

CLUB 80

Clubinfo

der

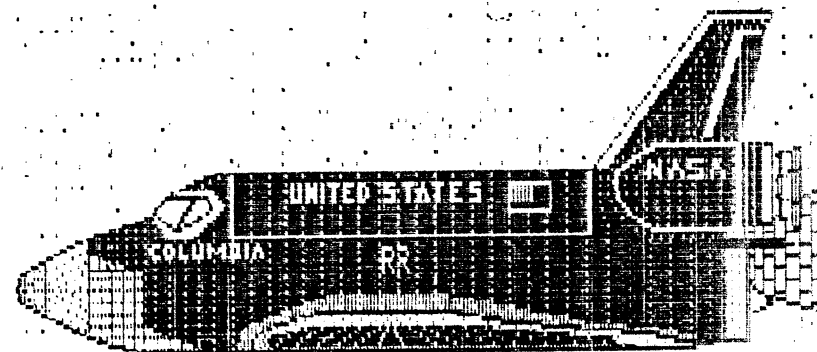
MANOV -

GENIE -

UND KOMBIER

ANWENDER

12. AUSGABE



Kontaktadresse: CLUB 80 / Günther WÄGNER / Zellenstraße 4 / 82071 Neubauern

Telefon: 08915/3351 (19 - 20 Uhr)

Inhaltsverzeichnis

Seite:

Clubinternes

Der Vorstand informiert	81
Clubtreffen in Holzhausen	82 - 84

Software

Nachtrag zu INPUT	85
Programmierwettbewerb	87 - 89
Verbessert: IO-Port	11 - 12
JKL - druckerschonend	13 - 14
Volle Library ?	15 - 16
DDE - Genie IIIs	17 - 20
HRG - Genie IIIs	21 - 26
Zeichensatz neu - Genie IIIs	28 - 32
Cell mit BASCOM	33 - 35
Modell 4 intern	37 - 43
Modell 4 soft	45 - 59

Hardware

CLUB 88 Hard	61
Datenübertragung per Fernsehbild	62
88-Zeichenkarte	63 - 64
Speed Writing	65 - 68
Modell 4 hard	69 - 78
Bildschirmausgabe aufpolieren	79 - 84
ECB-Bus	91

Seite:

Börse

Wer hat was ???--Wer sucht was ???	85 - 98
Fragen, Fragen, Fragekasten	92

Sonstiges

In Kürze	68
Absehnfälle	94
Diskussion	95 - 96

Programmbibliothek

Neue Programmbibliothekenliste liegt diesem INFO bei

Club -Bücherei

Bücherliste CLUB 88	97 -101
-------------------------------	---------

Die letzten Seiten

Impressum	103
Schluß	104
Clubmitgliederadressen	am INFO-Ende
Persönliches Kärtchen	liegt dem INFO bei

Liebe Mitglieder,

zum letzten Mal an dieser Stelle als Vorstand des Clubs möchte ich zunächst allen Mitgliedern recht herzlich danken, die mich in meine Arbeit unterstützt haben. Mein besonderer Dank geht an den Jens, dem Hartmut und dem KaiJott. Sie haben sich bereit erklärt, verantwortlich ein Stück des Clubs zu übernehmen und Ihre Freizeit dafür zu opfern.

Wir brauchen nun Jemanden (oder auch mehrere), der bereit ist, für das Amt des Vorstandes zu kandidieren. Die Satzung sagt ganz klar aus, daß der Vorstand von der Gesamtheit der beim Clubtreffen anwesenden Mitglieder gewählt wird. Ich hoffe, daß sich bis zum Clubtreffen Kandidaten finden, denn eines ist gewiß: Ich lasse mich als Vorstand nicht mehr aufstellen.

Ich will nun einen kurzen Abriß der Tätigkeiten des Vorstandes geben. Der Vorstand ist derjenige, welcher den Club nach außen hin vertritt. Das heißt, er hat Anfragen von Interessenten zu beantworten. Die neuen Mitglieder hat er in den Club aufzunehmen. Er ist für die Kassenführung der Hauptkasse verantwortlich. Der Vorstand ist Anlaufstelle für viele Clubangelegenheiten. Er soll auch den Kontakt zu den Mitgliedern halten. Der Vorstand regelt den Bücherverleih.

Das war dann schon alles, was mir derzeit einfällt. Ich habe in den letzten Monaten im Schnitt wöchentlich 3-5 Stunden für den Club aufgebracht. Wer jetzt noch Fragen zum Vorstand hat, kann sich ja bei mir melden.

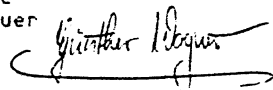
Bitte nehmt alle das Problem Vorstand ernst - wir brauchen einen Vorstand für einen funktionsfähigen CLUB 80 - und an dem ist uns doch allen gelegen. Sollte unter uns ein Mitglied sein, das bereit wäre, den Vorstand zu übernehmen, aber aus beruflichen oder privaten Gründen am Clubtreffen nicht teilnehmen kann, so muß er mir dies unbedingt mitteilen. Ich hoffe ferner, daß sich die Teilnehmer am Clubtreffen auch Ihre Gedanken darüber machen, wie das Treffen ablaufen soll und was besprochen gehört.

Wie bereits im letzten Info angedeutet wurde, haben wir einen Teil des Überschusses von 1985 für den Ankauf von insgesamt 22 Büchern verwendet. Hierbei konnten wir 19 Bücher zum m.E. sehr günstigen Komplettpreis von 171,- DM (inklusive aller angefallenen Unkosten) kaufen. Die komplette Bücherliste des CLUB 80 ist in diesem Info abgedruckt.

Ich hoffe jetzt nur noch auf ein gelungenes Clubtreffen.

Viele Grüße

Euer



0 1

P.S.: Zum Clubtreffen findet Ihr weiter hinten noch einen Beitrag.

Clubtreffen in Holzhausen a. d. Heide

Um allen Teilnehmern am Clubtreffen die Anfahrt möglichst zu erleichtern, veröffentlichen wir hier noch einmal eine Straßenkarte der weiteren Umgebung von Holzhausen a. d. Heide. Die günstigsten Anfahrtswege sind

- von Süden kommend:

Richtung Wiesbaden, dort auf die A66 in Richtung Rüdeshcim. Die Autobahn geht direkt in die B260 über, welche direkt durch Holzhausen führt!

- von Norden kommend:

entweder linksrheinisch in Richtung Koblenz, dort Richtung Bad Ems/Nassau (B417). In Nassau auf die B260 in Richtung Wiesbaden oder rechtsrheinisch Autobahn A3 bis Limburg a. d. Lahn. Dort auf die B417 in Richtung Diez a. d. L./Nassau. In Nassau auf die B260 in Richtung Wiesbaden.

Die B260 führt, wie schon erwähnt, direkt durch Holzhausen hindurch. In Holzhausen (übrigens der Geburtsort von Nikolaus August Otto) ist der Weg zum Hotel "Taurusblick" sehr gut ausgeschildert. Für die von Süden kommenden Teilnehmer der Hinweis: Die Hinweisschilder (mehrere) "Taurusblick" befinden sich auf der rechten Straßenseite! Für alle aus nördlicher Richtung ankommenden Teilnehmer gilt demnach: Die Hinweisschilder sind links zu suchen und zwar erst nach dem Passieren des Geburtshauses (deutlich sichtbar) von N.A. Otto!

Ich glaube mit dieser Beschreibung dürfte jeder Teilnehmer den Ort des Treffens mit Leichtigkeit finden!

Nun zu den Teilnehmern, die ev. mit der Bahn anreisen wollen. Da Holzhausen keine Bahnanschlus hat, ist es ohne eigenes KFZ nur schwer zu erreichen. Um trotzdem auch Mitgliedern ohne eigenen oder geliehenen fahrbaren Untersatz die Teilnahme zu ermöglichen, erbitte ich mit, sie vom Bahnhof Mainz oder Wiesbaden abzuholen und nach Beendigung des Treffens dort auch wieder abzuliefern. Damit ich nicht dauernd nur auf Achse bin, wäre mir natürlich ein "Sammeltransport" am liebsten. Ich bitte also alle Bahnfahrer, sich sobald wie möglich, jedoch bis spätestens 31.03. telefonisch bei mir zu melden, damit wir die Ankunftszeiten möglichst aufeinander abstimmen können!

Noch ein letzter Hinweis:

Mitglieder, die schon abgesagt oder ihren Anmeldezettel nicht abgeschickt haben und sich kurzfristig doch für die Teilnahme entscheiden, sollten mich telefonisch davon in Kenntnis setzen. Sollte nämlich der "Taurusblick" voll belegt sein, kann ich ohne vorherige telefonische Anmeldung keine Garantie dafür übernehmen, daß ihr in einem anderen Gasthof noch ein Zimmer bekommt!

Doch damit genug der guten Ratschläge. Ich wünsche allen Teilnehmern am Treffen eine gute und vor allem unfallfreie Anfahrt, Euer

Karstaut Obermann

Hotelanschrift: "Taurusblick", 5429 Holzhausen ☎ 06772/8343

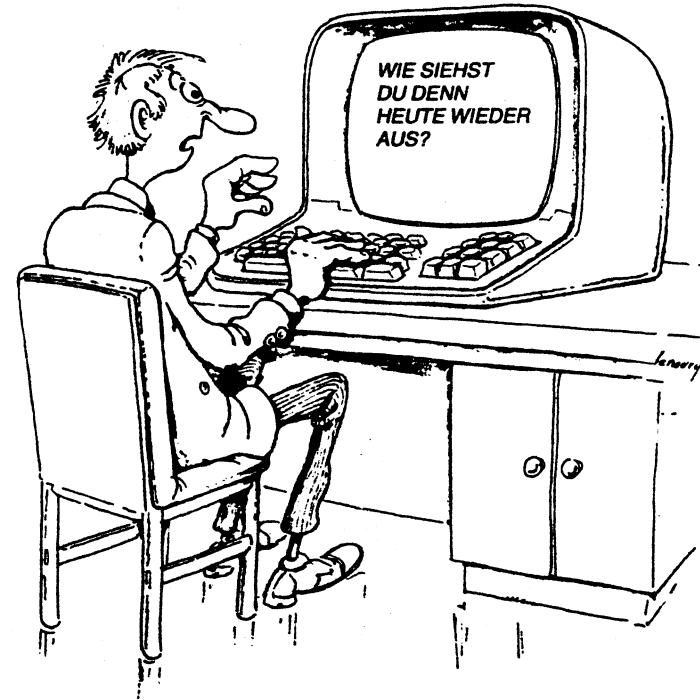
HEFT
12
März
1986

02

Nachtrag zu "INPUT"

Ihr erinnert Euch sicherlich noch an meinen wenig hilfreichen Hinweis, den Bildschirmditor zu laden, damit beim INPUT die nächste Zeile nicht gelöscht wird. Nun habe ich den Weg des geringsten Widerstandes gesucht und wiederum in "Basic Faster and Better" auch gefunden. Das nachstehende Programm löst das Problem der verschwindenden Zeile selbst, indem es sie einfach über eine Assembler-Routine in einen String rettet und nach der Eingabe (und dem Löschen) wieder zurückschreibt. Nicht genial, aber einfach.

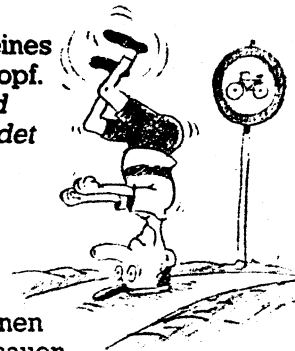
```
10 CLEAR 5000 : DEFINT A-Z
20 DIM US(4) 'Assembler-Routine 1: Eingabe
30 DIM UV(6) 'Assembler-Routine 2: Zeile retten
40 US(0)=1030 : US(1)=8448 : US(3)=-9779 : US(4)=-14075
50 UV(0)=8448 : UV(2)=4352 : UV(4)=16385 : UV(5)=-4864 :
   UV(6)=-13904
60 ' irgendein Hauptprogramm
70 ' ...
80 ?$I1; ' I1 = Position, an der die Eingabe erfolgt, könnte
90 ' auch über Abfrage von (4020H) und (4021H)
100 ' realisiert werden
110 GOSUB 1000
120 ' Reaktionen auf die Eingabe, entweder A$ oder I
130 ' auswerten
140 ' ...
140 END
1000 ' Unterprogramm "Eingabe"
1010 ' -das Programm benötigt die Cursorposition in I1
1020 ' -es gibt die Eingabe in A$ und den ASCII-Wert des
1030 ' ersten Buchstaben in I aus
1040 I2=(INT(I1/64)+1)*64 '5-Position am Anfang der nächsten
1050 ' Zeile
1060 ' Retten der nächsten Zeile:
1070 F$=STRING$(65,32) : UV(1)=15360+I2
1080 UV(3)=CVI(CHR$(PEEK(VARPTR(F$)+1))+CHR$(PEEK(
   VARPTR(F$)+2)))
1090 I=0 : DEFUSR=VARPTR(UV(0)) : I=USR(0)
1100 ' Eingabe:
1110 A$=STRING$(5,32)
1120 US(2)=CVI(CHR$(PEEK(VARPTR(A$)+1))+CHR$(PEEK(
   VARPTR(A$)+2)))
1130 I=0 : DEFUSR=VARPTR(US(0)) : I=USR(0)
1140 ' Gelöschte Zeile wieder zurück:
1150 UV(1)=CVI(CHR$(PEEK(VARPTR(F$)+1))+CHR$(PEEK(
   VARPTR(F$)+2)))
1160 UV(3)=15360+I2 : I=0 : DEFUSR=VARPTR(UV(0)) : I=USR(0)
1170 ' Auswertung
1180 IF INSTR(A$,CHR$(13))>0 THEN A$=LEFT$(A$,
   INSTR(A$,CHR$(13))-1) 'String ohne CR
1190 I=ASC(A$+CHR$(128))
1200 IF PEEK(14400)=4 THEN END ' Break-Taste gedrückt
1210 IF I<32 OR I>127 THEN I=0 ' Code zu klein oder zu groß
1220 RETURN
```



Französisch: Il a un petit vélo dans la tête.

Er hat ein kleines
Fahrrad im Kopf.
Er ist als Kind
zu heiß gebadet
worden.

Un chien
regarde bien
un évêque.
Ein Hund
kann doch einen
Bischof anschauen.
Ansehen kostet nichts.



Griechisch: Oponi laloun polli kori
argi na ximerosi.
Wo viele Hähne krähen, verspätet
sich das Tagwerden.
Viele Köche verderben den Brei.

Eipo (Papua-Neuguinea): Asok ki-
kye binil.
Mir ist nach Kratzen an der Stirn.
Ich habe mich wohl getäuscht.

Günther hat mit seinem Wettbewerb-Vorschlag im letzten INFO sicher alle müden Hacker anregen wollen, auch mal etwas zum INFO beizutragen; und seien es "nur 10 Zeilen"!

Im ersten Augenblick war ich ob dieser neuen Philosophie etwas verwundert und fragte mich, worin dabei die "Leistung" bestehen könnte! Denn seit ich die notdürftige Beleuchtung dieser UN-Welt wahrnahm (um nicht wiederholt zu behaupten, daß ich "das Licht der Welt erblickte" - denn wo ist es denn noch außerhalb des 'CLUBBO'??...) habe ich den Leitspruch beherzigt:

In der Kürze liegt die Würze!

Also war ich der Meinung, dieser Wettbewerb sei kein Wettbewerb um eine edle Sache...

Hier könnt Ihr nun die "Entwicklung eines unterentwickelten Entwicklungsfähigen" miterleben:

Bestärkt durch Harold B.Fink, einen klugen Mann - hat er doch das Schachspiel "SFINKS" erfunden, eines der besten! -, der im August 1984 in der **80 MICRO** (von manchen fälschlicherweise "MICRO 80" genannt) eine ganze Serie kluger Einzeiler unter dem Titel "ONE-LINERS" publizierte und dies als besonders vornehme (und elegante?) Programmierweise deklarierte - bestärkt also durch solches Vorbild, machte ich mich daran, ebenfalls einen Einzeiler zu verfassen, der das gleiche leistet wie ein X-Zeiler (z.B. X=10), egal wofür.

Was sollte als Beispiel herhalten? Wieder einmal das leidige Umwandlungsthema "DEZ/HEX"! Hatte ich doch gerade in INFO 3, vorletzte Seite (Seiten-Nummern gab es da noch nicht; welches Kind zählt schon die Tage ersten Spiels??) die Schöpfung von Kamerad Zwickel (Hallo Walter, ich grüße Dich und Deine Berge im schönen Österreich, besonders Dein Bergheim!) gelesen, der über 75 Stationen von DEZ nach HEX fährt, wobei die Entfernung von einer zur nächsten zugegeben auch zu Fuß zu überbrücken ist, so kurz sind deren Abstände (=Zeilen)...

"Ha!" - so frohlockte ich: "Was der Walter in 75 Schritten macht, das machst Du doch spielend in einem Durchgang!!"

Gesagt - getan. Da habt Ihr ihn, den EINZEILER! Seht selbst: er tut's!

Gewonnen? * Langsam! * Urteilen heißt Vergleichen. Ich hab da meinen eigenen Maßstab. Im Nachbarclub (wer kennt den BCC nicht?) hatte ich schon vor Jahresfrist ein Programm dem öffentlichen Lächeln preisgegeben, das ich bereits vor Urzeiten für's gleiche "Problem" geschaffen: eine 'DEZ/HEX/DEZ-Umwandlung'. Hier sei es mit seinen 10 Zeilen nochmals wiedergegeben. (Anm.: Auf die DEZ/HEX-Umwandlung selbst entfallen davon nur 5 Zeilen.)

Nun kommt's: Fahrt mal mit beiden Programmen die gleiche Strecke ab! Sagen wir irgendwas, z.B. von 47111(dez) nach BB07(hex)! Toll, was? Der "Normalzeiler" ist ein TEE, der "Einzeiler" ein Eummelzug dagegen!

Zur erstenmal hatte der Computer für mich etwas Menschliches an sich:

Je mehr man ihn quält, desto mehr er sich wehrt!

So hat er mich leise Bescheidung gelehrt...

Also steckt im "ZEHNZEILER" wohl doch so etwas wie ein "weises Optimum"? Dann wohlauf: folgt Günthers Aufruf!

* * Ehre wem Ehre gebührt!! * *

Der Naturphilosoph fühlt sich auch bestätigt:

Mutationen = winzigste Veränderungen können unübersichtliche Folgen haben! Nimm einen EINZELLER (der bekanntlich ewig lebt) das kleine Häkchen am ersten L, auf das er so stolz ist: und er lebt als EINZEILER kaum eine Stunde noch...

Und nun werden ihm obendrein auch noch die ZEPHE gereicht!!

* O jeeeeee - das tut ja weh! * * GRÜß Gott! * HaGott *

(Schlußfrage: Wann spielen wir "ALLE NEUNE"?)

Zu der Wettbewerbsausschreibung möchte ich ergänzenderweise noch folgendes anmerken:

Es muß kein ZEHNZEILER sein!

Eine geringere Zeilenzahl wird auf jeden Fall auch akzeptiert. Nur wollte ich durch die Beschreibung ZEHNZEILER sicherstellen, daß die Programme nicht mehr als 10 Zeilen haben.

Die Begründung für die Programmlänge liegt darin, daß die Programme leicht überschaubar sein sollen, und von jedermann leicht eingetippt werden können. Auch wollte ich durch die Programmkürze den Programmieraufwand für Euch gering halten und für uns, die das Programm beurteilen sollen, eine Möglichkeit schaffen, genauer über den Wert eines Programmes zu urteilen. Deshalb **MAXIMUM:**

ZEHNZEILER

Ich möchte Euch nun nochmals bitten, es diesem mutigen Vorreiter, dessen Programm hier folgt, nachzutun und eine Programmschöpfung -es können auch mehrere sein- aus eigener Feder hier in's Rennen zu schicken.

Eure CLUB 88 Redaktion



```

1 REM * * Beispiel eines 'E I N Z E I L E R s': DEZ/HEX-Umwandlung * *
2 REM * * ===== * *
3 REM   aufgestellt von KaJot Mühlenbein, Weinheim, Februar 1986
4 REM
5 REM
10 CLS:INPUT"Gib eine ganze Zahl unter 65336 ein :   ";D:FORI=3TO0STEP-1:S
(I)=INT(D/16^I):S$(I)=CHR$(48+S(I)-(S(I)>9)*7):D=D-S(I)*16^I:NEXT:PRINT"Die
entspr. Sedezimalzahl lautet : "S$(3);S$(2);S$(1);S$(0):END

```

* * * Die gleiche Aufgabe in einem 'ZEHNZEILER' : * * *

(der Teil 'DEZ/HEX-Umwandlung' beansprucht 5 Zeilen und ist >3mal schneller!)

```

1 CLS: CLEAR70: PRINT"   UMWANDLUNG VON HEX- IN DEZ-SYSTEM UND UMGEKEHRT
   < (C) Muehlenbein, Weinheim 1983 >
   "; STRING$(47, "*"): DEFINT A, I-N: DEFSTRH, X
2 INPUT"

   WIRD   <1>  HEX/DEZ-UMWANDLUNG
   ODER   <2>  DEZ/HEX-UMWANDLUNG
   GEWUNSCHT "; K: ONK GOTO 3, 6
3 CMD"LC N": INPUT"

HEXADEZIMALZAHL =   "; HEX
4 FOR I=1 TO LEN(HEX): A=ASC(MID$(HEX, I, 1))-48: IFA>16 THEN
A=A-7
5 DEZ=DEZ*16+A: NEXT: PRINT: PRINT CHR$(214)HEX" (HEX) = "DEZ" (DEZ)": CMD"1c"
: END
6 INPUT"

DEZIMALZAHL =   "; DEZ: D(I)=DEZ
7 D(I+1)=INT(D(I)/16): B(I)=D(I)-D(I+1)*16: IFD(I+1)>0 THEN I=I+1: GOTO 7
8 FOR J=0 TO I: IFB(J)<10 THEN X(J)=RIGHT$(STR$(B(J)), LEN(STR$(B(J)))-1): ELSE X(J)
=CHR$(E(J)+55)
9 NEXT: FOR J=I TO 0 STEP -1: HEX=HEX+X(J): NEXT: N=4-LEN(HEX): NULL$=STRING$(N, "0")
10 PRINT"

"CHR$(212)DEZ" (DEZ) = "NULL$+HEX" (HEX)": END

```



„Weißt du, welchen Code ich gerade geknackt habe?“

☹☹ Der menschliche Ver-
stand ist in der Praxis nicht
verlässlich, am wenigsten in
größter Not. ☹☹

☹☹ Früher litten wir an
Verbrechen, heute an Ge-
setzen. ☹☹

☹☹ Autovertreter ver-
kaufen Autos, Versiche-
rungsvertreter Versicher-
rungen. Und Volksvertre-
ter? ☹☹

☹☹ Wenn die Affen es
dahin bringen könnten,
Langeweile zu haben, so
könnten sie Menschen wer-
den. ☹☹

Verbessert: IO.port#(,xxx,yy,...

Im Info Nr. 11 stellt Hartmut den (nicht mehr ganz) neuen Library-Befehl OUT vor, den ich laut Listing offenbar schon vor bald zwei Jahren verbrochen habe. Irgendwie gefiel mir die Routine beim Durchlesen nicht mehr; sie erschien mir zu brav, amateurhaft. Auf der nächsten Seite ist deshalb eine heute entstandene Alternative vorgestellt, deren reiner OUT-Teil rd. 20% kürzer und schneller ist.

Zusätzlich gibt es jetzt die Möglichkeit, einen Input von einer Port einzuholen. Dazu wird hinter dem Befehlswort einfach nur die Portadresse ohne weitere Parameter (Outputs) angegeben. Die Routine "weiß" dann, daß sie den Port nur auslesen soll. Durch diese Erweiterung wurde das Programm nun insgesamt 9 Bytes länger als die alte Version. Wegen der Input-Möglichkeit heißt der Library-Befehl nun IO (Input/Output) statt nur OUT.

Im OUT-Teil unterscheiden sich das alte und das neue Programm nicht. Erklären will ich deshalb nur die Verbesserungen, und seien sie auch nur ästhetischer Art: Der Befehl RLCA ist hier vorzuziehen, denn aus der Routine getchr kann durchaus auch mit gesetztem Carry-Flag zurückgekehrt werden. Das würde mit RLA den Akku verändern. Als Zwischenspeicher für das höherwertige Nibble der Hexzahl dient hier B, weil halt C den Fort hält. Sieht besser aus. Die meisten Unterschiede gibt es bei der Umwandlung der ASCII-Zahl der Eingabe in eine Binärzahl zur Ausgabe. Hier wurden auch die meisten Bytes gespart. An der Stelle error war in der alten Version ein Bug. Es genügt nicht, mit NZ-Bedingung über RET zurückzukehren. Es muß über 4409h geschehen.

Der Dump zeigt den letzten Sektor von SYS3/SYS. Die OUT-Routine (unterstrichen) überschreibt hier die ehemalige JKL-Routine, die beim Genie IIIs überflüssig geworden ist. Wer einen anderen Computer hat, kann OUT in jedem beliebigen SYS-File unterbringen, wo Platz dafür übrig ist, z. B. in SYS28, wie Hartmut vorschlägt oder in SYS29 wie in H-DOS. Die jeweilige Ladeadresse ist dann natürlich von verfügbarem Platz abhängig. Sie wird mit ORG variiert. Dabei ändern sich auch die Sprungadressen der CALLs. Man sollte daher vorsichtshalber die Routine zu diesem Zweck mit EDTASM o. ä. eintippen und neu assemblieren.

Eine interessante Variante ist es, die Routine als CMD-File zu benutzen, nicht als SYS-File. Um dennoch den Arbeitsspeicher sauber zu halten, wird dazu die ORG-Adresse auf den Beginn des Sektorpuffers 4200h gelegt. Für dieses kleine Programm würde sogar der DOS-Eingabepuffer ab 4318h genügen (Vorsicht, der Aufrufbefehl selbst braucht davor noch genügend Platz!). Das ist auch eine gute Methode, das Programm zunächst auf einwandfreie Funktion zu überprüfen, bevor man sich das DOS damit zerschreibt.

Arnulf Sopp

```
000400: 81C2 8744 E57D C605 01EA F850 4F06 08CD 2 D u EA PoFH
000410: 59E1 7EFE 2028 0A3E 2FCD 2300 0403 CD59 Y08 (J)/ 36FC Y
000420: 5121 6751 CD67 44CD 4900 CBAF FE45 2808 31g0 gD 18 E(H
000430: FE4E 2804 FE4A 20EF F5CD 3300 3E0D CD33 N(D J 3EM J
000440: 00F1 E1FE 45C8 FE4E 2811 1180 443E 8012 8 E N(DD D) R
000450: CD17 4ED0 E579 CD36 49E1 C07D E6E0 C420 MN y 6I ü
000460: 6FD2 BC50 0C79 FE00 DAB7 50AF C97E FE20 o PLY 8 P B
000470: 23C4 3300 10F7 C900 2020 20C2 0944 6174 # 36F 8 IDat
000480: 6569 206D 7073 6768 656E 203F 2028 4AD5 ei löschen ? (J/
000490: 4EDF 4529 2003 CD43 514F CD05 4020 11E5 N(E) C DG L D
0004A0: 782A 2040 CD68 403E 0AC3 2300 CD05 4020 M 8 H5D 3E L
0004B0: CD43 51E8 7916 FECD B0E1 0707 0707 470E C y 668666
0004C0: B0E1 B0C9 7E27 D630 3B0C FE0A FE26 07FE D 8 0L J 8
0004D0: 0A76 4008 670E 2E2F C1C1 CD09 4000 0000 0000 0000
0004E0: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0004F0: 0000 0000 0202 0042 0000 0000 0000 0000 0000 0000
```

```
00001 :Tap in SYS28:B für den Befehl IO.port#(,xxx,yy,....
00002 ORG 5182h ;hier ehemalige JKL-Rout.
00003
00004 :den zu adressierenden Port ermitteln
518C CDA351 00005 CALL getval ;eine Hexzahl einlesen
518E 4F 00006 LD C,A ;das ist der Port
518A CDD54C 00007 CALL 4cd5h ;Trennz. und CR erkennen
5189 2011 00008 JB NZ,out ;falls noch etwas folgt
00009
00010 :nein, nur Portadresse angegeben, Input ist gefragt
518E ED7E 00011 IN A,(C) ;Port auslesen
518D 2A2040 00012 LD HL,(4020h) ;Cursorposition
5190 CD6840 00013 CALL 4068h ;Input anzeigen
5193 3E0A 00014 LD A,0ah ;Zeilenvorschub
5195 C33300 00015 JP 0033h ;ausgeben und raus
00016
00017 :die jeweiligen Outputs ermitteln und ausgeben
519E CDD54C 00018 loop CALL 4cd5h ;Trennz. und CR erkennen
519F C8 00019 RET Z ;falls alles abgearbeitet
519C CDA351 00020 out CALL getval ;nächste Hexzahl(en)
519F ED79 00021 OUT (C),A ;Output auf den Fort
51A1 18FE 00022 JB loop ;immer so weiter
00023
00024 :eine Hexzahl einlesen und auswerten
51A3 CDB051 00025 getval CALL getchr ;eine Hexziffer einlesen
51A6 07 00026 RLCA --;ins obere Nibble
51A7 07 00027 RLCA ;schieben, also zuerst
51A8 07 00028 RLCA ;die höhere Stelle der
51A9 07 00029 RLCA ;Zahl nehmen
51AA 47 00030 LD B,A ;zwischenspeichern
51AB CDB051 00031 CALL getchr ;untere Dezimalstelle
51AE B0 00032 OR B ;mit d. oberen vereinigen
51AF C9 00033 RET ;zur. mit kompletter Zahl
00034
00035 :eine Hexziffer einlesen und binär angleichen
51B0 7E 00036 getchr LD A,(HL) ;nächst. Zeichen einlesen
51B1 23 00037 INC HL ;weiter im Befehlsstring
51B2 D630 00038 SUB '0' ;Binärangleichung
51B4 3B0C 00039 JB C,error ;falls Zeichen < '0'
51B6 FE0A 00040 CF 0ah ;Zeichen > '9'?
51B8 DB 00041 RET C ;fertig, falls Deziffer
51B9 D607 00042 SUB 07h ;Ziffern A - F angleichen
51BB FE0A 00043 CF 0ah ;Hex A oder größer?
51BD 3B03 00044 JB C,error ;falls kleiner
51BF CB67 00045 BIT 4,A ;Wert > F?
51C1 CE 00046 RET Z ;fertig, falls höchst. F
00047
00048 ;Fehlercode "falsche Parameter" in den Akku, NZ-Bedingung
51C2 3E2F 00049 error LD A,2fh ;Fehlercode
51C4 C1 00050 POP BC ;Stack korrigieren (CALL)
51C5 C1 00051 POP EC ;(2. CALL-Ebene)
51C6 C30944 00052 JP 4409h ;Fehleranzeige und zurück
00053
0000 00054 END ;na Endlich!
```

00000 Fehler

```
error 51C2 getchr 51B0 getval 51A3 loop 5198 out 519C
```

Arnulf Sopp

JKL - aber druckerschonend!

Die Idee zu diesem Zap in SYS26/SYS (G-DOS 2.4) bzw. in SYS3/SYS3 (G-DOS 2.1b und 2.2; bei NEWDOS je nach Computermodell) wuchs auf keinem geringeren Mist als dem der weiland Fa. TCS Computer GmbH. Für G-DOS 2.4 hat man nämlich wegen der unterschiedlichen Bildschirmformate der Genie-Modelle III und IIIa die Routine zur Screen-Hardcopy völlig neu geschrieben und, ihres beträchtlichen Umfangs wegen, nach SYS26 ausgelagert. Daraus folgt übrigens, daß die alte Routine in SYS3 für diese G-DOS-Version überflüssig geworden ist und lustig überschrieben werden darf. Aufmerksame Leser erinnern sich, daß dort nun mein IO-Programm residiert.

In der neuen JKL-Routine dachte man einen Schritt weiter als damals und ersetzte nicht druckbare Codes nicht etwa durch den wenig informativen Punkt, sondern durch ein Leerzeichen. Das ist keinen Deut informativer, schon aber den Druckkopf. Da muß der Programmierer jedoch mitten im Gedankengang eingeschlafen sein, denn das eben Gelobte gilt nur für Codes ab 7Fh. Die Codes zwischen 00 und 1Fh kriegen kurzerhand das Bit 6 gesetzt und sehen nun aus wie gewöhnliche Buchstaben, was sie nun einmal nicht sind. Der erste der drei Sektor-Dumps zeigt, wie das aussieht. Breulich, wenn ihr mich fragt.

Die Abhilfe ist simpel: Wo das Bit 6 mit dem Befehl OR 40h gesetzt wird, kann man stattdessen mit dem Befehl LD A,20h den Akku mit einem Blank laden. Beide Befehle haben zwei Bytes, so daß akrobatische Verrückungen entfallen. Der zweite Dump zeigt den dateirelativen Sektor 01 von SYS26 (G-DOS 2.4) in der geänderten Form, der nächste den Sektor 04 von SYS3 (sonstige). Die Änderung ist jeweils unterstrichen. Diese beiden Dumps sind bereits mit der neuen Routine erstellt.

Wer zu diesem Zweck SYS3 ändern müßte, aber in seinen Punkt verliebt ist, möge alles beim alten lassen. Sollte jemand mit einem entsprechenden SYS26 den Punkt bevorzugen, so wird in dem zu ändernden Befehl der Akku einfach mit 2Eh statt 20h geladen. Die beiden Bytes lauten dann 3E-2E statt 3E-20. Das ist bereits alles.

Es gibt eine dritte Möglichkeit. Wer als User von G-DOS 2.1b oder 2.2 unterhalb 20h gerne anstelle der Buchstabenentsprechungen Punkte sehen möchte, ändert ADD A,40h in LD A,2Eh (3E-2E statt C6-40). Das ist die obere der beiden unterstrichenen Modifikationen.

Arnulf Sopp



```

000100: 673A FD37 0102 004E 6F19 E53A F637 6F3A g: 7 No : 7o:
000110: FB37 E603 FE03 2002 CB05 3AF1 3732 474E 7 : : 726N
000120: CD92 4CEB E1EB 19EB AF32 DD4D 32D9 4D01 L 2 M2 M
000130: 0000 CDD1 0528 070B 78B1 2B4C 18F4 EB2B ( x (L +
000140: DF2B 08CD D54D FE20 28F5 23EB D5E5 1E40 ( M ( # 5
000150: 1600 1943 2BCD D54D FE20 2003 10F6 04E1 C+ M
000160: 2277 4ED1 CDD5 4DFE 2030 023E 20CD 844E "wN M O > N
000170: 233A 4038 E604 2012 10EA 3A47 4E4F 2100 #:56 :GNO!
000180: 0009 CD82 4EDF 38C4 AFC9 3E0D D55F 3A70 N B > :p
000190: 43BB 3002 1E20 7BCD 3B00 D1C9 5049 4F20 C 0 ä ; P10
0001A0: 034E 4F50 520D 3A07 43E6 07FE 043E 2ADB NOFR : C > *
0001B0: 7EFE 0D2B 0BFE 5028 33FE 4E2B 1AC3 604D B ( F(3 N( 'M
0001C0: 2194 4ECD 6744 2199 4E3A BD05 FED4 2002 ! N gD! N:
0001D0: 2323 CD67 44AF C921 E837 3E32 32BB 0522 ## gD ! 7>22 "
0001E0: BC05 3E3A 32D1 0522 D205 1BE9 3E07 D3D6 >:2 " >
0001F0: D3D7 3E0F D3D6 3ECF D3D7 3EFE D3D7 3E01 > > > >
    
```

```

000400: E544 6973 6E65 7474 65EE 4174 747D 6962 Dislette Attrib
000410: 01EA 00E1 75F4 7363 686C 6563 6874 65F2 A EQU schlechte
000420: 4B65 696E E518 1S18 6645 686C 65F2 6245 Kein XXXfehle be
000430: E975 6E7A 756C 6165 7373 6967 E508 F3FE unzulässig H
000440: 86C2 004D 5E23 5623 CDD5 4C38 19E3 0602 SM*#V# LBY FB
000450: D521 5651 5E23 56D7 23E3 4FDF 79E3 2804 IV0*#V # D v (D
000460: 10F2 3E74 E1E1 C209 44EF 4944 D2E0 4FAE P #4 ID ID=PC
000470: 0000 0000 0000 0000 0000 0000 0000 0000 EEEEEEEEEEEEEEEEE
000480: 0000 0000 0000 0000 0000 0000 0000 0000 EEEEEEEEEEEEEEEEE
000490: 0000 0000 0000 0000 0000 0000 0000 0000 EEEEEEEEEEEEEEEEE
0004A0: 0000 0000 0000 0000 0000 0000 0000 0000 EEEEEEEEEEEEEEEEE
0004B0: 0000 0000 0000 0000 0000 0000 0000 0000 EEEEEEEEEEEEEEEEE
0004C0: 0000 0000 0000 0000 0000 0000 0000 0000 EEEEEEEEEEEEEEEEE
0004D0: 0000 0000 0000 0000 0000 0000 0000 0000 EEEEEEEEEEEEEEEEE
0004E0: 0000 0000 0000 0000 0000 0000 0000 0000 EEEEEEEEEEEEEEEEE
0004F0: 0000 0000 0000 0000 0000 0000 0000 0000 EEEEEEEEEEEEEEEEE
    
```

```

000400: 8132 8744 E57D C605 01F2 FB50 6F06 08CD 2 D ü Po
000410: 5951 7EFE 202B 0A3E 2FCD 3300 0603 CD59 Y0B ( > / 3 Y
000420: 5121 6751 CD67 44CD 4900 CBAF FE45 2808 0!g0 gD I E(
000430: FE4E 2804 FE4A 20EF F5CD 3300 3E0D CD33 N( J 3 > 3
000440: 00F1 E1FE 45CB FE4E 2811 1180 443E 8012 E N( D>
000450: CD17 4E00 E579 CD36 49E1 C07D E6E0 C620 N v 6I ü
000460: 6FD2 BC50 0C79 FE00 DAB7 50AF C97E FE20 o F y F B
000470: 23C4 3300 10F7 C900 2020 20C4 4461 7465 # 3 Date
000480: 6920 6C7C 7363 6865 6E20 203F 202B 4A2F i löschen ? (J/
000490: 4E2F 4529 2003 2100 3C3A 4038 E604 2805 N/E) ! <:5B (
0004A0: 2100 4018 F47D E63F 3E0D CC3B 007C E63F ! s ü ?> : 8 ?
0004B0: C97E FE20 2330 023E 2047 3A70 43BB 7830 B #0 > G:pc x0
0004C0: 023E 20CD 3B00 12D1 0D2B CB3A 803B E601 > : ( : B
0004D0: 4720 0D21 FF3B D7FE 003B FB7C FE40 30B6 E ! : B 6 50
0004E0: 3E07 CD3B 00CD 4900 E65F FE41 28AB FE50 > : I - A( F
0004F0: 2804 FE4E 20EA 4F3E DBEF 0000 0202 004D ( N D) M
    
```


In unseren Kirchenblättchen sind schon verschiedentlich Artikel mit Vorschlägen erschienen. wie man zusätzliche DOS-Befehle in die Library am Ende von SYS1/SYS implementieren kann. Wenn sie aber schon voll ist? Da gibt es DOSes, die für diesen Fall eine zweite Library in einem anderen SYS-File installieren. Wenn aber, wie beim Genie IIIs, nur noch eins frei ist, mit dem man Größeres vorhat?

Der letztere Weg ist gar nicht einmal so verkehrt. Da hierbei aber SYS1 überschrieben wird, müssen wesentliche Teile davon in der neuen SYS-Datei wiederholt werden. Das kostet Platz. Deshalb wurde hier ein Kompromiß gewählt: Ein neuer Library-Befehl leitet alle ganz-ganz neuen Library-Befehle ein. Er verzweigt in eine andere Systemdatei, wo sein Argument, nämlich der eigentliche Befehl, ausgewertet wird. Das hört sich umständlich an, ist es aber nicht:

Das Programm ist sehr einfach und kurz (s. Listing). Die Bedienung ist es nicht minder: Nehmen wir an, ein neuer Befehl soll IO lauten. Man gibt jetzt nicht IO,... ein, sondern SYS,IO,... Diese kleine Unbequemlichkeit ist wesentlich weniger unangenehm als der Verlust von viel Speicher- oder Diskettenplatz in einem teuren SYS-File. Aber zum Programm selbst:

Es residiert in SYS4/SYS, wo noch ausreichend Platz vorhanden war. Beim Einsprung wird zunächst geprüft, ob der Requestcode im Akku 66h lautet. Ist das nicht der Fall, muß es sich um eine andere DOS-Anforderung für SYS4 handeln. Also verzweigt das Programm in diesem Falle nach 4D00h, wo SYS4 beginnt und seine eigenen Checks durchführt. War es aber 66h, wurde also der Befehl SYS,... eingegeben, dann wird geprüft, was dem Befehl folgt.

Wegen der Platzökonomie sind hier ein paar Vereinbarungen verbindlich: Der Folgebefehl darf nur zwei Zeichen lang sein. Dann läßt er sich nämlich mit RST 18h, einem Vergleich von HL und DE identifizieren. Er darf keinen Restriktionen unterliegen wie beispielsweise einem Verbot unter Mini-DOS. Zwar wäre auch das machbar, aber es gälte dann für alle Zusatzbefehle.

Wie der zusätzliche Befehl erkannt wird, geht aus den Kommentaren des Listings hervor. Es muß erwähnt werden, daß bei einer Verlängerung der Zweit-Library jeweils auch die neue Anzahl der verfügbaren Befehle in Zeile 11 aktualisiert werden muß. Dort wird B als Zähler für die Suchdurchläufe geladen. Die neue Befehlstabelle selbst (cmdtab) besteht außer dem Befehlsword noch aus dem Requestcode für den Akku, der einen Wegweiser in das zutreffende SYS-File zu seiner Bearbeitung darstellt. Das genügt in allen Fällen, denn in keiner Systemdatei werden alle möglichen Codes voll ausgeschöpft. Platzersparnis, wie gesagt.

Die beiden Sektordumps zeigen das neue Ende von SYS4 mit der Serviceroutine für den SYS-Befehl sowie die Library in SYS1, wo der Befehl SYS eingezapt ist. Dort stand früher LIB, das aber überflüssig ist. Erstens kann es durch ? ersetzt werden, zweitens sollte man seine DOS-Befehle auswendig kennen. Und wer sie nicht kennt, ist ohnehin besser beraten, für nähere Informationen das Handbuch zu konsultieren.

