antwortseite zum Zurücksenden gedacht.

Als letztes möchte ich Euch allen zu Weihnachten und zum Jahreswechsel alles Gute und weiterhin Happy Computing wünschen.
(Das nächste INFO kommt erst im Januar 86)

So, nun am Ende des 10. INFO angelangt, hoffe ich, daß Ihr auch diesmal wieder Gefallen an Eurer Clubinfo habt, und verbleibe --

-- bis zum nächsten INFO Euer

J. Neudecker

CLUB 80 Mitgliederadressenliste

Name	Vorname	Straße	Plz	Stadt	Telefon
Alber	Herbert	Alemannenstr. 20	7732	Niedereschach	07721 /7102
Baldes	Hans	Johann-Strauss-Str. 6	8025	Unterhaching	089 /6115179
Beckhausen	Wolfgang	Vuerfelder-Kaule 30	5060	Bergisch-Gladbach 1	02204 /62781
Boecker	Dieter	Lehmweg 4	2930	Varrel 1	04451 /7640
Boeckling	Ulrich	Am Sonnenhang 11	5414	Vallendar	0261 /69522
Buskowiak	Thomas	Eschersheimer Landstr. 257	6000	Frankfurt 1	069 /5601621
Dreyer	Gerald	Am Speiergarten 8	6200	Wiesbaden-Bierstadt	06121 /508218
Drowälder	Bernd	Buchentalsweg 8	4939	Steinheim	
Fröhlich	Burghard	Brandströstraße 11	4350	Recklinghausen	02361 /63416
Grajewski	Werner	Zedernweg 29	4220	Dinslaken	02134 /54573
Hallupp	Matthias	Stockumer Straße 405	4600	Dortmund 50	0231 /756413
Held	Manfred	Stirnerstr. 22	8835	Pleinfeld	09144 /6563
Hermann	Klaus	Gartenstr. 22	7401	Pliezhausen	07127 /70024
Hummel	Anton	Schubertstr. 2	7612	Haslach	07832 /8289
Jablotschkin	Rainer	Thiekamp 29	4780	Lippstadt 8	
Kasper	Dieter	Zeppelinstr. 9	8952	Marktoberdorf	08342 /1630
Koenig	Hans J.	Hebbelstr. 25	2080	Pinneberg	04101 /209444
Konrad	Josef	Anzengruberstraße 35	8038	Gröbenzell	08142 /8494
Kuhn	Eckehard	Im Dorf 14	7443	Frickenhausen 1	07022 /45417
Marx	Andreas	Mecklenburgring 48	6600	Saarbruecken	0681 /812983
May	Holger	Marienstr. 9	5768	Sundern 2	02935 /1668
Neueder	Jens	Panoramastr. 21	7178	Michelbach/Bilz	0791 /42877 (dienstl.44-667)
Obermann	Hartmut	Schwalbacher Str. 6	6209	Heidenrod 1	06124 /3913
Perschbach	Patrick	Waldstr. 52	5000	Koeln 91	0221 /872118
Piller	Walter	Ronenstrasse 8	CH-8835	Feusisberg	01 /7847418
Preuss	Lothar	Lautshof 13	2940	Wilhelmshaven	04421 /84247 (dienstl. 804-1)
Rank	Heinrich	Fruehlingstr. 2	8080	Fuerstenfeldbruck	08141 /3791
Retzlaff	Bernd	Kleiner Sand 98	2082	Uetersen	04122 /43551
Schaefer	Walter	Rathausstr. 4	8160	Miesbach	08025 /1631
Schneider	Manfred	Rheinkasseler Weg 11	5000	Koeln 71	0221 /707044
Schrewe	Christian	Fliederweg 32	4000	Duesseldorf 31	0203 /740897
Schroeder	Gerald	Am Schuetzenplatz 14	2105	Seevetal 1	04105 /2602
Smerling	Frank	Tangstedter Str. 5	2080	Pinneberg	04101 /207284
Sopp	Arnulf	Wakenitzstr. 8	2400	Luebeck 1	0451 /791926
Spiess	Peter	Trugenhofenerstr. 27	8859	Rennertshofen	08434 /454
Stephan	Hans-Martin	Am Glasesch 9a (Postf. 1207)	4506	Hagen a.TW.	05401 /99585
Stevens	Peter	Postfach 56	4600	Dortmund 1	
Trapp	Harald	Kranichstr. 46	4270	Dorsten 1	02362 /42497
Troesch	Eberhard	Altenessener Str. 414	4300	Essen 12	0201 /342324
Voigtlaender	Holm	Haselnussweg 30	6940	Weinheim	06201 /65241
Wagner	Alexander	Theresienstr. 21c	8224	Chieming	08664 /1500
Wagner	Guenther	Gartenstr. 4	8201	Neubeuern	08035 /3361
Weiß	Dieter	Bürglestraße 3	7209	Wehingen	07426 /7194
Wies	Jean-Claude	Harthweg 9	6600	Saarbruecken	0681 /582513
Wucherer	Jürgen	Brauneggerstraße 14	7750	Konstanz	07531 /29145
Zwickel	Walter	Lengfelden 123	A-5101	Bergheim	0043662/51130

Wegen Systemwechsel ausgetreten
Jürgen Bozek (zum 1.1.86)

Neuzugang :
Bernd Drowälder

Adressenänderung
Peter Stevens

überprüft bitte Eure Adresse
auf Richtigkeit !!!