Anruf: Soco

512B		00001	SFS	512B	:ab hier Platz in SYS4	
512E	FE84	00002	start	66h	:SYS-Befehl?	
512D	00004F	00003	CF	NC,4200h	:falls nein	
5130	5E	00004	LD	E,(HL)	:1. Buchstabe des Argur.	
5131	23	00005	INC	HL	:2. Buchstabe	
5132	56	00006	LD	D,(HL)	:jetzt SYS-Argument in DE	
5133	23	00007	INC	HL	:auf Delimiter stellen	
5134	0DDE4C	00008	CALL	4cd5h	: " " oder " " oder CR?	
5137	3819	00009	JF	C,error	:Fehler, falls anderes Z.	
513F	E5	00010	FUSH	HL	:Befehlszeiger retten	
513A	0602	00011	LD	B,02h	:bisher 2 neue Befehle	
513C	DE	00012	PUSH	DE	:Argument auf den Stack	
513D	215651	00013	LD	HL,cmdtab	:Anfang Befehlstabelle	
5140	5E	00014	loop	LD	E,(HL)	:1. Zeichen
5141	23	00015	INC	HL		:2. Zeichen
5142	56	00016	LD	D,(HL)		:DE <- Befehl aus Tabelle
5143	D7	00017	RST	10h		:Requestcode in den Akku
5144	23	00018	INC	HL		:Zeiger auf nächsten Bef.
5145	E3	00019	EX	(SP),HL		:Zeig. m. Arg. vertausch.
5146	4F	00020	LD	C,A		:Akku retten
5147	DF	00021	RST	18h		:Befehl gefunden?
5148	79	00022	LD	A,C		:Akku restaurieren
5149	E3	00023	EX	(SP),HL		:zurücktauschen
514A	2804	00024	JR	Z,exit		:falls ja
514C	10F2	00025	DJNZ	loop		:falls noch nicht
514E	3E34	00026	LD	A,34h		:Fehlercode
5150	E1	00027	POP	HL		:Stack bereinigen
5151	E1	00028	POP	HL		: (2 Ebenen)
5152	C20944	00029	error	JF	NZ,4409h	:Fehleranzeige und zurück
5155	E5	00030	RST	28h		:zur Bearbeitungsroutine
5156	49	00031	DM	'ID',3dh		:DOS-Befehl ID in SYS27
5159	49	00032	DM	'ID',0a5h		:DOS-Befehl IO in SYS3
512B		00033	END	start		

000400:	E544	6973	6B65	7474	65EE	4174	7472	6962	Diskette	Attrib
000410:	01EA	0051	75F4	7363	686C	6563	6874	65F2	Qu	schlechte
000420:	4B45	696E	E518	1818	6665	686C	65F2	6265	Kein	fehler be
000430:	E975	6E7A	756C	6165	7373	6967	E508	F3FE	unzulaessig	
000440:	<u>B6C2</u>	<u>004D</u>	<u>5E23</u>	<u>5423</u>	<u>0DDE</u>	<u>4D3B</u>	<u>19E5</u>	<u>6602</u>	M^#V#	LB
000450:	<u>D521</u>	<u>5651</u>	<u>5E23</u>	<u>56D7</u>	<u>23E3</u>	<u>4FDF</u>	<u>79E3</u>	<u>2804</u>	!VD^#V # 0 y (
000460:	<u>10F2</u>	<u>3E34</u>	<u>E1E1</u>	<u>C209</u>	<u>44EF</u>	<u>4944</u>	<u>3D50</u>	<u>4FA5</u>	>4	D ID=FO
000470:	0000	0000	0000	0000	0000	0000	0000	0000		
000480:	0000	0000	0000	0000	0000	0000	0000	0000		
000490:	0000	0000	0000	0000	0000	0000	0000	0000		
0004A0:	0000	0000	0000	0000	0000	0000	0000	0000		
0004B0:	0000	0000	0000	0000	0000	0000	0000	0000		
0004C0:	0000	0000	0000	0000	0000	0000	0000	0000		
0004D0:	0000	0000	0000	0000	0000	0000	0000	0000		
0004E0:	0000	0000	0000	0000	0000	0000	0000	0000		
0004F0:	0000	0000	0000	0000	0000	0000	0000	0202	2E51	+Q

000300:	4D82	E900	4980	2A00	494E	464F	0102	0050	M	I * INFO	P
000310:	81FF	004A	4E4C	B07C	104B	494C	4CB0	4590		JKL & KILL	E
000320:	4C43	88EE	004C	4681	FE00	<u>5359</u>	<u>5380</u>	<u>8600</u>	LD	LF	<u>SYS</u>
000330:	4C49	5354	85F0	864C	4F41	4480	A450	4D3E	LIST	LOAD	PM>
000340:	82EB	B04E	51E4	B04E	4446	C02B	0050	4155	N	NDF (FAU	
000350:	5745	95EB	01E0	44E3	E900	5049	4FB0	9C00	SE	FD	P10
000360:	504F	5254	81FF	0050	5249	4E54	B6F0	B850	PORT	PRINT	P
000370:	524F	54E6	E900	505E	5247	4589	E900	5280	ROT	PURGE	R
000380:	2000	5381	E900	5349	4FB0	B000	53E4	4D54	* S	SID	STMT
000390:	89EB	0055	485D	82E6	005A	28E4	E500	SAB3		UFR	V+ Z
0003A0:	FE10	544E	4954	8AE9	005A	4CB2	F888	7084	ZEIT	ZL	0
0003B0:	F000	34C4	E1E1	0038	3082	F800	2187	EB8A	#4	90	!
0003C0:	2327	B788	0021	83E8	0040	81F0	007B	B6E7	&	&	:
0003D0:	002F	8587	003E	0049	003F	82E3	0000	0021	/	> H ?	!
0003E0:	564F	8E47	0008	7E09	7E23	200E	DDE7	5110	XC	\$	B # 0
0003F0:	FE07	207F	F7C4	E5E1	0000	E5E1	28E4	0DAD	#3	0	00

Seit ein paar Tagen steht in meiner Gummizelle ein neuer Komposter namens G3s. Sein Betriebssystem, G-DOS 2.4, steckt zwar so manches andere in die Tasche, aber gewisse H-DOS-Features vermisse ich doch sehr. Bei der ewigen Zapperei ist es natürlich der Library-Befehl DDE in SYS15/SYS, den ich am meisten brauche. Und so ist denn logischerweise SYS15/SYS das erste File, das in Richtung H-DOS gestreamlined wurde:

Im Gegensatz zur Version für das G1/2 lädt diese SYS-Datei in den DOS-Overlay-Bereich. Man kann sie deshalb sogar unter Mini-DOS benutzen. Das Argument, man solle den Disketten-Editor lieber als CMD-File fahren, zieht deshalb beim G3 nicht mehr.

Beim Einsprung wird zunächst geprüft, ob der Speicherbereich ab 3000h benutzt werden darf oder gerade von einem anderen Modul des Betriebssystems belegt ist. Wenn er nicht frei ist, wird automatisch die Datei DDE52/SYS aufgerufen, die mit dem SYS15/SYS für das G1/2 praktisch identisch ist. Klarer Fall: In DDE52/SYS wurde sofort SYS15/SYS aus H-DOS mit geringfügigen Änderungen einkopiert. Diese Änderungen werden hier nicht näher benannt, denn sie erscheinen auch weiter unten im modifizierten SYS15/SYS für das G3s.

Wenn in 3000h ff. aber Platz ist, verlagert sich der Hauptteil der Datei dorthin. Das Bildschirmformat wird auf 16 X 64 Zeichen eingestellt und das Programm an seinem neuen Platz schließlich angesprochen.

Ein paar Programmschritte zuvor hakt die erste Änderung ein: Im M-Modus, wenn also ein Sektor modifiziert werden soll, erscheint ein Cursor. Das ist in der Originalversion ein Graphikblock, der das betreffende Zeichen natürlich überdeckt. Schade drum! Das G3s kann Zeichen invers darstellen. Also ruft die neue Version von DDE zunächst in SYS22/SYS den Befehl Z,YR auf, der von Graphik auf inverse Zeichen ab 80h umstellt. Das sind im anhängenden Programmlisting die Zeilen 56-64. Der Z-Befehl braucht dabei nicht zu erscheinen, da das Programm ökonomischer direkt mit RST 28h arbeitet. Nur das Argument YR, gefolgt von CR (ENTER), erscheint explizit an der Stelle invar (Zeile 479, Parameter für Inversdarstellung). Übrigens sind diejenigen Teile des Listings, die hier nicht von Belang sind, mit LIST OFF ausgespart. Die Auslassungen sind durch Striche zwischen den Zeilennummern gekennzeichnet.

Die Änderung in den Zeilen 169-171 und 178-180 bezieht sich direkt auf die Form des Cursors: Das gerade angezeigte Zeichen wird jetzt nicht mehr vom Graphikblock mit dem Code 8Fh zugeschaufelt, sondern stattdessen wird sein Bit 7 invertiert, so daß das inverse Zeichen jetzt im Akku steht. Es wird statt seiner in den Bildschirm geladen.

Im gleichen Zusammenhang sind die Zeilen 380 und 381 zu sehen: der Code, der aus dem Sektorpuffer geholt wird (DE als Zeiger), wird direkt und ohne Änderungen in den Bildschirm geladen (HL als Zeiger). Da alle 256 Codes mit dem G3s anzeigbar sind, ist es nicht mehr erforderlich, einzelne von ihnen durch Punkte zu ersetzen.

Bis hierher beziehen sich die Modifikationen nur auf das optische Erscheinungsbild, sind also nur ebenso sinnvoll oder sinnlos wie ein auf Diskette sortiertes Directory (das beim nächsten Dateieintrag ein wieder durcheinandergerät). Eine wirklich sinnvolle, geradezu notwendige Änderung ist in den Zeilen 300-308 programmiert:

Beim Umschalten mit CLEAR von linken Hex- ins rechte ASCII-Feld geht im M-Modus der Cursor in der Originalversion immer in die jeweilige linke obere Ecke. Modifiziert man also gerade ziemlich weit unten im Sektor und will beispielsweise zwischen zwei Wörtern ein Steuerzeichen eingeben, das nicht über die Tastatur erreichbar ist, dann darf man zunächst mit den

Pfeiltasten Schiffversenken spielen. In der hier vorgestellten Version geht der Cursor im Nachbarfeld immer genau an dasselbe Byte, so daß man sofort weitertippen kann. Und das geht so:

Beim Ansprung dieses Programmsegments steht die Cursoradresse gerade in HL. Der Anfang der aktuellen Bildschirmzeile wird durch UND-Verknüpfung des Cursor-LSB mit COh gewonnen und vorübergehend in L verstat. Jede Bildschirmzeile bezieht sich auf 16 Bytes des Sektorpuffers. Dessen Zeiger-LSB ist E. Mit OFh verknüpft, ergibt es die aktuelle Stelle in einem jeden Block von je 16 Bytes. Sie wird in B kühl und trocken aufbewahrt. Jetzt sind die passende Videozeile und das laufende Byte in ihr dingfest gemacht. Ein schlichtes LD A,(HL) o. ä. hätte leider nicht genügt, denn wäre der Cursor gerade im Hexfeld gewesen, hätte diese Operation nur ein Nibble erbracht, obendrein in ASCII-Darstellung.

Jetzt wird in Zeile 306 das LSB der aktuellen Videoadresse aus ihrem Puffer geladen. Nun geschieht der eigentliche Wechsel von ASCII nach Hex bzw. umgekehrt. Weshalb er ausgerechnet durch die Exklusiv-ODER-Verknüpfung mit 38h vollzogen wird, wäre einen eigenen Artikel über das Liebesleben der Bits wert. Freuen wir uns einfach, daß mir dieser kurze Weg eingefallen ist und reden wir nicht weiter drüber! Das neue Video-LSB wird anschließend in den Puffer geschrieben.

Für das Weitere muß das Programm nun wissen, ob das gerade ASCII oder Hex wurde. Wenn im LSB der Cursoradresse das Bit 4 oder 5 gesetzt war (= 30h), dann war es wohl das ASCII-Feld, denn im rechten Viertel jeder Bildschirmzeile sind diese Bits immer 1. Das LSB des Zeilenanfangs wird aus dem Datenfriedhof L in den Akku zurückgeladen. Falls ASCII, erfolgt nun der Sprung zum Label ascii (Zeile 325). Auch was dort geschieht, ist wieder binäre Akrobatik, die ich hier nicht erläutern will. Letztenendes kommt das neue Bildschirm-LSB in den halben Zeiger L, und die Arbeit ist getan.

Ein bißchen schwieriger ist es, wenn soeben ins Hexfeld umgeschaltet wurde. Die erste Hex-Ziffer erscheint an der achten Stelle, weil am Zeilenanfang zuerst der Kopf mit Sektor- und Zeilennummer steht. Vorab kommt dieses vorläufige Cursor-LSB nach L. Weiter oben war die Stelle in einem laufenden 16er-Block des Sektorpuffers nach B gerettet worden. Wenn es die nullte Stelle war, ist es damit erledigt. Andernfalls wird zunächst die halbe Videostellenzahl ermittelt, um sie später als Anzahl der trennenden Blanks hinzuaddieren zu können. Danach werden für jedes Byte im Puffer zwei Stellen addiert, denn es werden ja je zwei Nibbles angezeigt. Die Summe der Nibbles und Blanks ist schließlich das LSB der tatsächlichen Bildschirmadresse.

Uff! In einer letzten mörderischen Anstrengung habe ich schließlich noch in den Zeilen 411 und 412 klargemacht, daß dies eine neue Version von DDE ist, und wer da wann mitgemischt hat.

Abschließend ein Wort zum Listing. Da sich die Versionen von SYS15/SYS für G1/2 und G3/3s unterscheiden, mußte zunächst eine Disassembly erstellt und näher untersucht werden. Die große Mehrzahl der absoluten Adressen (bei CALLs, JPs, Speicherladebefehlen usw.) bezog sich darin auf den Bereich ab 3000h nach der Relokation des Hauptprogramms. Da aber die Modifikationen nahezu alle Adressen veränderten, mußte jede dieser Adressen auf die Ladeadresse vor dem Programmstart bezogen werden. Deshalb wurde das Label offs (Offset zur Arbeitsadresse) eingeführt. Als Subtrahend eingesetzt, korrigierte es alle Verschiebungen durch die Manipulationen.

Der Quelltext wurde mit ZEUS/CMD erstellt. Dieser Assembler erlaubt eine sehr bequeme Programmierung. Es dürfte für den EDTASM-gewohnten Assembler-Crack kein Problem sein, die Programmlogik zu verstehen.

Das komplette Listing des Programms SYS15/SYS wäre für unsere Poststelle nicht nur zu lang, sondern es würde sich garantiert keiner der Mühe unterziehen, es abzutippen. Recht so! Wenigstens ein Hexdump der ersten vier Sektoren ist beigefügt, damit der geneigte Leser und Selbst-modifizierer an der Umgebung erkennt, was sich wo gerade abspielt. Im Dump sind die Änderungen gegenüber Version 2.1 unterstrichen. Da Teile gelöscht, andere Teile hinzugefügt sind, finden sich die unterstrichenen Segmente nicht unbedingt an derselben Stelle wie im TCS-Original.

Wer für sein G3s die Version 2.2 haben will, mag mir einen frankierten und adressierten Umschlag mit einer Leerdiskette zuschicken (bitte mit der üblichen G-DOS-Formatierung, sonst Zettelchen nicht vergessen!). Ohne das G3 (ohne s) zu kennen, vermute ich, daß das Programm auch auf dieser Maschine ohne größere Änderungen laufen müßte.

Arnulf Sopp

```

4D48 E5 00058 m4d48 PUSH HL ;Befehlszeiger retten
4D49 21DF51 00059 LD HL,invpar ;YR für Z-Befehl
4D4C 3EF8 00060 LD A,0f8h ;für SYS22/SYS
4D4E 0E01 00061 LD C,01h ;für Z,YR
4D50 CD0244 00062 CALL 4402h ;dort RST 28h
4D53 E1 00063 POP HL ;Befehlsz. restaurieren
4D54 11D732 00064 LD DE,m32d6-offs ;weiter wie gehabt
4E31 3E80 00169 LD A,80h ;Bit 7 = 1
4E33 AE 00170 XOR (HL) ;Zeichen invertieren
4E34 77 00171 LD (HL),A ;und neu laden
4E3D 3E80 00178 LD A,80h ;Bit 7 = 1
4E3F AE 00179 XOR (HL) ;Zeichen invertieren
4E40 77 00180 LD (HL),A ;und neu laden
4F00 7D 00300 m31a8 LD A,L ;LSB der Cursoradresse
4F01 E6C0 00301 AND 0c0h ;auf Zeilenanfang
4F03 6F 00302 LD L,A ;zwischen speichern
4F04 7B 00303 LD A,E ;LSB Pufferstelle
4F05 E60F 00304 AND 0fh ;Stelle im 16er-Block
4F07 47 00305 LD B,A ;zwischen speichern
4F08 3ABD32 00306 LD A,(m32bc-offs) ;Video-LSB
4F0B EE3B 00307 XOR 3Bh ;ASCII-Feld (->) Hexfeld
4F0D 32BD32 00308 LD (m32bc-offs),A ;Video-LSB neu ablegen
4F10 E630 00309 AND 30h ;jetzt ASCII-Feld?
4F12 7D 00310 LD A,L ;LSB d. Cursoradr. zurück
4F13 2011 00311 JR NZ,ascii ;falls jetzt ASCII-Feld
4F15 F60B 00312 OR 0Bh ;Hexfeld, auf B. Stelle
4F17 6F 00313 LD L,A ;Hexfeld-Zeilenanfang
4F18 AF 00314 XOR A ;A <- 00
4F19 B0 00315 OR B ;0. Stelle im 16er-Block?
4F1A CB 00316 RET Z ;erledigt, falls ja
4F1B 48 00317 LD C,B ;C <- Stellenzahl
4F1C CB39 00318 SRL C ;die Hälfte = Blanks
4F1E 7D 00319 LD A,L ;Zeilenanfang
4F1F C602 00320 hex: ADD A,02h ;je 2 Stellen/Byte
4F21 10FC 00321 DJNZ hex ;fortgesetzt addieren
4F23 B1 00322 ADD A,C ;+ Anzahl der Blanks
4F24 6F 00323 LD L,A ;neue Cursoradresse
4F25 C9 00324 RET ;erledigt
4F26 F630 00325 ascii OR 30h ;auf letzte 16 Stellen
4F2E B0 00326 OR E ;+ Stelle im 16er-Block
4F29 6F 00327 LD L,A ;neue Cursoradresse
4F2A C9 00328 RET ;erledigt
4F87 1A 00380 m4f5B LD A,(DE) ;ASCII-Code laden
4F8E 77 00381 LD (HL),A ;unverändert anzeigen
4FAA 1C 00410 m324d DE 1ch,1fh
4FEC 47 00411 DM *Genie-DDS Diskdaten-Editor Vers. 2.2*
4FE0 0A 00412 DM 0ah,0ah,'(C) 1986 TCS/HACKTORY',0dh
51D7 0000 0047B m51e0 DW 0000h,0000h,0000h,0000h
51DF 59 00479 invcar DM 'YR',0dh
1D48 00480 offe EQU m4d48-3000h

```

```

000000: 0102 004D 3A07 43E6 OFFE 0238 1BE5 214B AB6M:GC 0 EBA 1H
000010: 4D11 0030 0100 04ED B021 0030 E33A 1419 MD0A6E !50 :TY
000020: FE40 3E10 C233 00C9 3A6A 43CB 7F3E 38C0 9>P 39 :;C 3B
000030: E501 0A00 213E 4D11 D751 D5ED B0D1 E1C3 AJ5!>MO 0
000040: 3344 4444 4535 322F 5359 530D E521 DF51 3DDDE52/SYSM ! 0
000050: 3EF8 0E01 CD02 44E1 11D7 32CD 1C44 2827 > NA BD 0 2 6D(*)
000060: 2172 32CD 6744 21B0 32CD 6744 0618 21BE !r2 gP! 2 gDFX!
000070: 32CD 4000 D87E FE23 281A FE2A 2830 11D7 2 55 B #:(Z *(00
000080: 32CD 1C44 C251 3206 0021 0033 CD24 44C4 2 6D 02F5!53 $D
000090: 5A32 1807 237E E60F 32DD 32ED 4BE1 3211 Z2XG#B 2 2 K 20
0000A0: D732 CD42 44CD 3644 C45A 32CD 4544 CDOA 2 BD 6D 22 ED J
0000B0: 32CD 4832 FE01 28AB FE3B 2836 FE2B 2815 2 K2 A : (6 + (U
0000C0: FE3D 281E FE2D 281F C8AF FE53 2839 FE4D =( ^ - ( _ ( 5 (9 M
0000D0: CAD7 3018 D9ED 4BE3 323A DF32 E720 C00B 0X k 2: 2 K
0000E0: 18BD 0100 0018 B8ED 4BE1 3278 B12B BFOB X A55X K 2x ( K
0000F0: 18AD ED4B E132 032A E332 3ADF 32B7 2001 X K 2C* 2: 2 A
000100: 2BED 4230 0102 004E 9A18 A701 0000 CD4E + BOAB5N X A55 K
000110: 32FE 2138 E6C5 CDF7 31C1 3BF2 6069 2929 2 18 1 B 1))
000120: 2929 854F 4418 E711 0033 263C 3AED 326F )) ODX 053&<: 2o
000130: CD0A 327C E603 C63C 673E B0AE 773A BD32 J26 C <g> w: 2
000140: D608 2006 233E B0AE 772B 01E0 30C5 CD4E H F#> w+A 0 K
000150: 32FE 0928 46FE 0928 64FE 0828 6CFE 0ACA 2 I(F K(d H(1 J
000160: A031 FE40 CAAC 31FE 1FCA B831 FE0D CAE3 1 5 1 _ 1 M
000170: 31FE 01CA F331 473A BD32 D608 7820 1ACD 1 A 16: 2 Hx Z
000180: F731 38CA 7017 1717 174F CD4B 32CD F731 1B pWWW0D K2 1
000190: 38FB 2370 2B81 1218 0277 123A BD32 D608 B #p+ RXBwR: 2 H
0001A0: 2010 7B0F 7DCE 026F 1C7B E60F C001 1800 PÄDÜ Boä 0 AXB
0001B0: 09C9 231C 7BE6 OFC0 0130 0009 C901 CÖFF I #ää 0 A051 A
0001C0: 097B D610 5FD0 263F C93A BD32 D608 2013 Iä P_ &? : 2 H S
0001D0: 7B0F 7D3F DE02 6F1D 7BE6 OFEE OFC0 01E8 äÜ? Boü 0 0 A
0001E0: FF09 C92B 1D7B E60F EE0F C001 DOFF 09C9 I +üä 0 0 A I
0001F0: 0140 0009 7BC6 105F D026 3CC9 D53A BD32 A55Iä F_ &< : 2
000200: E620 121C 20FC D1C9 0102 004F 7DE6 C06F R6 AB50ü o
000210: 7BE6 0F47 3ABD 32EE 3832 BD32 E630 7D20 ä 0G: 2 B2 2 0ü
000220: 11F6 086F AFB0 CB4E CB39 7DC6 0210 FC81 0 Ho H 9ü BP
000230: 6FC9 F630 B06F C9C1 11D7 32CD 3944 C45A o 0 o 0 2 9D Z
000240: 32CD 4544 C362 30C1 C34F 3047 D630 D8FE 2 ED b0 006 0
000250: 0A3B 09CB AFFE 11DB D607 FE10 3FC9 0921 JB1 0 G P? !
000260: 003C 1100 33D5 0608 3AE2 32CD 6840 3AE1 5<053 FH: 2 h5:
000270: 32CD 6840 7BCD 6840 3E3A 7723 3E20 7723 2 h5ä h5: w#> w#
000280: 1ACD 6840 1C1A CD68 401C 10F0 3E20 7723 Z h5Z h5F > w#
000290: D106 101A 7723 1C2B 0410 FB18 CBD9 C9D5 FPZw#6(DP X
0002A0: CD49 00D1 C921 B932 CD67 44C3 1A30 F6C0 15 ! 2 gD ZO
0002B0: FEC6 2808 3333 CD09 44C3 1A30 3AD7 32F6 (H33 ID ZO: 2
0002C0: 0132 D732 AFC9 1C1F 4765 6E69 652D 444F A2 2 ä_Genie-DO
0002D0: 5320 4469 736B 6461 7465 6E2D 4564 6974 S Diskdaten-Edit
0002E0: 6F72 2056 6572 732E 2032 2E32 0A0A 2843 or Vers. 2.2JJ(C
0002F0: 2920 3139 3836 2054 4353 2F4E 4143 4B54 ) 1986 TCS/HACKT
000300: 4F52 590D 0A44 6174 6569 3F20 0102 0050 ORYMJDat? AB5P
000310: 0307 1B1B 0330 3031 3233 3435 3637 3839 CGAAC00123456789
000320: 4142 4344 4546 3031 3233 3435 3637 0DE2 ABCDEF01234567M
000330: 2800 0033 0000 0000 0000 00FF FFFF FFFF (5535555555
000340: FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF 5
000350: 0000 0000 0000 0000 0000 0000 0000 0000 55555555555555
000360: 0000 0000 0000 0000 0000 0000 0000 0000 55555555555555
000370: 0000 0000 0000 0000 0000 0000 0000 0000 55555555555555
000380: 0000 0000 0000 0000 0000 0000 0000 0000 55555555555555
000390: 0000 0000 0000 0000 0000 0000 0000 0000 55555555555555
0003A0: 0000 0000 0000 0000 0000 0000 0000 0000 55555555555555
0003B0: 0000 0000 0000 0000 0000 0000 0000 0000 55555555555555
0003C0: 0000 0000 0000 0000 0000 0000 0000 0000 55555555555555
0003D0: 0000 0000 0000 0000 0000 0000 0000 0000 55555555555555
0003E0: 0000 0000 0000 0000 0000 0000 0000 0000 55555555555555
0003F0: 0000 0000 0000 0000 0000 0000 0000 0000 55555555555555

```

Bis ich mich endlich entschlossen hatte, bei meinem alten Genie I die HRG 1b einzubauen, vergingen seit dem Kauf des Computers über zwei Jahre. "Spielkram!" war mein Argument dagegen. Aber die zusätzlichen 12 kB Speicher waren schließlich doch zu verlockend. Beim G3s ist von Hause aus eine Baugruppe für hochauflösende Graphik drin. Sie umfaßt sogar zweimal 32, also zusammen 64 kB. So kann sich der geneigte Leser lebhaft meinen Ärger vorstellen, als in der Dokumentation (wie bei Trommeschläger üblich nicht ausreichend) keine brauchbare Erklärung darüber zu finden war, wie man diesen Speicher anspricht.

Es wird ein sehr interessantes Graphik-BASIC auf der G-DOS-Diskette mitgeliefert. Aber es braucht den Speicher für die absonderlichsten Luxusroutinen auch dann, wenn man bloß eben mal irgendwo einen Punkt hinsetzen möchte. Und letztendes ist es natürlich nur unter BASIC nutzbar; für den Assembler-Crack ein gewichtiger Grund dagegen. Also war der Mann mal wieder selbst:

Aus der technischen Beschreibung geht (zumindest nach mehrmaligem Lesen und Probieren) hervor, daß der Graphikspeicher von 8000-FFFF geht. Er kann anstelle des Hauptspeichers durch einen Output auf den Port FA mit gesetztem Bit 3 der CPU zugänglich gemacht werden. Da dieser Port noch weitere Funktionen des Systems steuert, muß er zunächst gelesen und an diesem Input nur bitweise getummelt werden, damit der bisherige Betriebszustand nicht in fataler Weise verändert wird. Jetzt kann in 8000-FFFF beliebig geschrieben und gelesen werden. Dieser Adreßraum ist jetzt nach dem Umschalten über Port FA der Graphikspeicher.

Der logische Aufbau der HRG des G3s ist der HRG 1b ganz ähnlich. Auch hier entspricht ein bestimmtes Byte des Speichers der ganzen Breite einer Punktzeile einer normalen Anzeigestelle des Bildschirms. Deshalb sind auch hier die LSB der Graphik- und Bildschirmadressen gleich. Die Reihenfolge der Bits (Graphikpunkte) ist ebenfalls dieselbe: Bit 0 wird ganz links angezeigt, Bit 7 rechts (die HRG 1b zeigt nur bis Bit 5 an). Wegen der großen Ähnlichkeit mit der HRG 1b, die mir sehr vertraut ist, erschien deshalb die Aufgabe lösbar, der HRG des G3s auf die Schliche zu kommen.

Aber da ist eine Merkwürdigkeit, die vorab geklärt werden mußte: Jeder Anzeigeodus (25 X 80, 16 X 64, 24 X 64 und 32 X 64 Zeichen) benutzt einen unterschiedlich großen Bildschirmspeicher. Außerdem werden unterschiedlich tiefe Punktspalten pro Bildschirmbyte angezeigt. Nur im Modus 32 X 64 kommen 32 X 64 X 16 X 8, also 512 X 512 = 262144 Punkte, das sind 32 kB, zur Anzeige. Was geschieht in den anderen Modi mit dem übrigbleibenden Graphikspeicher? Ein Testprogramm, das im Anschluß an diesen Artikel aufgelistet ist, brachte Aufschluß.

Das Programm hat seinen Zweck erfüllt. Es mündete in die beiden Tabellen hinter dem Listing. Da es deshalb niemand abzutippen braucht, seien hier nur diejenigen Teile erläutert, die die Graphik ansprechen, denn diese Teile kann der Leser in seine eigenen Programme einbauen:

Im DOS liegt der Stack ziemlich tief unten. Um aber das Programm notfalls auch fahren zu können, wenn der Stack im Himmel liegt, war es notwendig, den Stackpointer zu retten und neu zu laden. Nach dem Umschalten würde er zwar normal arbeiten, wäre aber in der Graphik sichtbar.

Der nächste Schritt (Unterprogramm change) sollte an wichtigen Stellen des Bildschirm-RAMs Marken setzen, so daß die jeweilige Video-Adresse ablesbar ist. Es sind die Anfangs- und Endadressen des Bildschirms bei jeden der vier möglichen Bildschirmformate. Da im unteren Adreßraum der großer Formate gleichzeitig die Tastatur memory mapped ist,

mußte sie zunächst über das Bit 4 des Ports FA weggebankt und anschließend wieder enabled werden. Der einleitende CALL nach 01C9h löscht den Bildschirm und aktiviert gleichzeitig Formatänderungen (s. u.).

Das Löschen des Graphikspeichers ist einfacher als mit der HRG 1b: Da die CPU direkten Zugriff hat, braucht er nur mit LDIF ausgenullt zu werden. Die anschließend folgende Abfrage der P-Tasten, mit denen das Programm zwischen den Bildschirmformaten hin- und herspringen kann, soll hier nicht weiter erklärt werden. Wie aber die Formate gewechselt werden, ist wieder wichtig für den Graphik-Programmierer:

Ab 3760h stehen vier Tabellen zu je 16 Bytes, die die Bildschirmparameter enthalten. Die Parameter für das gerade benutzte Format stehen ab 37F0h. Deshalb werden beim Umschalten des Formats die neuen Parameter, wieder mit LDIR, dorthin geladen. Das alleine genügt allerdings nicht. Aktiv werden sie erst, wenn gewisse Control-Codes ausgegeben werden. Das ist z. B. auch beim Löschen des Bildschirms der Fall, deshalb wird nach dem Formatwechsel erneut change aufgerufen.

Das eigentliche Arbeitsprogramm steht ab loop. Es sei nur so viel dazu gesagt, daß ein Graphik-Cursor mit den Pfeiltasten über den Bildschirm gejagt werden kann. Mit der Blank-Taste kann er verlangsamt werden, um einzelne Bildschirmstellen gezielt ansteuern zu können. Die jeweilige Adresse des Cursors im Graphikspeicher wird ständig angezeigt. Wenn dieser Cursor nun eine der im UF change gesetzten Marken durchläuft, kann man sich die Bildschirmadresse der Marke und die Graphikadresse des Cursors notieren und erfährt so den Zusammenhang zwischen beiden.

Für den Computergraphiker wird es nun ab der Stelle exit wieder interessant: Der alte Stackpointer wird wieder aus dem Puffer geladen. Die Bits 1 und 3 des Systembytes 1 (Port FA) werden wieder auf 0 gesetzt, wodurch die Graphik vom Bildschirm verschwindet und der Hauptspeicher anstelle des Graphikspeichers in die obere Adreßhälfte eingeblendet wird. Die Interrupts waren die ganze Zeit über ausgeschaltet, denn bei zeitweise ausgebankter Tastatur und manipuliertem Stack ist Vorsicht angebracht. Mit EI werden sie wieder enabled. Mit RET geht es schließlich zurück ins Betriebssystem.

Die beiden Tabellen sind folgendermaßen zu interpretieren: Wie auch bei der HRG 1b, so kann man sich für die jeweils oberste Punktzeile eines Zeichens den ganzen Bildschirm als eine ununterbrochene Perlenkette vorstellen. Wenn die oberste Zeile durchlaufen ist (also rechts unten im Bildschirm!), dann dann geht es in der zweiten Punktzeile (oben links!) weiter. Beim Anzeigeformat 32 X 64 geschieht das lückenlos.

Die anderen Formate zeigen aber Unterbrechungen beim Zeilenwechsel. Besonders die Graphik auf der letzten Seite dieses Artikels dürfte das anschaulich machen. Aus beiden Tabellen geht auch hervor, daß pro Bildschirmstelle bei den kleineren Formaten auch nicht alle Punktzeilen zur Anzeige kommen. Die Punktzeilen sind von 0 - F durchnummeriert.

Zur praktischen Anwendung: Die Tabellen zeigen, welche Bereiche des Graphikspeichers überhaupt im Bildschirm sichtbar werden können, je nach Format. Die Abszisse eines Graphikbytes errechnet sich nun ebenso wie die eines Buchstabens im normalen Bildschirm. Um aus diesem Beitrag kein Buch zu machen, verweise ich hierzu auf die Vielzahl von Artikeln zur HRG 1b, die an dieser Stelle erschienen sind.

Die Ordinate, also die Lage eines Bytes in senkrechter Richtung, ist vom darüber-/darunterliegenden Byte immer genau 2 kB im Graphikspeicher entfernt, solange es sich um dieselbe Videozeile handelt. Beim Sprung in die nächste Videozeile (nicht zu Verwechseln mit den Punktzeilen eines Zeichens!) ist einfach zu einer Adresse die Zeilenlänge hinzuzusaddieren, also 80 oder 64.

Jedes Graphikbyte hat selbstverständlich ebenfalls 8 Bits. Jedes Bit repräsentiert einen Punkt in der Waagerechten. Wie auch bei der HRG 1b, so kann man auch beim G3s durch Setzen, Löschen oder Prüfen einzelner Bits die BASIC-Entsprechungen SET, RESET und POINT durchführen. Auch zu diesem Punkt mag es genügen, auf die bisher erschienen Artikel zur HRG 1b zu verweisen. Dabei muß lediglich bedacht werden, daß alle acht Bits beim G3s zur Anzeige kommen.

Arnulf Sopp

5300	00001	ORG	5300h	;darunter Platz für Stack
5300	00002	stack EQU	\$;hier fängt er an
0002	00003	spbuf DS	2	;Puffer für Stackpointer
5302	F3	00004	start DI	;nichts riskieren
5303	ED730053	00005	LD (spbuf),SP	;Stackpointer retten
5307	310053	00006	LD SP,stack	;eigener Stack f. Progr.
530A	CD8553	00007	CALL change	;Bildschirm markieren
530D	210080	00008	LD HL,8000h	;Beginn des Graphiksp.
5310	110180	00009	LD DE,8001h	;eine Stelle weiter
5313	01FF7F	00010	LD BC,7fffh	;Zähler zum Löschen
5316	75	00011	LD (HL),L	;1. Graphikstelle löschen
5317	EDB0	00012	LDIR	;alle löschen
5319	11023C	00013	LD DE,3c02h	;Bildschirmbereich
531C	3A8038	00014	loop LD A,(3880h)	;P-Tasten abfragen
531F	E6F0	00015	AND 0f0h	;eine gedrückt?
5321	281A	00016	JR Z,ctrlkey	;falls nein
5323	E5	00017	PUSH HL	;Graphikzeiger retten
5324	D5	00018	PUSH DE	;dto. Bildschirmzeiger
5325	210637	00019	LD HL,3706h	; (Adr. Video-Par.-tab.)
5328	11F037	00020	LD DE,37f0h	;dto. aktive Parameter
532B	0E20	00021	LD C,20h	;16 Parameter
532D	CDAS53	00022	CALL pkey	;Parameter-LSB ermitteln
5330	7D	00023	LD A,L	;Parameter-LSB
5331	07	00024	RLCA	;sum 4 Bits rot. (*16)
5332	07	00025	RLCA	
5333	07	00026	RLCA	
5334	07	00027	RLCA	
5335	6F	00028	LD L,A	;jetzt kompl. Adr. in HL
5336	EDB0	00029	LDIR	;neue Parameter setzen
5338	D1	00030	POP DE	;Register restaurieren
5339	E1	00031	POP HL	
533A	CD8553	00032	CALL change	;Änderungen aktivieren
533D	3A4038	00033	ctrlkey LD A,(3840h)	;Tastatur auslesen
5340	E6FC	00034	AND 0fch	;Pfeile, Blank oder BRK?
5342	28DB	00035	JR Z,loop	;falls nichts dergleichen
5344	07	00036	RLCA	;Blank?
5345	063F	00037	LD B,03fh	;für Verzögerung
5347	3802	00038	JR C,goloop	;falls Blank
5349	0607	00039	LD B,07h	;sonst weniger Verzög.
534B	0EFF	00040	goloop LD C,0fffh	;Zähler BC komplettieren
534D	D5	00041	PUSH DE	;retten
534E	110004	00042	LD DE,0400h	;in d. Br.-split. darunter
5351	3600	00043	LD (HL),00h	;alten Strich löschen
5353	07	00044	RLCA	;Rechtspfeil?
5354	DCAD53	00045	CALL C,right	;falls ja
5357	07	00046	RLCA	;Linkspfeil?
5358	DCAF53	00047	CALL C,left	;falls ja
535B	07	00048	RLCA	;Abwärtspeil
535C	DCB153	00049	CALL C.down	;falls ja
535F	07	00050	RLCA	;Aufwärtspeil?
5360	DCB353	00051	CALL C.up	;falls ja

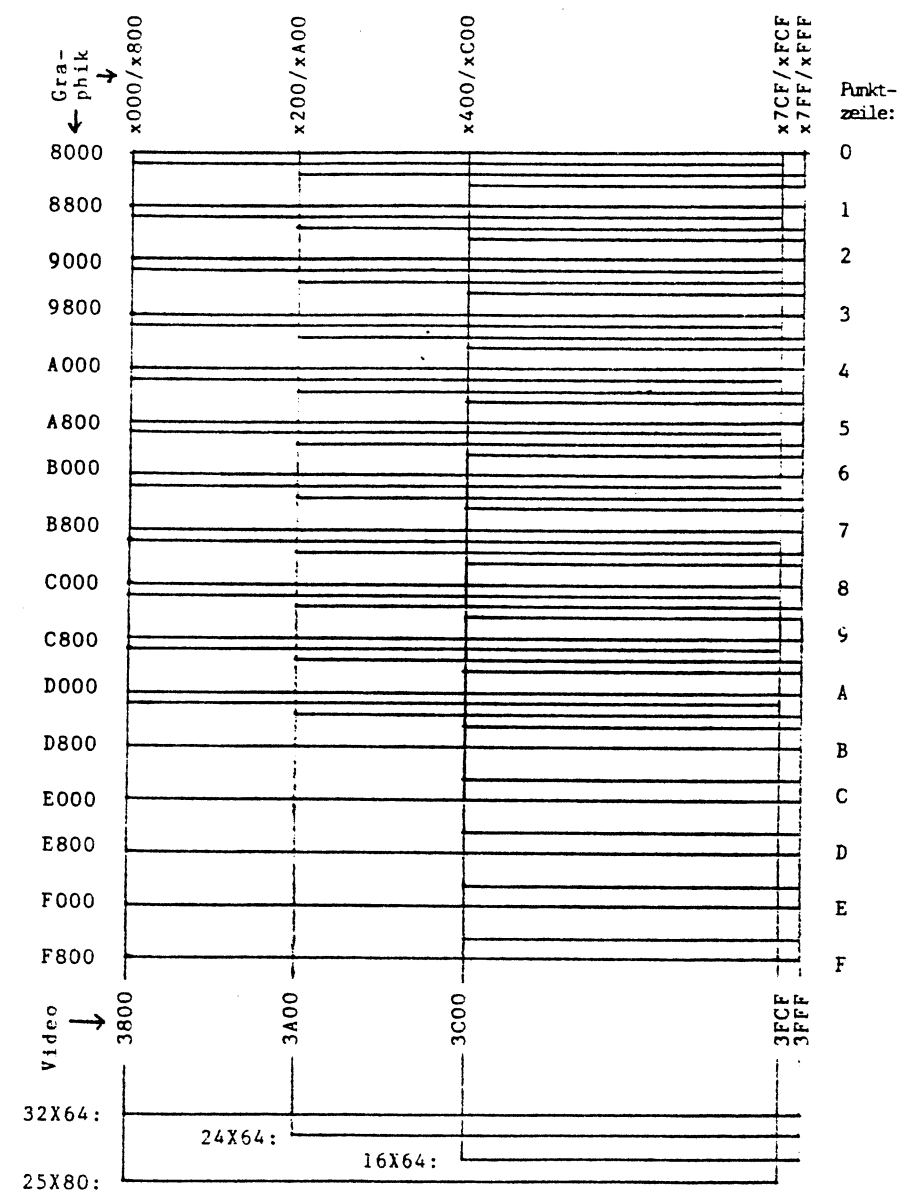
5363	D1	00052	POP DE	;restaurieren
5364	07	00053	RLCA	;BREAK
5365	3B12	00054	JR C,exit	;falls ja
5367	CEFC	00055	SET 7,H	;nur B000 - FFFF zulassen
5369	36FF	00056	LD (HL),0fffh	;dort neuen Strich laden
536B	EB	00057	EX DE,HL	;austauschen
536C	E5	00058	FUSH HL	;wird verändert
536D	CD6340	00059	CALL 4063h	;Stelle in Hex anzeigen
5370	E1	00060	POP HL	;restaurieren
5371	EB	00061	EX DE,HL	;zurücktauschen
5372	0B	00062	delay DEC BC	;Verzögerungszähler dekr.
5373	7B	00063	LD A,B	;nachsehen, ob
5374	B1	00064	OR C	;Zähler abgelaufen
5375	20FB	00065	JR NZ,delay	;falls noch nicht
5377	18A3	00066	JR loop	;o. k., von vorne
5379	ED7B0053	00067	exit LD SP,(spbuf)	;Stackpointer restaur.
537D	DBFA	00068	IN A,(0fah)	;Systembyte 1 lesen
537F	E6F5	00069	AND 0f5h	;Bits 7 und 5 wieder aus
5381	D3FA	00070	OUT (0fah),A	;Graphik ausblenden
5383	FB	00071	EI	;keine Gefahr mehr
5384	C9	00072	RET	;ins Betriebssystem
5385	CDC901	00073	change CALL 01c9h	;Änderungen aktivieren
5388	DBFA	00074	IN A,(0fah)	;Systembyte 1 lesen
538A	F61A	00075	OR 1ah	;Bits 1, 3 und 4 setzen
538C	D3FA	00076	OUT (0fah),A	;Graphik fitmachen
538E	3EBF	00077	LD A,0bfh	;Grobgraphikblock
5390	32003B	00078	LD (3B00h),A	;Anfang Großbildschirm
5393	32003C	00079	LD (3c00h),A	;Anfang Kleinbildschirm
5396	32CF3F	00080	LD (3fcfh),A	;Ende 80-Zeichen-Bildsch.
5399	32FF3F	00081	LD (3fffh),A	;Ende sonstiger Bildsch.
539C	DBFA	00082	IN A,(0fah)	;Port wieder lesen
539E	CBA7	00083	RES 4,A	;Tastatur wieder aktiv.
53A0	D3FA	00084	OUT (0fah),A	;mit Systembyte 1
53A2	C9	00085	RET	;oben weiter
53A3	07	00086	pkey RLCA	;P1-Taste? (80 X 25 Z.)
53A4	DB	00087	RET C	;falls ja
53A5	2C	00088	INC L	;Adr.-LSB d. Tab. erhöhen
53A6	07	00089	RLCA	;P2-Taste? (64 X 16 Z.)
53A7	DB	00090	RET C	;usw.
53A8	2C	00091	INC L	
53A9	07	00092	RLCA	;P3-Taste? (64 X 24 Z.)
53AA	DB	00093	RET C	
53AB	2C	00094	INC L	
53AC	C9	00095	RET	;P4: 64 X 32 Z.
53AD	23	00096	right INC HL	;nächste Graphikstelle
53AE	C9	00097	RET	;oben weiter
53AF	2B	00098	left DEC HL	;vorige Graphikstelle
53B0	C9	00099	RET	
53B1	19	00100	down ADD HL,DE	;tiefer in selber Spalte
53B2	C9	00101	RET	
53B3	B7	00102	up OR A	;Cy löschen
53B4	EDE2	00103	SBC HL,DE	;höher in selber Spalte
53B6	C9	00104	RET	;erledigt
5302	00105	END	start	;dort Einsprung

00000 Fehler

change	5385	ctrlkey	533D	delay	5372	down	53B1	exit	5379
goloop	534E	left	53AF	loop	531C	pkey	53A3	right	53AD
spbuf	5300	stack	5300	start	5302	up	53B3		

Tabelle der Graphik- und Bildschirmadressen des Genie III s:

Bildschirm- darstellung:	Bildsch.- adressen:	Zei- le:	Graphik- adressen:
25 X 80 Zeichen:	3800 - 3FCF	0	8000 - 87CF
		1	8800 - 8FCF
Beispiel für einige Zeilen:		2	9000 - 97CF
3800 - 384F <-	8000 - 804F	3	9800 - 9FCF
3850 - 389F <-	8050 - 809F	4	A000 - A7CF
.		5	AB00 - AF CF
usw.		6	B000 - B7CF
.		7	BB00 - BF CF
3F30 - 3F7F <-	D730 - D77F	8	C000 - C7CF
3F80 - 3FCF <-	D780 - D7CF	9	CB00 - CF CF
		A	D000 - D7CF
16 X 64 Zeichen:	3C00 - 3FFF	0	B400 - B7FF
		1	BC00 - BFFF
Beispiel für einige Zeilen:		2	9400 - 97FF
3C00 - 3C3F <-	8400 - 843F	3	9C00 - 9FFF
3C40 - 3C7F <-	8440 - 847F	4	A400 - A7FF
.		5	AC00 - AFFF
usw.		6	B400 - B7FF
.		7	BC00 - BFFF
3F80 - 3FBF <-	F780 - F7BF	8	C400 - C7FF
3FC0 - 3FFF <-	F7C0 - F7FF	9	CC00 - CFFF
		A	D400 - D7FF
		B	DC00 - DFFF
		C	E400 - E7FF
		D	EC00 - EFFF
		E	F400 - F7FF
24 X 64 Zeichen:	3A00 - 3FFF	0	B200 - B7FF
		1	BA00 - BFFF
Beispiel für einige Zeilen:		2	9200 - 97FF
3A00 - 3A3F <-	8200 - 823F	3	9A00 - 9FFF
3A40 - 3A7F <-	8240 - 827F	4	A200 - A7FF
.		5	AA00 - AFFF
usw.		6	B200 - B7FF
.		7	BA00 - BFFF
3F80 - 3FBF <-	CF80 - CFBF	8	C200 - C7FF
3FC0 - 3FFF <-	CFC0 - CFFF	9	CA00 - CFFF
		A	D200 - D7FF
32 X 64 Zeichen:	3800 - 3FFF	0	8000 - 87FF
		1	8800 - 8FFF
Beispiel für einige Zeilen:		2	9000 - 97FF
3800 - 383F <-	8000 - 803F	3	9800 - 9FFF
3840 - 387F <-	8040 - 807F	4	A000 - A7FF
.		5	AB00 - AFFF
usw.		6	B000 - B7FF
.		7	BB00 - BFFF
3F80 - 3FBF <-	FF80 - FFBF	8	C000 - C7FF
3FC0 - 3FFF <-	FFC0 - FFFF	9	CB00 - CFFF
		A	D000 - D7FF
		B	DB00 - DFFF
		C	E000 - E7FF
		D	EB00 - EFFF
		E	F000 - F7FF
		F	FB00 - FFFF



Korrespondierende Graphik- und Bildschirmadressen des Genie III s bei unterschiedlichen Bildschirmformaten

**Irren ist menschlich,
aber wenn man
richtigen Mist
bauen will,
braucht man einen
Computer.**

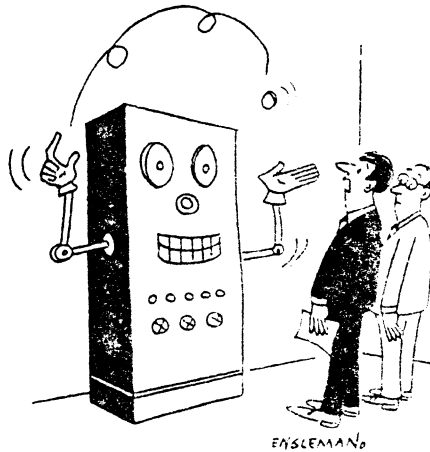


**Um die Wahrheit
wahrscheinlicher zu ma-
chen, tut man gut daran, ein
wenig Lüge hineinzumi-
schen.**

**Am liebsten esse ich
immer nur Pellkartoffeln.
Allerdings gern mit einem
Klacks Kaviar und einem
Glas Champagner dazu.**

**Die Welt ist über-
haupt nur dadurch weiter-
gekommen, daß irgend je-
mand die Courage gehabt
hat, an Dinge zu rühren, von
denen die Leute, in deren
Interesse das lag, durch
Jahrhunderte behauptet ha-
ben, daß man nicht an sie
rühren darf.**

**Wenn man Intelli-
genz als die Fähigkeit defi-
niert, neue Dinge zu lernen
und Lösungen für Proble-
me zu finden, die das erste
Mal auftauchen - wer ist
dann intelligenter als das
Kind?**



*"I don't like the way this computer
makes decisions!"*

© Creative Computing

Eine sehr erfreuliche Eigenschaft des 63c ist es, daß sein Zeichensatz sofort vorliegt und von der Systemdiskette (oder sonstwoher) geladen wird. Zur Auswahl stehen ab Werk vier verschiedene: STD (Standard deutsch), HIREC (verlängert, um beim Bildschirmformat 32 X 64 Zeichen normal auszusehen), ASCII (amerikanisch) und Tandy (so ähnlich). Die beiden Letzteren haben für die Praxis hierzulande keine nennenswerte Bedeutung. Also ärgerte ich mich nur darüber, daß STD und HIREC ästhetisch und praktisch ihre Nachteile hatten, die jetzt behoben sind:

Die Codes 01-1A waren nur eine Wiederholung des Großbuchstaben-Alphabets 41-5A. Nun sind sie unterstrichen. Man kann damit nicht nur unterstrichene Texte ausgeben, sondern auch etwa unter CP/M einen Code wie CTRL-C als zweifelsfrei unterscheidbares Zeichen ausgeben lassen.

Überflüssigerweise gab es zwei Cursorzeichen (1F und 5F), obwohl beim 63c der Cursor vom Video-Controller hard erzeugt wird. Sie flogen raus und sind jetzt das Grad-Zeichen (1F) und der griechische Buchstabe x (5F).

Die übrigen Änderungen, von denen es noch ungefähr zwanzig gibt, haben nur rein geschmackliche Bedeutung. So sind z. B. die unmögliche Form der großen Umlaute, des p und des q mit viel zu kleinen Bäuchen usw. gestreamlined. Im Zeichensatz HIREC, der noch phantasieloser als STD war, sind die Modifikationen etwas zahlreicher.

In den Dumps im Anschluß an diesen Artikel sind die neuen Zeichensätze dargestellt (zunächst STD, dann HIREC). In der ersten Spalte steht der ASCII-Code, danach die einzelnen Bytes für die Matrixzeilen des Zeichens, am Ende ist das Zeichen selbst mit Hilfe der Drucker-DRG ausgeprintet. Es liegt auf der Seite, weil mein Drucker nun mal nur neun Nadeln hat. Zwischen je 16 Zeilen ist eine Leerzeile, um die Grenzen der Disketten zu markieren.

Die neuen Zeichen unterscheiden sich von den alten mitunter nur durch ein Bit. Man muß deshalb schon sehr genau hinsehen, um die Unterschiede festzustellen. Wer die Modifikationen durchführen will, hat beim Abtippen reichlich zu tun. Deshalb möchte ich anbieten, den Interessenten die fertigen Zeichensätze auf Diskette zuzuschicken. Sendet mir hierzu bitte einen selbstadressierten Freiumschlag mit einer Leerdiskette (oder formatiert a la G-DOE 2.4).

Arnulf Sepp

00 : 00 3C 42 B9 BE BE B9 42 3C 00 00 00 00 00 00 00
01 : 00 00 0E 14 22 22 22 3E 22 22 00 FF 00 00 00 00 00
02 : 00 00 1E 22 22 1E 22 22 1E 00 FF 00 00 00 00 00
03 : 00 00 1C 22 02 02 02 22 1C 00 FF 00 00 00 00 00
04 : 00 00 1E 22 22 22 22 22 1E 00 FF 00 00 00 00 00
05 : 00 00 3E 02 02 0E 02 02 3E 00 FF 00 00 00 00 00
06 : 00 00 3E 02 02 0E 02 02 02 00 FF 00 00 00 00 00
07 : 00 00 1C 22 02 3A 22 22 1C 00 FF 00 00 00 00 00
08 : 00 00 22 22 22 3E 22 22 22 00 FF 00 00 00 00 00
09 : 00 00 1C 08 08 08 08 08 1C 00 FF 00 00 00 00 00
0A : 00 00 3C 20 20 20 20 22 1C 00 FF 00 00 00 00 00
0B : 00 00 22 12 0A 06 0A 12 22 00 FF 00 00 00 00 00
0C : 00 00 02 02 02 02 02 3E 00 FF 00 00 00 00 00
0D : 00 00 22 36 2A 2A 22 22 22 00 FF 00 00 00 00 00
0E : 00 00 22 22 26 2A 32 22 22 00 FF 00 00 00 00 00
0F : 00 00 1C 22 22 22 22 22 1C 00 FF 00 00 00 00 00

10 : 00 00 1E 22 22 1E 02 02 02 00 FF 00 00 00 00 00
11 : 00 00 1C 22 22 22 2A 12 2C 00 FF 00 00 00 00 00
12 : 00 00 1E 22 22 1E 0A 12 22 00 FF 00 00 00 00 00
13 : 00 00 1C 22 02 1C 20 22 1C 00 FF 00 00 00 00 00
14 : 00 00 7F 08 08 08 08 08 08 00 FF 00 00 00 00 00
15 : 00 00 22 22 22 22 22 22 1C 00 FF 00 00 00 00 00
16 : 00 00 22 22 22 22 22 14 08 00 FF 00 00 00 00 00
17 : 00 00 22 22 22 2A 2A 36 22 00 FF 00 00 00 00 00
18 : 00 00 22 22 14 08 14 22 22 00 FF 00 00 00 00 00
19 : 00 00 22 22 14 08 08 08 08 00 FF 00 00 00 00 00
1A : 00 00 3E 20 10 08 04 02 3E 00 FF 00 00 00 00 00
1B : 00 00 08 1C 2A 08 08 08 08 00 00 00 00 00 00
1C : 00 00 08 08 08 08 2A 1C 08 00 00 00 00 00 00
1D : 00 00 00 08 04 3E 04 08 00 00 00 00 00 00 00
1E : 00 00 00 08 10 3E 10 08 00 00 00 00 00 00 00
1F : 00 00 1B 24 1B 00 00 00 00 00 00 00 00 00 00

20 : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
21 : 00 00 08 08 08 08 08 00 08 00 00 00 00 00 00
22 : 00 00 14 14 14 00 00 00 00 00 00 00 00 00 00
23 : 00 00 14 14 3E 14 3E 14 14 00 00 00 00 00 00
24 : 00 00 08 3C 0A 1C 2B 1E 08 00 00 00 00 00 00
25 : 00 00 46 26 10 08 04 62 61 00 00 00 00 00 00
26 : 00 00 04 0A 0A 04 2A 1A 24 00 00 00 00 00 00
27 : 00 00 1B 1B 08 04 00 00 00 00 00 00 00 00 00
28 : 00 00 10 08 04 04 04 08 10 00 00 00 00 00 00
29 : 00 00 08 10 20 20 10 08 00 00 00 00 00 00 00
2A : 00 00 00 08 2A 1C 2A 08 00 00 00 00 00 00 00
2B : 00 00 00 08 08 7F 08 08 00 00 00 00 00 00 00
2C : 00 00 00 00 00 00 1B 1B 08 04 00 00 00 00 00
2D : 00 00 00 00 00 3E 00 00 00 00 00 00 00 00 00
2E : 00 00 00 00 00 00 0C 0C 00 00 00 00 00 00 00
2F : 00 00 40 20 10 08 04 02 01 00 00 00 00 00 00

30 : 00 00 1C 22 32 2A 26 22 1C 00 00 00 00 00 00
31 : 00 00 0E 0C 0A 08 08 08 1C 00 00 00 00 00 00
32 : 00 00 1C 22 20 10 08 04 3E 00 00 00 00 00 00
33 : 00 00 1C 22 20 18 20 22 1C 00 00 00 00 00 00
34 : 00 00 10 18 14 12 3E 10 10 00 00 00 00 00 00
35 : 00 00 3E 02 1E 20 20 22 1C 00 00 00 00 00 00
36 : 00 00 1B 04 02 1E 22 22 1C 00 00 00 00 00 00
37 : 00 00 3E 20 20 10 08 0E 08 00 00 00 00 00 00
38 : 00 00 1C 22 22 1C 22 22 1C 00 00 00 00 00 00
39 : 00 00 1C 22 22 3C 20 20 1C 00 00 00 00 00 00
3A : 00 00 00 00 1E 1B 00 1B 1B 00 00 00 00 00 00
3B : 00 00 00 00 1E 1B 00 1B 1B 08 04 00 00 00 00
3C : 00 00 10 08 04 02 04 08 10 00 00 00 00 00 00
3D : 00 00 00 00 3E 00 3E 00 00 00 00 00 00 00 00
3E : 00 00 04 0E 10 20 10 0E 04 00 00 00 00 00 00
3F : 00 00 1C 22 22 22 22 22 22 00 00 00 00 00 00

40 : 00 00 1C 22 20 2C 2A 2A 1C 00 00 00 00 00 00
41 : 00 00 0E 14 22 22 3E 22 22 00 00 00 00 00 00
42 : 00 00 1E 22 22 1E 22 22 1E 00 00 00 00 00 00
43 : 00 00 1C 22 02 02 02 22 1C 00 00 00 00 00 00
44 : 00 00 1E 22 22 22 22 22 1E 00 00 00 00 00 00
45 : 00 00 3E 02 02 0E 02 02 3E 00 00 00 00 00 00
46 : 00 00 3E 02 02 0E 02 02 02 00 00 00 00 00 00
47 : 00 00 1C 22 02 3A 22 22 1C 00 00 00 00 00 00
48 : 00 00 22 22 22 3E 22 22 22 00 00 00 00 00 00
49 : 00 00 1C 08 08 08 08 08 1C 00 00 00 00 00 00
4A : 00 00 3C 20 20 20 20 22 1C 00 00 00 00 00 00
4B : 00 00 22 12 0A 06 0A 12 22 00 00 00 00 00 00
4C : 00 00 02 02 02 02 02 3E 00 00 00 00 00 00 00
4D : 00 00 22 36 2A 2A 22 22 22 00 00 00 00 00 00
4E : 00 00 22 22 26 2A 32 22 22 00 00 00 00 00 00
4F : 00 00 1C 22 22 22 22 22 1C 00 00 00 00 00 00

50 : 00 00 1E 22 22 1E 02 02 02 00 00 00 00 00 00
51 : 00 00 1C 22 22 22 2A 12 2C 00 00 00 00 00 00
52 : 00 00 1E 22 22 1E 0A 12 22 00 00 00 00 00 00
53 : 00 00 1C 22 02 1C 20 22 1C 00 00 00 00 00 00
54 : 00 00 7F 08 08 08 08 08 08 00 00 00 00 00 00
55 : 00 00 22 22 22 22 22 22 1C 00 00 00 00 00 00
56 : 00 00 22 22 22 22 22 14 08 00 00 00 00 00 00
57 : 00 00 22 22 22 2A 2A 36 22 00 00 00 00 00 00
58 : 00 00 22 22 14 08 14 22 22 00 00 00 00 00 00
59 : 00 00 22 22 14 08 08 08 08 00 00 00 00 00 00
5A : 00 00 3E 20 10 08 04 02 3E 00 00 00 00 00 00
5B : 00 22 08 14 22 22 3E 22 22 00 00 00 00 00 00
5C : 22 00 1C 22 22 22 22 22 1C 00 00 00 00 00 00
5D : 22 00 22 22 22 22 22 22 1C 00 00 00 00 00 00
5E : 00 00 08 14 22 00 00 00 00 00 00 00 00 00 00
5F : 00 00 00 40 3E 25 24 24 24 00 00 00 00 00 00

60 : 00 1C 22 02 1C 22 1C 20 22 1C 00 00 00 00 00
61 : 00 00 00 00 1C 20 3C 22 3C 00 00 00 00 00 00
62 : 00 00 02 02 1A 26 22 26 1A 00 00 00 00 00 00
63 : 00 00 00 00 1C 22 02 22 1C 00 00 00 00 00 00
64 : 00 00 20 20 2C 32 22 32 2C 00 00 00 00 00 00
65 : 00 00 00 00 1C 22 3E 02 1C 00 00 00 00 00 00
66 : 00 00 1B 24 04 0E 04 04 04 00 00 00 00 00 00
67 : 00 00 00 00 1C 22 22 22 3C 20 1E 00 00 00 00
68 : 00 00 02 02 1A 26 22 22 22 00 00 00 00 00 00
69 : 00 00 0C 00 0C 08 08 08 1C 00 00 00 00 00 00
6A : 00 00 30 00 3B 20 20 20 20 22 1C 00 00 00 00
6B : 00 00 02 02 22 12 0E 12 22 00 00 00 00 00 00
6C : 00 00 0C 08 08 08 08 1C 00 00 00 00 00 00 00
6D : 00 00 00 00 16 2A 2A 2A 2A 00 00 00 00 00 00
6E : 00 00 00 00 1A 26 22 22 22 00 00 00 00 00 00
6F : 00 00 00 00 1C 22 22 22 1C 00 00 00 00 00 00

70 : 00 00 00 00 1A 26 22 26 1A 02 02 00 00 00 00
71 : 00 00 00 00 2C 32 22 32 2C 20 20 00 00 00 00
72 : 00 00 00 00 1A 26 02 02 02 00 00 00 00 00 00
73 : 00 00 00 00 3C 02 1C 20 1E 00 00 00 00 00 00
74 : 00 00 04 04 0E 04 04 24 1B 00 00 00 00 00 00
75 : 00 00 00 00 22 22 22 32 2C 00 00 00 00 00 00
76 : 00 00 00 00 22 22 22 14 08 00 00 00 00 00 00
77 : 00 00 00 00 22 22 2A 2A 14 00 00 00 00 00 00
78 : 00 00 00 00 22 14 08 14 22 00 00 00 00 00 00
79 : 00 00 00 00 22 22 22 3C 20 1E 00 00 00 00 00
7A : 00 00 00 00 3E 10 08 04 3E 00 00 00 00 00 00
7B : 00 00 22 00 1C 20 3C 22 3C 00 00 00 00 00 00
7C : 00 00 22 00 1C 22 22 22 1C 00 00 00 00 00 00
7D : 00 00 22 00 22 22 22 32 2C 00 00 00 00 00 00
7E : 00 00 1C 22 22 3A 42 42 32 02 02 00 00 00 00
7F : 00 00 55 AA 55 AA 55 AA 55 00 00 00 00 00 00 00

Call mit BASCOM

Vor kurzem schrieb ich ein BASIC-Programm zur rationellen Erstellung von ILF-Files und zum sicheren Kopieren von "großen" (z.B. 80/DS/DD) auf "kleine" (z.B. 40/SS/DD) Disketten. Das Programm hier in seinem genauen Ablauf darzulegen ist nicht meine Absicht. Ihr bekommt es in der Programmabibliothek bei Kajott und könnt es euch auseinanderfriemeln oder einfach nur benutzen! Es geht mir vielmehr um ein Problem, das beim Kompilieren des Programms mit dem BASIC-Compiler BASCOM von Microsoft auftrat.

BASCOM kontra ZBASIC

BASCOM erzeugt, genau wie der ZBASIC-Compiler, direkt aufrufbare /CMD-Files. Er hat gegenüber ZBASIC jedoch den Vorteil, sich in seinem Syntax sehr eng an das Disk-BASIC anzulehnen und so z. B. auch sehr komplizierte Stringoperationen übersetzen zu können. Dafür kann er, im Gegensatz zu ZBASIC, mit der Funktion CMD"S=DOSCMD" nichts anfangen.

Da mein Programm fast nur von Stringoperationen (MID\$,LEN\$ usw.) lebt, aber nur ein CMD"S=DOSCMD" enthält, wurde mir die Wahl des Compilers leichtgemacht. BASCOM musste es sein!

Aber wie schon erwähnt, BASCOM kann mit dem Befehl CMD"S=DOSCMD" nichts anfangen. Es mußte also ein anderer Weg gefunden werden, z. B. ein COPY direkt von einem kompilierten Programm aus zu starten!

Die Rettung: CALL

BASCOM unterstützt neben den Disk-BASIC-Funktionen des TRSDOS zusätzlich Befehle, die normalerweise nicht verwendbar sind. Unter anderem gibt es da den Befehl CALL Routine (Par,Par,Par).

Mit diesem Befehl lassen sich Maschinenroutinen von einem BASIC-Programm aufrufen. Im BASCOM Handbuch steht dazu folgende, frei ins deutsche übersetzte, kurze Anmerkung:

Der CALL-Befehl erlaubt es, ein Maschinensprache- oder TRS80-Fortran- Unterprogramm aufzurufen.

Das Format des CALL-Befehls lautet:

*CALL <variable name> [(argument list)]
wobei <variable name> und <argument list> vom Benutzer bestimmt werden können.*

<variable name> ist der Name des Unterprogramms, welches aufgerufen werden soll. Dieser Name muß zwischen einem und sechs Zeichen lang sein (mein BASCOM verkraftet nur zwei Zeichen!!!) und muß von LINK 80 (LBO/CMD) als "Global Symbol" erkannt werden! (<variable name> muß entweder der Name eines Unterprogramms in einem FORTRAN SUBROUTINE Statement oder ein PUBLIC Symbol in einer Maschinenspracheroutine sein.)

<argument list> ist optional und enthält die Argumente, die einer Maschinen- oder Fortran- Subroutine übergeben werden sollen.

Beispiel: 120 CALL MYROUT (I,J,K)

Das ist alles, was mein BASCOM-Handbuch zu dieser Möglichkeit der Einbindung von Maschinenroutinen in BASIC-Programme sagt. Ein praktisches Beispiel zur Nutzung des CALL-Befehls ist nirgends vorhanden und so ist diese wertvolle Eigenschaft von BASCOM praktisch nutzlos.

Gänzlich nutzlos ist er nicht, mit einem kleinen Trick kann man ihn schon benutzen. Wie man das bewerkstelligt, will ich euch kurz aufzeigen!

Das folgende, kurze BASIC-Programm soll nichts weiter tun, als den Bildschirm zu löschen, einen kurzen Text auszugeben und dann den freien Speicherplatz auf der Diskette anzuzeigen.

```
10 CLS
20 PRINT"DIESES PROGRAMM DEMONSTRIERT DEN AUFRUF DES
30 PRINT"DOS-KOMMANDOS -FREE- VON EINEM MIT BASCOM
40 PRINT"KOMPILIERTEN BASIC-PROGRAMM AUS!
50 PRINT:PRINT
60 CALL FR
70 END
```

Die folgenden Zeilen sind ein "Livemitschnitt" (Bildschirmausgaben in ein Diskfile geroutet!) eines Kompilerlaufs von BASCOM:

```
DOS> BASCOM TEST=TEST
```

```
00000 FATAL ERROR(S)
13040 BYTES FREE
```

```
DOS> LBO TEST,TEST-N-E
```

```
DATA 5200 75AA
FR* 5256
(5211 75AA)
```

Wie man sieht, tritt beim Kompilierungsvorgang keinerlei Fehler auf. Einzige "abnormale" Anzeige ist die Zeile:

```
FR* 5256
```

Hier wird angezeigt, daß an dieser Stelle im erzeugten /CMD-File der Aufruf eines, dem Linker nicht bekannten Unterprogramms steht. Wie man im untenstehenden Hexdump des erzeugten /CMD-Programms deutlich sehen kann, trägt der Linker an der angezeigten Stelle (5256H) als Adresse für die unbekannte Routine den Wert 0000H ein. Dies führt natürlich beim Lauf des Programms unvermeidbar zum Absturz!

```
TEST/CMD Drive 1 Record 0 X'0000'
```

```
0123456789ABCDEF BYTE 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
.....
...R..... <00> 01 02 00 52 00 00 00 00 00 00 00 00 00 00 00 00
.....!XR.S.R... <10> 00 00 00 00 00 21 25 52 C3 19 53 DE 52 00 00 E0
R.R.R...R.hT..X! <20> 52 00 52 07 52 00 00 11 52 CD 68 54 CD BC 5B 21
.R.gZ..X!.R.gZ.. <30> AD 52 CD 67 5A CD BC 5B 21 81 52 CD 67 5A CD BC
X!.R.gZ..X!.R.gZ <40> 5B 21 5E 52 CD 67 5A CD BC 5B 21 DB 52 CD 67 5A
..X!.R.gZ.....V. <50> CD BC 5B 21 DB 52 CD 67 5A CD 00 00 CD 8E 56 CD
.V ARKOMPILIERTE <60> 8E 56 20 61 52 4B 4F 4D 50 49 4C 49 45 52 54 45
N BASIC-PROGRAMM <70> 4E 20 42 41 53 49 43 2D 50 52 4F 47 52 41 4D 4D
AUS!).RDOS-KOMM <80> 20 41 55 53 21 29 84 52 44 4F 53 2D 4B 4F 4D 4D
ANDOS -FREE- VON <90> 41 4E 44 4F 53 20 2D 46 52 45 45 2D 20 56 4F 4E
EINEM MIT BASCO <A0> 20 45 49 4E 45 4D 20 4D 49 54 20 42 41 53 43 4F
M+.RDIESES PROGR <B0> 4D 2B B0 52 44 49 45 53 45 53 20 50 52 4F 47 52
AMM DEMONSTRIERT <C0> 41 4D 4D 20 44 45 4D 4F 4E 53 54 52 49 45 52 54
DEN AUFRUF DES. <D0> 20 44 45 4E 20 41 55 46 52 55 46 20 44 45 53 00
.R...AXXA BASLIB <E0> DE 52 00 00 00 5B 5B 5B 5B 20 42 41 53 4C 49 42
#000000 5.2 - 0 <F0> 20 23 30 30 30 30 30 30 20 35 2E 32 20 2D 20 4F
```

HEFT
12
März
1986

34

Um den CALL-Befehl doch noch nutzen zu können, bin ich auf folgenden Trick gekommen:

1. Man sucht sich im erzeugten /CMD-Programm eine vom Programm selbst nicht benutzte Stelle, die für das gewünschte Unterprogramm groß genug ist. Eine solche Stelle kann man sich selbst dadurch schaffen, daß man eine Dummyvariable mit soviel Text auffüllt, wie man für sein Unterprogramm braucht.
2. Nach dem Kompilierungsvorgang wird die Maschinenroutine dann in diese Variable gezapft und der ursprüngliche CALL nach 0000H in einen CALL auf die eingezapfte Unterroutine umgeändert.

In unserem Beispiel habe ich keine Dummyvariable sondern die Copyrightmeldung von BASCOM für die Unterroutine ausgenutzt. Das kurze Maschinenprogramm sieht folgendermaßen aus:

```

21F752 LD HL, FREE 'HL-Register auf den DOS-Befehl
CD1944 CALL 4419H 'DOSCALL aufrufen
C9 RET 'zurück zum Hauptprogramm
0000 DEFN 0000H 'nur Füllung, keine Funktion
0000 DEFN 0000H '
46524545 DEFN 'FREE' 'der auszuführende DOS-Befehl
OD DEFN ODH 'nicht vergessen: OD (CR) als
'Endzeichen!

```

Die durchzuführenden Änderungen sind im untenstehenden Hexdump unterstrichen!

TEST/CMD Drive 1 Record 0 X'0000'

```

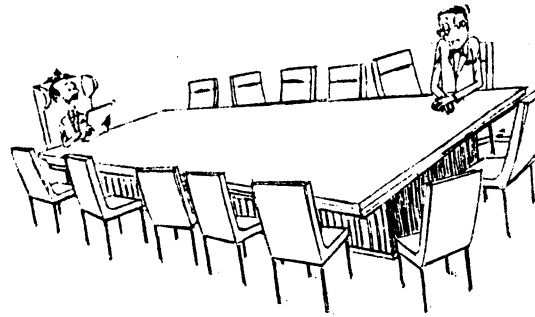
0123456789ABCDEF BYTE 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
-----
...R..... <00> 01 02 00 52 00 00 00 00 00 00 00 00 00 00 00 00
.....!ZR..S.R... <10> 00 00 00 00 00 21 25 52 C3 19 53 DE 52 00 00 E0
R.R.R...R.hT..X! <20> 52 00 52 07 52 00 00 11 52 CD 68 54 CD BC 5B 21
.R.gZ..X!.R.gZ.. <30> AD 52 CD 67 5A CD BC 5B 21 81 52 CD 67 5A CD BC
X!.R.gZ..X!.R.gZ <40> 5B 21 5E 52 CD 67 5A CD BC 5B 21 DB 52 CD 67 5A
..X!.R.gZ..R..V. <50> CD BC 5B 21 DB 52 CD 67 5A CD EC 52 CD 8E 56 CD
.V aRKOMPILIERT <60> 8E 56 20 61 52 4B 4F 4D 50 49 4C 49 45 52 54 45
N BASIC-PROGRAMM <70> 4E 20 42 41 53 49 43 2D 50 52 4F 47 52 41 4D 4D
AUS!).RDOS-KOMM <80> 20 41 55 53 21 29 84 52 44 4F 53 2D 4B 4F 4D 4D
ANDOS -FREE- VON <90> 41 4E 44 4F 53 20 2D 46 52 45 45 2D 20 56 4F 4E
EINEM MIT BASCO <A0> 20 45 49 4E 45 4D 20 4D 49 54 20 42 41 53 43 4F
M..RDIESES PROGR <B0> 4D 2B 80 52 44 49 45 53 45 53 20 50 52 4F 47 52
AMM DEMONSTRIERT <C0> 41 4D 4D 20 44 45 4D 4F 4E 53 54 52 49 45 52 54
DEN AUFRUF DES. <D0> 20 44 45 4E 20 41 55 46 52 55 46 20 44 45 53 00
.R...XXXX BASLIB <E0> DE 52 00 00 00 5B 5B 5B 5B 20 42 41 53 4C 49 42
!.R..D.....FREE. <F0> 21 F7 52 CD 19 44 C9 00 00 00 00 46 52 45 45 0D

```

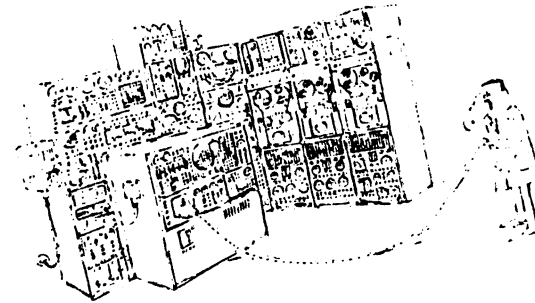
Dieses kurze Beispiel kann natürlich nur das Prinzip darlegen. Eine immer gültige Vorlage kann es schon deshalb nicht sein, da sich je nach Länge des BASIC-Quellprogramms die Dummyvariable (und auch die Copyrightmeldung) verschiebt. Man muß sich also die jeweiligen Adressen selbst suchen.

Wie ihr selbst seht, ist diese Methode natürlich reichlich umständlich und ich wäre sehr froh, wenn mir jemand eine bessere Möglichkeit aufzeigt. Am liebsten wäre mir natürlich eine Anleitung, die aufzeigt, wie man Assemblerfiles erstellt, in denen der Linker die Call's als "Global Symbol" erkennt! Sollte sich also jemand in dieser Richtung auskennen, wäre ich für eine Information sehr dankbar!

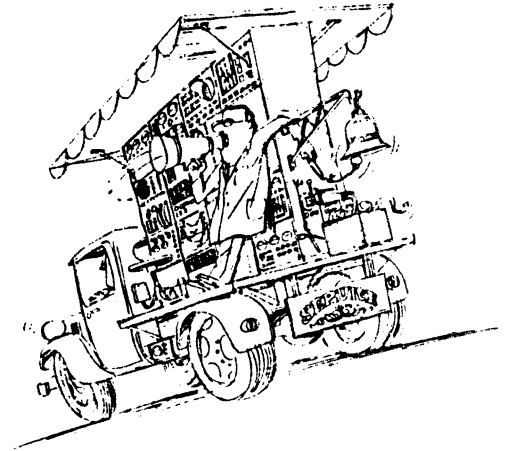
Euer
Karlmut Obermann.



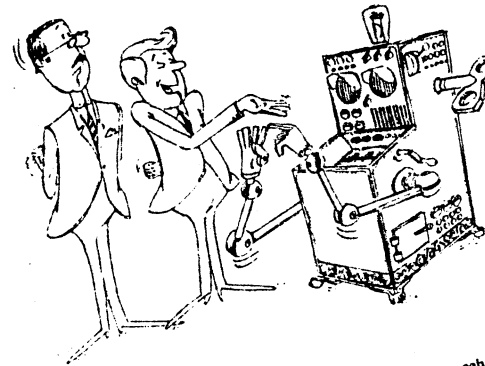
„Auf der letzten Sitzung kamen wir überein, den Betrieb zu automatisieren. Der Prozeß ist beinahe abgeschlossen ...“



„Und wenn du diesen Fehler jetzt noch einmal machst, dann schreibst du zur Strafe 5000mal: ICH BIN NUR EINE MASCHINE UND HABE KEINEN FREIEN WILLEN!“



„Daten. frische Daten!“



„Gewiß - es gibt bessere Modelle. Aber das da war sehr preisgünstig.“

Model 4 intern: Dem Interpreter echt mit geschaut

Naturngemäß wendet sich dieser Beitrag in erster Linie an Assembler-Programmierer und solche, die es werden wollen. Doch heiße Tips gibt es auch für Basic-Fans, zum Beispiel wie der List-Schutz funktioniert.

Zuerst: Viele Routinen aus dem Level2-Basic findet man auch hier wieder. Der Arithmetik-Teil wurde fast unverändert übernommen, nur kräftig gemischt und im Adreßbereich verschoben. Kleine Verbesserungen sind festzustellen, die große Schwachstelle, reiner 8080-Code, blieb. In Listing 1 sind die wichtigsten Unterprogramme aufgeführt. Wo WRA1 und WRA2 hingewandert sind, können sie aus Listing 2 ab 6664 (Hex) entnehmen. Sofern nicht anders vermerkt, sind alle Werte Hex-Zahlen.

Bei der Entwicklung von Ausdrücken benutzt Basic eine Tabelle mit Adressen einiger Arithmetik-Routinen; diese finden Sie in Listing 3. Generell ist WRA1 der am meisten benutzte Speicherplatz. Er hält immer die »laufende Variable«, das heißt, wenn Basic irgendein Ergebnis erwartet, dann muß es da stehen. Um welchen Typ es sich handelt, ist in (60BE) notiert.

Mit PEEK sollte man da ruhig einmal nachschauen; es ist nicht alles »Double-Precision, wo man es vermutet. Die trigonometrischen Funktionen werden immer in einfacher Genauigkeit ausgeführt, und FIX ist noch lange nicht Integer, SGL bleibt erhalten, es wird nur abgeschnitten.

Sollten Sie ein Sortier-Programm schreiben wollen, so vergessen Sie die Subtraktions-Routinen, die »Compares« (siehe Listing 1) sind deutlich schneller.

Der Schlüssel zu einem Interpreter ist natürlich die Adreßtabelle der Basic-Befehle. Sie wurde (Listing 4) mit einem Assembler in lesbare Form gebracht, ein Disassembler zeigt da nur sinnlosen Code. Das Prinzip ist einfach: Trifft Basic auf ein Token, subtrahiert es 81H, multipliziert das mit 2, addiert das Ergebnis auf die Anfangsadresse der Tabelle und schon ist die Adresse der Routine gefunden.

Die Tabelle hat zwei Basis-Adressen, nämlich (5757) und (57F1). Letztere gilt für Doppel-Tokens, deren erstes Byte immer FF ist. Ab 692B können Sie das (mit einem Disassembler) nachlesen, bei 68EB startet der »Execution Drivers«.

Nicht alle Tokens sind in dieser Tabelle aufgeführt. Solche, die logisch zu einem Befehl gehören, wie TO zu FOR werden dort, solche die erst später in einem Ausdruck auftreten, werden in EVALEX aufgeführt. Die Routine zur Entwicklung von Ausdrücken (EVALEX) startet bei 7000, einige weitere Interpreter-Routinen in Listing 5.

Etwas anders als im Level-2-Basic werden Variable abgelegt. Da ein Name anstatt zwei jetzt bis zu 40 signifikante Zeichen haben kann, wird eine Variable in VARTAB so eingetragen:

Byte 1: Typ (2, 3, 4 oder 8)
Byte 2: Erstes Zeichen des Namens (Großbuchstabe)
Byte 3: Zweites Zeichen des Namens (Großbuchstabe oder Ziffer)
Byte 4: Anzahl weitere Zeichen (Länge des Namens 2)
Byte 5 und folgende: die weiteren Zeichen des Namens, ASCII-Code, aber Bit 7 gesetzt.

Hiernach ist also die Namenslänge nicht auf 40 begrenzt, wohl aber die Größe des Puffers, den Basic für die Behandlung des Namens zur Verfügung stellt (ab 5E4B).

Mit Byte 4 wird erreicht, daß Basic die Variablen-tabelle schnell scannen kann, der Test eines langen Namens kostet natürlich dennoch mehr Zeit als der eines kurzen.

Alle Ein-/Ausgabe-Operationen laufen ähnlich wie in CP/M über eine Funktionsnummer mit anschließendem »CALL BDOS«, allerdings mit folgenden Unterschieden:

Die Funktionsnummer wird im Register A, eine eventuelle Unterfunktion in B übergeben. Ein auszugebendes Zeichen muß immer in C stehen, die Rückgabe erfolgt ins Register B. Gegebenenfalls sind vor dem Aufruf weitere Register zu laden, auch stehen in weiteren Registern Rückgabeparameter bereit.

Der Aufruf erfolgt immer über RST 28. Das Z-Flag bedeutet, die Routine wurde (ohne Fehler) ausgeführt, NZ heißt Fehler oder »kein Erfolg«. Wird zum Beispiel die Ta-

```
Arithmetik-Routinen
=====
Bedeutung der Kurzzeit:
WRA1: Work Area1, WRA2: Work Area 2
INT: Integer, SF: Single Precision, DP: Double Precision
(HL)...: Bytes ab Adresse in HL, etc. (DE)...
TF: Typ-Flag, Inhalt von (60BE)
BCDE: hält einen SF-Operanden als Exp. MSB, LSB, LSR

-----
2607: DF-Add. : WRA1=WRA1+0.5
2617: SF-ADD. : WRA1=WRA1*(HL)....
261C: SF-Sub. : WRA1=(HL)...-WRA1
261F: SF-SUB. : WRA1=BCDE-WRA1
2622: SF-ADD. : WRA1=BCDE+WRA1
2746: LOG : WRA1=LOG(WRA1)
278B: SF-Mult.: WRA1=BCDE*WRA1
27ED: SF-Div. : WRA1=BCDE/WRA1
28C4: SGN-Test: Nur SF: Sgn(WRA1)->A. -1=neg. +1=pos.. 0=0
28E6: ABS : WRA1=ABS(WRA1)
28F3: NEG : WRA1=-WRA1
28F8: SGN : WRA1=SGN(WRA1)
2905: SGN-Test: wie 28C4, aber alle Typen lt. TF
2917: Push WRA: PUSH WRA1 -> Stack, POP mit POP BC, POP DE
2924: Get SF : WRA1=(HL)...
2927: Get SF : WRA1=BCDE
2932: Get SF : BCDE=WRA1
2935: Get SF : BCDE=(HL)...
293E: Transfer: WRA1->(HL) 4 Bytes
2941: Transfer: (DE)...->(HL)... 4 Bytes
2945: Transfer: (HL)...->(DE)... TF Bytes
2946: Transfer: (DE)...->(HL)... TF Bytes
2949: Transfer: (DE)...->(HL)... A Bytes
294A: Transfer: (DE)...->(HL)... B Bytes
2970: Transfer: WRA1 -> WRA2 TF Bytes
Die folgenden Compare-Routinen sind schneller als Sub.
2982: Compare : SP : WRA1-BCDE Flags valid
299C: : wie vor, wenn Vorzeichen gleich
29AF: Compare : INT: HL-DE Flags valid
29C4: Compare : DP : WRA1-WRA2 Flags valid
29ED: Compare : DP : WRA2-WRA1 Flags valid
29F4: CINT : WRA1=HL=CINT(WRA1), alle Typen lt. TF
2A53: MAKINT : HL->WRA1, setzt TF=2, USR-RET
2A6A: CSNG : WRA1(DP) -> WRA1(SF)
2A87: Convert : WRA1(INT) -> WRA1(SF)
2A8A: Convert : (HL)...-> WRA1(SF)
2A96: CDBL : WRA1(alle Typen lt TF) -> WRA1(DP)
2AA9: Set TFB : TFB=8 (DP)
2AAC: Set TF4 : TF=4 (SP)
2AB0: Test TF : IF TF=3 THEN exit to "Type mismatch error"
2AE4: TM-Error: Print "type mismatch error"
2AE2: FIX : WRA1=FIX(WRA1)
2AF5: FRCINT : WRA1=HL=INT(WRA1) (SF & DP)
2B1A: : wie vor, aber nur DP
2B97: Sub. INT : WRA1=DE-HL
2B9E: Add. INT : WRA1=DE+HL
2BBE: Mult. INT : WRA1=DE*HL
2C8B: Abs(INT) : WRA1=ABS(WRA1)
2CB1: Sub. DP : WRA1=WRA1-WRA2
2C8B: Add. DP : WRA1=WRA1+WRA2
2DB5: Mult. DP : WRA1=WRA1*WRA2
2DF0: REF : Konstante 10 in DP
2EBC: Div. DP : WRA1=WRA1/WRA2 (Siehe auch Entry 2EBB)
2F16: ASCII : Convert ASCII-String (TF) -> WRA1
(HL)=Begin of String, Delimiter 00 or ","
3331: : Print "in", weiter bei 313D
313D: Print # : Print (WRA1)=(HL), i.d.R. Zeilennummer
3147: Convert : WRA1 -> ASCII, formatiert lt. Bits set in A
Bit muß auf 1, sonst unformatiert
Bit 6 5 4 3 2
. . . + trailing blanks

26D4: SQR : WRA1=SQR(WRA1)
26E0: Exp. : Basis in BCDE, Exp. in WRA1, Result->WRA1
278F: EXP : WRA1=EXP(WRA1)
2876: RND(N) : WRA1=RND(1..N)
289C: RND(0) : WRA1=RND(0..1)
die folgenden trigonometr. Funktionen ergeben immer SF
28E4: COS : WRA1=COS(WRA1)
28F0: SIN : WRA1=SIN(WRA1)
28BD: TAN : WRA1=TAN(WRA1)
29AC: ATN : WRA1=ATN(WRA1)
```

Listing 1. Die Arithmetik-Routinen des Model 4, das alte Level 2-Basic kaum verändert

Systemvariable & Buffer übersicht

5753-5806: Adressen der Basic-Routinen
5787-588A: Adressen für Anfangsbuchstaben A-Z in folgender Liste.
588B-5AC0: Basic-Worte alphabetisch sortiert.
5AC1-5AE1: Liste Operator Precedence
5AE2-5B0B: Adressen Arith-Routinen
5B0C : NOP
5B0A-5B0F: Texte der Error Meldungen
5D04-5E07: Adressen USR0-USR9 oder 6A45 (FC-Error)
5E08-5E74: Variable & Buffer
5E75-5FB4: Buffer für Tokenizing
5FB5-60B4: Input Buffer
60B5-62A4: Variable und Buffer
62A5-62A9: Message "in"
62AA-62B0: Message "ready"
62B1-62B6: Message "Break"

Die wichtigsten Variablen

5E0C: Code last error
5E0E: Druckerposition für LPDS
5E3F: Ausgabe-Flag, 00=DRT
5E11: Zeilenlänge Printer
5E12: Zeilenlänge CRT
5E17: String Data Pointer
5E19: Aktuelle Zeilennummer
5E1B: Head Pointer, Beginn Basic-Pgm
5E1D: Adresse aktueller Error Text
5E24: Pointer durch Tokenized Buffer
5E4B-5E71: Variable in VARPTR
5E4B: Pointer durch Video-Zeile (auch TAB)
60B9-60BB: RND Use
60ED: Flag locate(00)/create(FF) für Find Var Routine
60BE: Typ-Flag (2=INT, 3=STRINGS, 4=SGL, 8=DBL)
60C1: 16tste Zeilennummer vor aktueller
60CF: HIMEM-Pointer
6051: Adr. next free Entry to LSTPA
6053-60F0: Literal String Pool Area, 1083 Bytes
60F1-60F3: Dope Vector aktueller String
60F4: Adresse aktuelles Token
60F5: Aktuelle DATA-Zeile
60F6: CALL-Flag
60FE: Read/Input Flag, 00=Input, else Read
610C: Auto Flag

6104: Zeilennummer in Eingabe
610e: Auto-Inkrement
610B: Encoded Statement Pointer
610A: Stack Pointer
610C: ON ERROR-Zeile
610E: EDIT-Zeile, lfd. Zeile in LIST
6110: Text-Pointer in Error-Zeile
6112: ON ERROR Adresse
6114: RESUME-Flag (00) else FF
6115: Adresse next Token (wenn Backspacing)
6117: CONT-Zeile
6119: CONT-Adresse
611B: Tail Pointer (Ende Basic-Pgm)
611D: Pointer einfache Variable
611F: Pointer Array-Variable
6121: DATA-Pointer
6123-613C Typ-Tabelle A-Z für DEFxxx
621F: Option Base (0 oder 1)
6220: Option Base Flag (Option Base+1)
6247: Prot Flag (>0: no LIST, no Direkt Mode)
6259: Trace Flag

Work Area 1 WRA1

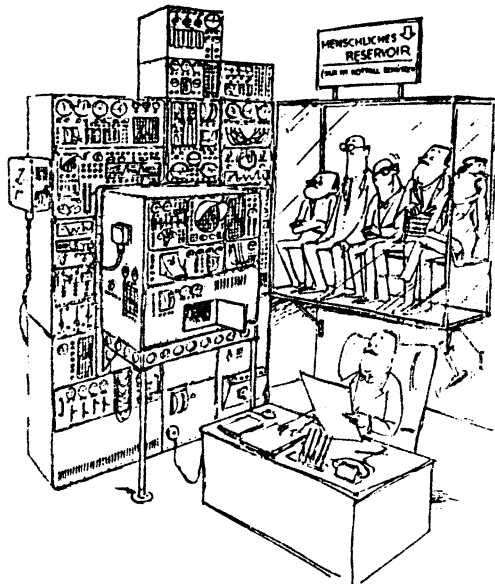
Adr.	if INT	if SGL	if DBL
6264			LSB
6265			LSB
6266			LSB
6267			LSB
6268	LSB	LSB	LSB
6269	MSB	LSB	LSB
626A		MSB	MSE
626B		EXP	EXP

Work Area 2 WRA2

6272	LSB	LSB	LSB
6273	MSB	LSB	LSB
6274		MSB	LSB
6275		EXP	LSB
6276			LSB
6277			LSB
6278			MSB
6279			EXP

627A-6294: PRINT USING Buffer

Listing 2. Systemvariable, Texte und Puffer, Adressen in der Übersicht und im einzelnen



00100 ;ADRESS-TABELLE FUER ARIHT-ROUTINEN
00110 ;=====

00120			
00130	ORG	5AE2H	
00140	BASEDP	00150	DEFW 2A96H ;BASIS-ADR. WENN DEL PRECIS.
00150		00160	DEFW 0000H ;CDBL
00160		00170	DEFW 29F4H ;NOT USED
00170		00180	DEFW 2AB0H ;CINT
00180		00190	DEFW 2A6AH ;TESTE TYP
00190		00200	DEFW 2C88H ;CSNG
00200		00210	DEFW 2C81H ;ADD (DBL)
00210		00220	DEFW 2DB5H ;SUB (DBL)
00220		00230	DEFW 2E8CH ;MULT (DBL)
00230		00240	DEFW 29EDH ;DIV. (DBL)
00240		00250	DEFW 2622H ;COMPARE (DBL)
00250	BASSGL	00260	EQU \$;BASIS ADR. SGL PRECIS.
00260		00270	DEFW 261FH ;ADD (SGL)
00270		00280	DEFW 278BH ;SUB (SGL)
00280		00290	DEFW 27EDH ;MULT (SGL)
00290		00300	DEFW 2982H ;DIV. (SGL)
00300		00310	DEFW 2B9EH ;COMPARE (SGL)
00310	BASINT	00320	EQU \$;BASIS-ADRESS INTEGERS
00320		00330	DEFW 2B93H ;ADD
00330		00340	DEFW 2BBEH ;SUB
00340		00350	DEFW 716CH ;MULT
00350		00360	DEFW 29AFH ;DIV.
00360		00370	DEFW \$;COMPARE
00370	ERROR	00380	EQU \$;BEGINN TEXTE ERROR MELDUNGEN
00380		00390	END
0000			
00000	TOTAL ERRORS		
34294	TEXT AREA BYTES LEFT		

BASEDP 5AE2 00140
BASINT 5B00 00310
BASSGL 5AF6 00250
ERROR 5B0A 00380

Listing 3. Nach dieser Adreßtabelle werden Arithmetik-Routinen aufgerufen

statur abgefragt, heißt NZ »keine Taste gedrückt«, Z dann »Taste betätigt«, der Code steht in A. In Listing 6 sind einige Routinen praktisch demonstriert. Sie sind ein Teil des DOS, das hier aus Platzgründen nicht behandelt werden kann. Nur soviel: Auch jede Disk-Operation läuft über RST 28. Sie sollten nie eine Routine direkt aufrufen. Der Sinn dieser Organisation ist es, das System beliebig erweitern (und ändern) zu können, womit sich absolute Adressen verschieben, die Funktionsnummern aber bleiben. Den Drucker hätte ich fast vergessen, Funktion 6, Zeichen in C, oder direkt über Port F8. Die RST-Calls des Level2-Basics sind hier entfallen. Das kostet zwar Speicher, macht das System aber schneller. Aus RST 8 (Syntax-

Check) wurde CALL 430E, RST 10 (Scanner) ist jetzt CALL 6943, RST 18 (HL-DE) wurde zu CALL 4308 und RST 20 (get vartyp) zu CALL 735E.

Interpreter mit Error-Meldungen knacken

Ein weiterer Weg, so einen Interpreter zu knacken, sind die Error-Meldungen, die hier auf vier Arten erzeugt werden. Meistens geschieht dies über JP in die Sprungliste (Listing 7). Mit einem Disassembler sieht man da nur Umengen von »LD BC,nn«, ein beliebiger Trick, den Microsoft häufig anwendet. Wenn Sie in der Adreßtabelle nur 3-Byte-Abstände vorfinden, geht das auf diese Art. Bei

2-Byte-Abständen steht da »OR AF«, womit A > 0 wird. Steigt man beim zweiten Byte ein (AF), heißt das »XOR A«, also A = 0. Zu den Error-Meldungen, Weg 2: »LD E,Code/JP 6343«. 6343 ist die Adresse, auf die die vorher genannte Liste durchfällt. Weg 3: ein JP zur Adresse, wo ein »Weg 2« beginnt, und Weg 4: HL wird mit der Adresse des Fehlertextes geladen und dann wird 4660 (Print Text ab (HL)) gerufen. Als Hilfe dazu Listing 8.

Störend ist oft, daß bei Syntax-Fehlern sofort EDIT aufgerufen wird, womit leider alle Variablen gelöscht werden. Wollen Sie das vermeiden, poken Sie NOPs (00) auf 63F5-63F7. Da wir gerade beim Poken sind. Ist Ihr Drucker grafikfähig, Sie wollen also die Punkte nicht im Screenprint, dann reicht ▶

SMARTSCREEN, SMARTKEY und SMARTPRINT: nützliche Utilities — oder Spielerei?

Auch den neueren Versionen des CP/M-Betriebssystems, CP/M 2.2 und sogar CP/M 3.0, sieht man noch an, daß es seine Wurzeln in der Zeit der Computer-Veteranen hat. Die Ein- und Ausgabe ist zeilenorientiert — schließlich standen damals nur Lochstreifenleser und -stanzer (daher die »reader«- und »puncher«-Routinen im BIOS) sowie Fernschreibterminals zur Verfügung. Mit zunehmender Verbreitung der Bildschirmsichtgeräte wurden diese auch für den Personal Computer interessant.

Steigende Integrationsdichte und der Verfall der Preise für integrierte Schaltungen sorgen bald dafür, daß Bildschirmsichtgeräte mit erheblicher Intelligenz ausgestattet wurden. Heute führt meist ein eigener Mikroprozessor Befehle wie »Bildschirm löschen«, »Löschen von Cursor bis Zeilenende«, »Löschen von Cursor bis Bildende«, »Einfügen einer Leerzeile«, »Ausfüllen einer Zeile«, und so weiter aus. Damit waren die Voraussetzungen geschaffen für bildschirmorientierte Editoren, Datenbanksysteme mit Bildschirmmasken und nicht zu vergessen die Computerspiele sowie viele andere Anwendungen. Doch damit tauchte bereits ein neues Problem auf, denn die Hersteller (der Terminals) hatten sich, wie so oft in der Vergangenheit, nicht auf einen Standard geeinigt. Daher können verschiedene Terminals nicht nur verschiedene Funktionen ausführen, oftmals verwenden verschiedene Terminals für eine identische Funktion verschiedene Steuercodes.

Beispiel: Direkte Cursoradressierung.
Steuercode Terminal
(ESC)=(Zeile)(Spalte) Lear Siegler
ADM 3/5/31,
Soroc IQ
120,
Televideo
912-950,
NEC, Osborne
I.

(ESC)Y(Zeile)(Spalte) Heath/Zenith
19, DEC VT
52, ADDS,
IBM 3101,
Intertube,
Superbrain,
Telaray
(DC4)(Zeile)(Spalte) ACT IV/V
(RS)(Zeile)(Spalte) Datamedia
(ESC)(Zeile)(Spalte)(Dynabyte 57
(ESC)&a(Zeile) HP 2621
(Spalte)C
(ESC)I(Spalte)(Zeile) Info-10
(ESC)Y(Zeile)(ESC)X Perkin Elmer
(Spalte)

Tabelle 1. Beispiele der Cursorsteuerung verschiedener Terminals

Bildschirmorientiert arbeitende Programme müssen daher in der Regel (meist vom Anwender) an das eigene Terminal angepaßt werden. Um diesem Umstand weitestgehend abzuhelfen, wurde von FBN Software das Programm SMARTSCREEN entwickelt. Dieses Programm läßt sich direkt unter den CP/M Console Command Processor (CCP) und verändert die Bios-Sprungtabelle so, daß bei der Ausgabe über die Console (Bildschirm) zunächst die jeweiligen Übersetzungsroutinen in SMARTSCREEN angesprochen werden, die dann den ADM-3A-Code in den Terminalspezifischen Code übersetzen. Damit sieht es für ein Anwenderprogramm so aus, als würde ein ADM-3A-Terminal ange-

CP/M 2.2 Adresse	
FFFFH	BIOS
FA00H	BDOS
EC00H	CCP
E400H	SMARTSCREEN
E100H	jp bdos
	T P A
0100H	
0005H jp SMARTSCREEN	
0000H jp bios	
0000H	

Bild 1. Speicherverteilung bei einem 64k-CP/M-System mit SMARTSCREEN:

steuert. Dadurch, daß SMARTSCREEN direkt unter dem CCP liegt und auch den BDOS-Sprung bei Adresse 0005H verändert, bleibt SMARTSCREEN auch beim Warm-Boot im Speicher aktiv und muß daher nicht vor jedem Programm neu geladen werden. Das hat jedoch zur Folge, daß der Speicherplatz, der vom CCP und von SMARTSCREEN belegt wird, nicht dem Anwenderprogramm zur Verfügung steht. Der Verlust beträgt 2,75 KByte Speicherplatz.

Das »virtuelle« ADM-3A-Terminal erkennt folgende Steuercodes:
IK, (VT), 0BH: Cursor 1 Pos. nach oben
IJ, (LF), 0AH: Cursor 1 Pos. nach unten
IH, (BS), 08H: Cursor 1 Pos. nach links
IL, (FF), 0CH: Cursor 1 Pos. nach rechts
II, (RS), 1EH: Cursor in Zeile 1, Spalte 1
IZ, (SUB), 1AH: Cursor in Zeile 1, Spalte 1 und Bildschirm löschen
IM, (CR), 0DH: Cursor an den Zeilenanfang
IG, (BEL), 07H: Bell (Klingel)

Zur Cursor-Adressierung benutzt das ADM-3A-Terminal folgende Sequenz: (ESC)= in HEX: 1BH, 3DH, gefolgt von der Zeilen- und Spaltenadresse. Die Adresse ist in HEX, begonnen wird in Zeile 1, Spalte 1 mit 20H, 20H. Zeile 1, Spalte 2 wäre also 20H, 21H.

Wie nicht anders zu erwarten, muß man SMARTSCREEN vor der

ersten Benutzung an das eigene Terminal anpassen. Das geschieht mit dem mitgelieferten Programm SMSPATCH. Für eine Anzahl von Terminals kann SMSPATCH die notwendigen Änderungen in SMARTSCREEN selbst durchführen. Dies sind:
Microterm ACT IV/ACT V, DEC VT-52, Heath/Zenith H19/Z19, ANSI X3 standard, ADDS Viewpoint, Hazeltine 1500, HP 2621, TRS-80 (P&T), IBM 3101, NEC, Vector Graphic und Superbrain.

Für alle anderen Terminals ist eine spezielle Anpassung notwendig, die aber mit dem SMSPATCH keine Schwierigkeiten macht. SMSPATCH fragt nach der maximalen Spalten- und Zeilenanzahl.

Bei der Cursoradressierung ist beispielsweise folgende Form vorgegeben: (string1) (Adress1) (string2) (Adresse2) (string3), wobei jeder string aus einer Folge von 0 bis 9 Zeichen bestehen kann. Es kann wahlweise als erste Adresse die Zeilen- oder Spaltenadresse gesendet werden. Der Zeilen- und Spaltenoffset kann ebenfalls getrennt eingegeben werden. Beispiel für eine selektierte Cursor-Adressierungs-Sequenz: (ESC)Y (Spalte + 20H)(Zeile + 20H).

In entsprechender Weise werden die notwendigen Steuerzeichen für Bell (Klingel), Cursor 1 Pos. nach links, Cursor 1 Pos. nach unten, Cursor 1 Pos. nach oben, Cursor 1 Pos. nach rechts, Cursor an Zeilenanfang, Cursor Home und Bildschirm löschen eingegeben. Die Eingaben können dezimal, hexadezimal oder direkt, als Tastendruck, erfolgen.

Da nach Aktivieren von SMARTSCREEN das Programm nicht nur sich selbst, sondern auch den CCP im Speicher vor Überschreiben schützt, ist es nicht mehr notwendig, beim Warmboot den CCP von der Diskette zu laden. Auch das kann man bei der Programm Anpassung durch SMSPATCH vorgeben. Bei einigen Computern wird es sogar notwendig sein, ein Nachladen des Systems zu unterbinden. Wird nämlich beim Warmboot auch das BIOS neu geladen (wie beim Superbrain), bleibt SMARTSCREEN ohne diese Vorsichtsmaßnahme natürlich nicht im Speicher.

Nachdem SMARTSCREEN angepaßt ist, kann man die Anpassung mit dem ebenfalls mitgelieferten Programm SMSTEST testen. SMSTEST erstellt mit Hilfe der ADM-3A Steuercodes Testbilder,

die mit den schriftlichen Beschreibungen beziehungsweise mit den Abbildungen im Manual übereinstimmen müssen. Hier fehlt allerdings der Hinweis darauf, daß das Testprogramm nur für ein Terminal mit 80 Zeichen pro Zeile und 24 Zeilen geeignet ist. Bei anderen Bildschirmformaten sind die Angaben nicht mehr korrekt. Bei der Benutzung arbeitet SMARTSCREEN so, wie man es von ihm erwartet. Die für ADM-3 adaptierte Version eines bildschirmorientierten Editor funktionierte einwandfrei. Einen Nachteil sollte man aber nicht verschweigen: Das ADM-3-Terminal kennt nur wenige und nicht sehr leistungsfähige Kommandos. Die meisten neueren Terminals können bereits selbständig Zeilen ein- und ausfüllen und andere leistungsfähigere Befehle ausführen. Die meisten Programme (wie Wordstar) kennen diese Befehle ebenfalls und können sie mitbenutzen. Der Geschwindigkeitsvorteil, gerade bei seriell angeschlossenen Terminals ist beträchtlich. Die meisten kommerziellen Programme lassen sich außerdem auch an eine große Anzahl weiterer Terminals anpassen, notfalls sogar im Dialog, Kommando für Kommando.

Auch Anpassung an Tastatur und Drucker möglich

Wer diesen (geringen) Aufwand scheut, eine größere Anzahl bildschirmorientierter Programme hat, oder wer gar Programme hat oder benötigt, die sich nicht anpassen lassen und die für das ADM-3A-Terminal ausgelegt sind, für den kann und wird sich dieses Programm lohnen.

Einen ähnlichen Zweck wie SMARTSCREEN verfolgen auch die Programme SMARTKEY und SMARTPRINT. Der Unterschied ist, daß sie nicht die Funktion eines »virtuellen« Terminals, sondern die einer »virtuellen« Tastatur beziehungsweise eines »virtuellen« Druckers haben. Wer hat nicht eine Tastatur, auf der es einige von CP/M nicht, oder falsch genutzte Funktionen gibt? Oder eine Tastatur, die keinen Standard-ASCII-Code ausgibt? Diese Probleme kann (in den meisten Fällen) SMARTKEY lösen. SMARTKEY funktioniert ähnlich wie SMARTSCREEN. Einmal aktiviert, liegt es unterhalb des CCP und fängt die Eingaben über die Tastatur ab, übersetzt sie so, wie vom Benutzer gewünscht und leitet sie weiter. Hiermit lassen sich

nicht nur Tasten umdefinieren (man denke nur an die Verwechslung von »Z« und »Y« bei vielen Tastaturen), oder allen Tasten neue Codes zuordnen, sondern es lassen sich einer Taste ganze Strings, wie zum Beispiel »DIR(CR)«, zuordnen. Ist das Programm erst einmal aktiv, können die Zuordnungen ständig verändert werden. Das geschieht durch Eingabe eines bestimmten Control-characters (normalerweise (ESC), läßt sich aber verändern). Damit gelangt man in den SMARTKEY-Input-Modus. Nun betätigt man die Taste, die umdefiniert werden soll und gibt anschließend den zuzuordnenden String ein, abgeschlossen mit einem weiteren (ESC). Die neue Zuordnung wird in einer Tabelle innerhalb des SMARTKEY-Programmes gespeichert und wird ab sofort berücksichtigt.

Beispiel: »ID« soll durch »DIR(CR)« ersetzt werden.

Eingabe: (ESC)ID DIR(CR)(ESC) (Das Blank nach dem ersten D wird automatisch von SMARTKEY erzeugt) Wird hiernach ID betätigt, so erscheint nicht ID, sondern DIR auf dem Bildschirm und der Befehl wird durch das folgende (CR) ausgeführt. Um diese Zuordnung wieder rückgängig zu machen, muß man nur (ESC)ID (ESC) eingeben, dann wird wieder der Originalcode ausgegeben.

Die weitergehende Verwaltung der Zuordnungstabellen und des Programmes SMARTKEY geschieht durch das Programm FIXKEY. Ist SMARTKEY aktiv und man startet FIXKEY so erscheint folgendes Menü:

```
COMMAND MENU
0 Exit to CP/M
1 Pack and Save current definitions
2 Load a definition file
3 List contents of a definition file
4 List current definitions
5 Clear current definitions
6 Alter Escape character
7 Terminate SMARTKEY
```

Dadurch ist es auch möglich, die Zuordnungstabellen auf Diskette abzuspeichern und später direkt als Tabelle zu laden. So kann man mehrere Übersetzungstabellen abspeichern und jeweils die benötigte laden.

SMARTPRINT hat eine entsprechende Übersetzerfunktion für die Ausgabe über die »List«-BIOS-Routine. Hier hat man ebenfalls die Möglichkeit, Drucker, die nicht ASCII-Code-kompatibel (zum Beispiel alte Fernschreib-Druckwerke) sind, entsprechend anzupassen. Die Generierung der Zuord-

nungstabellen geschieht genauso wie bei SMARTKEY. Man kann SMARTPRINT auch dazu benutzen, vom Drucker nicht druckbare Zeichen in äquivalente umzusetzen. Kann ein Drucker beispielsweise die Zeichen »*« und »*« nicht drucken, so kann man sie mit SMARTPRINT durch »LT.« beziehungsweise »GT.« ersetzen lassen. Die Verwaltung der Zuordnungstabellen und des Programms erfolgt ebenfalls durch FIXKEY. Man kann also auch hier die Zuordnungsta-

bellens permanent auf Diskette speichern.

Die Handhabung dieser Programme wird erst dann kompliziert, wenn man mehrere dieser Utilities gleichzeitig benutzen muß. Beispielsweise SMARTSCREEN und SMARTKEY. Dann ist es wichtig, das Programm, welches länger aktiv bleiben soll, zuerst zu starten.

Bleibt noch anzumerken, daß SMARTSCREEN, SMARTKEY und SMARTPRINT zur Zeit nicht unter

CP/M 3.0 und unter Microshell lauffähig sind.

Die in Deutschland ausgelieferten Handbücher sind in englischer Sprache und dem Programm durchaus angemessen. Nachteilig fiel nur auf, daß die Handbücher nicht auf einem Standard-ASCII-Drucker erstellt wurden. Dadurch sind an Stelle der eckigen Klammern beispielsweise deutsche Umlaute abgedruckt, was manchmal etwas verwirrt.

(Andreas Walter)

POWER — ein Nachbrenner für CP/M

Eigentlich wäre dieser Ausdruck falsch, denn dieses Programm ist in Richtung Benutzer dem CP/M vorgeschaltet, aber was die verstärkende Wirkung des Nachbrenners betrifft, ist er richtig. Die Sammlung von Dienstroutinen übernimmt die Programmsteuerung durch ein nummeriertes Directory. Sie eignet sich für alle »täglichen« Gegebenheiten. Die — laut Handbuch — 55 Befehle werden hier zusammenfassend besprochen.

Die zuvor angeführten Gegebenheiten kann man in fünf Gruppen einteilen:

1. Datei- und Directoryaktivitäten
 2. Diskmanipulationen und Reparaturen
 3. Programmschutz
 4. Debug-Arbeiten
 5. Diskentestest
- Zum Punkt Datei und Directoryaktivitäten werden folgende Funktionen ausgeführt:
- a) Umbenennen
 - b) Löschen
 - c) Wiederherstellen dieser gelöschten Dateien
 - d) Größe von Dateien feststellen
 - e) Vergleichen
 - f) Kopieren zwischen verschiedenen Laufwerken
 - g) Kopieren zwischen Usernummern
 - h) Schreibschutz setzen für Dateien
 - i) Aufheben des Schreibschutzes
 - j) Dateinamen als systemeigen erklären (nicht sichtbar mit »DIR«)
 - k) Systemprogrammschutz aufheben
 - l) Starten von Programmen
 - m) Starten von bestimmten Speicherbereichen
 - n) Einloggen von Disketten (Reset)
- Zu Punkt 2 — Diskmanipulationen und Reparaturen:
- a) Laden von Dateien
 - b) Wegschreiben von Dateien
 - c) Laufwerkstatistiken
 - d) Laufwerkstest
 - e) CP/M-Gruppen listen für eine Datei

f) Gruppendaten auf dem Bildschirm anzeigen (in ASCII und HEX)

g) Gruppendaten in den Speicher lesen und auf die Disk schreiben

h) Sektor- und Spurdaten auf dem Bildschirm anzeigen (in ASCII und HEX)

i) Sektor- und Spurdaten in den Speicher lesen und auf die Disk schreiben

j) Programme und Daten von der Disk lesen und auf dem Bildschirm anzeigen (in ASCII und HEX)

k) Markieren von Dateinamen zum Beispiel zum gemeinsamen Kopieren (ohne Umbenennung)

l) Sortieren des »DIRECTORIES« nach bestimmten Kriterien

Zu Punkt 3 — Dateischutz:

- a) Bestimmen von Dateien, die nur mit Schlüsselworten bearbeitet werden können
- b) Dateien als »nur zu lesen« kennzeichnen
- c) Lesen und Schreiben erlauben

Zu Punkt 4 — Debug-Arbeiten:

- a) Speicherinhalt anzeigen und einsetzen (in HEX, ASCII, BINÄR)
- b) Speicherbereiche vergleichen
- c) Speicher »dumpen«
- d) Speicher mit einem bestimmten Byte füllen
- e) Speicherzonen verschieben
- f) Bytemuster im Speicher suchen
- g) Ausführen von im Speicher befindlichen Maschinencoderroutinen

Zu Punkt 5 — Diskentestest:

- a) Testen der Diskette auf fehler-

hafte Sektoren, Reparieren dieser Stelle(n) oder Ablegen in einer Sammeldatei, um sie von der Benutzung auszuschließen

Zu diesen Punkten kommen noch zwei weitere hinzu. So gibt es für den Umgang mit POWER noch

- a) HELP (Ausdruck von POWER-Befehlen)
- b) EXIT (Rückkehr zu CP/M)
- c) LOG (Programmparameter)
- d) SPEED (Geschwindigkeit, mit der Daten auf dem Bildschirm »geblättert« werden)

Vier weitere (USER1 bis USER4) sind vorbereitet, aber noch nicht belegt. Hier können Benutzerwünsche verwirklicht werden.

Der Umfang der obigen Sammlung zeigt, daß sich hier mehrere Programme, die sonst einzeln aufgerufen werden mußten, alle unter einem »Hut« befinden. Dies erleichtert beträchtlich die Arbeit. Man muß sich nur auf die Einarbeitung in ein Programm konzentrieren. So gibt es zum Beispiel für die »wilden Karten« unter CP/M eine vereinfachte Sequenz, statt ** ist es möglich, nur * einzugeben, was eine Erleichterung gegenüber der Eingabe »Shift Stern-Punkt-Shift-Stern« darstellt. Einige Befehle sind auch schon durch CP/M und dessen Dienstprogramme bekannt, jedoch wurden sie aus dem Verband (STAT — hier die Befehle DISK, SETDIR, SETSYS, SETRO, SETWR, SIZE, STAT oder PIP — COPY) herausgelöst und einzeln verwendet.

Den großen Pluspunkt stellt jedoch ein Menüsystem dar, daß das Arbeiten mit diesem Programm zu einer wahren Freude für eilige Benutzer macht. Anstelle von achtstelligen Programmnamen wird die Eingabe von nur ein, zwei oder höchstens drei Ziffern benötigt. So wird zum Beispiel ein einfaches, selektiertes Kopieren möglich, ohne wie bereits gesagt, lange Buchstabenkolonnen eintippen zu müssen, das gleiche gilt für den Löschvorgang, das Ausdrucken von Dateien auf dem Bildschirm, wobei einzelne Dateien nur angedrückt werden können und nach Eingabe eines 1 Ks (dieses Zeichen kann auf ein gewünschtes geändert werden), die nächste aufgerufen wird. Es gilt weiterhin für den Vergleich mehrerer Dateien, Neubenennung von Dateien, Kennzeichnen von Bits in Dateinamen zum selektierten Kopieren, Schreibschutz von Dateien, Ausschließen von der Darstellung von Dateinamen beim »DIR«-Befehl, Größenfeststellung von Dateien, Transfer zwischen verschiedenen Benutzernummern. In Bild 1 sehen Sie ein Beispiel dieser Menüarchitektur.

Der Programmname XDIR.COM ist mit einem Stern versehen. Dies kennzeichnet eine »nur lese«-Datei. Nun noch ein Beispiel für eine Übertragung von einem Benutzerbereich in einen anderen. Mit dem Befehl XUSER wird der Transfer in einem anderen Bereich eingeleitet (Bild 2).

A) POWER

POWER 3.03 on CP/M 2.2

Copyright (c) 1981, 1982 by PAVEL BREDER
All rights reserved. 01/03/83

AD=DIR

AD: 1= POWER .COM | 2= RESOURCE.COM | 3= SDIR .COM
AD: 4= SURVEY .COM | 5= SWEEP37 .COM | 6= SPOOL .COM
AD: 5= XDIR .COM*

AD=XUSER 3

AD-3=COPY

AD: 1= POWER .COM | 2= RESOURCE.COM | 3= SDIR .COM
AD: 4= SURVEY .COM | 5= SWEEP37 .COM | 6= SPOOL .COM
AD: 5= XDIR .COM*

select? 6

destination drive : A

(Copy or Move): M

AD: A: SPOOL .COM

AD: DIR [U]

AD: 1= POWER .COM | 2= RESOURCE.COM | 3= SDIR .COM
AD: 4= SURVEY .COM | 5= SWEEP37 .COM | 6= SPOOL .COM
AD: 7= XDIR .COM*
AD: 1= SPOOL .COM

Bild 1. Beispiel für die Menüarchitektur

In dem letzten Beispiel wird noch einmal die Frage gestellt, ob kopiert oder transferiert werden soll. Der Unterschied besteht darin, daß "M" nur den Namen in das jeweilige »DIRECTORY« einträgt, die Datei aber nicht kopiert wird. Bei einer Kopie würde die gesamte Datei auf der Diskette dupliziert werden! Das U hinter »DIR« ist der Code für die Darstellung aller Benutzerbereichseinträge in das »DIRECTORY«. Durch Hinzufügen eines X können auch die Benutzerbereiche auf anderen, bis dahin angesprochenen Disketten-Laufwerken angezeigt werden. Da es für die meisten Befehle Schlüssel dieser Art gibt, ergeben sich mehr als »nur« die oben erwähnten 55 Befehle. Zusätzlich zu diesen gibt es noch eine Datei, die die Programmsteuerung übernimmt. Der Inhalt kann mit einem »LOG«-Befehl sichtbar gemacht werden. Die in dieser Datei vorhandenen Parameter kann man immer ändern, wenn das Power-Zeichen vorhanden ist (AO=):

— so kann hier bestimmt werden, ob beim Kopieren die auf dem Ziellaufwerk vorhandene Datei auf den Typ .BAK umbenannt, überschrieben oder übersprungen werden soll oder die Abfrage der genannten Möglichkeiten in dem Fall selbst durchgeführt werden soll — in wieviele Spalten seitenweise umgeblättert wird — ob Original und Backup nach dem Kopieren markiert werden sollen

— kann der Autostart von Power nach Lauf eines (T)ransient (P)rogramm (A)rea Programms an und ausgeschaltet werden

— es können bestimmte Attribute bei Systemprogrammen (R/W, R/O, SYS, DIR) gesetzt werden.

Etlche weitere Hinweise zum »Zuschneiden« des Programms auf persönliche Wünsche findet man in dem recht guten und umfangreichen (etwa 123 Seiten) Handbuch (Kapitel Anpassung = Customization), das mit dem Programm geliefert wird. Dort findet man auch die Erklärung eines Hilfsprogramms, das im Lieferumfang von Power enthalten ist — PWRETOOL. Mit Hilfe dieses Programms ist es möglich, die Befehle umbenennen, sie als Funktion zu streichen oder sie mit einem Schlüsselwort zu belegen. Zum Beispiel ERA wird in LÖSCHEN umbenannt und mit einem Schlüsselwort versehen. Nun kann nur derjenige ein Programm oder eine Datei löschen, der im Besitz dieses Wortes ist.

Besonders viel Interesse dürfte die genaue Erklärung, wie man eine zerstörte Diskette (BAD SECTOR Meldung) wieder brauchbar macht, finden. Auch Tips, wie man Daten nach einem Systemcrash retten, Dateien für ein schnelleres Laden neu organisiert und wie man Programme modifizieren kann, findet man im Anhang.

Dieses Programm lief auf einem CP/M-System Version 2.2 einwandfrei, auf einem System mit Version 1.41 (Icom-Floppy-Disk-System) erfolgte die Fehlermeldung, daß eine spätere CP/M-Version vonnöten sei. Das Handbuch ist leider in Englisch, das einigen nicht so perfekt »Computerenglisch« sprechenden Lesern einiges Kopfzerbrechen bereiten wird. Störend wirkt sich auch das Fehlen einer Bildschirmsteuerung aus. So werden zum Teil gleiche Informationen über den Bildschirm geschoben, anstatt nur die sich ändernden Daten neu abzubilden.

Dieses umfangreiche Programm ist leicht bedienbar. Es eignet sich daher (bis auf Einschränkungen — siehe vorhergehenden Abschnitt) für den kommerziellen Einsatz, auf den es gut zugeschnitten werden kann. Aber auch der Hobbyist, dem es in erster Linie auf Programmabarbeitung und weniger um Betriebsysteminteresse geht, ist mit dieser Vielfalt von Hilfsprogrammen sicher gut bedient.

(Axel Pohl)

DISK DOCTOR

Es kommt alle Nase lang vor, daß man irrtümlich die falsche Datei löscht. »ERA.BAK« ist ein Beispiel, das wohl jeder CP/M-Benutzer kennt.*

Bemerkt man den Irrtum sofort nach Ausführung des Befehls, dann kann Disk Doctor das Inhaltsverzeichnis (Directory) der Diskette so verändern, daß die Datei dem Benutzer wieder vollständig zugänglich gemacht wird. Es sollte jedoch bereits an dieser Stelle bemerkt werden, daß durch den Gebrauch von Disk Doctor unter Umständen mehr kaputt gemacht als repariert wird.

Disk Doctor bietet auch für den Fall Hilfestellung, daß man das Löschen erst später bemerkt. In diesem Fall ist es jedoch nicht immer möglich, die ganze Datei zu retten. (Die Gründe hierfür liegen beim CP/M-Betriebssystem. Eine Erklärung würde an dieser Stelle zu weit führen. Man kann sie im Disk-Doctor-Handbuch nachschlagen.) In Disk Doctor ist ein »DIR«-ähnlicher Befehl vorhanden, mit dem auch gelöschte Dateien angezeigt werden können. Anhand dieser Liste kann man sich dann die »hervorzuhebenwertigen« Dateien aussuchen. Ein anderes Anwendungsgebiet liegt bei defekten Disketten. Hier sind mehrere Fälle zu unterscheiden, die alle von Disk Doctor abgedeckt werden:

— Defekte Sektoren im Directory. Sämtlicher Zugriff auf Dateien wird hierdurch unmöglich gemacht. Disk Doctor kann nun ganze Disketten kopieren, indem einfach Sektor für Sektor kopiert wird. Die defekten Sektoren der zu kopierenden Diskette werden beim Kopieren einfach übersprungen.

— Defekte Sektoren in Dateien. Auch hier kann wieder die gesamte Diskette kopiert werden. Meistens reicht es jedoch aus, die noch lesbaren Sektoren einer Datei auf eine andere Diskette zu kopieren. Auch hierfür steht eine Routine von Disk Doctor bereit.

— Defekte Sektoren in Bereichen, die keiner Datei zugeordnet sind. Beim Anlegen der nächsten Datei könnte ein Fehler gemeldet werden. Disk Doctor durchsucht die ganze Diskette nach defekten Sektoren und erzeugt dann eine Datei, in der diese Sektoren sozusagen gesammelt werden. Falls der Benutzer nun eine neue Datei anlegt,

werden die defekten Sektoren von CP/M automatisch übersprungen. In diesem Fall geht kein Byte an Information verloren.

Disk Doctor ist auf verschiedenen 5,25-Zoll-Diskettenformaten und auch auf 8-Zoll-SS-SD-Format erhältlich.

Das Handbuch ist hervorragend — aber nicht aktuell

Das 20seitige englische Handbuch ist — von wenigen sachlichen Fehlern und Schönheitsfehlern abgesehen — hervorragend. Nach einer kleinen Einführung wird die Art, wie das CP/M-Betriebssystem Dateien und Directory handhabt, ausführlich beschrieben. Aber das kann man von der Dokumentation eines Diskettenreparier-Programmes ja wohl erwarten. Anschließend wird das Installieren von Disk Doctor auf das Diskettenformat des Benutzers beschrieben (ein Relikt aus den Zeiten der CP/M-Version 1.4.). Die folgenden Kapitel beschreiben den Aufbau von Disk Doctor und das Arbeiten mit den fünf Routinen des Disk Doctors (genannt »Ward A.« bis »Ward E.«, siehe »Anwendungsbereiche«). In diesen Kapiteln wird dem Benutzer auch ein tiefergehender Einblick in die Arbeitsweise des CP/M-Betriebssystems gegeben, zumindest was das Verhalten von CP/M im Falle eines Diskettenfehlers betrifft.

Nun zu den Schönheitsfehlern: Die sonst übliche Liste von Dateien auf der »Distribution-Diskette« fehlt. Man muß sich die Dateinamen aus dem Kapitel »Installation« herausuchen. Aber auch diese Liste ist nicht komplett. In einer Datei namens READ.ME wird erläutert, daß man das Installieren von Disk Doc-

tor auch einem Programm namens INSTALL2.COM (ebenfalls auf der Diskette) überlassen kann. INSTALL2 sucht sich dann die notwendigen Installationsparameter (Anzahl Sektoren je Track, Anzahl Tracks je Diskette) aus den ohnehin schon in CP/M 2.2 vorhandenen Tabellen heraus.

Ferner stehen das Programm HELP.COM und die Datei HELP auf der Diskette. Beide Dateien stellen praktisch ein gekürztes Handbuch auf Diskette dar. (Die Ausgabegeschwindigkeit von HELP.COM ist so gewählt, daß ein altersschwacher Fernschreiber keine Probleme beim Mitprotokollieren bekommt.) Von diesen Dateien steht im Handbuch kein Wort. Mit anderen Worten: Das Handbuch ist seit einiger Zeit nicht mehr überarbeitet worden.

Die gravierendste Unannehmlichkeit sei zuerst genannt: Besitzt man nur ein einziges Diskettenlaufwerk, dann sollte man sich den Disk Doctor erst gar nicht kaufen. Selbst für das Installieren werden zwei Laufwerke benötigt. Abhilfe könnte allerdings ein (irgendwie zu beschaffendes) Programm schaffen, das es gestattet, ein einziges Laufwerk so zu verwenden, als hätte man zwei oder vier (man wird dadurch im wahrsten Sinne des Wortes zum Disk-Jockey). Ein anderer Weg, Disk Doctor mit nur einem Laufwerk zum Laufen zu bringen, besteht darin, im Programmcode herumzufummeln. In diesen Fällen lohnt sich vielleicht der Kauf von Disk Doctor doch noch.

Ein weiterer Kritikpunkt ist die Notwendigkeit, Disk Doctor überhaupt zu installieren. In einem Anhang werden die Parameter für die wichtigsten Formate angegeben. Trotzdem kann der Benutzer Fehler machen, wenn er nicht INSTALL2 verwendet. Außerdem ist man irgendwie auf halbem Weg stehen geblieben: Der Installierungsteil hätte genauso gut in DOCTOR.COM eingebaut werden können. (Vor der Verwendung des Formates 8 Zoll, 512-Byte-Sektoren sollte man sich hüten. In diesem Fall meint Disk Doctor, die Diskette habe nur 76 statt 77 Tracks. Der Fehler tritt auch bei Verwendung von INSTALL2.COM auf.)

Die Behauptung, CP/M speichere die Directory-Einträge einer Datei immer in aufsteigender Reihenfolge ab (siehe Handbuch, letzter Abschnitt Seite 14), ist schlichtweg falsch. Dies ist zwar normalerweise der Fall, es gibt allerdings auch

Ausnahmen. (Löschen einer Datei von einem Texteditor aus, während man eine andere Datei bearbeitet.) Diesem Irrtum sind aber auch schon andere Utilities zum Opfer gefallen, hauptsächlich Directory-Sortierprogramme. Infolgedessen verweigert Disk Doctor unter Umständen das »Un-ERAsen« von Dateien, die noch vollkommen intakt sind.

Hat man häufiger mit defekten

Disketten zu tun (ohne, daß der Fehler beim Laufwerk liegt), so ist Disk Doctor bestimmt ein gutes Hilfsmittel, Dateien zumindest teilweise zu retten.

Falls die Hauptanwendung von DISK DOCTOR darin liegen sollte, »ERA« rückgängig zu machen oder defekte Sektoren auf der Diskette zu suchen, so ist einem systemerfahrenen Benutzer wahrscheinlich eher mit den kostenlosen — da Pu-

blic Domain — Programmen DISKUTIL, RECOVER, RECLAIM, UNERASE, FINDBAD und so weiter geholfen. Dem reinen CP/M-Anwender jedoch ist mit Disk Doctor ein kompaktes sowie leicht verständliches und anwendbares Programm gegeben. Es nimmt ihm vielleicht auch ein wenig die Angst, durch eigene Fehler große Dateien zu verlieren.

(Joachim Schäfer)

System-Checker und Diagnostics-II

Computer sind komplizierte Maschinen. Fehler, besonders solche sporadischer Natur, sind schwer aufzuspüren. Unter Umständen aber kann sich der Computer selbst diagnostizieren. Hier werden die beiden System-Testprogramme »Diagnostics-II« und »System-Checker« vorgestellt. System-Checker ist eine abgemagerte Version von »Diagnostics II«.

Sowohl System-Checker als auch Diagnostics-II bestehen aus mehreren Programmen. Jedes einzelne testet einen bestimmten Bereich der Hardware:

- Speicher,
- CPU,
- Disketten-Laufwerke (nicht die Disketten selber),
- Terminal (nur die Ausgabe, nicht die Eingabe) und
- Drucker.

Ich werde nun im folgenden die Art und Weise beschreiben, wie Diagnostics-II die einzelnen Komponenten testet. Die Unterschiede zwischen System-Checker und Diagnostics-II werden am Ende eines jeden Abschnitts aufgeführt.

□ Speicher

Es gibt vier Testarten:
— Der »Quick-Test« schreibt den Speicher voller Nullen oder OFFH-Bytes. Er testet also nur, ob sich jedes Bit setzen und rückschalten läßt.
— Der »Walking Bit Test« schreibt den zu testenden Speicherbereich voller Nullen und »schiebt« ein einzelnes Bit mit Wert »1« durch den Speicherbereich. Durch diesen Test können Adreßdecodierungsfehler und »Übersprecher« zwischen einzelnen Bits ausfindig gemacht werden.

— Der »Burn-In Test« soll testen, ob dynamische Speicherchips gegen »Einbrennen« von Bits resistent sind.

— Der »Speed Test« soll Auskunft darüber geben, ob der Prozessor mit maximaler Geschwindigkeit arbeitet, oder ob er auf langsame RAMs warten muß.

Der zu testende Speicherbereich läßt sich jeweils angeben. Mit System-Checker ist nur der »Quick Test« durchführbar.

□ CPU

Dieser Test stellt fest, ob der im Computer verwendete Prozessor ein Z80 oder ein 8080 beziehungsweise 8085 ist. Einige kleinere Testroutinen prüfen, ob der Prozessor die wichtigsten Befehle noch ausführen kann. Auftretende Fehler werden nur bei Diagnostics-II mit recht ausführlichen Fehlermeldungen gemeldet.

System-Checker testet nicht ganz so viele CPU-Befehle wie Diagnostics-II. Auch werden keine Fehlermeldungen abgegeben.

□ Disketten-Laufwerk

Getestet wird, ob sich der Schreib-/Lesekopf einwandfrei positionieren läßt und ob Lesen- und/oder Schreiboperationen fehlerfrei ausgeführt werden.

Bei Diagnostics-II kann der Benutzer den Testablauf steuern (zum Beispiel durch Angabe der zu testenden Sektoren). Bei System-Checker hat der Benutzer keine Einflußmöglichkeit.

□ Terminal

Dieser Test »zeigt« jedes druckbare Zeichen in jeder Bildschirmposition. Dies ist der bekannte Test, bei dem die Zeichen auf dem Bildschirm ein diagonales Muster bilden. Falls der Bildschirm es erlaubt, werden direkte Cursor-Positionierung und andere Spezialfunktionen (inverses Schriftbild und so weiter) getestet. Dazu muß das Testprogramm erst installiert werden. Dies wird jedoch vom Testprogramm auf Wunsch durchgeführt und ist im Handbuch sehr ausführlich beschrieben.

Bei dieser Testroutine unterscheiden sich System-Checker und Diagnostics-II nicht.

□ Drucker

Auch hier wird ein Diagonalmuster gedruckt. Besitzer von Typenradruckern können sogar Vor- und Rückwärtsdruck, hoch- und tiefgestellten Druck und einiges mehr testen.

Auch hier gibt es keinen Unterschied zwischen System-Checker und Diagnostics-II.

Abgesehen von Umfang und Komfort der Testroutinen unterscheiden sich System-Checker und Diagnostics-II auch in der Art des Aufrufes: System-Checker wird mit dem Speichertest-Programm gestartet. Ist der Speicher getestet, wird automatisch das nächste Testprogramm von der Diskette nachgeladen und ausgeführt.

Bei Diagnostics-II muß jede Testroutine einzeln aufgerufen werden. Hierbei ist dann allerdings auch die Angabe der Testparameter als »Command-String« möglich. Ferner besteht die Möglichkeit, alle Testprogramme durch SUBMIT ausführen zu lassen.

Das Diagnostic-II-Handbuch umfaßt 38 Seiten, das System-Checker-Handbuch 18 Seiten sowie ein Sseitiges Glossar, in dem die wichtigsten Begriffe des Computers aufgeführt sind. Beide Handbücher sind in Englisch. Die Testroutinen werden sehr ausführlich beschrieben, und es dürfen keine Mißverständnisse aufkommen.

Diagnostics-II bietet neben den oben genannten Testroutinen noch zwei weitere wichtige Programme: Ein Schnelltest für die wichtigsten Systemkomponenten und ein Programm, das beim Justieren des Schreib-/Lesekopfes von Diskettenlaufwerken gute Dienste tut.

Die Diagnostic-II-Testroutinen schreiben auf Wunsch die auftretenden Fehlermeldungen in eine Datei DIAG-LOG. (Sehr angenehm im Zusammenhang mit SUBMINT.)

Falls beim Disktest Fehler auftreten, so kann Diagnostic-II diejeni-

gen Ein-/Ausgabereports oder Speicherstellen »dumpen«, die dem Floppy-Controller zugeordnet sind.

Der Speichertest von Diagnostic-II kann auch Speicherbereiche testen, die in einer anderen 64-K-Bank liegen.

Angaben über den Zeitbedarf der Testroutinen werden zwar bei Diagnostics-II gemacht, sind jedoch viel zu pauschal, um aussagefähig zu sein. (Mein 2,106-MHz-Z80 braucht 112 Sekunden für einen Test, der mit einer Minute veranschlagt ist). Für System-Checker gilt dasselbe.

Falls die Hardware fehlerhaft ist und die Fehler von System-Checker oder Diagnostics-II erkannt werden, so kann der Benutzer mit den System-Checker-Fehlermeldungen (sofern sie überhaupt ausgegeben werden) kaum etwas anfangen. Die Diagnostics-II-Fehlermeldungen lassen schon

eher auf die genaue Fehlerursache schließen.

Benutzer, die ihre Hardware für fehlerfrei halten, sich dies aber bestätigen lassen wollen, sind wahrscheinlich mit dem billigeren System-Checker besser bedient. (Ob-

Zusammenfassung

wohl der »Quick-Test« meiner Meinung nach nur sehr grobe Fehler feststellt). Legt man jedoch Wert auf hilfreiche Fehlermeldungen, dann sollte man sich für Diagnostics-II entscheiden.

Diagnostics-II ist wohl ein Leckerbissen für Hobby-Reparateure, die öfter einmal den Mikrocomputer ihrer Bekannten nachsehen müssen. Denn ein einzelner Benutzer kann die Möglichkeiten gar nicht ausreizen.

(Joachim Schäfer)

sucht, und wenn es da nicht gefunden wird, auf Benutzer 0. 3B)ZSID

Hier beginnt die Suche auf Benutzer 3, Laufwerk B:, danach Benutzer 0, Laufwerk B: und als letztes Benutzer 0, Laufwerk A:.

CCP darf bei Warmboot nicht nachgeladen werden

Die Modifikation des CP/Ms durch Extend hat auch noch eine andere Nebenwirkung: Zu einer Warmbootsequenz gehört der Reset der BDOS-Parameter (FCB Adresse, DMA-Adresse und die Rückkehr zu Laufwerk A als Be-

zugslaufwerk) sowie das Laden des CCP von den Systemspuren der Diskette. Diese letzte Ladefunktion wird durch das Vorhandensein von Extend unterbunden. Man muß hier also bei einigen Programmen vorsichtig sein, die den Speicherplatz vom CCP belegen. Die Gefahr der Benutzung des CCP-RAMs wird jedoch heute im Zeitalter der 48- bis 64-KByte-Computer relativ gering sein. Früher, mit 24 bis 32 KByte war sie jedoch oft notwendig. Um den CCP und Extend dann wieder zu laden, ist ein Systemreset notwendig. Ein Systemcoldboot ist dann auch der notwendige Weg, um »Extend« zu entfernen.

Durch den Einfluß von »Extend« auf den CCP ist es nicht mehr möglich, ein CP/M-Hilfsprogramm — XSUB — zu benutzen. Anstelle dessen gehört eine neue Routine — FBNSUB — zum Lieferumfang. Diese wird in gewohnter Weise wie XSUB aufgerufen, das heißt es wird als erstes Argument in einer Submitdatei aufgeführt.

Da Extend transparent für den Benutzer abläuft, also nicht interaktiv, ist seine Benutzung innerhalb von wenigen Minuten gelehrt. Drei Seiten Begleitdokumentation geben eine kurze Einführung in die Funktion und Handhabung des Programms. (Axel Pohl)

Extend — ein Programm für Diskettenplatz-Fetischisten

CP/M bietet, in einfacherer Form als MP/M, die Möglichkeit, in verschiedenen, sogenannten »User«-Bereichen zu arbeiten, beziehungsweise das »Directory« in verschiedene Unterteilungen zu gliedern. So kann man zum Beispiel Hilfsprogramme wie PIP, STAT, (Z)SID, Kopierprogramme und so weiter im Benutzerbereich 0 ablegen; Sprachen — wie Basic, Pascal, Pilot und andere mehr im Bereich 1 und Textverwaltung im Bereich 2. Jedoch wird man in allen Bereichen die Programme PIP und eventuell STAT benötigen. Kopiert man diese nun in die »USER«-Bereiche hinein, so belegen sie jedesmal weiteren Speicherplatz auf der Diskette. Diesem Übel hilft Extend ab. »Bitfreaks« werden sicherlich die gefährliche Methode der »Directory«-Manipulation kennen, aber auch diesen erleichtert Extend das Arbeiten.

Extend wird aufgerufen wie eine normale (COM)mand Datei. Statt sich jedoch auf den Anfang der (T)ransient (P)rogramm (A)rea (100H) zu laden, errechnet das Programm seinen Speicherplatz, indem es vom Anfang des (C)onsole (C)ommand (P)rocessors seine Länge subtrahiert. Dort oben lagert es sich ein und verändert anschließend die Einsprungadressen ins CP/M. Außerdem werden die Parameter verändert, die die Länge der TPA angeben. Da die Größe von Extend nur 2,5 KByte beträgt, ist die Verkleinerung zu vernachlässigen. Damit ist das Programm resident geworden. Um zu

vermeiden, daß es durch andere Programme, die sich ebenfalls dort oben einlagern (DDT, (Z)SID, DESPOOL, XSUB und die »SMART«-Programme »rausgeschmissen« wird, bedienen Sie sich am besten des »AUTOLOAD«-Features von CP/M. Hierdurch wird direkt nach dem »booten« des Systems Extend geladen. Nachdem die Kontrolle an das CP/M zurückgegeben wurde, meldet sich der CCP jetzt statt mit dem üblichen »A« mit »0A«. Wechselt man die »User«-Nummer, ändert sich zum Beispiel »0A« zu »2A«. Nun ist es möglich, Programme, welche sich in der User-Nummer 0 befindet, von anderen Bereichen

anzurufen. Eine Einschränkung bildet jedoch die Tatsache, daß dies nur für die Kommandoausführung durch den CCP gilt. Benutzerprogramme können diesen Effekt nicht nutzen, sie können nur Programme oder Overlays aufrufen, die sich in dem jeweiligen Benutzerbereich befinden.

Hier einige Beispiele für das Aufrufen von Programmen:
3A)PIP

PIP wird zuerst in der jetzigen Benutzernummer 3 gesucht, anschließend unter 0.

3B)A:STAT

STAT wird in der Benutzernummer 3 auf dem Laufwerk A: ge-

Microshell, für CP/M-Gourmets genau das richtige

Keine Frage, ohne CP/M wäre die Mikrocomputerwelt heute nicht das, was sie ist. CP/M gab den Softwarehäusern die Möglichkeit, Programme für eine breite Masse von Benutzern zu schreiben. Und langsam wurden auch andere Betriebssysteme bekannt — darunter Unix, das hauptsächlich auf Minirechnern läuft. Dann kam man auf die Idee, die Benutzerschnittstelle von Unix auf CP/M zu übertragen. Eine Sammlung von diesen übertragenen Routinen stellt Microshell dar.

Microshell ersetzt den (C)onsole (C)ommand (P)rocessor von CP/M. Es lagert sich unterhalb von dem CCP ein und verändert die Einsprungadresse 0005H und die Größe der zur Verfügung stehenden (T)ransient (P)rogramm (A)rea (um 8 KByte reduziert).

Nun zu den Funktionen von Microshell. Man kann sie in drei Gruppen aufteilen:

1. Umleitung von Ausgabedaten (Datensichtgerät und Drucker) auf die Diskette, Eingaben für Programme werden statt vom Datensichtgerät von einer Diskettendatei angefordert (ähnlich CP/Ms

SUBMIT- und XSUB-Programmen).
2. »PIPELINE« Verarbeitungsstruktur, das heißt Ausgaben des einen Programms dienen als Eingabeparameter für das nächste. Dessen Ausgaben werden wiederum von einem vierten Programm verarbeitet und so weiter.

3. SHELL- oder COMMAND-Dateien steuern die Abarbeitung von zum Beispiel Dienstprogrammen (Backup von Dateien, die seit dem letzten Backup neu beschrieben wurden, oder Modifizierung von Texten, ohne den Editor oder das Textverarbeitungsprogramm aufzurufen).

Unter Microshell sind solche Kommandos wie DIR, ERA, SAVE, TYPE, REN, USER also alle CP/M-eigenen Kommandos, weiterhin gültig. Hier einige Beispiele für die Umleitung von Daten:

B(0)DIR > > + SAMMELDIR

Das Directory wird zu den bereits in der Datei »SAMMLDIR« stehenden Daten hinzugefügt, dies wird durch die Verdoppelung von »>« erreicht.

B(0)Dir <+ SAMMLDIR

Zu der obigen Funktion der Umleitung kommt hier noch die Darstellung auf dem Bildschirm hinzu. Aber nicht nur in der Umleitung von Datenausgaben zur Diskette für Drucker oder Bildschirm, das zusätzliche Echo auf dem Bildschirm, sondern auch die Anordnung mehrerer Befehle in einer Zeile stellen die Besonderheit von Microshell dar.

Der zweite Punkt, das »pipelining«, kann auf einem CP/M-System nur simuliert werden (es entsteht eine Zwischendatei) Unix arbeitet diese Prozesse parallel ab (ohne Datei).

Als wesentlichstes Feature bietet Microshell eben die SHELL-Verarbeitung. Diese SHELL-DATEIEN (vielleicht besser Programme) führen viele Standardfunktionen in einem wesentlich komplexeren Rahmen aus als SUBMIT/XSUB. So gibt es zum Beispiel die Möglichkeit, Dateien und Programme im gesamten Disksystem suchen zu lassen (dieser Weg wird vom Benutzer über eine Datei bestimmt), SHELL-Dateien lassen sich gegenseitig verknüpfen (nesting), Argumente

Computer in den USA

USA, das Land der »hartnäckigen« Computer:
 Ein Mann war längere Zeit von zu Hause fort und hatte daher keinen Strom verbraucht.
 Der Computer der Elektrizitätsgesellschaft war auf Draht, er schickte dem Mann eine Rechnung über 0,00 Dollar. Sie landete im Papierkorb. Vierzehn Tage später: Wieder die gleiche Rechnung über 0,00 Dollar mit dem Zusatz »überfällig«. Auch sie wurde von dem Empfänger müde lächelnd zerrissen und weggeworfen. Der Computer reagierte sauer. Er schickte vierzehn Tage später nochmals eine Rechnung über 0,00 Dollar und drohte, dem Mann das Licht zu sperren.
 Der geplagte Stromverbraucher resignierte und sandte der Elektrizitätsgesellschaft einen Scheck über 0,00 Dollar. Der Computer war versöhnt und schickte dem Mann eine Karte, mit der er höflich für die Zahlung dankte.

ZEICHEN	BEDEUTUNG	BEISPIEL
>	Ausgabenumleitung vom Bildschirm	stat > dateiname
>*	Ausgabenumleitung vom Drucker	pip lst:=Datei >* prdatei
<	Eingangsumleitung	ed text.txt < korrektur
	leitet Ausgaben eines Befehls als Eingaben an den nächsten	Datei 1 Datei 2 stat *.* pip lst:=con:
	" " dieses Zeichen verändert den folgenden Buchstaben zum ASCII-Kontrollzeichen	C oder c wird ASCII Wert 03
:	als erstes Zeichen wird es als Kommentar interpretiert	:so wird kommentiert :oder so auch
+	Echo der Umleitung erscheint auf dem Bildschirm	Datei 1 +Datei 2 stat *.* > +Dateiname
-	bei Abfrage des Konsoleingangs-Status = "Zeichen erhalten"	Datei 1 -Datei 2

ZEICHEN	BEDEUTUNG	BEISPIEL
;	Kommandos in einer Befehlszeile trennen	dir *.com;era *.bak
""	behandelt den Text innerhalb der Anführungszeichen als ein Char.	echo "dies wäre ein argument"
\$	gleiche Verarbeitung wie bei SUBMIT/XSUB-Argumentersatz	copieren bak ;aufruf pip b:=a:*. \$1 ;datei pip b:=a:*.bak ;ausführen
-u -U	erlaubt ohne Konvertierung die Weitergabe von Kleinbuchstaben	A(0) -U A(0)echo kleinbuchstaben <RETURN> Kleinbuchstaben
+v +V	gibt Echo von allen Kommandos vor der Ausführung	A(0) +V löschen test spiel era test.com era spiel.com
-x -X	Rückkehr zu CP/M	A(0) -x A>

Einige Funktionen von Microshell

können beim Aufruf von SHELLS auch nicht belegt werden, MICRO-SHELL gewährleistet trotzdem einwandfreie Funktion. Unter SUBMIT würde folgendes Beispiel eine Fehlermeldung auslösen:

```
Submitdatei:
DIR $1.COM
ERA $1.BAK
PIP $2.ASM=$3.ASM,B:$4.ASM
```

Befehlszeile:
SUBMIT CLEAN MIC MAC MUC
Hier fehlt das vierte Argument. Die Fehlermeldung »Datei nicht gefunden« wäre unausweichlich.

Hier gibt es unzählige Möglichkeiten, in einer Art Programmiersprache, die Elemente wie GOTO, IF... THEN, IFF (F=FILE).. THEN, RETURN (CONDITION)

enthält den Programmfluß zu steuern. So gibt es insgesamt 33 Variablen

numerisch %A - %G
String %A\$ - %C\$
sowie die von SUBMIT bekannten Übergabeparameter \$0 - \$18

Hierbei ist \$0 für die Shelldatei reserviert. Es können also noch weitere 17 Argumente an die aufgerufene Shelldatei übergeben werden. Zu der Stringfunktion ist noch zu sagen, das eine Anweisung existiert, die ein Löschen der in den

Zeichenketten vorhandenen Werte vornimmt (%clear).

In der Tabelle sind einige Funktionen von Microshell aufgelistet.

Die Tabelle ist nicht vollständig, aber sie gibt schon einen Eindruck von der Leistungsfähigkeit von Microshell.

Ein Beispiel, welches die Anwendung solcher Anweisungen wieder spiegelt, ist eine Schleife. Mit einem Texteditor oder über die direkte Eingabe in eine Datei unter Zuhilfenahme von PIPEST.SUB=CON: (LINEFEED muß nach dem [CR] extra eingegeben werden) würde wie folgt aufgebaut werden:

```
:Beispiel für die Anweisungen
:Kommentar, %Label, %getchar,
:Terminaldirekteingabe $T,
:numerische und Stringvariable,
:%if .. then .., %goto
```

```
: % Schleife
%getchar %a$ < $1
%b$=%b$+%a$;%a=%a+1
%if %a=7 then ende
%goto schleife
```

Da die Eingaben in einer Zeile bis zu 84 Zeichen lang sein können, wurde eine recht praktische Editerroutine implementiert. Sie benutzt Wordstar ähnliche Kommandos, was das Erlernen vereinfacht. Es ergibt sich die Möglichkeit vorhergehende Anweisungen ein zweites Mal auszuführen.

Viele der verwandten Anweisungen lassen sich auf Benutzerwünsche anpassen. So kann man statt des Ausdrucks \$T (für Terminaldirekteingabe) eigene Wünsche verwirklichen (statt \$P vielleicht \$DRK, oder für den Aufruf des Editors statt ! ein @). Zu beachten ist nur, daß Einzelzeichen (! oder @) dann für diese Aufgaben reserviert sind. Weiter gibt es Definitionsdateien (zum Beispiel automatische Suchprozedur) worin die erforderlichen Parameter definiert werden (wie viele Laufwerke, Reihenfolge des Suchvorganges), Echo auf den Bildschirm an- und ausschalten, automatisches Einloggen von gewechselten Disketten und so weiter. Aufgezeigt werden diese Möglichkeiten sehr genau in dem mitgelieferten Handbuch. Dort werden die einzelnen Anweisungen auch sehr detailliert besprochen, Tips für die Anpassung gegeben, Fehlermeldungen besprochen und Programmiersprache angegeben, wenn jemand spezielle Programme für Microshell schreiben möchte. Leider muß hier noch bemerkt werden, daß das Handbuch in Englisch geschrieben ist, was vielen Lesern eine Erschwernis bereiten wird, bis sie endlich die Vorteile dieses hilfreichen Programms genießen können.

(Axel Pohl)

»Wieso füttern Sie den CHROMWELL 3X-4 - Computer mit Eiswürfeln; mag er die denn?«
 »Scheinbar ja; über Nacht sind sie immer weggeputzt!«

Der Computer ist in die Gewerkschaft eingetreten: Er verlangt zweimal täglich eine Kaffeepause.

Computer-Identifizierung

Der neue Computer steht am Firmeneingang. Er identifiziert morgens bei Arbeitsbeginn die Mitarbeiter optisch durch Erfassen des Gesichtes und gibt dann den Weg frei. Der Firmeneinhaber verfügt schon am zweiten Tag nach der Aufstellung des Computers dessen Abschaffung. Am ersten Tag hatte der Computer beim Anblick des Firmenchefs unter »Besondere Bemerkungen« ausgedruckt: »IHR GESICHT IST SO EHRlich WIE IHRE SPESENABRECHNUNGEN!« Am zweiten Tag schrieb er: »IHR GESICHT IST SO EHRlich WIE IHRE EINKOMMENSTEUERERKLÄRUNGEN!«

CP/M-Disassembler gibt es eigentlich »wie Sand am Meer«: In jedem Debugger ist ein Disassembler-Modul enthalten. Dagegen sind Resource, Disilog und Revas eigenständige Programme, die »nur« disassemblieren können.

Sogenannte Assembler übersetzen in symbolischer Form (LDA, 3AH = Lade das Register A mit der Hexzahl »3A«) geschriebene Maschinenprogramme in Binärzahlen, wie sie vom Prozessor verstanden werden. Disassembler kehren diesen Vorgang um: Aus dem Programmcode, wie er in einem EPROM oder bei einem von Disk oder Kassette geladenen Programm vorliegt, erzeugen sie ein für den Menschen lesbares Programm in Assembler-Sprache. Leider hat dies aber so seine Tücken, denn nicht alles, was in einem Codefile vorkommt, sind Maschinenbefehle — es können zum Beispiel Texte sein, die vom Programm ausgegeben werden, oder irgendwelche Daten, die in einer Tabelle für das Programm griffbereit vorliegen. Disassembler können das von sich aus nicht unterscheiden.

•Gewöhnliche• CP/M-Disassembler verlangen neben dem Namen einer Programmdatei (.COM-FILE) die Angabe von Byte- und Text-Speicherbereichen, bevor der eigentliche Disassemblierungsvorgang stattfindet. Man kann nun nach einem von Murphys Gesetzen sicher sein, daß man mindestens einen solchen Speicherbereich übersehen hat. Ein anderes Gesetz besagt, daß dies erst auffällt, nachdem man das bereits disassemblierte Programm stundenlang kommentiert, verbessert oder sonstwie editiert hat. Es lohnt sich also nicht, von neuem zu disassemblieren, sondern man muß frustriert in einer Tabelle nachsehen, daß die Befehle LD C.A.; LD L.E.; LD L.ODH den Text »Ok« gefolgt von einem CR darstellen. (Dies Beispiel ist vielleicht ein wenig an den Haaren herbeigezogen, zeigt aber, worum es im Prinzip geht.)

Resource dagegen ist ein »interaktiver« Disassembler: Man kann sich ansehen, was passiert, wenn man einen Speicherbereich versuchsweise als Byte-Speicherbereich definiert anstelle als Programmcode.

Einige weitere Eigenschaften von Resource in Stichworten:

Kommentare können bei Resource bereits während des Disassem-

blierens eingefügt werden. Symbolnamen anstelle von Adressen werden automatisch erzeugt. Texte, die länger als acht Byte sind, werden erkannt. Man kann sich die Angabe von Text-Bereichen also sparen. Das Disassemblieren kann zu jedem beliebigen Zeitpunkt unterbrochen und später wieder fortgesetzt werden. Man kann nach 16-Bit-Wörtern suchen. Anstelle von Hex-Adressen können auch Symbole angegeben werden.

Steuer-Tabelle

Das »Was wäre, wenn« ist deshalb möglich, weil Resource die sogenannte Steuertabelle verwendet: Jeder Eintrag dieser Tabelle besteht aus einem Adreßteil und einem Byte, das den Disassembler steuert. Diese Tabelle kann vom Benutzer jederzeit verändert werden.

Zum Beispiel: Die Speicherbereiche 0100-0102, 012C-0147, 014C-017F enthalten Programmcode, der Bereich 0103-012B einen Text, der Bereich 0148-014B zwei Datenwörter. Der Stackbereich des zu untersuchenden Programmes liege von 0180-019F. Adressen ab 01A0 kommen nicht vor.

Dann sieht die zugehörige Steuertabelle folgendermaßen aus: 0100,I Instruktion ab Adresse 0100 0103,B Byte-Speicherbereich (*) 012C,I Instruktionen (*) 0148,W Wort-Speicherbereich 014C,I Instruktionen 0180,S Space, das heißt im Assemblerfile wird: »DEFS 20H« gelistet (20H = 01A0 minus 0180) 01A0,E Ende des Programms

Dabei gibt die Adresse immer den Anfang des jeweiligen Bereiches an. Der erste Eintrag ist von Resource vorgegeben, da CP/M-Programme ab Adresse 0100Hex gestartet werden. Ab Adresse 0100 ist demnach mindestens eine Instruktion zu finden. Die beiden mit (*) gekennzeichneten Einträge wurden von Resource erzeugt, da ab Adresse 0103 ein Text steht, der länger als acht Byte ist. Am Ende des Textes »schaltet« Resource wieder auf Programmcode-Modus um. Die Programm-Ende-Markierung erzeugt im Assemblerfile an der

entsprechenden Stelle den Pseudo-Befehl »END 0100H«.

Nun kann man zum Beispiel den Eintrag »014C,I« löschen. Zusätzlich zu den beiden Wörtern ab Adresse 0148 werden nun auch alle Bytes zwischen 014C und 017F als Wörter gelistet. Indem man die Steuertabelle gezielt verändert, kann man also ein Assemblerfile erhalten, das von der Struktur her mit dem Original übereinstimmt.

Kommentare und Symbole im Listing

Die Steuer-Tabelle selber macht dem Komfort von Resource jedoch nicht allein aus: Man kann an jede beliebige Stelle des Listings bereits beim Disassemblieren Kommentare einfügen. Diese werden sowohl in der fertigen Assemblerdatei ausgedruckt als auch auf dem Bildschirm ausgegeben. Zum Speichern der Kommentartexte wird ein Speicherbereich verwendet, dessen Beginn der Benutzer selbst festlegen muß.

Mit dem »B«-Befehl wird Resource dazu veranlaßt, eine Standard-symbol-Tabelle zu erzeugen: Jede Adresse, die als Sprungziel, Unterprogramm-Adresse oder als 16-Bit-Konstante in einem Ladebefehl verwendet wird, wird in die Symboltabelle aufgenommen: Adresse xyzt — — Symbol »Lxyzt«. Zum Beispiel: Adresse 04FA — — Zeichenreihe »L04FA«.

Die Symboltabelle kann jederzeit vom Benutzer verändert werden. Das Symbol »L005C« zum Beispiel kann in »FCB1« geändert werden. Ebenso ist folgende Konstruktion möglich:

»L05F1« — »KONVRT«
»L05F9« — »KONVRT«+8

Damit das Disassemblieren zu einem späteren Zeitpunkt fortgesetzt werden kann, müssen die Symbol-, Control- und Kommentar-Tabellen abgespeichert werden. Eine ASM-Datei braucht noch nicht gespeichert zu werden, da, wie gesagt, der Disassemblierungsvorgang noch nicht beendet ist. Das Erzeugen einer Assemblerdatei ist demnach die letzte Taugkeit, zu der man Resource veranlaßt.

Es sollte beachtet werden, daß Resource weder eine Warnung, noch sonst einen Hinweis ausgibt, wenn die Tabellen nicht abgespeichert wurden.

Resource-Versionen gibt es für folgende Assemblersprachen: 8080: Standard-Intel-Format Z80 : Standard-Zilog-Format, TTL-Z80-Format.

Resource wird CP/M-gemäß ab Adresse 0100H geladen, verschiebt sich jedoch nicht unter das BDOS, wie man es von DDT her kennt. Das hat zur Folge, daß man einen sogenannten Offset angeben muß. Diese Offset-Adresse wird als scheinbare »Adresse 0« angesehen. Das heißt im einzelnen: Das zu disassemblierende .COM-File wird

ab RAM-Adresse Offset+100H geladen. Von nun ab ist der Offset weniger wichtig. Gibt man nun den Befehl »L100« oder D1234«, dann wird ab wahrer Adresse Offset+100H gelistet beziehungsweise ab wahrer Adresse Offset+1234H gedumpt. Der Offset und Resource selber werden für den Benutzer also »unsichtbar«.

Falls man am Ende einer Sitzung die Assemblerdatei schreiben will, erfolgt die Ausgabe sowohl auf die Datei als auch auf den Bildschirm. Dies ist zeitaufwendig. Deshalb kann man vor jedem Befehl, den man enttippt, den Buchstaben »Q« setzen (Quiet-Modus). Die Bildschirmausgabe wird unterbunden, bis man den nächsten Befehl gibt.

Resource besitzt folgende Anweisungen:

- A Suche im angegebenen Bereich nach Texten (»A.SCI1)
- B Erzeugen der Standard-Symboltabelle (»B.UILD«)
- C Control-Tabelle ändern
- D Dumpen eines Speicherbereichs
- E Einfügen eines Symbols in die Symboltabelle
- F Finde 16-Bit-Worte
- K Symbol löschen (»K.UILL)
- L Listen eines Speicherbereichs
- O Offset angeben
- Q Quiet-Modus
- R Lesen von Dateien (»R.READ)
- S Schreiben der Dateien
- : Kommentare einfügen (Joachim Schäfer)

DPATCH ist ein bildschirmorientiertes menügesteuertes Hilfsprogramm für Floppy- und Harddisksysteme. Es gestattet die Datensicherung bei Dateien mit I/O-Error, die Weiterbenutzung von Disketten und Hard-Disks mit Hard-Fehlern sowie die Manipulation eines jeden Bytes auf der Diskette oder Platte.

Nach dem Start von Dpatch meldet sich das Programm mit Menü. Man kann nun eine der sechs Funktionen durch Eingabe der entsprechenden Ziffer aufrufen.

1. Disk-Edit-Funktion
2. Datei-Manipulation
3. Reaktivierung einer Datei mit I/O-Error
4. Disk-Oberflächen-Analyse
5. Reaktivieren einer gelöschten Datei
6. Programmende.

Die oberste Statuszeile nach Aufruf der Disk-Edit-Funktion informiert, welcher Sektor von welchem Laufwerk gerade in hexadezimaler und ASCII-Form dargestellt wird, während die letzten beiden Kommandozeilen die möglichen Befehle zur Anwahl eines beliebigen Sektors auf allen angeschlossenen Laufwerken das Lesen, Modifizieren und Zurückschreiben auf die Diskette oder Platte ermöglichen. Mit den Kommandos »IN« und »OUT« kann trackweise fortgeschritten werden.

Die nicht dokumentierten Kommandos der Version 2.2. erlauben es, nach bestimmten Stringfolgen zu suchen (LOCATE, LOCATE NEXT) und einen kompletten Sektor in einem von acht »Sektor-Registern« zwischenspeichern (PUT)

und von dort bei Bedarf in einen anderen Sektor zurückzuladen (GET). Mit CNTRL-P kann der Bildschirm jederzeit auf einem angeschlossenen Drucker ausgegeben werden.

Als nachteilig wurde empfunden, daß im Disk-Edit-Mode mit den Kommandos »Sektor« und »Sektor« jeweils der physikalische Sektor um ±1 verändert und dargestellt wird und nicht unter Berücksichtigung des »Skew-Faktors« die logischen Sektoren. Ein gezieltes Suchen, zum Beispiel durch den Bereich des Inhaltsverzeichnisses, wird dadurch erschwert, daß immer sechs Sektoren weitergeschaltet werden muß.

Bei der direkten Dateimanipulation besteht dieser Nachteil jedoch nicht. Dieses Programmsegment arbeitet ähnlich wie die Disk-Edit-Funktion, nur muß hier zu Beginn der Name der zu analysierenden Datei eingegeben werden. Im Gegensatz zur Disk-Edit-Funktion, mit der sich jedes Byte auf einer Diskette oder Platte manipulieren läßt, ist dies mittels der direkten Dateimanipulation nur innerhalb der gewählten Datei möglich.

Dateien, die aufgrund eines I/O-Errors nicht mehr lesbar sind, können unter Umständen wieder korrigiert werden. Zu diesem Zweck

wird die Datei umkopiert und bei Erreichen des fehlerhaften Sektors der Lesezyklus so lange wiederholt, bis er entweder erfolgreich abgeschlossen oder vom Anwender abgebrochen wird. Als Erfahrungswert wird im Handbuch angegeben, daß nach 40 bis 50 Versuchen die Funktion abgebrochen werden sollte. Bei den Leseversuchen wird die Spur, auf der sich der fehlerhafte Sektor befindet, aus verschiedenen Richtungen angefahren, in der Hoffnung, so eventuell ein gewisses Spiel in der Mechanik auszugleichen und den Kopf exakt so zu positionieren wie beim Schreibvorgang.

Mittels der »Oberflächenanalyse« werden BAD RECORDS ausgespürt, in einer Datei »BADBLKS« zusammengefaßt und unter der User-Nummer 31 abgelegt, auf die unter Standard-CP/M-Kommandos nicht zugegriffen werden kann. Zwei Analysefunktionen können angewählt werden:

1. Zerstörungsfreie Disk-Analyse
2. Test-Pattern-Analyse

Während bei 1. die Diskette nur gescannt und Fehler ausgespürt werden, werden bei 2. verschiedene Bitmuster geschrieben, gelesen und verglichen. Hierbei werden sämtliche Informationen auf der Diskette zerstört. Dpatch arbeitet in allen Funktionen fehlerfrei. Die interaktive menügesteuerte Bedienung ist, insbesondere aufgrund der guten Dokumentation mit umfangreichen Fehlermeldungen, im Klartext schnell zu beherrschen. Für den ambitionierten Anwender ist Dpatch sicherlich ein wertvolles Hilfsmittel.

(Rainer Markgraf)

Peter Wollschlaeger

TRS-80/Modell 4:

Kleine Zipperlein rasch kuriert

Der TRS-80/Modell 4 wird serienmäßig mit TRSDOS 6.1 und einem erweiterten MBasic geliefert. Dieser Software ist durchaus Lob zu zollen. Doch es gibt nichts, was man nicht besser machen kann, und so ein paar kleine Zipperlein, die im Programmieralltag manchmal ganz schön zwicken, sind auch hier vorhanden. Lesen Sie, wie man sie kuriert und einige schummernde Eigenschaften der Software zum Leben erweckt.

Vorab, wir greifen mit Peek und Poke in das Betriebssystem ein. Einige der Adressen sind nicht offiziell dokumentiert, andere wurden dem technischen Manual [1] entnommen, das aber in Deutschland (noch) nicht erhältlich ist. Theoretisch besteht also die Gefahr, daß sich mit einer neuen Version die inoffiziellen Adressen ändern. Die neuen dann wieder aufzuspüren, dürfte aber für findige mc-Leser kein Problem sein. Auch in [1] sind keine absoluten Adressen zu finden. Dokumentiert sind sogenannte SVCs (Supervisor Calls) – ein neuer Name für einen alten (CP/M-)Hut. Sprich, DOS-Routinen werden über eine Funktionsnummer (im A-Register) immer mit RST 26H aufgerufen. Ein SVC (85H) erbringt im IX-Register die Basisadresse einer Tabelle, in der diverse Systemvariable gehalten werden. Bild 1 zeigt die zwei Assembleranweisungen, die dafür erforderlich sind. Demonstriert werden soll aber etwas anderes: Früher

hätte man einfach, wenn man einen Wert von der USR-Routine an Basic zurückgeben wollte, geschrieben: LD HL, WERT, JP CINT. Jetzt ist für CINT nur noch ein Vektor dokumentiert (2805H); der Wert muß aber immer noch in HL stehen. Daher die Lösung ab Zeile 8. Nämlich: Wert auf den Stack, HL mit Adresse von CINT (aus Vektor) laden und "EX (SP),HL". Danach steht der Wert in HL und die Adresse von CINT auf dem Stack, das folgende RET wirkt als indirekter Sprung zu CINT. In Bild 2, Zeilen 100...130, finden Sie das Programm noch einmal, diesmal in einen Basic-Lader eingebunden. Danach steht die gesuchte Adresse in der Variablen BASE. BASE+10 zeigt auf die Versionsnummer, in diesem Byte steht 61H für „6.1“. In Zeile 150/160 finden Sie die praktische Anwendung. Wichtiger ist schon das K-Flag (Keyboard) in BASE+10. Interessant sind

```

0000 0000 00001 USR-ROUTINE, ERGIBT BASIS DER SYSTEM-FLAGS-TABELLE
0000 0000 00002
0000 3E45 00003 LD A,45H ;SVC TO GET FLAG-
0000 0002 EF 00004 RST 26H ;TABLE BASE -> IX
0000 0003 F0E5 00005 PUSH IY ;IVAL TO RETURN
0000 2A0526 00006 LD HL,(2805H) ;ADR CINT-ROUTINE
0000 E3 00007 EX (SP),HL ;-> STACK
0000 C9 00008 RET ;VIA CINT -> BASIC
0000 0000 00009 END
0000 0000 00010 TOTAL ERRORS
34831 TEXT AREA BYTES LEFT

```

Bild 1. So gibt eine USR-Routine ein Ergebnis an Basic zurück

Bit 7 (1, wenn noch ein Zeichen im „Type Ahead Buffer“), Bit 5 (1, wenn „CAPS locked“) und Bit 0 (Break gedrückt).

CAPS-Taste vom Programm bedient

Wesentlich erscheint dem Autor Bit 5. Da in String-Variablen sauber zwischen Groß- und Kleinschrift unterschieden wird, ist man gezwungen, Antworten auf Fragen wie „Ja oder Nein (J/N)?“ so zu prüfen: IF A\$="J" OR A\$="j" THEN... ELSE IF A\$="N" OR A\$="n" THEN... Läßt man in der Antwort viele Buchstaben und andere Zeichen zu, wird auch das Testen via ASC mühsam. Deshalb ab Zeile 180 die Demonstration, wie man mit Hilfe von „Bit 5“ die CAPS-Taste im Programm abfragen und betätigen kann.

Cursor in allen Varianten

Viele Möglichkeiten eröffnet das Video-Flag. Vorab (Zeilen 350...510) einige Spiele mit der Uhr. Das Stellen wäre zwar auch über einen SYSTEM-Aufruf möglich, nur muß dazu eine Systemdiskette eingelegt sein. Mögen Sie lieber einen blinkenden oder stehenden Cursor, hätten Sie ihn gern etwas größer oder kleiner oder ganz unsichtbar? Möchten Sie das alles mischen, um in Masken bestimmte Stellen herauszuheben? Kein Problem, in den Zeilen 520...690 werden alle Varianten vorgeführt.

Hardcopy mit einem Befehl

Das Modell 4 erlaubt es, im Direkt-Modus mit „Control Shift Doppelpunkt“ den Schirminhalt auf den Drucker zu kopieren. Per Programm geht das auch, Sie müssen nur die BS-Routine kennen (siehe Zeile 730). Sollte es Sie dann stören, daß anstatt Ihrer mühsam erzeugten Grafikzeichen nur schnöde Punkte gedruckt werden, dann hilft von Basic aus: SYSTEM("SYSTEM(GRAPHICS)"). Zeile 740 zeigt, wie man die Break-Taste aus- und einschalten kann, und Zeile 750, ein CLS mit Ready-Meldung, man weiß dann wenigstens, was Sache ist. Scroll-Protect von maximal sieben Zeilen ist entgegen anderslautenden Gerüchten auch beim Modell 4 möglich. Offiziell läuft das über SVC 0FH, wobei vor dem Aufruf das B-Register mit 7 und C mit der Zeilenzahl zu laden ist. Ist Ihnen das zu mühsam, so geht das auch mit einem Poke-Befehl, wie in Zeile 800 gezeigt. Der etwas umständliche Ausdruck dort ist die Kopie der BS-Routine: LD A,C:AND 7:LD C,A:LD A,(0BF6):AND F8:OR C:LD (9BF6),A.

LIST-Befehl deaktiviert oder geheim

Man kann zwar ein Programm über die P-Option schützen, nur hat die den Nachteil, daß jeder Hacker weiß, wie man sie ausschalten kann. Wenn Sie sich nach den Vorschlägen ab Zeile 820 Ihre Spezialversion schaffen, wird es etwas schwieriger. Wenn Sie ganz sicher gehen wollen, müssen Sie allerdings diese mc-Ausgabe komplett aufkaufen.

Basic-Programm gerettet

Nach einem versehentlichen Ausflug ins DOS (bei SYSTEM Textstring vergessen) kommen Sie leider nicht mehr ins Basic zurück, ohne das Programm dort zu zerstören. Abhilfe bringt die Routine in Bild 3. Der Basic-Teil generiert einen Cmd-File, so daß Sie dann nur noch BOLD tippen müssen, und die Welt ist wieder in Ordnung. Notfall- geht es auch so: Tippen Sie im DOS: MEMORY-(GO=X'63DD').

Sie kommen dann auch ins Basic, finden auch Ihr Programm intact vor, nur alle Zeiger sind verbogen, und FRE(0) meldet einen freien Speicher von 8.054 Bytes, also 3 Bytes und noch ein halbes Bit oder so. In diesem Falle sollten Sie Ihr Programm sichern und dann Basic neu laden.

Zeilen duplizieren geht doch

Die kleine Routine in Bild 4 kostet kein einziges Byte, sie ändert nur den Editor. Der K-Befehl ist bekanntlich redundant (mit Delete geht es auch), weshalb ihm die Bedeutung KOPIE zugewiesen wurde. Wollen Sie eine Zeile duplizieren, so holen Sie sie in den Edit-Mode und tippen K. Es erscheint ein „?“ als Aufforderung, die Nummer der neuen Zeile einzugeben. Sobald das erfolgt ist, wird die neue Zeile im Edit-Mode angezeigt. Sie können dann <Enter> geben oder die Zeile gleich editieren, ganz original kopiert man ja meistens nicht.

Deutsche Sonderzeichen auch gedruckt

Sollten Sie auch zu geizig gewesen sein, sich den Tandy-Drucker zu kaufen und sich statt dessen mit einem Epson oder ähnlichem bescheiden, dann haben Sie ein Problem. Die deutschen Zeichen erscheinen zwar auf dem Schirm, bringen aber auf dem Drucker nur Nonsense. Abhilfe schafft das Programm in Bild 5. Das ist ein Basic-Lader einmal anders. Dies

```

Tricks und Tips für den TRS-80 Modell 4
(C) 1984 Peter Wollschlaeger, Hildesheim
Die folgende Routine ermittelt die Basis der System-Flags-Tabelle
DATA 3E,65,EF,FD,ED,2A,05,26,E3,C9
FOR I=0 TO 9:READ A$:POKE &H5FB5+I,VAL("&H"+A$):NEXT
DEF USR=&H5FB5:BASE=USR(0)
V=H$ HEX$(PEEK(BASE+27))
PRINT "Sie arbeiten unter TRSDOS Version ";LEFT$(V$,1)"; "RIGHT$(V$,1)
CAPS-Taste
*****
KF=BASE+10: 'Keyboard Status Flag
' Status der CAPS-Taste testen:
IF PEEK(KF) AND 32 THEN PRINT "CAPS on Uppercase"
ELSE PRINT "CAPS on Lowercase"
CAPS-Taste betätigen:
POKE KF,PEEK(KF) XOR 32
' Uppercase erzwingen:
POKE KF,PEEK(KF) OR 32
' Lowercase erzwingen
POKE KF,PEEK(KF) AND 323
' Echtzeit-Uhr auf dem Bildschirm
*****
VF=BASE+21: 'Video Status Flag
' testen ob Uhr sichtbar oder nicht
IF PEEK(VF) AND 16 THEN PRINT "Uhr an" ELSE PRINT "Uhr aus"
' Uhr einschalten:
POKE VF,PEEK(VF) OR 16
' Uhr ausschalten:
POKE VF,PEEK(VF) AND 239
' Uhr stellen
INPUT "Uhrzeit in h.m.s " ;H,M,S
POKE 43,S:POKE 44,M:POKE 47,H
PRINT "Ihre Zeit eingab: ";TIME$
' Cursor ändern
*****
CSA=&H5BF: 'Adresse des Cursor-Symbols
VF=BASE+21: 'Video Status Flag
' Cursor-Symbol testen/ändern
PRINT "Cursor hat g.L. Code ";PEEK(CSA),HEX$(PEEK(CSA)) ;"H"
' Cursor neu Symbol (neu mit hhhh) "C": POKE CSA,C
REM Code 32 oder 128 macht den Cursor unsichtbar
' testen ob Cursor blinkt
IF PEEK(VF) AND 64 THEN PRINT "Cursor stent" ELSE PRINT "Cursor blinkt"
' Cursor auf stehend schalten:
POKE VF,PEEK(VF) OR 64
' Cursor auf blinkend schalten:
POKE VF,PEEK(VF) AND 191
' Verschiedene Tricks
*****
X=&H5A:CALL X: ' erzeugt ein "Klick"
X=&H8A:CALL X: ' Hardcopy Schirm -> Drucker
VF=BASE+10:POKE B,PEEK(B) XOR 16: ' Schaltet Break-Taste aus bzw. wieder an
X=&H55:CALL X: ' Wie CLS, aber mit Ready-Prompt
Die folgende Routine erlaubt, max. 7 Zeilen auf dem Schirm zu schützen
(Kein Scrollen dieser Zeilen)
INPUT "Wie viele Zeilen schützen (max 7) ";I
POKE &HFB, (PEEK(&HFB) AND &HFB) OR (I AND 7)
' Der ausgeschaltete oder geheime LIST-Befehl
*****
' in den Adressen &H599A &H599B &H599C &H599D
' steht ASC von "I" "S" "T" *128 &H93-Token LIST
*****
' mit POKE &H599A,ASC("A") wird aus LIST LOST
' mit POKE &H599D,129 wird aus LIST END
' mit POKE &H599D,140 wird aus LIST NEW
*****
' Wenn es Sie auch stört, daß im EDIT die ESC-Taste (Shift ~) schlecht
' zu bedienen ist, dann legen Sie doch die Funktion auf F1, und zwar so:
POKE &H3E,129

```

Bild 2. Eine Sammlung von Tricks: Eingriffe in das Betriebssystem über Peek und Poke

```

000010 (RE-ENTER BASIC FROM TRSDOS 6.1) TRS-80, M 4
000020 (C) 1983/12 PETER WOLLSCHLAEGER, HILDESHEIM
000030
000040
000050 (C) 1984/3 PETER WOLLSCHLAEGER, HILDESHEIM
000060
000070
000080
000090
000100
000110
000120
000130
000140
000000 TOTAL ERRORS
34818 TEXT AREA BYTES LEFT
10 'Programm generiert einen Com-File. Name: BOLD/CMU
20 'Zweck: Ruckkehr von TRSDOS (uber SYSTEM erreicht) zu BASIC.
25 'Aufruf: in TRSDOS: BOLD
30
40 DATA CD,ED,7D,CD,96,42,CD,3D,43
50 FOR I=0 TO 25:READ A$:POKE 486000+I,VAL("BH"+A$):NEXT
60 SYSTEM"DUMP BOLD/CMU:0" (S="6000",E="6008",TRA="6000")"
    
```

Bild 3. So kann man nach einem Ausflug ins DOS sein Basic-Programm retten

```

000010 ZIDUP ZEILEN DUPLIZIEREN
000020
000030 DAS EDIT K-KOMMANDO HEISST JETZT KOPIE
000040
000050 (C) 1984/3 PETER WOLLSCHLAEGER, HILDESHEIM
000060
000070
000080
000090
000100
000110
000120
000130
000140
000150
000160
000170
000180
000190
000200
000210
000220
000230
000240
000250
000260
000270
000280
000290
000300
000310
000320
000330
000340
000350
000360
000370
000380
000390
000400
000000 TOTAL ERRORS
34536 TEXT AREA BYTES LEFT
1 'BASIC-Lader ZIDUP
10 DATA 3E,3F,CD,AF,40,21,00,5F,01,00,05,3E,09,EF,3E,60,EF,ED,43,0E,61,
F1,CS,CD,3D,3D
20 FOR I=0 TO 25:READ A$:POKE 483000+I,VAL("BH"+A$):NEXT
    
```

Bild 4. Den Editor verändert: Jetzt ist das Duplizieren von Zeilen möglich

Assemblerzeilen erscheinen als Kommentare („" heißt REM). Diese Art der Darstellung ist hier angebracht. Sie sehen ab Zeile 290 die Umsetztabelle, wobei immer das erste Zeichen (in hex.) der Modell-4-Code, das zweite der Code des Druckers ist. Sie können damit jeden Drucker anpassen. Sollten Sie mehr oder weniger als sieben Zeichen konvertieren wollen, müssen Sie die Zahl Sieben in Zeile 160 ändern.

Das Programm „patch“ direkt den Drucker, obwohl das die ideale Anwendung für ein Filterprogramm wäre. Nur ist es mir leider nicht gelungen, ein solches zum Laufen zu bringen, auch nicht, als ich ein Beispiel aus [1] vom Header bis zur Verschiebetabelle so weit zeichengetreu abgetippt habe, wie das mit einem fremden Assembler möglich ist (der Modell-4-Assembler ist noch nicht lieferbar). Auf jeden Fall muß ich einen intelligenten Fehler gemacht haben, denn die sonst so cleveren Experten bei Tandy-Deutschland konnten das Problem auch nicht lösen und haben es

an die Zentrale weitergegeben. Der Autor hofft, in mc vom Ergebnis noch berichten zu können.

```

100 '
110 DATA C3,80,02,' ORG 00E0H :PATCH KB-DRIVER
120 ' JP START
130 DATA E5 ' START PUSH HL
140 DATA C5 ' PUSH BC
150 DATA 21,98,02,' LD HL, TABLE-1
160 DATA 04,07 ' LD SP, 7 :TAB LEN
170 DATA 79 ' LD A,C :CHAR TO TEST
180 DATA 23 ' LOOP INC HL :NEXT CHAR TO TEST
190 DATA BE ' CP (HL) :IN TABLE?
195 DATA 23 ' INC HL :POINT TO NEW CHAR
200 DATA 28,09 ' JR Z, FOUND :IF
210 DATA 10,F9 ' RSTM LOOP :ELSE KEEP LOOKING
220 DATA C1 ' EXIT POP BC
230 DATA E1 ' POP HL
240 DATA DC,FB ' OUT (DE),A :POINT CHAR
250 DATA 79 ' LD A,C
260 DATA BF ' CP A :KEEP TRSDOS HAPPY
270 DATA C9 ' RET
275 DATA 7E ' FOUND LD A,(HL) :REPLACE
276 DATA 1B,F6 ' JR C,INT
280 '
290 DATA D1,5B ' TABLE DEFW 5B01H :A
300 DATA D2,5C ' DEFW 5B02H :B
310 DATA D3,5D ' DEFW 5B03H :C
320 DATA D4,78 ' DEFW 7B04H :D
330 DATA D7,7C ' DEFW 7C07H :E
340 DATA D8,7D ' DEFW 7D08H :F
350 DATA D9,7E ' DEFW 7E09H :G
360 ' END
400 FOR I=0 TO 2 :READ A$:POKE 483000+I,VAL("BH"+A$):NEXT
410 FOR I=0 TO 28 :READ A$:POKE 483000+I,VAL("BH"+A$):NEXT
    
```

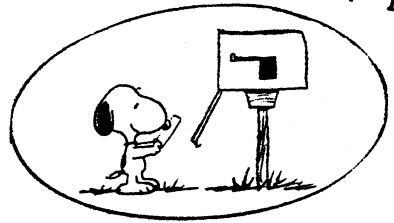
Bild 5. Es muß nicht der Tandy-Drucker sein. Mit diesem Patch kann alles angeschlossen werden, was einen deutschen Zeichensatz hat

Wie Ihnen bei allen Programmen aufgefallen sein dürfte, wurde nicht mit Mem-Size ein Speicherbereich für die Assembler-routinen geschützt. Nun, der Editor-Patch braucht das nicht, die Routine von Bild 1 nutzt den I/O-Buffer, und die letzte Routine schließlich tummelt sich in Seite 2. Dort werden die Parameter für die „Geräte“ KI DO usw. eingetragen, und zwar bis 0237H. Jedes „User-Device“ belegt acht Bytes, folglich ist mit dem ORG auf 0280H noch Platz für neun davon gelassen worden. Meines Erachtens kann man auch noch Teile der Seite 3 benutzen, wieviel, ist reine Nervensache. Das Problem: Ab 0380H wächst Ihnen ein Stack entgegen, der allerdings nicht allzu tief werden dürfte. Normalerweise liegt nämlich der Stack „hoch oben“, der untere wird nur benutzt, solange der obere infolge Bank-Switching nicht greifbar ist. Zum Schluß sei noch erwähnt, wie ohne Modell-4-Assembler die Listings erstellt wurden. Ich habe das Modell 4 im Modell-3-Modus unter NEWDOS gefahren und den NEWDOS-Assembler benutzt. Dann wurden die Objekt-Files auf eine TRSDOS-1.3-Diskette kopiert und von da (wieder im Modell-4-Modus unter TRSDOS 6.1) mittels CONV herübergezogen. Damit ist bewiesen, daß sich die TRS-80-Modelle 1, 3 und 4 im Objektcode-Format (Lader) nicht unterscheiden.

Literatur
 [1] Model 4 technical reference manual. Tandy Corporation 1983.
 [2] Röckmuth, Ludiger: Der geknackte TRS-80. mc 1981, Heft 1, Seite 46.

IN KÜRZE

nächster Redaktionsschluß ist der 28. April 1986



Programmabibliothek
 Die Programmabibliothek erscheint diesmal wieder als unsere jährliche Sonderausgabe und liegt diesem 12. INFO bei.

Programmkurzanleitungen

Endlich ist es soweit!
 Die erste Resonanz auf Eure Kurzanleitungseinsendungen wird ein Grafiksonderheft werden.
 Ob es mir möglich ist, diese Kurzanleitungssonderausgabe dieser INFO mit beizulegen, weiß ich noch nicht. Ihr werdet es aber spätestens beim Clubtreffen erhalten. Wer am Clubtreffen nicht teilnehmen kann bekommt es mit dem 13. INFO zugesandt.

CLUB GO HARD

E s h a b e n :

Video Genie

Alber Herbert
Albers Herbert
Boecker Dieter
Froehlich Burghard
Grajewski Werner
Held Manfred
Hummel Anton
Jablotschkin Rainer
Kasper Dieter
May Holger
Preuss Lothar
Rank Heinrich
Schaefer Walter
Schroeder Gerald (II S)
Sopp Arnulf (III S)
Spiess Peter
Stephan Hans-Martin
Trapp Harald
Voigtlaender Holm
Volz Oliver (IS)
Weiss Dieter
Wucherer Juergen

TRS-80 Mod. 1

Boeckling Ulrich
Koenig Hans J.
Konrad Josef
Kuhn Eckehard
Muehlenbein Klaus Juergen
Neueder Jens
Obermann Hartmut
Perschbach Patrick
Piller Walter
Raggan Hans
Retzlaff Bernd
Schneider Manfred
Wagner Alexander

TRS-80 Mod. 3

Buskowiak Thomas
Schrewe Christian
Smerling Frank
Stevens Peter
Troesch Eberhard
Wagner Guenther
Spiess Peter

TRS-80 Mod. 4

Dreyer Gerald
Hermann Klaus
Stevens Peter
~~Spiess Peter~~

Komtek

Beckhausen Wolfgang
Zwickel Walter

Star - Drucker

Albers Herbert
Grajewski Werner
Jablotschkin Rainer
May Holger
Schaefer Walter
Schroeder Gerald
Weiss Dieter
Wucherer Juergen
Zwickel Walter

HRG

Boeckling Ulrich
Dreyer Gerald
Drowaelder Bernd
Jablotschkin Rainer
Muehlenbein Klaus Juergen
Neueder Jens
Schneider Manfred
Sopp Arnulf
Spiess Peter
Weiss Dieter

Epson - Drucker

Beckhausen Wolfgang
Dreyer Gerald
Hermann Klaus
Koenig Hans J.
Muehlenbein Klaus Juergen
Piller Walter
Smerling Frank
Stevens Peter
Troesch Eberhard
Voigtlaender Holm

* DATENUEBERTRAGUNG PER FERNSEHBILD *

(C by CORNER BOX)

Der WDR-Computer-Club plant eine neue Art der Datenübertragung. Statt über den Tonkanal werden die Informationen mit dem Fernsehbild übertragen. Die Übertragung per Tonkanal hat folgende Nachteile:
1. Durch die aufwendige Anpassung des Cassetten-Recorders ist die Übertragung sehr unsicher.
2. Während der Übertragung ist der Tonkanal blockiert.

Das WDR CC-VIDEO DAT-SYSTEM erlaubt es nun, während der ganzen Sendezeit des Computer-Clubs parallel Daten zu übertragen. Diese können auch vom Video-Recorder in den Rechner geladen werden. Es ist möglich, in 30 Minuten ca. 26 DIN-A4 (das sind 50.000 Buchstaben = 50 K) zu übertragen. Gesendet werden Informationen, Programme in Ascii-Code, Basicode, Pascal und Schaltbilder.

Das Verfahren:

Im Fernsehbild werden die Zeilen 25 bis 36 entsprechend dem Ascii-Code hell getastet. So kann bei jedem 2. Halbbild ein Datenbild übertragen werden.

Der Anschluß:

Die Scart Buchse oder die AV-Buchse des Fernsehers oder Video-Recorders wird über das Interface 'Videodat 300' mit der RS 232 Schnittstelle des Computers verbunden. Als Software kann jedes DFÜ-Programm verwendet werden. Die Parameter sind: 300 Baud / 1 Startbit / 8 Datenbit / 2 Stoppbits / keine Parität.

Das Interface:

Das Interface 'Videodat 300' muß man sich selbst zusammenlöten. Die Kosten der Bauteile belaufen sich auf ca. DM 50. Ein Bausatzt ist in Vorbereitung. Dieser soll mit einem speziellem IC arbeiten. Damit können die Kosten auf ca. DM 30 gesenkt werden. Das Schaltbild bekommt ihr beim WDR Computer-Club und demnächst in unserer Box.

Viel Spaß beim Löten! Ede u. Heribert

Die Firma Schmidtke electronic machte 1984 eine bemerkenswerte Ankündigung: es sollte Mitte des Jahres (wohlgemerkt: noch befinden wir uns im Jahr 1984) eine 80-Zeichen- und Grafikkarte für ca. 500 DM angeboten werden. Diese Karte sollte neben 80 Zeichen, 25 Zeilen auch 720 x 480 Grafikpunkte, einen eigenen Prozessor, einen 64K-Speicher, eine parallele und zwei serielle Schnittstellen bieten.

Zuerst verschob sich die Vorstellung auf den September, dann wurden diesbezügliche Anfragen gar nicht mehr beantwortet.

Mitte 1985 erfuhr ich dann, daß Schmidtke electronic endlich die Karte anbot. In der Anzeige war nun von 80 Zeichen bei 25 Zeilen und 4 Zeichensätzen die Rede: Deutsch + Invers oder Grafik, Amerikanisch + Invers oder Grafik. Außerdem wurde auf die Software und ein umfangreiches Handbuch hingewiesen. Es hörte sich gut an und war mit 245 DM (inzwischen auf 295 DM erhöht) noch erschwinglich. Also her damit.

Mitte November 1985 bestellte ich die Karte mit einem Scheck. Mitte Januar 1986 kam sie dann endlich. Diese Wartezeit ist bei Schmidtke noch guter Durchschnitt. Die Karte selber sieht dürftig aus: Es hätte doppelt so viel darauf gepaßt.

Der Einbau in mein Genie IIS war recht einfach, aber er nichts ist gegen den in das Genie I/II. Zuerst mußten zwei ICs ausgetauscht werden. Kein Problem. Dann sollte die Karte in einen freien Slot gesteckt und mit dem Videokabel verbunden werden. Dieses Videokabel war nur leider zu kurz. Selbst nach Umstecken der anderen Karten erreichte es nur unter äußerster Spannung die Karte. Bei dieser Spannung brachen dann auch die Lötstellen am Stecker nach außen. Ein neues Kabel und eine Videobuchse mußten besorgt werden, denn die alte war Hongkong-made, sprich: verklebt statt geschraubt.

Nun lief die Karte endlich. Aber das Bild auf dem Monitor mußte erst langwierig nachgestellt werden, bis sowohl im 80- als auch im 64-Zeichen-Modus alles erkennbar war. Beim Umschalten zwischen den beiden muß jedes Mal die Helligkeit verändert werden, denn im 80er-Modus sind die Zeichen heller.

Nun wurde das "Handbuch" studiert. Eigentlich ist es kein Handbuch, sondern eine Montageanleitung. 21 der 32 Seiten bestehen aus Schaltplänen (dabei wollte ich die Karte eigentlich nicht nachbauen) und der Einbauanleitung für das Genie I/II. Der Rest beschreibt kurz die drei Programme auf der Diskette und noch kürzer die Programmierung der Karte. Auf der ersten Seite wird extra gesagt: Januar 1986 erscheint das "richtige" Handbuch, mit dem beiliegenden "Gutschein" kann man es anfordern. Nur: erstens lag kein Gutschein bei und zweitens habe ich die Karte im Januar 1986 bekommen, also dürfte sich das Handbuch noch im Kopf des Autors befinden (falls es jemand inzwischen irgendwo aufgetrieben hat, wäre ich stark daran interessiert).

Die Programme sind schlicht, um nicht zu sagen simpel. Eines testet die Karte und zeigt dabei die verschiedenen Zeichensätze an. Dabei stellt sich auch heraus, daß das oben erwähnte "Grafik" eine geschickte Abkürzung für "Grafikzeichen" ist, die genauso öde wie nutzlos sind (bis auf die Kartensymbole für Kreuz, Pik, Herz und Karo). Es gibt auf der Karte keine Grafik, nicht mal eine Entprechung für die SET/RESET-Grafik. Die 720 x 400 Punkte sind verschwunden. Ebenso der 64K-RAM-Speicher: Die 2048 Zeichen werden bei B000H oder F000H eingeblendet. Die Karte hat genau 4K Speicher. Dafür kann die Karte aber die vollen 64K des Computers für CP/M verfügbar machen. Erweiterungen wie MBA / EG64 / LSS1 werden damit überflüssig.

Von Schnittstellen keine Spur: Es gibt nur einen Anschluß für ein Netzgerät, falls man eins benötigt. Im Genie IIS entfällt dies.

Weiter mit den Programmen: Das eine schaltet den 80-Zeichen-Modus an, das andere ihn wieder aus. Interessant ist nur das erste. Es belegt den Speicher über

F000H und leitet alle normalen Bildschirmausgaben auf die Karte. Wenn ein Programm allerdings nicht schön gesittet diese Routinen benutzt, sondern lieber poked, ist Sensor 80 Zeichen ade.

Dieses Programm ermöglicht es, über PRINT CHR\$(7) + Steuercode im Basic die verschiedenen Schriftsätze anzuwählen und sogar ein PRINT\$ bis 2047 durchzuführen. Dafür laufen dann andere Hilfsprogramme vielleicht nicht mehr. Ich habe das bisher nur mit dem Bildschirmeditor ausprobiert, und der meldet sich ab. Was soll ich mit fantastischen 80 Zeichen und diesem alten, total untauglichen Zeileneditor? Zum Heulen!

Andere Programme laufen von Fall zu Fall. Edtasm läuft im Editormodus, aber Assembeln tut er nicht immer, Scripsit meldet sich ab.

Dem könnte ja abgeholfen werden, indem man diese Programme so abändert, daß sie selbst die Karte programmieren. Würde vielleicht klappen, wenn man wüßte, wie! Aber im "Handbuch" steht nur, daß die Ports D0-D2 benutzt werden und die Werte, die man in das Steuerregister D2 schreiben muß, um bestimmte Reaktionen zu erreichen. Über D0 und D1 schweigt es sich aus, aber sie werden von dem Treiberprogramm angesprochen. Nur was es mit ihnen macht und warum, bleibt im Dunkeln. So, ich glaube, jetzt habe ich die Karte genug verrissen. Noch einige Anmerkungen:

Es gibt die Karte für TRS80, Speedmaster, Genie I/II/IIS und das Colour Genie. Löhnen dürfte sie sich nur, wenn man intensiv CP/M 2.2 nutzen will. Dann ist es D I E Erweiterung. Ansonsten würde ich beim Genie IIS wohl eher die TCS-Karte und für die anderen Computer eine Grafikkarte empfehlen.

Falls mir jemand mehr Informationen über die Karte geben kann oder schon Programme angepaßt hat: könnte er sich freundlicherweise bei mir melden? Das würde meine Laune stark verbessern.

Gerald Schröder

Wer Interesse an einer ausführlichen Anleitung zu dieser Zeichenkarte hat, melde sich bitte bei der Redaktion.

Der Chefprogrammierer liest die Computerdiagnose und erbleicht: Sie hat ergeben, daß sich die Firma keinen Computer mehr leisten kann.

*»Was hältst Du von Großrechnern, Albert?«
»Von diesen Strebern halte ich gar nichts; ich jedenfalls war in der Schule immer ein kleiner Rechner.«*

Speed Writing

by Joe Lindsey

If you own an Epson MX-series printer, you're wasting time. That's because your Epson prints at a fraction of its potential. Imagine a 40 percent increase in printing speed—that's what you'll get with this modification. By replacing your printer's 6-megahertz (MHz) crystals with ones rated at 10 MHz, you can print out both text and graphics faster. And you can make this simple modification for about \$6 and a half hour of your time.

Here's how the modification works: The Epson's 6-MHz crystals govern the printer's master clock. When you change the crystals, you change the clock frequency. The faster the system clock, the more instructions-per-second the printer's central processing unit (CPU) executes. And when the CPU goes faster, the Epson goes faster.

Printer Fix

You first need to buy two 10-MHz crystals from an electronics supply house. You also need a Phillips screwdriver, some masking tape, and a soldering iron.

Disconnect the Epson's external cable and power cord, put your printer on a table or other level surface, and you're ready to begin.

Pull the manual paper feed knob off the right-hand side of the printer and remove the smoked plastic cover. Flip it up to a vertical position and lift it out, left side first.

On the MX-100, reach inside the right side of the printer (by the control buttons) and unplug the cable to the buttons. This will make it easier to remove the case. Look along the front of the printer; you'll see three screws where the base of the gray cover was, and one screw in each of the back corners. Remove all five screws and lift off the case.

Turn the printer upside down, making sure you don't damage anything that sticks out of the top, and unscrew the four Phillips screws at each corner of the bottom of the case. Cover the holes with masking tape to keep the screws from falling out.

Turn the printer right side up and remove the top cover by lifting up on the left side and sliding the cover off the paper feed shaft that sticks out the right side.

You'll see some tied wires running up to the printer's control panel on the right side of the machine (see Photo 1). Unplug the connector in the top cover at the operator panel and the whole cover should come free. Set it aside for now.

Turn the printer around so the back

faces you. I'll refer to the right and left sides as you view the printer from the rear; I'll use "front" and "rear" in their true senses.

Look carefully at your printer from the rear to get acquainted with its components. The power supply appears on your left. A large control circuit board lies at the bottom of the machine behind the paper feed mechanism. A smaller board, called the driver circuit board, sits on the right side and on top of the control board.

Remove the driver board from the control board by removing the two Phillips screws along the right side of the driver board (see Photo 2). The left side of the driver board plugs into a couple of sockets (CN4 and CN5) on the control board. Carefully work the pins out of the sockets underneath, making sure you don't bend them. You may want to remove the big, blue male plug (CN6) on the right side of the driver board. I left mine plugged in and let the whole board dangle by the attached wiring. Remove the driver board by grasping it and working it loose while pulling straight up.

A single ground wire connector (FG) attaches to the rear of the control board, just to the left of the external printer connector (CN1). Slip it off the pin onto which it plugs. You need to remove two power supply connectors (CN2 and CN6), both of which are on the left rear corner of the control board, near the largest of the two capacitors mounted there (C1) (see Photo 3).

To free the control board, you have to remove three Phillips screws. One is in the center of the printer just behind the paper feed assembly, between the integrated circuits labeled 2C and 3C (midway between the two white standoff posts protruding up through the control board from the bottom). The other two screws sit at the left side of the board in front of the power transformer (see Photo 4).

The only things holding the control board in place now are three plastic tabs molded into the base, two along the rear of the board, the third on the right side. Push each plastic tab out of the way enough to clear the board, then lift the corner of the board out from under the tab (see Photo 5).

System Requirements

Epson MX-series printer

A simple, inexpensive modification to Epson MX-series printers that increases printing speed by up to 40 percent.

Raise the control board over the standoffs molded to the base and slide it out from under the paper feed assembly. The board is made of a paper laminate material that breaks and chips easily, so be careful when removing it. Also, use caution when soldering or unsoldering, as excessive heat will destroy the board.

Unsolder the two 6-MHz crystals (CR1 and CR2) from the control board and solder the 10-MHz crystals in their place, keeping the lead lengths the same as the original (see Photo 6). CR1 is located toward the left and on the front of the control board just to the left of the large integrated circuit labeled 1C. CR2 is on the right side of the control board just in front of the large integrated circuit labeled 9B.

The crystals in my printer were also held in place by double stick tape. I had to pry up on them slightly to break the tape bond, but the tape was still sticky enough so that I didn't have to replace it. The new crystals came encased in metal, so I stuck a small piece of black electrical tape to the bottoms to keep them from shorting out to any circuitry close by.

Now you can reassemble your printer, reversing the above procedure. Be sure you put all connectors back in their original positions without bending any of the pins. Make sure that the connectors line

up properly between the driver board and main printed circuit board. Don't force anything. Additionally, when you reinstall the PC board you need to push each of the plastic tabs out of the way before you gently push the board into place. Don't expect the board to move the tabs. Finally, don't forget the control panel plug in the top cover of the printer.

Printer Ready

The operation I've just described for speeding up your MX-80 is reversible—you can put the old 6-MHz crystals back in and your printer will work the way it did before. If you ever need to take your printer in for service, you may need to put the original crystals back in to return it to "normal" speed. Epson won't fix a modified printer without unmodifying it, and you can't exchange any boards to which you've soldered.

I do offer one caveat: I wouldn't try to increase the speed over 10 MHz, since the CPU and memory can't safely drive the Epson any faster than that.

I've noticed no change in print quality and no skipping of dots after increasing the speed of my MX-80, even in heavy graphics printing. You'll hear a noticeable change in pitch as it prints, but I don't think it's any louder. Another positive re-

sult is that the skip and space speed of the paper feed has also increased. ■

Support für Genie

Den Support der Recher Genie I und II übernimmt ab sofort die Firma Schmidtke Computersysteme, Sandkaulstr. 41, 5100 Aachen. Produktion, Vertrieb und Support der aktuellen Modelle (Genie II, Genie III und Genie 16) bleiben unverändert in den Händen der Firma Phoenix-Computer.

Phoenix-Computer, Computer GmbH & CO KG, Gewerbegebiet Klarensplatz, 5461 Windhagen, 0 26 45 3222

Tandy-Computer mit 80286

In den gehobenen Bereich der MS-DOS-Computer ist der neue Tandy-3000 einzuzordnen. Diese Tatsache belegt nicht nur die Wahl der CPU (80286 mit 8 MHz Taktfrequenz), sondern auch der Arbeitsspeicher von 512 KByte in der Standardversion, der ohne Erweiterungskarten auf 640 KByte ausgebaut werden kann. Das eingebaute Diskettenlaufwerk bringt auf jeder Scheibe 1,2 MByte unter, kann aber auch IBM's Standardformat mit 360 KByte lesen. Das Betriebssystem, MS-DOS 3.1, gehört nicht zur Grundausstattung, kann aber getrennt bezogen werden. Für Mehrplatzanwendungen bis zu sechs Benutzern wird künftig Xenix 5.0 auf dem Tandy-3000 laufen. Als lokales Netzwerk für

MS-DOS-Computer bietet Tandy ViaNet an. Das Gerät hat insgesamt zehn Erweiterungssteckplätze: Sieben davon sind kompatibel zum IBM-PC/AT und zwei zum IBM-PC/XT. Ein weiterer „XT“-Steckplatz kann für kurze Karten verwendet werden; er ist normalerweise von der Parallel-/Seriel-Schnittstelle belegt, die zur Grundausstattung gehört. Wer viel Massenspeicher braucht, kann sich für das Modell HD entscheiden, das eine 20-MByte-Festplatte enthält. Aber auch beim Grundmodell kann der Massenspeicher auf über 40 MByte ausgebaut werden. Für rechenintensive Aufgaben gibt es den Arithmetikprozessor 80287 als Zusatz.



Mit der Taktfrequenz von 8 MHz läuft der Prozessor 80286 im Tandy-3000

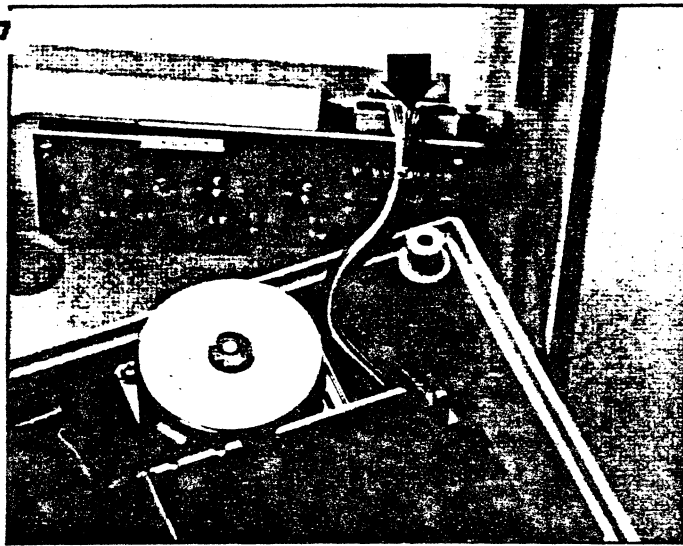


Photo 1. Unplug the connector at the operator panel to remove the cover.

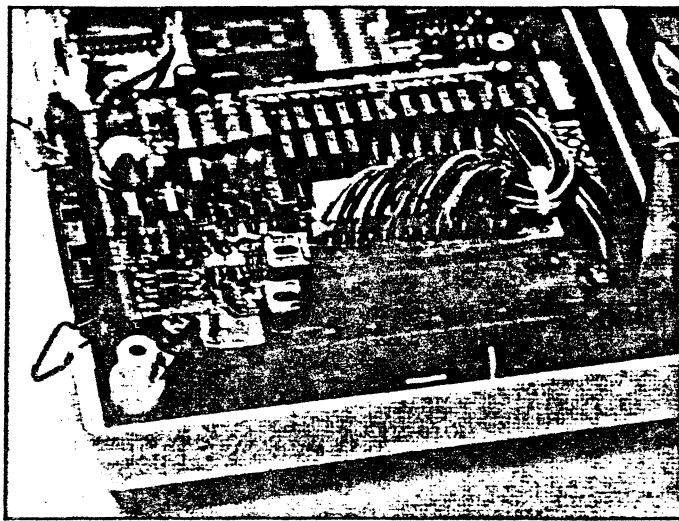


Photo 2. Remove the driver board from the control board by removing the two Phillips screws along the right side of the driver board.

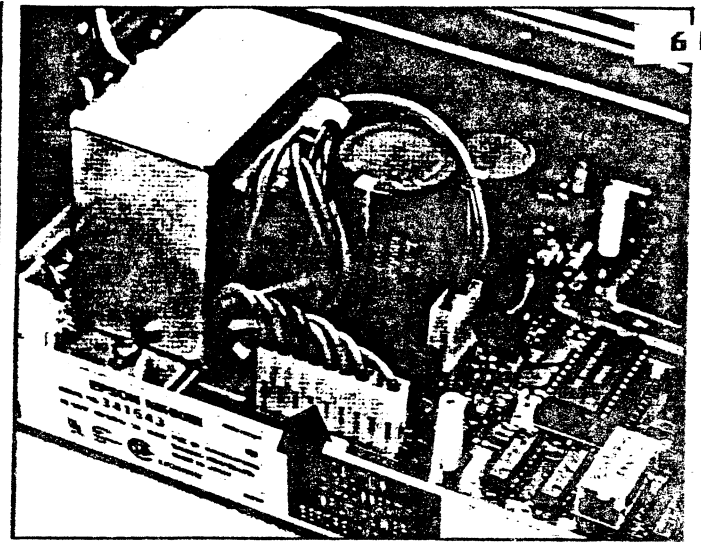


Photo 3. Remove CN2 and CN6 from the left rear corner of the control board.

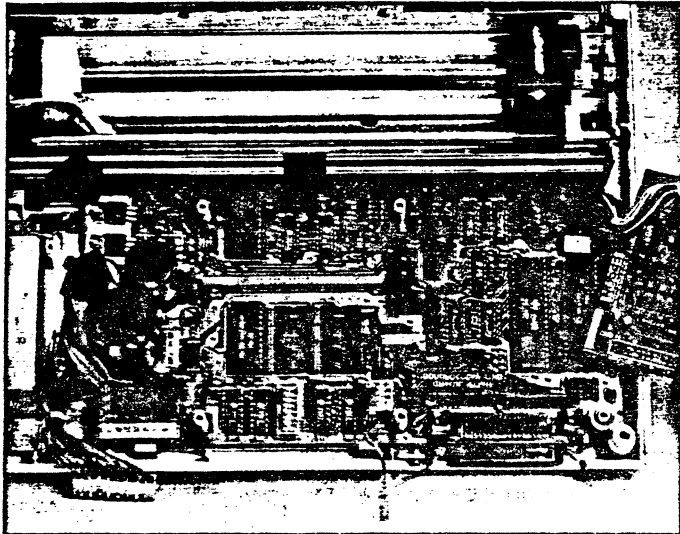


Photo 4. Free the control board by removing the three Phillips screws at the left side of the control board.

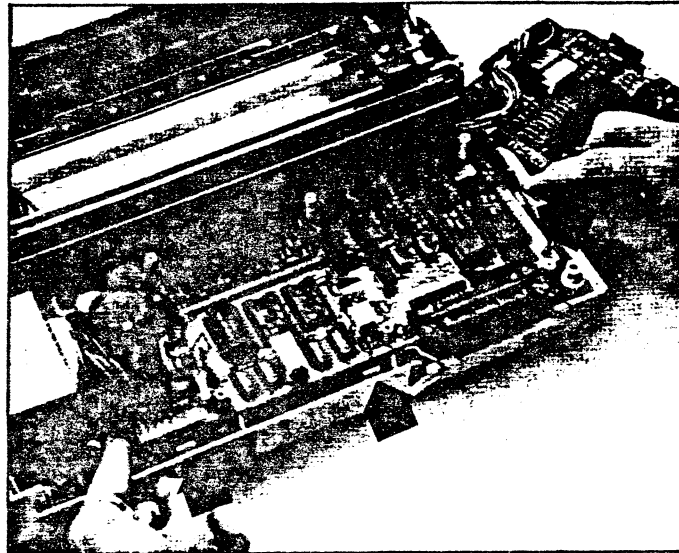


Photo 5. Push each of the three plastic tabs on the base of the printer out of the way.

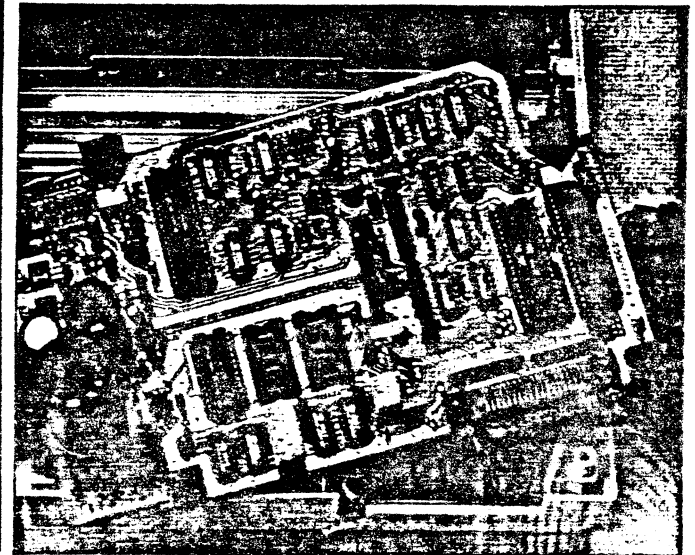


Photo 6. Unsolder the two crystals CR1 and CR2 from the control board.

TRS-80 Modell 4P: Der schönste Portable der Welt



Daß ein Portabler nicht immer ein häßlicher schwarzer Kasten sein muß, hat Tandy mit dem Modell 4P eindeutig bewiesen. Ich habe noch keinen Portablen gefunden, der besser aussieht. Blicke die Frage, was hinter der schönen Fassade steckt? Ein Anwender des großen Modell 4 hat den Kleinen getestet.

Eine englische Fachzeitschrift schrieb unter der Überschrift 'Tandy's Sweet Pea', das 4P sei ein M4 'with wanderlust'. Längere Wanderungen dürften damit allerdings nicht sehr lustig werden, denn immerhin wiegt der Kleine 11 kg, so klein ist er also gar nicht. Das liegt an der sehr soliden Konstruktion, ein Portable muß schließlich rauhe Behandlung vertragen.

Die Bildröhre hat 9 Zoll Diagonale und damit ein Format, das als angenehm bezeichnet werden kann. Womit auch gesagt sein soll: die 7-Zoll-

Formate der Mitbewerber sind bei 80 Zeichen/Zeile meines Erachtens unerträglich.

Die Frontseite ist durch einen Deckel abgedeckt, der nach Lösen von zwei Schnellspannern abgenommen wird. Der Tragegriff ist auf der Rückseite, das Gerät wird also mit dem Schirm nach unten transportiert. Im Deckel ist Platz für das Netzkabel, ein Fach für das Pocket-Manual und 4 bis 5 Disketten.

Auf der Rückseite verbergen sich unter einer Klappe die Buchsen für das Netzkabel, eine RS232- und eine Centronics-Schnittstelle, Anschlüsse für den Rechner-Bus und für bis

zu vier Festplatten. Eine Tonband-Buchse fehlt.

Auf der Vorderseite links vom Schirm sind der beleuchtete Netzschalter und die Regler für Helligkeit und Kontrast. Die Reset-Taste ist unter dem Netzschalter angebracht und damit so gut versteckt, daß man sie garantiert nie versehentlich auslöst.

Auch innen sieht es gut aus

Das Gehäuse ist recht einfach zu entfernen: Zwei Schrauben auf jeder Seite und die beiden unter dem Griff lösen, Gehäuse abziehen. Nach dem Lösen zahlloser Schrauben müßte man an die Einzelteile herankommen, mit wenig Aufwand läßt sich aber feststellen, daß das CPU-Board die Unterseite belegt, links steht senkrecht die Video-Platte, rechts die Laufwerk-Elektronik, den Deckel bildet sozusagen das Netzteil. Die Baugruppen sind untereinander nochmals mit Blechen abgeschirmt; jetzt weiß man langsam, wo die 11 kg herkommen. Immerhin, lieber eine solide Konstruktion, die auch mal einen Bums verträgt, als ein empfindliches Leichtgewicht. Auf der Rückseite sieht man nach Abnehmen eines Blechdeckels in einen Schacht, der noch eine Platine aufnehmen kann, nämlich die für das in Deutschland nicht zugelassene Modem. Auch der CPU-Bus ist mit einem Blechdeckel abgedeckt.

Vorbildliche Tastatur

Die Tastatur ist beim Transport in einem Fach unter dem Bildschirm gelagert, man muß sie da herausziehen. Leider ist das Kabel etwas kurz geraten. Zur ergonomischen Anordnung auf dem Tisch reicht es noch, nicht aber für die amerikanische Kreativ-Haltung, sprich Füße auf dem Tisch und Tastatur auf dem

Schoß. Um gleich bei der Tastatur zu bleiben: Sie ist extrem flach, kann mit zwei ausschwenkbaren Füßen schräg gestellt werden und hat 70 beziehungsweise 75 (deutsche) Tasten. Die Anordnung ist sehr gut (besser als beim M4), Shift und Enter sind deutlich abgesetzt. Die Tasten lassen sich gut bedienen und haben einen deutlich hörbaren Anschlag, allerdings nicht so blechern, wie beim IBM-PC. Auch schnelle Tipper können die Tastatur nicht schlagen. Alle Tasten haben eine Wiederholautomatik, die Ansprechverzögerung und die Wiederholfrequenz sind per Software einstellbar, die Grundeinstellung ist aber meines Erachtens schon genau richtig.

Beim Bildschirm hat sich viel getan

Das M4 hat als deutsche Version immer noch den Fehler, bei großen Grafik-Flächen ständig und bei Texten im 80-Zeichen-Mode manchmal zu flimmern. Beim 4P ist das Bild gestochen scharf, und steht wie eine Eins. Obwohl der Schirm kleiner ist (9 statt 12 Zoll), läßt er sich deshalb leichter ablesen. Die Option, auf 40 Zeichen/Zeile umzuschalten, habe ich mal probiert (sie funktioniert), aber sonst nie angewendet. Tatsächlich hat man die Qual der Wahl zwischen 24 Zeilen mal 80 Zeichen, 24 mal 40 und 16 mal 64.

Erste Inbetriebnahme in drei Sprachen

Das Modell 4P soll international sein, was an sich kein Problem ist, man muß halt nur die entsprechende nationale Diskette einlegen, von der dann der Zeichensatz geladen wird. Dafür hat der Computer einen Bootstrap-ROM (Urlader). Alle Fehlermeldungen und Prompts erscheinen in Englisch, Deutsch und Französisch, womit wir wissen, was der

Hersteller für die drei wichtigsten Sprachen der Welt hält. Daß sich damit der Bootstrap-ROM vom M4 unterscheidet, ist klar, allerdings noch in einem anderen Punkt. Er erwartet TRSDOS 6.1.1 oder neuer. Eine Diskette der Version 6.0.0 oder 6.1.0 bootet nicht. Solche Disketten können allerdings ohne Probleme geladen, und mit einem mitgelieferten Programm auch 'upgedatet' werden. Letzteres spielt nur das neue DOS auf, läßt aber alle anderen Dateien unberührt.

Ein weiteres Problem: Das Testgerät hat noch keine deutsche Tastatur. Die Lösung: Ich schreibe die Umlaute aus und lasse sie auf dem M4 über die globale Ersetzfunktion wieder in 'stöuß' wandeln.

Laut Prospekt besteht kein Unterschied zwischen dem M4 und dem 4P, vom Äußeren einmal abgesehen. Im M4 ist ein Modell 3 physikalisch echt enthalten. Tatsächlich fehlt dem 4P aber das Modell-3-ROM. Beim M4 ist es so, daß das ROM dem RAM adreßmäßig parallel liegt, und per Bank-Select automatisch das eine oder andere gewählt wird, je nachdem, welche Diskette man einlegt. Beim 4P funktioniert das im Prinzip auch so, nur daß das sogenannte ROM-Image auf der Diskette vorhanden sein muß. Diese wird beim Systemstart zuerst in das RAM geladen, was natürlich einige Sekunden dauert. Hat man kein ROM-Image auf der Diskette, kann man mit der zum Lieferumfang gehörenden Software ein solches darauf kopieren. Ist dafür kein Platz mehr, muß man mit einer anderen Diskette starten und dann die Scheibe wechseln. Dies mag auf den ersten Blick ein Nachteil sein, Hacker aber werden jubeln. Haben Sie doch jetzt die Möglichkeit, ein Modell 3 mit maßgeschneidertem 'ROM' anzufertigen.

Ich habe einige Modell-3-Software getestet, auch unter einem fremden



DOS (NEWDOS/80), verschiedene Programme eingesetzt, die exzessiv ROM-Ware benutzen: kein Fehler feststellbar. Im Modell-4-Mode läuft das System von Haus aus (Lieferumfang) unter TRSDOS 6.1, ein Betriebssystem, das keine Wünsche offen läßt. Computer Nr. 3 ist eine reine CP/M-Maschine, vorausgesetzt, Sie starten mit einer passenden Diskette. Von Tandy lange angekündigt, ist CP/M 3.0.

Drei Computer in einem

Dazu muß man wissen, daß das 4P von Haus aus 64 KByte RAM hat, das M4 ist darauf aufrüstbar.

Beide bieten die Möglichkeit, auf 128 K zu erweitern (329 Mark). Diese letzten 64 KByte sind aber nur über Bank-Select zu adressieren, was TRSDOS 6.1 perfekt unterstützt, nicht aber CP/M so ohne weiteres. Hier dürfte das Problem in der Anpassung von CP/M 3.0 gelegen haben. Ungeduldige haben inzwischen CP/M 2.2 gekauft. Vielleicht war die Idee gar nicht so schlecht. Dieses CP/M soll laut Tandy zwar nicht perfekt sein, es hat aber einen Vorteil: Es liest und schreibt im Format von IBM, Kaypro, Osborne und anderen.

Welches Betriebssystem, das ist die Frage!

TRSDOS 6.1 ist für das Modell 4 maßgeschneidert und eindeutig benutzerfreundlicher als CP/M, letzteres hat aber den Vorteil, daß es ein riesiges Softwarepotential dafür gibt. Das heißt, wenn Sie unbedingt Software benötigen, für die unter TRSDOS nichts Gleichwertiges angeboten wird, müssen Sie wohl oder übel in den teuren Apfel beißen.

Eine Zukunftsinvestition dürfte aber der Kauf von CP/M nicht sein. Der Schwerpunkt liegt eindeutig auf TRSDOS, was durch eine Open-System-Politik unterstützt wird. Das technische Manual das M4 (und 4P) ist ein Musterbeispiel dafür. Die Hard- und Software ist völlig offen gelegt, für einen Systemprogrammierer bleibt keine Frage unbeantwortet. Daß der Hersteller inzwischen selbst exzellente Software entwickelt, zeigt das ALDS, ein Super-Assembler, der beim Autor gerade im Test ist; aber auch die Open-System-Politik trägt Früchte, wie Alcor-Pascal und Multiplan zeigen.

Mein Rat: Kaufen Sie zuerst TRSDOS 1.3, das Betriebssystem des Modell 3. Damit steht Ihnen schon einmal eine riesige Programm-Bibliothek zur Verfügung, in der Regel wesentlich preiswerter als CP/M-Software. Dann schauen Sie sich das M4-Angebot an, und wenn dann noch was fehlt, versuchen Sie Passendes unter CP/M zu finden.

Disk-Drives und Geräusche

Zwei Disketten und dann nur noch Festplatten als Option. Das heißt, für unterwegs muß man sich mit 360 KByte Massenspeicher (einseitige Aufzeichnung) »bescheiden«, zu Hause darf man dann auf 20 MByte aufrüsten. Das M4 erlaubt, noch zwei externe Laufwerke anzuschließen, die auch »double sided« sein dürfen, es ist also im Prinzip preiswerter mit Massenspeicher auszurüsten. Im M4 klappern die Laufwerke ganz schön, was das große Gehäuse als Resonanzboden noch kräftig unterstützt, die 4P-Laufwerke sind dagegen wohlthuend ruhig und deutlich schneller. Ich habe ein 1100-Zeilen-Programm mit einem Assembler übersetzt, der dazu ständig auf die Disk, wo der Source-Text liegt, zugreift. Auf dem M4 dauert das 80 Sekunden, auf dem 4P nur 66.

Was gibt es an Software?

Mitgeliefert wird TRSDOS 6.1.1 und Basic. Vom TRSDOS 6.1.0 gibt es auch eine Version für die deutsche Tastatur, Sie sollten allerdings auf der Version 6.1.1.D bestehen, die noch besser eingedeutscht ist, zum

Beispiel beim Datum. TRSDOS 6.x ist ein exzellentes Betriebssystem, das keine Wünsche offen läßt. Wenn Sie NEWDOS kennen, wissen Sie, daß man da lange Parameterlisten mit unverständlichen Kürzeln eingeben muß, oder Ersatz-Werte durch viele Kommas zu definieren hat. Die vielen Parameter sind bei TRSDOS6 auch möglich, nur daß dafür Klartext oder auch nur die Anfangsbuchstaben zugelassen sind. Aber das Wesentliche: Wissen Sie die Parameter nicht, geben Sie einfach den Befehl, zum Beispiel BACKUP ein, und Sie werden gefragt.

Besonders zu erwähnen wäre noch MEMDISK, eine sehr einfache zu bedienende RAM-Floppy und SPOOL, ein Spooler, der im einfachsten Fall durch Nennung seines Namens aktiviert wird. Allein diese beiden Programme sind Grund genug, die 128-KByte-Erweiterung, die jetzt auch in Deutschland erhältlich ist, zu kaufen. Es ist schon ein erhebendes Gefühl, einen großen File auf »Disk« zu schreiben, und er ist da, sobald man die Enter-Taste wieder losgelassen hat.

Ansonsten gibt es noch einige Programme für Fortgeschrittene, wozu zum Beispiel die Communication-Software (RS232) zu zählen ist.

Die läßt zwar das Herz eines jeden Hackers höher schlagen — es fehlt nichts, was einen daran hindern könnte, fremde Großrechner anzupapern — aber die Bedienung ist doch arg umständlich. Diese Umstände könnte man zwar auch einem ICL-File anvertrauen, die Job Control Language ist auch einsame Spitze, aber leider von Profis für Profis gemacht. Wer sich nicht scheut, den ausführlichen Anhang zu diesem Thema zu studieren (lesen reicht nicht), hat dann allerdings auch ein

mächtiges Instrument zur Verfügung. ICL ist nicht nur ein Textfile, der anstelle der Tastatur gelesen wird, sondern schon eine Programmiersprache für sich.

Gut gelöst ist in TRSDOS 6.x das Konzept der »logischen Geräte«. Das heißt, Tastatur, Bildschirm, Drucker, die RS-232 und Disk-Files sind im Prinzip alle gleich zu behandeln. Mit LINK kann man die »Geräte« miteinander verbinden, zum Beispiel Schirm und Drucker, mit dem Erfolg, daß man sieht, was gedruckt wird. Mit ROUTE wird umgeleitet, zum Beispiel der Drucker auf den Schirm, womit sich die Formatierung überprüfen läßt, ohne Papier zu verbrauchen. In alle »Links« kann man sogenannte Filterprogramme schalten, um Zeichen umzuwandeln, welche hinzuzufügen (zum Beispiel zu jedem CR ein LF) oder andere zu »strippen«, sprich zu unterdrücken. Einige Filterprogramme werden mitgeliefert, wovon FORMS/FLT (Druckformat) und KSM/FLT (vom Anwender zu definierende Tasten-Makros) besonders hervorzuheben sind.

Das Basic ist mehr als erfreulich

Was sich da im Copyright mit Basic 1.0 meldet, ist Microsoft-Basic Version 5.

Die besonderen Eigenschaften dieses Basics erkennt man erst, wenn man sehr tief (mit einem Disassembler) hineinschaut. So werden zum Beispiel Konstanten, auch hexadezimale und oktale, sofort nach der Eingabe in das rechnerinterne Format umgewandelt und so im Programmtext zusammen mit einem Typ-Byte gespeichert. Das heißt,

der alte Trick, häufig benutzte Konstanten als erste Variable zu definieren, ist hier unangebracht, Konstanten sind immer schneller!

Aber auch wenn man das Basic nur von außen (laut Handbuch) kennenlernt, wird man erfreut sein. Zuerst: Wer Level 2 oder Basic-80 beherrscht, kann sofort loslegen, wenn er sich bitte nur eines merkt: Reservierte Worte sind mit Blanks zu trennen. Tatsächlich ist Level 4, wenn ich das mal so nennen darf, eine Erweiterung von Level 2 oder 3.

Über die neuen Befehle sollte man allerdings gründlich nachdenken. Auf Seite 32 unten links sehen Sie zum Beispiel, wie man mit WHILE/WEND und SWAP (vertauschen von zwei Variablen) ein Sortierprogramm mit einer in Basic bisher nicht möglichen Effektivität schreiben kann. Die langen Variablennamen sind echt, bis zu 40 signifikante Zeichen sind jetzt möglich. Das Bemühen, volle Kompatibilität zu den Vorgängern zu erhalten, führt allerdings auch zu einiger Redundanz, sogar an falscher Stelle.

So gibt es nach wie vor USR, obwohl der neue Befehl 'CALL Name (Parameterliste)' erlaubt, beliebig viele Routinen aufzurufen (andere Namen) und alle mit soviel Parametern, wie in einer Basic-Zeile von 250 Zeichen unterzubringen sind.

Diesem sehr guten Basic nicht mehr angemessen ist der Editor, der genau der des alten Level 2-Basics ist. Ich würde empfehlen, Tandys ALDS zu kaufen. Dieses Assembler-Entwicklungs-System beinhaltet einen erstklassigen Editor, der auch auf Basic-Files (ASCII-Format) anwendbar ist. Einschränkung: Basic-Zeilen dürfen dann nicht länger als 80 Zeichen sein, deutsche Umlaute werden verfälscht.

Alle Programme der Modelle 1 und 3 laufen auch auf dem 4, sofern Sie jedes PEEK und POKE und alles was sonst mit den Maschinen-Internas zu tun hat, strikt vermeiden. Wenn Sie die Programme vom Modell 1 mit REPAIR beziehungsweise vom M 3 mit CONV übertragen, dann bitte nur im ASCII-Format. PRINT heißt zwar immer noch PRINT, ist aber intern anders codiert, die sogenannten Tokens haben alle eine andere Bedeutung. Vergißt man das, erhält man sehr lustige Listings, nur laufen tun sie nicht. REPAIR und CONV versagen aber bei Modell 3-Disketten, die unter NEWDOS erstellt wurden.

```

100 ' Demo Bubblesort mit WHILE/WEND und SWAP
110 '-----
120 '
130 'Array A mit Zufallszahlen laden:
140 FOR I=0 TO 10:A(I)=RND(100):NEXT
150 '
160 'Start Sort
170 '-----
180 FLAG=1
190 WHILE FLAG
200   FLAG=0
210   FOR I=0 TO 9
220     IF A(I)>A(I+1) THEN
                SWAP A(I),A(I+1):
                FLAG=1
230   NEXT I
240 WEND
250 '
260 'Print sorted data
270 '-----
280 FOR I=0 TO 9:PRINT A(I):NEXT
290 '
300 END

```

◀ So einfach kann man mit dem neuen Basic ein Sortierprogramm schreiben.

Die Anpassung eines Druckers ist an sich kein Problem, sollte aber im Manual besser beschrieben sein. ▼

```

.LA180/JCL
.Set up for RS-232 -> DEC-Printer LA-180

SET #CL TO COM/DVR
SETCOM (BAUD=2400,CTS=YES)
ROUTE #PR TO #CL

.LA-180 ready to use
//EXIT

```

Also schnell eine TRSDOS-1.3-Diskette besorgen (27 Mark) und Zwischenkopien erstellen. Für Grafik-Fans eine schlechte und zwei gute Nachrichten, zuerst die schlechte: SET, RESET und POINT sind entfallen. Die erste gute Nachricht: Es gibt eine Grafik-Karte für 768 Mark mit einer extrem hohen Auflösung und mächtigen Grafik-Befehlen.

Die zweite gute Nachricht: Computer persönlich bringt demnächst ein Programm, das die Blockgrafik-Befehle wieder installiert, und das mit doppelt so hoher Auflösung als beim Modell 3.

Beim M4 hat man die Buchse für den Kassetten-Recorder noch belassen, wohl, weil es das Gerät als Billig-Version auch ohne Disks gibt; beim 4P fehlt sie und folglich auch CLOAD und CSAVE. Somit dürfen 4P-Besitzer auch TAPE100 löschen, was Tandy vergessen hat. Das ist das Programm zum Einlesen von Modell-100-Kassetten. Sie können das nicht löschen? Nun, das Paßwort heißt UTILITY.

Die Dokumentation: Qualität.

In diesem Punkt war Tandy vom Inhalt her schon immer gut, jetzt auch in der äußeren Form. Mein M4-Manual wurde noch als Kopie von Schreibmaschinenseiten im Pappdeckel geliefert, die 4P-Anleitung ist ein Ring-Ordner in Kunstleder mit Goldeindruck. Die Seiten sind richtig gesetzt und offensichtlich von einem Layouter gestaltet, das heißt angenehm und klar zu lesen, sofern man Englisch beherrscht. Da Tandy aber inzwischen Deutschland entdeckt hat, was Tastatur und DOS schon beweisen, ist auch hier noch Besserung zu erwarten.

Vom Inhalt her ist fast nichts zu bemängeln, bestenfalls zuviel Ausführlichkeit, aber da gibt es ja noch das Pocket-Manual, mit dem ich gern praktisch arbeite.

Das »fast« mußte ich geben, weil ich die Kapitel über die RS232 nie verstanden hätte, wenn ich nicht schon einiges darüber wüßte und nicht wie wild im Manual geblättert hätte. COM/DVR, SETCOM und COMM/CMD gehören nämlich zusammen, sind aber im Handbuch völlig getrennt.

Tatsächlich sind nur drei Befehle nötig die Zeilen mit dem Punkt:

Kommentare. Legt man die in einem Do-File ab, reicht »DO Name« und der Drucker ist benutzbar. SETCOM selbst ist ein gutes Programm, sein Aufruf zeigt den Ist-Stand, mit »SETCOM(Parameter)« kann man jederzeit ändern, und sich so sehr einfach an die Besonderheiten einer wilden Norm wie die der RS232 herantasten.

M4 oder 4P, das ist noch eine Frage

Theoretisch kosten beide Computer dasselbe, praktisch ist das 4P in gleicher Ausbaustufe um den Preis der dort serienmäßigen RS232-Schnittstelle billiger (rund 300 Mark). Wenn Ihnen zwei Diskettenlaufwerke reichen (oder sie sich Harddisks leisten wollen), ist vom technischen her kein Unterschied festzustellen. Blicke der kleinere Schirm (9 statt 12 Zoll). Wie gesagt, der kleinere ist schärfer, und Apples Macintosh hat auch 9 Zoll.

Vorteilhaft ist die absetzbare Tastatur, und der Kleine blockiert nicht gleich den ganzen Schreibtisch. Er sieht besser aus und kann leicht weggeräumt werden.

(Peter Wollschläger)

Auf eine Blick

CPU: 280A, 4 MHz

RAM: 64 KByte, aufrüstbar auf 128 KByte

zwei eingebaute Diskettenlaufwerke, je 184 KByte Erweiterung um bis zu vier Festplatten, je 5 MByte Parallelschnittstelle und RS232 serienmäßig

Betriebssystem: TRSDOS 6.1

(Standard), CP/M 3.0 (Aufpreis)

Sprachen: Basic (Microsoft Ver. 5)

Gegen Aufpreis: Pascal, Assembler und anderes

Nachrüstbar: Hochauflösende Grafik

Bildschirm 9 Zoll, grün auf schwarz oder invers

24 Zeilen mal 80 Zeichen

umschaltbar auf 40 Zeichen

Absetzbare Tastatur, 75 Tasten

Besonderheiten: Kann TRS-80, Modell 3 emulieren

Tonausgabe

Gewicht: 11 kg



Bild 1. Ähnelt dem Vorgänger Modell III: Tandy's neues Modell 4

TRS-80 Modell 4 — Drei Computer in einem?

Das seit langem auf dem Markt befindliche Modell I und das sich kaum davon unterscheidende Modell III bedurften eines Nachfolgers auf dem neuesten Stand der Technik. Mit Rücksicht auf die umfangreiche Software-Basis sollte die Kompatibilität der Reihe erhalten bleiben, ein besseres DOS war nötig, und die CP/M-Welt durfte auch nicht länger ignoriert werden. Lesen Sie, ob und wie Tandy diesen gordischen Knoten mit seinem Modell 4 durchschlagen hat.

Das Modell 4 sieht aus wie ein Modell III (Bild 1), nur daß dessen abweisbarer Metallic-Look durch soliden, matweißen Kunststoff ersetzt wurde. Zum Öffnen sind zehn Schrauben auf der Unterseite zu lösen (man merke sich tunlichst, wo welche hingehören). Dann hebt man das Oberteil ab. Das erscheint einem sehr schwer, und das ist es auch, denn da hängt die ganze Bildröhre nebst Video-Platte dran (Bild 2). Jetzt hat man freie Sicht auf das Chassis und erblickt ganz hinten die große Hauptplatine, davor zwei kleinere für den Disk-Controller und das Netzteil. Die Hauptplatte ist durch

ein gleichgroßes Blech abgeschirmt, auch sonst dürfte die Post zufrieden sein. Jedenfalls funktioniert der beliebte Modell-I-Trick, »Ton« über ein neben dem Computer stehendes Kofferradio auszugeben, nicht mehr.

Auf der Hauptplatte sieht man fünf Drahtbrücken, von denen mindestens eine wild gebastelt ist, immerhin ein Zeichen dafür, daß das Gerät erprobt wurde (Bild 3).

Die Sound-Platine mit ihren nicht viel mehr als drei Bauteilen hätte auch noch bequem Platz auf der Hauptplatte gefunden, und man hätte die Steckverbindung gespart. Aber irgendwie muß man ja Kun-

den, die die abgemagerte Grundversion kaufen wollen, davon abhalten. Sound scheint ein gutes Verkaufsargument zu sein.

Ein weiterer Blick auf die Platine zeigt, daß Tandy den in der Einleitung genannten gordischen Knoten mit Geld, sprich Hardware, durchschlagen hat. Mit 64 KByte RAM ist das Modell 4 als CP/M-Maschine ausgerüstet, 14 KByte ROM extra sind nötig, um die Modell-III-Betriebsart zu ermöglichen, und noch einmal 2 KByte Video-RAM zusätzlich waren nötig, um nicht CP/M's TPA durch einen Bildspeicher ab 3C00H zu verwirren.

Wer selbst von 64 auf 128 KByte aufrüsten will, muß mindestens ein Teil bei Tandy kaufen. Die Adreßlogik wird mit einem PAL (Programmable Logic Array) gemacht, und PALs sind kundenspezifische Schaltkreise, die nur an diese ausgeliefert werden. Wie gesagt, es gibt nur noch ein Netzteil, früher wurden die Disklaufwerke (2 x 180 KByte Speicherkapazität) gesondert versorgt, und das merkt man als leichtes Zucken im Bild, wenn ein Diskmotor anläuft.

Insgesamt macht die Technik einen sehr soliden Eindruck, irgendwo müssen die 20 kg Gewicht schließlich herkommen. Platz für Erweiterungen ist genügend vorhanden. Tandy bietet zur Zeit eine RS232- und eine Grafikkarte zum Einbau an.

Die Tastatur ist deutsch (Bild 4) und hat mit 75 Tasten fünf mehr als die amerikanische, die im Handbuch beschrieben ist. Auf jeden Fall ist sie angenehm zu bedienen, prellfrei und hat Autorepeat.

Leider habe ich noch das amerikanische DOS, das deutsche soll in zwei Wochen hier sein. Das macht stutzig. Man sollte sich doch überlegen, ob man nicht bei der auch hier angebotenen ASCII-Tastatur bleibt, zumal, wenn man noch andere Computer bedient.

Der Zeichengenerator erzeugt 256 Zeichen, darunter die deutschen Umlaute, griechische Zeichen und anderes mehr. In Kombination mit Shift, Clear und Control sind sogar alle von der Tastatur aus erreichbar.

Zu loben ist der Type-Ahead-Buffer (nur im Modell-4-Modus) zu bedauern ist, daß die mit Control erzeugten Codes nicht wie beim Apple in Texte aufgenommen werden, man also weiterhin CHR\$(x) tippen muß. Die drei Funktionstasten werden vom Basic-Editor akzeptiert, ansonsten lösen sie keine besondere-

HEFT
12
März
1986

74

ren Aktivitäten auf. Der Anwender muß sie selbst (in Basic mit INKEY\$) abfragen.

Der Schirm (grün) bietet 24 Zeilen x 80 Zeichen im Modell-4-Mode, 16 x 64 im Modell-III-Betrieb. Bei Breitschrift wird auf 40 beziehungsweise 32 Zeichen/Zeile umgeschaltet. Reverse Video ist auch bereichsweise möglich. Bei größeren Flächen in Pixelgrafik stört ein deutliches Flimmern im Modell-4-Mode, in dem statt der 48 x 128 jetzt 72 x 160 Pixels möglich sind.

Eine Parallelschnittstelle für Drucker ist eingebaut, RS232 gibt es nur gegen Aufpreis.

Drei Betriebssysteme und zwei Computer auf einmal

Das Modell 4 kann sich als Modell 4 verhalten oder als Modell III oder als reine CP/M-Maschine. Was es gerade sein soll, merkt es selbst anhand der Diskette, mit der es hochgefahren wird. Fehlt sie ganz (oder ist nicht eingelegt) meldet es sich mit »Cass?«, womit es sich als Modell-III-Kassettenversion einstuft und nach der Baudrate fragt.

Im Modell-III-Mode arbeitet die Z80-CPU mit 2 MHz, kann aber per Software auf 4 MHz umgeschaltet werden. Der Cursor flackert dann recht nervös, was ihm mit einem POKE abgewöhnt werden kann, und die Echtzeit-Uhr zeigt Hausnummern an. Ansonsten läuft alles an Modell-III-Software dann auch, halt nur entsprechend schneller.

CP/M ist in der deutschen Version (Tastatur) noch nicht lieferbar (soll Ende 1983 kommen). In Amerika und hier habe ich bezüglich der US-Version widersprüchliche Aussagen gehört, der eine hat sie, der andere nicht. Ich fürchte, daß da einige CP/M-2.2 anbieten und nicht das von Tandy angekündigte CP/M-Pius (3.0). Von letzterem gibt es bekanntlich zwei Versionen (mit und ohne Bank-Select). Wie macht das nun Tandy, wenn jemand von 64 auf 128 KByte aufrüsten will?

TRS-DOS 6.1, das eigentliche Modell-4-DOS, ist absolute Spitze und sicherlich benutzerfreundlicher und komfortabler als CP/M. Für letzteres gibt es jedoch eine Umengung Software, darunter einiges, wofür ich auch im großen Tandy-Angebot nichts Gleichwertiges finde. Daß CP/M erst in einigen Monaten verfügbar ist, hat mich dennoch nicht vom Kauf abgehalten, ein neuer Computer und gleich drei Betriebssysteme auf einmal ist eh zu viel für den Anfang.

Das Modell 4 in Stichworten

Z80-CPU, 4 MHz, im Modell-III-Mode 2/4 MHz umschaltbar
64 KByte RAM, erweiterbar auf 128 KByte

14 KByte ROM-Basic, ROM abschaltbar

2 KByte Video-RAM extra
Grüner Bildschirm, 24 Zeilen x 80 Zeichen

256 darstellbare Zeichen
Umschaltbar auf 16 Zeilen x 64 Zeichen

Breitschrift als 40 beziehungsweise 32 Zeichen/Zeile
Reverse Video, Blockgrafik, hochauflösende Grafik nachrüstbar

Deutsche Tastatur mit 75 Tasten, Autorepeat-Funktion

1 oder 2 Diskettenlaufwerke eingebaut, je 180 KByte, 2 weitere Laufwerke oder Festplatte anschließbar

Besonderheit: kann das Modell III emulieren

Betriebssysteme: TRS-DOS-6.1 oder CP/M-Plus

Im Lieferumfang: TRS-DOS-6.1 und Microsoft-Basic

Im Angebot: Assembler, Compiler-Basic, Alcor-Pascal, Cobol

Software ist genügend vorhanden, nur das Interessante fehlt

Ich habe Umengungen Software auf Modell-I-Disketten und einiges (ganz schnell geliehen) im Modell-III-Format. Zuerst: Im Modell-III-Mode läuft alles, was Original-Modell-III ist, ohne Fehl und Tadel, zum Beispiel Scripsit, mit dem dieser Artikel geschrieben wurde, und dieses Programm macht exzessiv von Tastatur, Schirm und Systemsoftware Gebrauch. Leider mit 16 Zeilen zu 64 Zeichen, und nicht mit den 24 x 80 des Modell-4-Modus.

Bei der Übertragung von Modell-I-Software gibt es natürlich die bekannten Probleme, wenn man in Maschinensprache arbeitet, nahezu unmöglich wird dies Richtung Modell 4, da paßt halt nichts mehr. In Basic ist das alles viel einfacher, auf die Unterschiede wird noch eingegangen. Wichtig ist, daß Sie alle Modell-I-/Modell-III-Programme im ASCII-Format abgespeichert haben. Andernfalls hilft nur noch NEW-DOS/80, das in der Modell-III-Version noch ein Modell-I-Laufwerk emulieren kann.

Grund für das ASCII-Format ist, daß das Modell-4-Basic im gepackten Format die Befehle anders codiert. Eine Modell-I-Diskette kann man einfach mit REPAIR ins Modell-4-Format bringen, von einer TRS-DOS-1.3-Disk sind Files mit CONV auf Modell-4-Disketten zu übertragen. CONV akzeptiert aber keine Modell-III-NEW-DOS-Diskette.

Folglich habe ich, der ich nur NEW-DOS besitze, damit ein Modell-I-Laufwerk emuliert, dann die Modell-III-Software auf die Modell-I-Diskette überspielt, und schließlich diese mit REPAIR ins Modell-4-Format gebracht. Wundersamerweise liefern die Programme (in Basic) dann immer noch, allerdings erst nach Korrektur eines Hauptfehlers. Modell-4-Basic erwartet nämlich, daß alle Schlüsselwörter durch Blanks getrennt sind. »FORI=1TO10« enthält danach drei Syntaxfehler.

Typische Modell-4-Software, die die Vorteile des neuen DOS und des Modell-4-Basic voll ausnutzt, gibt es hier noch nicht, schlimmer, wichtige Tools, die im Katalog stehen, den ich mir aus Amerika mitgebracht hatte, sind in Deutschland noch nicht erhältlich. Ich vermisse besonders das Assembler-Entwicklungspaket (149 Dollar) und Alcor-Pascal (jetzt auch Tandys Pascal) für 250 Dollar. Auch laut US-Katalog dürfen deutsche Anwender auf Cobol hoffen (199 Dollar), Basic (195 Dollar), Visicalc (249 Dollar), Multiplan (199 Dollar) und ein auch im Singlestep arbeitendes Compiler-Basic mit Debugger für 149 Dollar.

Mitgelieferte Software wertvoll und nützlich

Zuerst: Es handelt sich nicht um die üblichen Soloprogramme, sondern alle sind schön in das System eingebunden. KSM/FLT zum Beispiel erlaubt eigene Texte unter 26 Tasten zu legen und mit SYSGEN kann das permanent gemacht (sozusagen ins DOS eingebunden) werden. Das gilt auch für die Communication-Software, die sehr komfortabel die RS232 unterstützt und in das Rout-/Link-Konzept paßt, womit zum Beispiel ein direkter Diskfile-Transfer möglich wird.

JCL, die Job Control Language, dürfte auch einen an Komfort gewohnten Arbeitsvorbereiter im Großrechenzentrum nichts vermissen lassen. Natürlich kann man auf einfachste Art mit BUILD den Standard-Batch-File schreiben, das ist aber noch keine JCL, obwohl sie

```
10 [TRSDOS61] 5" Floppy #1, Cyls= 40, Dden, Sides=1, Step=12ms, Div=5s
11 [No Disk] 3" Floppy #2, Cyls= 40, Dden, Sides=1, Step=12ms, Div=5s
12 [No Disk] 3" Floppy #4, Cyls= 40, Dden, Sides=1, Step=12ms, Div=5s
13 [No Disk] 3" Floppy #8, Cyls= 40, Dden, Sides=1, Step=12ms, Div=5s
**K1 <# [SDU] X'FEAC'
**D0 <# X'0BE9'
**FR #) [-#S0J] X'F7A9'
**S1 <# #1.1
**S0 <# #D0
**JL #) N1.1
**DU <# X'0BED'
**S0 #) X'0E0F'
Options: Spool, Type, KSM
```

Listing 1. Der Device-Befehl informiert über Geräte und Software im System

```
LIST
10 * Programm listet die reservierten Worte des Model 4
20 * (C) 1983 Peter Mollischlager. All rights released.
30 *
40 TABELLENZEIGER=22666
50 WORTANFANG=ASC("A")
60 IF PEEK(TABELLENZEIGER+1) THEN PRINT CHR$(WORTANFANG);
70 TABELLENZEIGER=TABELLENZEIGER+1
80 ZEICHEN=PEEK(TABELLENZEIGER)
90 IF ZEICHEN=0 THEN WORTANFANG=WORTANFANG+1; GOTO 60; Skip 1 oder 2 0-Bytes
100 IF WORTANFANG > ASC("X") THEN END
110 IF ZEICHEN=ASC("Z") THEN PRINT CHR$(ZEICHEN);;GOTO 70
120 PRINT CHR$(ZEICHEN) AND SH7F); " ";
130 IF POS(0) > 76 THEN PRINT;
140 TABELLENZEIGER=TABELLENZEIGER+1;
150 GOTO 60
Ready
RUN
AUTO AND ABS ATN ASC CLOSE CONT CLEAR CINT CSNG CDBL CVI CVS CVD COS CHR$ CALL
COMMON CHAIN CLS DELETE DATA DIM DEFSTR DEFINIT DEFNSG DEFDBL DEF DATE$ ELSE
END ERASE EDIT ERRORS ERROR ER$ ERR EXP EDP FOR FIELD FN FRE FIX GOTO GO TO
GOSUB GET HEX$ INPUT IF INSTR INT IMP IMP INKEY$ KILL LPRINT LLIST LPOS LET
LINE LOAD LSET LIST LOG LOC LEN LEFT$ LOF MERSE MOD MK18 MK36 MK48 MK54
NEXT NAME NEW NOT OPEN ON DR OCT$ OPTION PRINT PUT POKE POS PEEK RETURN
READ RUN RESTORE REM RESUME RES RIGHT$ RND REND RANDOM ROM STOP SNAP SAVE
SPC1 STEP SIGN SQR SIN STR$ STRING$ SPACES SYSTEM SOUND THEN TRON TROFF TAB1
TO TIMES TAN USING USR VAL VARPTR WAIT WHILE WEND WRITE XOR
Ready
```

Listing 3. Ein Programm und sein Ergebnis: Alle Befehle und Funktionen in alphabetischer Folge.

mancher Anbieter so nennt. Dinge wie Makros, Variable und IF-THEN-ELSE machen eine gute JCL aus, hier sind sie vorhanden.

TRS-DOS-6.1 ist L-DOS

Zuerst zur Namensverwirrung. Im Copyright meldet es sich mit »TRSDOS 06.01.00«, im Disk-Namen mit TRSDOS61 und ansonsten heißt es TRS-DOS-6.1. Wie auch immer, das erste TRS-DOS schrieb seinerzeit Bill Schroeder als Tandy-Mitarbeiter. Dann hat er sich selbständig gemacht und schuf das berühmte L-DOS. Offensichtlich hat er sich von seiner alten Firma in Frieden getrennt, denn TRS-DOS-6.1 ist L-DOS-5.1. Tandy bietet L-DOS jetzt auch für die anderen Modelle an (sinnigerweise auch für das Modell 4, wo es unter anderem Namen zum Lieferumfang gehört), und schreibt in dem 84er Katalog, daß man neuerdings bevorzugt Software für L-DOS liefern wird.

L-DOS ist mit NEW-DOS/80

```
10 INPUT"Eingabe in Hex (H) oder in Dezi (D)";E$
20 IF E$="D" THEN 80
25 ~~~~~
30 '///// hex-dezi Konversion \\\\\\\
40 INPUT "Hexzahl (stop mit 0)";H$
50 IF H$="0" THEN END ELSE PRINT VAL("&H"+H$); GOTO 40
60 '
70 '///// dezi-hex Konversion \\\\\\\
80 INPUT "Dezimalzahl (stop mit 0)";D
90 IF D=0 THEN END ELSE PRINT HEX$(D); GOTO 80
```

Listing 2. Mit dem neuen Basic wird vieles einfacher. Hier Hex/Dez und Dez/Hex-Konversion mit je nur einem Befehl.

Am frapierendsten werden die Unterschiede zwischen NEW-DOS und L-DOS am Beispiel Spooler sichtbar. Wer sich unter NEW-DOS schon mal mit ASPOOL/MAS versucht hat, wird staunen. Unter L-DOS reicht schlicht SPOOL. Damit wird automatisch ein i-KByte-Druckerpuffer geschaffen und alle Ausdrücke laufen ab sofort über diesen. Ist Ihnen 1 KByte zu wenig und sie hätten lieber 10 KByte, dann geben Sie einfach »SPOOL (MEM=10)« ein. Wird es im Arbeitsspeicher eng, können Sie ein Disk-File hinzunehmen und geben dazu nur den Namen und die File-Größe in KByte an.

Auch in diese Rubrik fällt MEMDISK, was erlaubt, das Memory als Disk zu benutzen. Richtig Spaß macht das zwar erst in der 128-KByte-Version, aber mit 64 KByte geht's auch. Ich habe so eine 10-KByte-Diskette im Background, wo ich blitzschnell Zwischenergebnisse speichere.

So schöne Dinge wie SPOOL und MEMDISK sollte man sich allerdings überlegen, bevor man vom DOS ins Basic geht. Das sind nämlich die einzigen DOS-Routinen, die man nicht unbeschadet von Basic aus initialisieren, wohl aber da benutzen kann. Noch ein Bug, und zwar ein großer: Ich habe noch nicht herausbekommen, wie man nach einem versehentlichen Verlassen von Basic da wieder hinkommt und sein Programm unbeschadet vorfindet. Der Befehl »BASIC*« von NEW-DOS fehlt. Das Problem dürfte lösbar sein. Ich habe es mir vorgenommen, sobald ich über einen Assembler verfüge.

Schon aus Gründen der Kompatibilität ist alles an TRS-DOS/alt-Befehlen auch hier vorhanden. Von den Extras wären hervorzuheben:

Nach dem Kaltstart erscheint die (abschaltbare) Frage nach dem Da-

tum als MM/DD/YY, die dann als Text quittiert wird, sofern man nicht versucht, den 29. Februar 1983 oder ähnliches einzugeben, was das Kalender-Programm nicht mag. Das Datum wird im File-Header mit aufgezeichnet und kann zur selektiven Behandlung von Files genutzt werden. Ein ebenfalls geführtes »Mod-Flag« erlaubt, nur Dateien zu kopieren, die seit dem letzten Datum geändert worden sind.

BACKUP kann als Spur für Spur-Kopie laufen oder als File-Kopie mit Krunch-Funktion (die Diskette wird aufgeräumt).

DEVICE liefert ähnlich CP/Ms STAT Informationen über alle Geräte im System, aber auch über User-Options wie Spool (Listing 1).

FILTER lenkt ein logisches Gerät über ein Filter-Programm.

FORMS ist ein vorhandenes Filterprogramm zur Druckereinstellung beziehungsweise Formatierung.

MEMORY erlaubt »Lomem« und »Himem« zu setzen.

PATCH ist ein Programm zum direkten Ändern von Disk-Sektoren; es arbeitet auch Patch-Files ab.

REMOVE entfernt Files; es darf eine Liste angegeben werden.

PURGE erlaubt globales »Remove« mit Rückfrage.

SYSGEN bindet eine Anwender-Konfiguration ins DOS ein und

SYSTEM erlaubt ihre Anpassung/Eingabe.

BLINK: Blink-Frequenz (auch 0) und Größe des Cursors.

SYSPRES: Laufwerke physikalisch anpassen.

System-Drive kann ein anderer als 0 sein.

TRACE gibt den Z80-PC aus.

ALIVE erzeugt einen sogenannten »alive bug«, ein Grafiksymbols, das sich in sich ändert, solange das System nicht »hängt«.

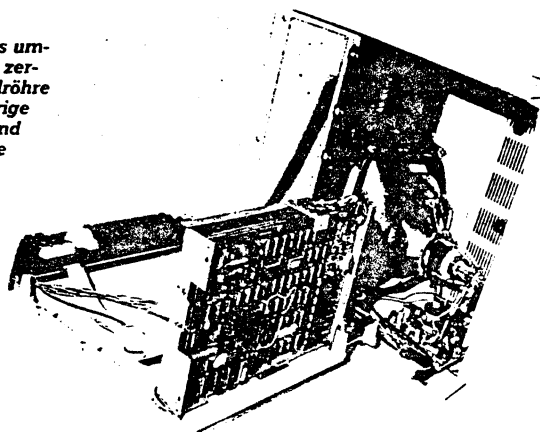
Ein neues Basic von Microsoft

Das Basic ist total RAM-resident und wird immer von der Diskette geladen. Damit sind Korrekturen und Erweiterungen leicht möglich und billig.

Nun zum Handling von Random-Files: Es läuft in schlechter M80-Tradition immer noch über Fielding, LSET und CVx. Ein CIM-Dump in Basic wäre auch noch zu installieren und der Editor ist der alte, also verbesserungsbedürftig, sofern man sich nicht Wordstar leisten will.

Den Kern bildet das alte Level 2-Basic. Einiges wurde verbessert und

Bild 2. Etwas umständlich zu zerlegen — Bildröhre und zugehörige Schaltung sind in der Haube eingebaut



vieles kam hinzu, aber es ging auch etwas verloren. CLOAD und CSAVE gehören dazu, was nicht zu beklagen ist, aber auch SET, RESET und POINT. Die Grafik-Symbole gibt es zwar noch, wenn auch wegen der jetzt 24 Zeilen leicht gestaucht, man kann sie aber nur noch mit PRINT\$ auf den Schirm bringen. Letzteres erlaubt jetzt auch die Angabe von Zeile/Spalte. POINT kann auch nicht mehr mit PEEK realisiert werden, weil der inzwischen ausgelagerte Video-RAM sich dem entzieht. Tandy bietet allerdings eine Supergrafik-Karte an mitsamt vielen mächtigen Grafik-Befehlen. Ein Testbericht soll folgen.

Gehen wir alphabetisch vor, so beginnen die interessanten neuen Dinge bei C wie CALL. Der Befehl erlaubt, Routinen in Maschinensprache aufzurufen und beliebig viele Variablen zu übergeben. USR blieb erhalten, wurde aber auch erweitert. Noch unter C finden Sie zwei ganz wichtige Dinge, nämlich CHAIN zum Hereinziehen von Overlays und Fortrans COMMON, damit dann die Parameterübergabe auch klappt.

CLEAR hat eine ganz neue Bedeutung, Stringspace muß nicht mehr reserviert werden. Es ermöglicht, im Basic-Programm Mem-Size und Stackgröße zu setzen. Mit letzterem kann man ein paar Byte sparen, wenn mangels geschachtelter FOR-NEXTs oder GOSUBs der Basic-Stack bescheiden bleibt. DATE\$ und TIME\$ ergeben die entsprechenden Daten.

Die globale Deklaration von Variablen-Typen wie DEFINT blieb (Gott sei Dank) erhalten, Variablen-namen dürfen jetzt aber bis zu 40 signifikante Zeichen haben, was wesentlich zur Lesbarkeit und

Wartbarkeit der Programme beiträgt und zahllose REMs erspart. ERL ergibt jetzt die echte Fehlerzeile, ERR gibt es noch, neu ist ERR\$, das die Fehlermeldung im Klartext ausgibt.

ERASE erlaubt dynamisches Löschen von Arrays. FOR-Schleifen werden nicht mehr sinnlos ausgeführt, wie es bisher zum Beispiel in »FOR I=1 TO 0« der Fall war.

Vielleicht störend bei der Übernahme von Modell 3-Programmen kann sein, daß bei der Umwandlung Fließkommata in Integer nicht mehr abgeschnitten sondern gerundet wird.

HEX\$ und OCT\$ als die Umkehr von &H und &O kamen hinzu und machen die so schönen Konversionsprogramme glatt überflüssig. INKEY\$(n) als Erweiterung dazu, macht auch Listings zu diesem Thema obsolet. Bei J gibts nichts, zu O wäre OPTION BASE zu erwähnen, was erlaubt, das selten gebrauchte nullte Element von Arrays zu streichen. POS() ergibt wie gehabt die Cursorposition, ROW ist neu und »returned« die Cursor-Zeile und LPOS() gilt für den Drucker. Mit WIDTH kann neuerdings die Zeilenlänge für Video und Drucker eingestellt werden.

RESTORE jetzt mit Zeilennummer und RENUM mit drei Argumenten (wohin, woher, Inkrement) ist fest eingebaut.

SWAP tauscht die Inhalte zweier Variablen, SYSTEM »DOS-Befehl« ersetzt CMD«, WAIT wartet auf ein zu definierendes Ereignis an einem Port.

WHILE/WEND bringt etwas Pascal ins Basic und WRITE, wie

Test

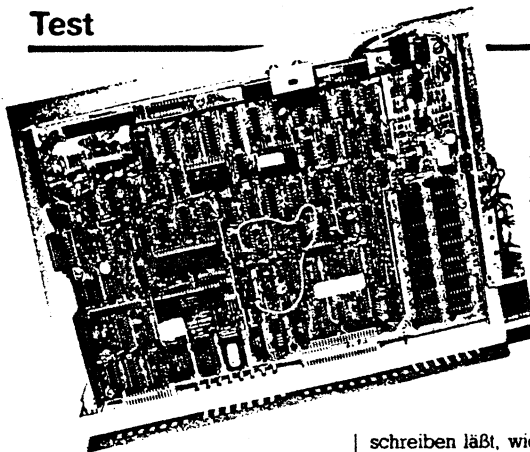


Bild 3. Die Hauptplatine weist noch von Hand durchgeführte Änderungen auf

PRINT anzuwenden, drückt die Variablen mit Kommas getrennt und Strings in Anführungszeichen. Auf dem Schirm sieht das gut aus, als WRITE# auf Files angewandt erspart es, selbst Delimeter vorzusetzen.

Befehle, die nicht im Handbuch stehen

In der Einleitung des Manuals wird dem Besitzer zum Kauf gratuliert und ein kurzer Abriss über die Vorteile des Modell 4 gegeben, und da steht auch »sound generation« und mehr nicht. Im Anhang findet man dann noch in der Liste der Tokens wiederum SOUND und das war's dann auch. Nach der Methode »trial and error« ergab sich folgende Syntax: »SOUND Ton, Dauer«. Dabei darf Ton die Werte 0 bis 7 annehmen, was in etwa einer Tonleiter G-Dur entspricht, und Dauer darf 0 bis 31 betragen in grob 0,5-Sekundenschritten. Klangqualität und Lautstärke sind nicht berühmt.

Auch in der Tokenliste habe ich XOR entdeckt, die Syntax ist klar, vielleicht auch noch EQV, das wie alle Logik-Operatoren auf Bit-Ebene arbeitet und bei gleichen Bits 1 sonst 0 erzeugt. Etwas länger habe ich an IMP geknobbelt. Was das heißt, weiß ich nicht. Vielleicht kann mir da ein Leser helfen, jedenfalls gilt »A IMP B = NOT (A OR B)«. (Anmerkung der Redaktion: Wahrscheinlich handelt es sich um die logische Implikation). Eine Anwendung dafür sehe ich noch nicht, jedoch für den Schrägstrich links, was DIV, also Ganzzahldivision heißt sowie für MOD, also Modulo.

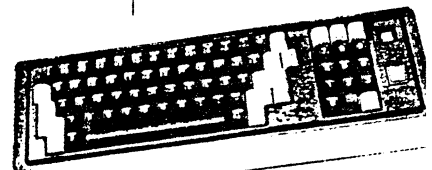
Auch nicht erwähnt wird, daß das altbekannte &H jetzt sogar in Ausdrücken erlaubt ist, womit sich eine höchst simple Dezi-Hex-Konversion

schreiben läßt, wie Listing 2 zeigt. Das neue Feature HEX\$ wird gleich mit demonstriert.

Daß Variablen-Namen jetzt bis zu 40 Zeichen (signifikant) lang sein dürfen, habe ich mit Freude registriert und prompt in CBasic-Art den Punkt mit eingebaut. Auch das funktioniert, obwohl es nicht im Manual steht. In die Rubrik gehört noch INPUT, was neuerdings (wozu?) als Direkt-Statement erlaubt ist.

So belehrt mißtrauisch und hoffnungsvoll wollte ich wissen, ob denn nun wenigstens die Token-Tabelle im Handbuch vollständig ist und fing an, das Pendant im RAM zu suchen. Sie war im Prinzip schnell gefunden, aber nicht so einfach zu knacken. Wie, zeigt Listing 3 mitsamt dem Ergebnis. Dahinter steht folgendes: Alle Texte sind alphabetisch sortiert und in Gruppen je Anfangsbuchstabe eingeteilt. Jede Gruppe endet mit einem 0-Byte, folglich bedeuten zwei aufeinanderfolgende Nullen, der Buchstabe, zum Beispiel Q, ist nicht besetzt. Da so der Anfangsbuchstabe je Wort vorgegeben ist, wird er denn auch weggelassen. Das Wortende wird durch ein gesetztes Bit 7 im letzten Buchstaben markiert, direkt dahinter folgt das Token, ein 1-Byte-Code, der anstelle der langen Basic-Worte im Programmtext eingetragen wird. Um Tokens von anderen Zeichen unterscheiden zu können, sind sie in der Regel ≥ 128 . Da ein Byte nicht mehr als 255 darstellen kann, sind maximal 128 Tokens möglich, das ging bisher ganz gut, das neue Basic hat aber 140 Tokens!

Bild 4. Sehr schön: deutsche Tastatur mit 10er-Block



Gelöst wurde das so: Das »Token« 255 bedeutet, das folgende ist das richtige, hat aber eine andere Bedeutung. Diese Sondertokens stehen in der Tabelle im Handbuch als Zahlen > 65280 . Das » ist $255 * 256 + \text{Tokenwert}$.

Im Programmtext sind auch diese Tokens ≥ 128 , nicht aber in der RAM-Tabelle, da sind von ihrem Wert jeweils 128 subtrahiert, womit man sie von den anderen unterscheiden kann, ohne die 255 davorzusetzen.

Nebenbei gesagt hatte ich auch Probleme, ein Basic-Programm im RAM zu finden, denn im Handbuch steht folgende Memory-Map:

```
0000H-2F5FH DOS
2600H-2FFFH Overlays
3000H-85FFFH Basic
8600H-Himem Programm
```

Das ist insofern falsch, als Basic-Programme schon bei 7FOCH beginnen. Mag sein, daß eine geänderte Basic-Version, zum Beispiel die für Grafik, mehr verbraucht, die kenne ich aber noch nicht. Zu den Handbüchern ist nicht viel zu sagen, bekannte Tandy-Qualität in Menge. Neu ist »Getting started« als extra Heftchen und eine Referenz-Karte. Zu bemängeln wäre: Das DOS-Manual ist nicht gedruckt, sondern direkt von der Schreibmaschine kopiert, vielleicht kommt's noch besser. Für den Nicht-Diskbasic-Teil gibt es ein extra Buch, gemeinsam für Modell III und 4, nach dem Motto, das alte Modell-III-Buch mit Hinweisen auf Modell-4-Features. Man hat allerdings versäumt zu sagen, daß die meisten der aufgeführten POKEs im Modell 4 Nonsens sind.

Fazit: Eine solide Maschine mit einem guten DOS und Basic, offen für CP/M und andere Betriebssysteme. Der Anschluß an die Vorgänger und ihr Softwarepotential ist auch gelungen. Vom Konzept und vom Preis her eindeutig Richtung kommerzielle Anwendung orientiert, aber noch preiswert genug als Homecomputer für gehobene Ansprüche.

(Peter Wollschlaeger)

Die folgende Schaltung stammt aus 80 MICRO Mai 81 und ermöglicht, daß man an beliebigen Stellen im Text einzelne oder mehrere Zeichen invertieren kann, d.h. schwarze Zeichen auf weißem Hintergrund. Leider hat die Sache aber einen Schönheitsfehler und der ist, daß inverse Zeichen und Grafik-Zeichen nicht gleichzeitig auf dem Bildschirm dargestellt werden können. Da aber die Anwendung der Invers-Darstellung vorwiegend im Textbereich liegt, ist dieser Nachteil meiner Meinung nach zu verschmerzen.

Da ich mit dieser Schaltung in der Lage bin ein einzelnes Zeichen zu 'adressieren' lag der Gedanke nahe die Schaltung so umzubauen, daß man an der gewünschten Stelle auch die Helligkeit des Zeichens umschalten kann und damit einzelne Wörter oder Textabschnitte in den Hintergrund schieben oder auch besonders hervorheben kann. Die gesamte Schaltung ist auf 2 Bilder verteilt. Bild 1 zeigt die etwas modifizierte Schaltung (für TRS-80!) aus 80 MICRO, Bild 2 zeigt die Adressierlogik. Inverse Darstellung und Helligkeitsumschaltung können HW-mäßig über die Schalter S1 bzw S2 und SW-mäßig über die Befehle:

OUT 251,2 = Helligkeit schwach

OUT 251,4 = Inverse Darstellung

umgeschaltet werden. Als Treiberprogramm muß entweder das Maschinenspracheprogramm (Listing1) oder das Basic Programm (Listing2) geladen werden. Nachfolgend habe ich die Beschreibung dieser Schaltung aus 80-MICRO übersetzt, damit Ihr auch wißt wie die Sache funktioniert.

Die Umschaltung erfolgt durch Bit 7. Sieht man sich das Schaltbild des TRS-80 an, so sieht man das Bits 0 bis 5 von den Video Memory ICs an IC 228 gehen und Bits 6 und 7 zusammen mit ein paar anderen Signalen an IC 227 gehen. Diese beiden ICs sind einfach Zwischenspeicher welche die Daten puffern die vom Video Memory kommen.

Bis hierhin ist Bit 7 immer noch im Takt mit den anderen Datenbits. Von 229 und von 28 aus gehen die Bits dann zu den Schieberegister 210 und 211 die dazu dienen, die von den Zeichengeneratoren kommenden Zeichen Bit für Bit auszugeben.

Mit Bit 7 wird entschieden, welches der Schieberegister die Daten erhalten soll und somit ob ein normales Zeichen oder ein Grafiksymbol ausgegeben werden soll. Das Problem ist nur, daß Bit 7 gerade solange ansteht um das Schieberegister zu laden. Zu der Zeit, in der die Bildpunkte vom Register ausgegeben werden, sind aber die Bits an dem Datenpuffer inclusive Bit 7 bereits die, des nächsten Zeichens welches ausgegeben werden soll. Die Konsequenz ist, daß das inverse Zeichen immer um eine Zeichenbreite nach links versetzt ist und damit nicht an der Stelle wo man es haben wollte.

```

00010 ;*****
00020 ;* LOWER CASE AND INVERSE CHARACTER DRIVER PROGRAM *
00030 ;* (DISK VERSION) *
00040 ;*****
00050 ;
00060 RVCODE EQU 7FH ; REVERSE CONTROL CODE
401E 00070 VIDEO EQU 401EH ; VIDEO DCB ADDRESS
4049 00080 HIMEM EQU 4049H ; DOS HIGH MEMORY POINTER
00090 ;
FFB0 00100 ; ORG 0FFB0H
00110 ;
FFB0 21BFFF 00120 LOADER LD HL,START ; GET DRIVER ADDRESS
FFB3 221E40 00130 LD (VIDEO),HL ; AND INSERT IT INTO THE
00140 ; VIDEO DEVICE CONTROL
00150 ; BLOCK
00160 ;
00170 ; THE FOLLOWING TWO INSTRUCTIONS AUTOMATICALLY SET THE
00180 ; DOS HIGH MEMORY POINTER. THIS ONLY APPLIES TO NEWDOS
00190 ; (2.1 OR 80) AND TRSDOS 2.2 OR LATER.
00200 ; FOR OTHERS, THEY SHOULD BE DELETED.
FFB6 21BEFF 00210 LD HL,START-1 ; GET MEMORY SIZE
FFB9 224940 00220 LD (HIMEM),HL ; AND SAVE IT
00230 ;
FFBC C32D40 00240 JP 402DH ; RETURN TO DOS
00250 ;
FFBF DD6E03 00260 START LD L,(IX+3) ; GET THE ADDRESS OF
FFC2 DD6604 00270 LD H,(IX+4) ; THE CURSOR
FFC5 DA9A04 00280 JP C,049AH ; IF READING FROM SCREEN,
00290 ; THEN RETURN TO ROM
00300 ;
FFC8 79 00310 LD A,C ; GET THE CHARACTER
FFC9 FE7F 00320 CP RVCODE ; IF IT'S NOT REVERSE CODE
FFCB 2089 00330 JR NZ,DISPLY ; THEN GO TO "DISPLY"
FFCD 3AEFFF 00340 LD A,(MASK) ; GET THE REVERSE MASK
FFD0 EE80 00350 XOR 80H ; AND TOGGLE IT
FFD2 32EFFF 00360 LD (MASK),A ; SAVE THE NEW MASK
FFD5 C9 00370 RET ; RETURN WITHOUT DISPLAY-
00380 ; ING THE CHARACTER
00390 ;
FFD6 DD7E05 00400 DISPLY LD A,(IX+5) ; IF THE CURSOR
FFD9 B7 00410 OR A ; IS ON
FFDA 2801 00420 JR Z,GETCHR
FFDC 77 00430 LD (HL),A ; THEN BLANK IT
00440 ;
FFDD 79 00450 GETCHR LD A,C ; GET THE CHARACTER
FFDE FE20 00460 CP 20H ; IF IT'S A CONTROL CHAR.
FFE0 DA0605 00470 JP C,0506H ; THEN JUMP BACK TO ROM
FFE3 FE80 00480 CP 80H ; IF IT'S A GRAPHIC CHAR.
FFE5 D2A604 00490 JP NC,04A6H ; THEN JUMP BACK TO ROM
FFE8 3AEFFF 00500 LD A,(MASK) ; GET THE INVERSION MASK
FFEB B1 00510 OR C ; AND 'OR' IT WITH CHAR.
FFEC C37D04 00520 JP 047DH ; JUMP BACK TO ROM
FFEF 00 00530 MASK DEFB 00 ; REVERSE MASK (INITIALLY
00540 ; SET FOR NORMAL DISPLAY)
00550 ;
FFB0 00560 END LOADER ; AUTO-EXECUTE DRIVER

```

Program Listing 1. Disk based inverse video driver.

Die Lösung ist der Einbau eines neuen Schieberegisters für Bit 7, welches das Signal für inverse Zeichen liefert. Das neue Schieberegister welches ich Z 10a nenne, liegt parallel zu Z10, sodaß sein Ausgangssignal gleich mit dem Signal für normale Zeichen ist.

Die Umschaltung zwischen normalen Zeichen und inversen Zeichen geschieht durch die Logik nach Bild 2. Im normalen Darstellungsmodus ist Pin 8 von IC X2 high, d.h. Pin 4 von Z 26 low und Pin 10 von Z26 ist high, wie bei der Darstellung von alphanumerischen Zeichen. Wenn im Invers Modus Bit 7 high ist, liegt Z von Z27 auf low. Der Unterschied zwischen diesem Signal und DLY 7 wird von Z42a erkannt, dessen Pin 3 dadurch high wird. Dieses High geht nun zu Z 10a welches dafür sorgt, daß die Bits im gleichen Takt wie bei Z10 ausgegeben werden. Die Ausgangssignale von Z10 und Z10a gehen dann auf ein Exklusiv-Oder wodurch die Zeichen invertiert werden.

Wird auf normale Darstellung umgeschaltet, d.h. Bit 7 an Z 27 ist low, so wird Pin 3 von Z 42a low, weil über die Gatter auf Bild 2 an Pin 1 und 3 von Z 42a das gleiche Signal liegt. Da jetzt der Ausgang von Z 10a ebenfalls low ist, werden die Bits von Z 42a nicht invertiert.

Z 26 bewirkt, daß Z10 alphanumerische Zeichen lädt, wenn die Signale CLK, DLY Bit 7, DLY BLANK und DLY L8 alle low sind. Z 26a bewirkt das gleiche für Z 10a, benutzt jedoch nicht das DLY L8 Signal, da DLY L8 dazu dient die 4 unbenutzten Zeilen unter jedem Zeichen zu erzeugen. Somit werden diese 4 Zeilen immer invertiert.

Voraussetzung für inverse Darstellung ist, daß Kleinschreibung installiert ist, da in einem TRS-80 ohne Kleinschreibung Bit 6 auf Low gezwungen wird wenn Bit 7 high wird. Deshalb werden alle inversen alphanumerische Zeichen als inverse Steuerzeichen dargestellt.

Helligkeitsumschaltung

Die Helligkeitsumschaltung funktioniert prinzipiell genauso wie die Invers-Umschaltung. Nur wird hier das Ausgangssignal des Schieberegisters Z10a Pin 13 über IC X5 auf die Basis des Video Transistors Q1 gegeben.

Beide Umschaltungen werden mit 'CHR' aktiviert.

BEISPIEL: PRINT CHR;"HALLO";CHR

Basic-Treiber

```

20 CLEAR200:CLS
50 MS=PEEK(13561)+PEEK(16562)*256
60 MS=MS-49
80 POKE16561,MS-(256*INT(MS/256))
90 POKE16562,INT(MS/256)
100 M=MS
110 IFM>32767THENM=M-65536
130 FORI=M+1TO M+49
140 READN
150 POKE I,N
160 NEXT I
180 MH=INT((MS+49)/256)
190 ML=(MS+49)-(256*MH)
200 POKE M+16,ML:POKE M+17,MH
210 POKE M+43,ML:POKE M+22,MH
220 POKE M+43,ML:POKE M+44,MH
250 POKE16414,(MS+1)-(256*INT((MS+1)/256))
260 POKE16415,INT((MS+1)/256)
270 CLEAR:END
290 DATA 221,110,3,221,102,4,218,154,4,121
300 DATA 254,127,32,9,58,0,0,238,128,50
310 DATA 0,0,201,221,126,5,193,40,1,119
320 DATA 121,254,32,218,6,5,254,128,210,166
330 DATA 4,58,0,0,177,195,125,4,0

```

HEFT
12
März
1986

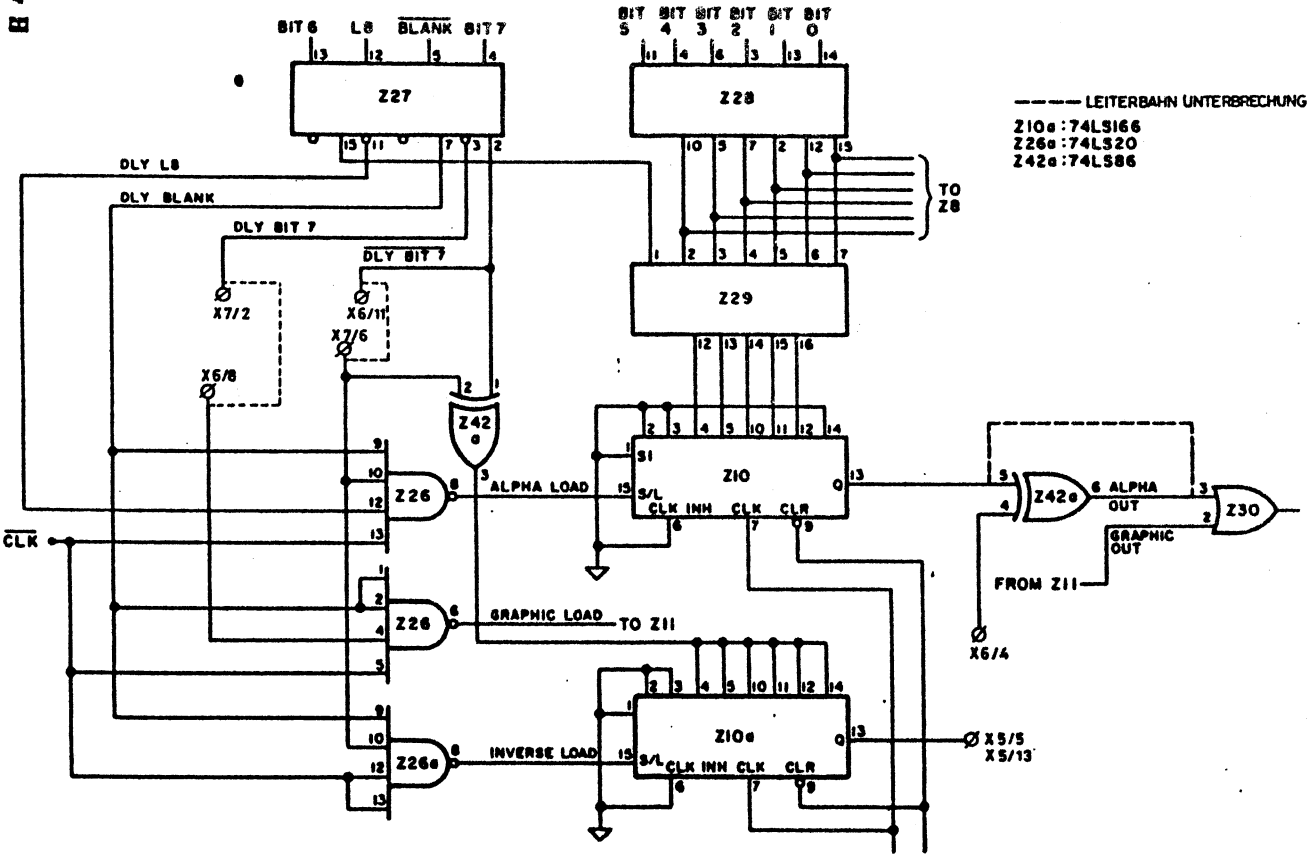


Bild 1

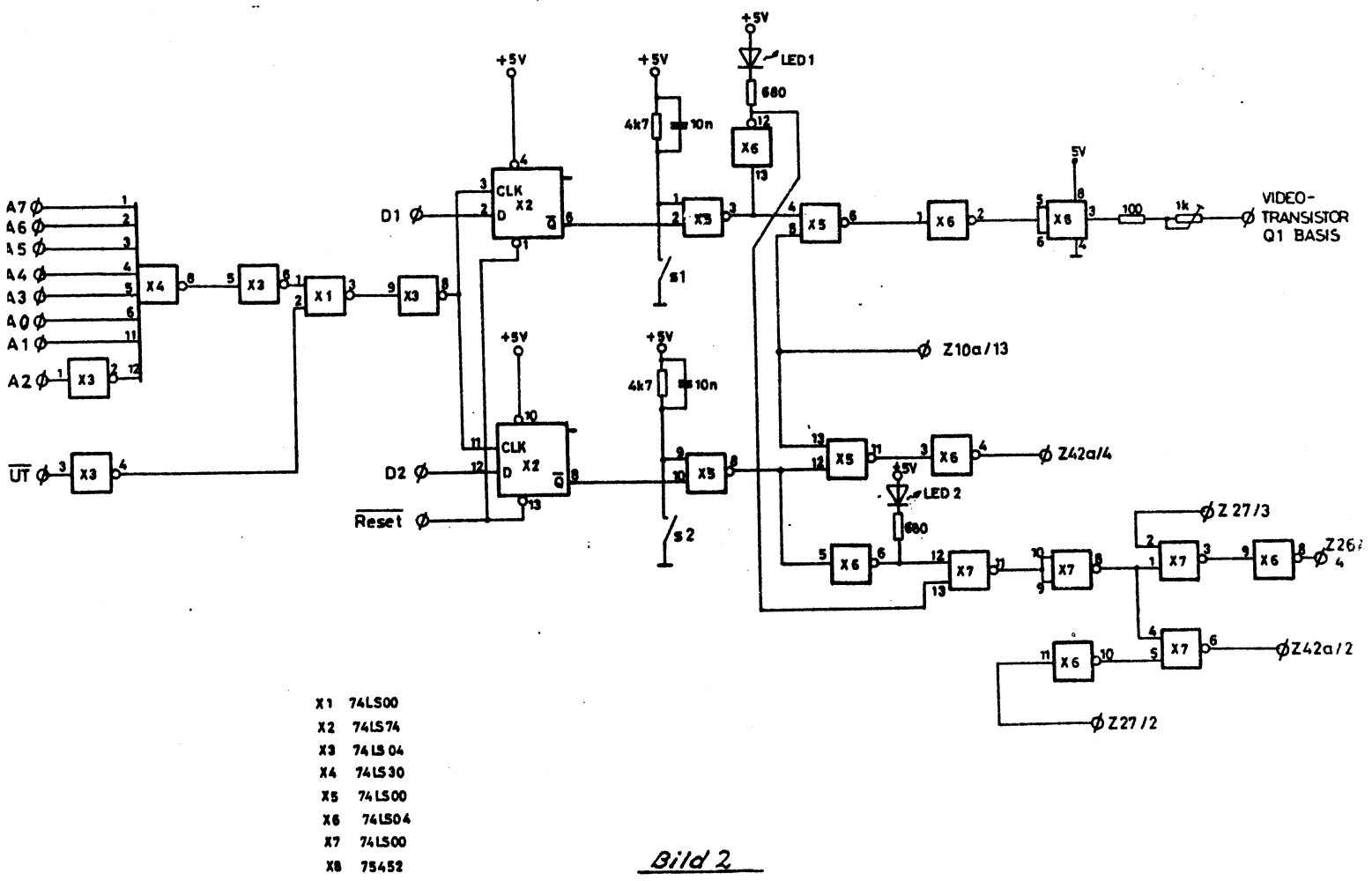


Bild 2

BÖRSE

TRS-80 Modell III zu verkaufen
 einschließlich:
 - Cassettenrecorder CTR-604
 - umfangreiche Software und Literatur (inkl. KUM-Listing)
 Preisvorstellung ca. 600 DM

Außerdem abzugeben ist ein Fernschreiber Lorenz C15 mit
 Beschreibung, diene lange Zeit als erster Drucker am
 GENIE II. Preisvorstellung ca. 100 DM
 Tel. 02948/542 R.Jablotschkin

RB - RS-232 Schnittstelle zu verkaufen;
 Kompl. mit Software und allen Kabeln;
 passend für TRS80 M1 und Genie I + II.
 Preis: 120,-DM. Peter Spieß, Trugen-
 hofenerstr. 27, 8859 Rennertshofen I

Vorkaufe

Genie I, 64K	400.-
Floppy-Contr.	200.-
Doppler	120.-

* Möglichst als Einheits
 für 700.-

80-Zeilenkarte für
 Genie I + Software 150.-
 Herbert Alber
 Niederschach

Das System besteht aus dem KONTEK-1s, einem ZENITH-Monitor, dem SEIKOSHA-Drucker GP100A und zwei BASF 6108 DS-Laufwerken incl. Controller, Netzteil und Gehäuse sowie Software, Büchern und Unterlagen.

Verhandlungsbasis für das ganze Paket wären 1200,- DM. Ich würde jedoch die Geräte auch einzeln verkaufen.

Sollte jemand Interesse an dem System haben, so kann er mich am Tage unter der Rufnummer 040/215011-15 erreichen.

Hans Jürgen Bozek

Verkaufe:
 TRS 80 Modell I, 48 KB RAM, 2 x 40 Tr. LW (SS,DD)
 VB: 1450,- DM
 Klaus Hermann (Tel. 07127/70024)

Nachdem ich mir kürzlich zwei Laufwerke angeschafft habe, ist bei mir das Druckerinterface EG 3016 über. Ich wäre Dir zu unendlichem Dank verpflichtet, wenn Du das gute Stück in der Börse anbieten könntest. Als Verhandlungsbasis habe ich mir DM 150,- vorgestellt. Dienstlich bin ich z.Zt. unter der Rufnummer 05401/2422 zu erreichen.
 Mit freundlichen Grüßen
 Martin

Verkaufe Apple-Schaltnetzteil
 Die technischen Daten wie folgt:
 +5V/3,5A .. -5V/0,5A .. +12V/2A .. -12V/0,5A
 Preis 78,-DM
 Ulrich Böckling Tel.: 0261/69522

Üblicherweise lassen wir bei einem Verkauf von Dreifachrechner eine entsprechende Notiz formulieren oder schicken für die Floppmarketecke zwei, drei selbstgedruckte Sätze. Die Note, die ich anzubieten habe, ist jedoch so heiß, daß sich eine längere Beschreibung lohnt. Schuld daran ist Helmut Bernhardt, der seine Hard-Ideen wohl vorsichtshalber gerne an meinem Computer ausprobierte. Das Gerät ist deshalb gespickt mit seinen digitalen Schmankerln. Schuld ist auch meine Sucht, aus dem Genie ein Universalgenie zu machen. Deshalb hat es auch dies und jenes von RE-Elektronik und von Udo Jourdan. Und schuld ist schließlich meine Bank, die mir nicht böse ist, wenn schon wieder ein gewisser Betrag ins Hobby fließt. Schließlich leben die davon, daß unsereiner Schulden hat.

Das Wichtigste und Beste in Kürze, falls ich nicht wortreich in Marktschreierei entgleite: Das Gerät ist ein ehemaliges Genie I. Der neue Name in der Überschrift ist natürlich ein Scherz, aber vom Genie I unterscheidet sich der Computer tatsächlich in so vielen Punkten, daß ich über den Scherz eigentlich nicht lachen kann: 256 KB freies RAM, vorbereitet auf 1 MB (wer fit mit dem Lötkolben ist, macht das in einer halben Stunde). Da die CPU nur 64 KB gleichzeitig erreichen kann, gibt es ein ausgeklügeltes Banking-System:

In Portionen zu 32 KB kann man wahlweise in der oberen oder unteren Speicherhälfte (0000-7FFF oder 8000-FFFF) einfach die Bank wechseln. Da es auch unten geht, ist der Computer für CP/M (jedwede Version!) geeignet. Der Banker kann außerdem bestimmte Bereiche des Speichers getrennt für Lese- und Schreibzugriffe enablen. Das bedeutet, daß man z. B. für LPRINT CHR\$(10) nicht extra eine Utility laden muß, sondern einfach der Interpreter direkt unpaket. Und schließlich kann Helmut's Banker die I/O-Adressen für Bildschirm, Tastatur, Floppy usw. ins Hirn legen, so daß CP/M oder ein x-beliebiges anderes System mit der Maschine klarkommt.

Da ist ein Speed-Up drin, bei dem man mitten in der Arbeit von 1,77 auf 3,54 MHz und zurück schalten kann. Der ziemlich lendenlahme Cursor von SCRIPSIT wird plötzlich ganz schön munter. Bei PACMAN hat der Spieler keine Chance mehr, es sei denn, erschaltet rechtzeitig auf 1,77 MHz.

Die HRE 1b ist ebenfalls eingebaut. Sie hat eine Matrix von 354 X 192 Punkten. Sie ist nachträglich auf 16 KB aufgemotzt (ursprünglich 12 KB). Dieser Speicher geht nicht den Hauptspeicher von 64 KB verloren, denn die RAMs befinden sich auf der Zusatzplatine. Wenn man nicht gerade eine Graphik bearbeitet, steht dieser Speicher für beliebige Zwecke als Datenfriedhof zur Verfügung. Außerdem ist die HRS nachträglich sauber dekodiert. Sie braucht nur 6 Ports, verseuchte aber ab Werk so ziemlich den kompletten I/O-Adreßraum bis zum Port 127. Jetzt sind durch sie nur noch die Ports 0-7 belegt.

Da ist auch ein Interrupt-Timer. Er generiert wahlweise nichtmaskierbare Interrupts (NMI) oder maskierbare (INT). Seine Frequenz läßt sich in weiten Grenzen einstellen. Mit Floppy kriegt man seine Interrupts von ihr. Aber dann ist das Ding dennoch interessant, weil man in regelmäßigen Abständen NMIs ausgeben kann, die irgendeine Funktion erfüllen. Sie verzweigen beim Z80 an die Speicherstelle 0000h, wo man dank des Bankers eine eigene Routine unterbringen kann. Damit lassen sich Programme überlisten, die die maskierbaren Interrupts abschalten, um sich nicht in die Karten gucken zu lassen.

Der Adreßraum 3600-3FFF ist bei der Computerfamilie unsereins durch den Tastatur vorbehalten. Helmut hat es fertiggebracht, die Tastatur auf 256 Bytes zurückzustutzen, so daß der Bereich 3600-3FFF für weiteres zur Verfügung steht. Er nannte das in der alt 8085 "Ein sicheres Plättchen". Wahrscheinlich kein Programm für TRS-80 oder Genie steht diesen Speicher-

reich (weiter H-DOS, versteht sich, aber davon später mehr).

Ein weiterer Speicherzugewinn ist dadurch entstanden, daß Helmut eine Platine entwarf, die den Adreßraum 3000-37DF versechzehnfacht. Das sind runde 32 KB. Auf dieser Platine sind 16 Steckplätze für EPROMs oder statische RAMs. Einer davon ist vom Level-IV-EPROM von TCS belegt. Der Rest (bisher gut zur Hälfte bestückt) kann andere Speicher aufnehmen, so daß man beispielsweise ständig einen HRS-Treiber verfügbar haben kann o. ä..

Der eingebaute Lautsprecher ist regel- und abschaltbar, falls das Gebrauchs bei DEBUG einmal nerven sollte. Eine Ohrhörerbuchse hat er auch, um den Ärger mit den Nachbarn und der Familie in Grenzen zu halten.

Die Tastatur ist vollständig mit Tasten bestückt. So sind die deutschen Umlaute, das B, der Cursor usw. über die Tastatur erreichbar. Der Interpreter, auch Programme wie TSCRIPS spielen dabei mit, so daß man sich die CTRL-Akrobatik für diese Zeichen schenken kann. Zusätzlich gibt es 7 programmierbare Funktionstasten, die ganze Befehlsstrings oder sonstwas mit einem Tastendruck ausgeben können. Die Strings dürfen auch Steuerzeichen wie NEW LINE etc. enthalten.

Es gibt eine Anschlußbuchse für eine komplette zweite Tastatur und eine für einen Joystick. Bei letzterer habe ich zwei Kabel beim Löten vertauscht, so daß bei einer Cursorrichtung stattdessen ein Buchstabe ausgegeben wird. Ich habe selbst keine Lust mehr dazu, aber die Behebung dieses Mangels dauert keine zehn Minuten. Gegf. reden wir noch darüber.

Am Gehäuse ist ein CPU-Resettknopf angebracht. Im Gegensatz zu dem an der Rückseite, der nur einen NMI ausgibt, funktioniert dieser immer, denn er versetzt den Z80 in den Einschaltzustand. Die Speicherinhalte bleiben allerdings erhalten, so daß man nach einem Hänger einfach weiterarbeiten kann. Nur Zeiger auf BASIC-Texte, Variable usw. werden in ihren Nullzustand versetzt, denn das System bootet natürlich neu.

Der Cassettenrecorder ist eingebaut (wird mitgeliefert; es sind weiterhin zwei Recorder anschließbar). An seiner Stelle ist ein Ziffernblock wie beim Genie II. Zusätzliche Tasten, die ihn zu einem Hexblock machen, sind bereits fertig zur Einbau vorbereitet und werden beigelegt. Dazu hatte ich selbst allerdings bisher keine Lust.

Das alte einfache Netzteil ist durch ein wesentlich stärkeres Schaltnetzteil ersetzt, so daß für eine Fülle weiterer Um- und Anbauten genügend Power zur Verfügung steht. Wie sich in der Praxis zeigte, nimmt es auch Kurzschlüsse nicht übel.

Die Floppies, die ich ebenfalls verkaufen möchte, sind auch modifiziert. Sie enthalten zwei Laufwerke mit 80/DS/DD und eins mit 40/SS/DD. Durch einen Umschalter kann wahlweise auf Laufwerk 0 (80) oder 2 (40) gebotet werden, so daß man praktisch jedes fremde DOS fahren kann. Evtl. wäre ich interessiert, eines der Laufwerke selbst zu behalten. Darüber wäre zu verhandeln.

Der Monitor ist bernsteinfarben.

Außer Helmut's Supertanker gehört zum Gerät auch noch der EG 64 MBA von TCS. Daran habe ich eine Platine angebaut, die alle notwendigen Anschlüsse zum Port-1-D hat. Es lassen sich damit gleichzeitig ca. 20 externe Geräte ansteuern, die über Ports betrieben werden.

Eine DOS-Fuchste und ein serielles Interface sind auch dabei. Beide sind aber noch nicht an den Computer angeschlossen. Da sich die Bastelerei in der letzten Zeit sehr gehäuft hatte, waren immer andere Dinge wichtiger. Der Einbau ist nicht schwierig. Gegf. kann ich Hinweise dazu geben.

Schließlich ist noch ein Zweifach-Eusextender dabei. So lassen sich außer den Floppies noch weitere Geräte auf den Euse stecken. Da die Steckleisten einen recht weiten Abstand haben, passen gut und gerne noch zwei weitere dazwischen, so daß ein Vierfach-Extender entsteht. Die Leisten stecken in Slots und können herausgenommen werden. Man kann deswegen Zusatzgeräte wahlweise mit Vater- oder Mutteranschluß betreiben. Letzteres ist bei den gängigen Geräten für Genie-Computer üblich.

Der Leser wird sich mit Recht fragen, weshalb ich ein solches Gerät nicht selbst behalte. Weil das Bessere der Feind des Guten ist. Inzwischen habe ich nämlich ein Genie III s, bei der all' das und noch einiges mehr von Hause aus schon drin ist.

Als Verhandlungsbasis für den Preis des kompletten Systems stelle ich mir DM 2.500,- vor. Allein die Floppies waren wesentlich teurer. Es dürfte sich rund um ein Drittel der Summe der Neupreise aller Einzelteile handeln. Insofern denke ich, daß dieses Angebot fair genannt werden kann. Wenn Interessenten nur einzelne Komponenten des Systems haben möchten, wird man sich über den Preis jeweils unterhalten müssen.

Umsonst gibt's dazu noch M-DOS, eine erweiterte Version von G-DOS. Seine Features möchte ich hier nicht beschreiben, denn das wären ein paar weitere Seiten. M-DOS holt aus den Hardware-Erweiterungen jeweils das Äußerste heraus und ist damit jedem anderen DOS, das für die Computer unseres Clubs im Umlauf ist, überlegen.

Arnulf Sopp

Z80 - Assemblerkurs zu verkaufen!!!

Die Z80-Maschinensprache kann man nur am praktischen Beispiel erlernen! Diese Tatsache ist mir das erstmal aufgegangen, als ich die Z80-Bibel (Programmieren des Z80 von Rodney Zaks, erschienen im SYBEX-Verlag) durchzuackern versuchte. "Da ich jedoch einen TRS80 mit einer Z80-CPU habe, kann das ja eigentlich nicht allzu schwer sein", dachte ich und versuchte mit allen möglichen Mitteln das Geheimnis der Mnemonics zu lüften. Vergebens!!! Leider gibt es trotz der beinahe schon in Computergenerationen zählenden Standzeit des TRS80 keinen einigermaßen brauchbaren Kurs zum Erlernen der Maschinensprache auf dieser Maschine. Was tun also, um doch noch zu einem einigermaßen gefestigten Grundwissen über die Maschinenspracheprogrammierung zu kommen?

Ein Kurs mußte her!

Nach einer gewissen Zeit der Umschau habe ich mich für den Z80-Kurs von Christiani entschieden. Grundlage der Entscheidung war dabei die Tatsache, daß ich nach dem Softwarekurs auch den anschließenden Peripherie-Lehrgang machen wollte.

Gesagt, getan! Beide Kurse sind beendet (jeweils ca. 4 Monate a' 20 Stunden) und ich glaube mich inzwischen nicht mehr zu den ganz feuchten Greenhorns unter den Assemblerfreaks zählen zu müssen. Natürlich reichen die erworbenen Kenntnisse nicht, um unserem Assemblerpapst Arnulf Sopp das Wasser zu reichen und müssen noch erheblich vertieft werden. Das jedoch kann ich ganz gut auf dem TRS80 machen!

Und so steht nun also ein Z80-Kurs (Microprofessor) und ein Peripheriebausteine-Lehrgang herum und verstaubt. Diesen Zustand möchte ich ändern! Ich möchte die ganze "Lernanlage" verkaufen und von dem Erlös einen Olivetti M10 oder ein Model 100 von Tandy anschaffen (logisch muß ich noch was drauflegen!). Es steht also folgendes zum Verkauf:

- | | |
|---|---------------|
| 1 Z80-Kurs (4 Lehrbriefe im Ordner und ein Microprofessor Lehrsystem) zum Original-Preis von | 676,- |
| 1 Peripherie-Bausteine-Kurs (4 Lehrbriefe im Ordner mit einer Erweiterungsplatine mit CTC, PIO und 8255) zum Original-Preis von | 642,- |
| | <u>1318,-</u> |

Das Ganze soll für ca. 900,- DM (VB) über den Tisch gehen, wobei ich auch an einem Tausch gegen einen M10 oder ein Model 100 interessiert bin (natürlich mit entsprechender Ausgleichszahlung)!

Interessenten melden sich am besten telefonisch bei mir. Meine Adresse und Telefon-Nr.: Hartmut Obermann

Schwalbacher Str. 6
6209 Heidenrod/Kemel
☎ 06124 / 3913

HEFT
12
März
1986

20

Berliner Putzfrau zum Computer: »Ick hör imma, Du kannst allet - staub Dir doch selba ab!«

»Frollein Rosi, laut Computer schreiben Sie nur ganze 1,348 Briefe an einem Tag, machen pro Zeile 0,0975 Fehler und verbringen 2,416% Ihrer Arbeitszeit auf der Toilette! Was haben Sie dazu zu sagen?«
»Wenn ich aufrunden darf, Chef, Sie gehen mir 100% auf den Wecker!«

Hardware im letzten Augenblick!!!

Am ersten Wochenende (Sa./So. 01/02) im März hatte ich Besuch von unserem Clubinfo-Macher Jens Neueder. Samstags war auch noch Ulrich Böckling anwesend, von dem ihr in diesem Info einen Hardwarebeitrag findet!

Während des Treffens kam unter anderem zur Sprache, daß man von dem laufenden ECB-Projekt schon lange nichts mehr gehört hat. Es gibt schon Gerüchte, die Sache sei ganz gestorben! Dem, das sei hier noch einmal ausdrücklich bemerkt, ist nicht so!

Nachdem mir der Initiator des Euro-Card-Bus-Projekts, Walter Zwickel, im Sommer die Layouts für die zu fertigende Platine geschickt hatte, wurden diese von Ulrich Böckling auf Folien umkopiert. Die Herstellung der Platine hat dann einer unserer eifrigsten Hardwarebastler, Manfred Held übernommen.

Und obwohl alle an der Entwicklung und Fertigung der Adapterplatine Beteiligten immer ihr Bestes gaben, waren Verzögerungen nicht ganz zu vermeiden. Schließlich gibt es für einige Mitglieder hie und da auch etwas wichtigeres zu tun, als immer nur am Computer zu hängen! Nichts desto trotz ist die Platine nun endlich (zumindest in zwei Prototypen) fertiggestellt.

Tester gesucht!!!

Nun muß noch die Prüfung auf ev. Fehler im Layout oder in der Schaltung erfolgen. Leider habe ich gerade erfahren, daß Walter Zwickel auf einen PC-Kompatiblen umgestiegen ist und uns damit der kompetenteste Mann zum Testen wegfällt. Sollte sich jemand als Tester zur Verfügung stellen, melde er sich bitte bei Manfred Held oder bei mir. Wenn alle Stricke reißen, werden Ulrich Böckling und ich das Auschecken übernehmen. Die Fertigstellung des Projekts wird dann aber noch länger auf sich warten lassen, da wir beide nicht allzuviel Zeit haben. Wenn alles glatt geht, können die ersten Karten nach dem Clubtreffen ausgeliefert werden. Die Platinen werden voraussichtlich ca. 10,- DM pro Stück kosten und auch für den technisch nicht allzu versierten (einen LötKolben sollte man schon bedienen können) zusammenbaubar sein. Die Bauteile können ev. in Sammelbestellung beschafft werden, darüber kann man jedoch auch auf dem Clubtreffen noch entscheiden. Sollte jemand es sehr eilig haben, kann er die Karte auch sofort (vor den letzten Tests) bei Manfred Held anfordern!

Manfred Held, der mich telefonisch bat diesen Bericht über den Entwicklungsstand in Sachen ECB-Bus zu schreiben, gab mir zusätzlich den Auftrag, folgende Hinweise bzw. Bitten zu formulieren:

1. Die von ihm vorgestellte Highspeed - Umschaltung kollidiert mit manchen CP/M-Erweiterungen! Aus diesem Grunde sollten CP/M-Benutzer die Adresse der Schaltung von FEh in FRh ändern. Wie man das macht ist im Hardwarebeitrag von Ulrich Böckling kurz beschrieben (es ist nur eine Adressleitung zu ändern)!
2. Um bei der weiteren Entwicklung von Karten für den ECB-Bus Kollisionen mit schon bestehenden Hardwareerweiterungen zu vermeiden, werden alle Mitglieder des Clubs aufgefordert, ihnen bekannte Adressen von Erweiterungen im Info zu veröffentlichen bzw. an - Manfred Held, Stirner Str. 22, 8835 Pleinfeld zu schicken! Z.B. hat Manfred eine Hardwareuhr für den ECB-Bus gebaut, welche die Adressen CBh-CBh belegt.

Ich hoffe, möglichst viele Mitglieder kommen dem Aufruf nach und verabschiede mich bis zum Clubtreffen mit den freundlichsten Grüßen,
Euer

Rainer Obermann

??? FRAGENKASTEN ???

Für den Fragekasten:

Wer hat ein kommentiertes Listing des Bildschirmditors (SEdit etc.) oder weiß, wie das Ding funktioniert. Jedes Mal, wenn ich mir das disassemblierte Listing anschau, finde ich heraus, daß das Programm überhaupt nicht laufen dürfte, obwohl es das zweifellos tut, nur fatalerweise zusammen mit der 80-Zeichen-Karte nicht!

Wer kann mir bei "Lucifer's Realm" im Ventilationsschacht weiterhelfen. Habe leider nichts dabei, um den CANVAS zu CUT-ten.

Wer hat Informationen, Programme oder das Handbuch für die 80-Zeichen-Karte?

Wer hat eine Assembler-Routine, die die Tastaturroutine ersetzen kann, so daß Umkodierungen vorgenommen werden könnten (Umlaute realisieren oder ESCAPE mit Clear-Taste, denn <Shift Pfeil Abwärts> ist nun wirklich grausam)?

Gerald Schröder

TECPIP3 scheint nicht alle Codes an den Drucker weiterzugeben, die zwischen 8-Zeichen eingeschlossen sind. Die Codefolge \$2401\$ (12h, 20h, 01h bzw. 27, 26, 11), mit der die freidefinierbaren Zeichen des Gemini-10X zugeschaltet werden sollen, wird nicht befolgt. Wer hat damit Erfahrungen und kann mir helfen?

Für den Fragekasten:
- Wie steht es mit dem Projekt 'ECB-Bus-System' ?
- Lange nichts mehr davon gehört, ist es eingeschlafen oder kurz vor der Vollendung ?

- Seit ein paar Tagen sind wir DFU-fähig. Akustikkoppler, V.24 und Software (Multicom) alles von RB-Elektronik. Wer Lust uns zu schreiben, Mailboxen und außerdem per Post oder über Corner Box (02927/650) an 'JABO'.

Rainer Jablotschkin

Ehe der große Stop kam

Das Volk der Bit im Lande Byte,
man schrieb 01 der Nanozeit,
erwählte sich aus eigenen Kreisen
einen König, welcher Flop geheißten.

Flop war ein schneller Interrupter,
er zog Register, hielt Adapter,
und an der Spitze der Regierung
gewann er manche Assemblierung.

Flop emulierte fulminant
und fing den Cobol mit der Hand.
Einst sprach der 1. Programmierer
der Bitpartei zu Flop, dem Führer:
Es fehlt dir in dualen Sinn
zum guten Glück die Königin!

Des ernstesten Rates Ironie
bemerkte nur das Parity;
die andern wählten klar und klipp
dem Flop ein Weib, die flotte Flip.

Flip-Flop, ein Hochzeitspaar in Pracht,
man schrieb die Nanozeit 08,
Das Bitvolk jauchzte - auf zum Run,
zum Doppeljob als Festprogramm.

Das war ein Flippen und ein Floppen,
ein Zyklus, nicht mehr abzustoppen,
denn kybernetisch digital
wie weiland hexadazumal
Klang die Musik der Elektronen
schon hoch bis in die Tier-Zonen.

Am tollsten trieben es die Timer,
der Multiplexer ging in Eimer.
Mit Dezimalen fuhr man Schlitten,
das Gleitkomma war ausgeflitten.

Im großen Overhead-Büro
kniff man Compiler in den Po,
und Bits addierten splinternackt
im Puffer zum Dreivierteltakt.

Ein Interface sang ordinär
die Watson-Hymne rein binär.
Man hatte Maß und Ziel verloren...
dem Supervisor kams zu Ohren:
Du lieber Dump, welch ein Gezeter -
Greif ein, oh großer Operator!

Ein Druck - es kam der große Stop.
"Halt" war das letzte Wort von Flop.
Abrupt erstarb der Bit-Gesang,
na, seht Ihr wohl? Cherchez la femme!

Im Lande Byte seitdem geht stumm
bei Tag und Nacht der Error um!

**Auch wenn Computer-Freaks Rech-
nerprogramme ohne kommerzielles
Interesse nur tauschen möchten,
bekommen sie es schnell mit dem
Staatsanwalt zu tun. Oder es mel-
den sich Rechtsanwälte, die durch
»Abmahnschreiben« schnelles Geld
einzustreichen gedenken.**

Abmahnfälle



Auch Computerprogramme unterliegen dem Urheberrechtsschutz.

An die »ges. Vertretung Avon Rolf Koch«*) war der lange Brief adressiert, in dem der Vater Holger Koch mit rund 1100 Mark für »Gebühren und Auslagen« zur Kasse gebeten wurde. Absender war eine Rechtsanwaltskanzlei im fernen München, die in dem offensichtlich vervielfältigten Schreiben Sohn Rolf »Software-Piraterie« anlastete und nun vom Vater per Abmahnung die Versicherung erheischte, daß der Sprößling Kopien seiner Programme für den Heimcomputer niemals wieder zum Tausch oder Verkauf anbiete. Widrigenfalls seien für jeden Verstoß gegen § 97 des Urheberrechtsgesetzes 2000 Mark Vertragsstrafe fällig. Der vorläufige »Gegenstandswert« betrage 50 000 Mark. Als so hoch sei der Schaden einzuschätzen, den »Pirat« Rolf

dem Hersteller seiner Computerspiel-Programme mit Raubkopien zufügen könne. Und nach dieser Schadenshöhe berechnet sich die Abmahngebühr von 1100 Mark. Die Kochs hielten leicht verstört Familienrat, vermochten sich jedoch keines Unrechts bewußt zu werden: Rolf besitzt einige Computerspiele für seinen Rechner, tauscht die relativ teure Software mit Freunden, hatte aber auch auf eine Annonce hin seine Programme einem anderen Freak angeboten. Eigene Kopien hatte Rolf nicht angefertigt. Und Geld war bei Rolfs Tauschaktionen zu keinem Zeitpunkt im Spiel gewesen. Allerdings: Die zuständige Staatsanwaltschaft hatte einige Zeit zuvor gegen Rolf ermittelt, ob er Raubkopien von Computerprogrammen

rechtswidrig unter die Leute bringe. Und spätestens hier beginnt der Fall Rolf allgemeine Bedeutung zu erhalten. Der Verdacht war nämlich aufgekommen, nachdem Rolf auf die Annonce reagiert hatte; denn beim Inserenten hatte die Polizei wegen des dringenden Verdachts der Software-Piraterie Hausdurchsuchung gehalten - und war fündig geworden. Was die Beamten auch fanden, war eine große Menge an Adressen weiterer Computerbesitzer, darunter die von Rolf.

Mitgefangen, mitgegangen: Nach der Ergänzung des Urheberrechtsgesetzes im Mai des Jahres 1985 haben Ermittlungsbehörden nicht mehr nur auf Antrag eines Geschädigten tätig zu werden, sondern auch bei vermutetem öffentlichen Interesse an Strafverfolgung. So geriet auch Rolf in die Ermittlungsmühle. Für ihn verlief die Geschichte allerdings unproblematisch: Der Staatsanwalt stellte das Verfahren ein.

So klar die Situation danach war, so vertrackt ist die Rechtslage nach dem Abmahnschreiben aus München: Vater Koch hatte sich nach der ersten Rechtsanwaltspost vehement gewei- gert, die Unterwerfungserklärung zu unterschreiben und damit ein rechtswidriges Verhalten seines Sohnes anzuerkennen: folgerichtig hielt er auch die 1100 Mark Gebühren für gegenstandslos. Er hat vielmehr den dringenden Verdacht, daß die Schuldfrage die Rechtsanwälte gar

nicht sonderlich interessiert: Vielmehr sieht er - je länger die Sache sich hinzieht, desto deutlicher - die Forderung aus München als »Abmahnmasche«, mit der sich die Kanzlei an alle Adressaten wendet, die sie über Ermittlungsverfahren aufspindig machen kann.

Vater Kochs entschlossene Weigerung zog weitere Rechtsanwaltspost nach sich: Werden die Gebühren bis zu einer letzten »Nachfrist« nicht gezahlt, soll Vater Koch auf dem Wege einer Unterlassungsklage zur Zahlung bewegt werden. Nach nochmaligem Gespräch mit dem Staatsanwalt will Holger Koch nun erst recht standhaft bleiben. Eine Zuwiderhandlung hätte nur erwiesen werden können, wenn der Staatsanwalt eine Hausdurchsuchung angeordnet hätte. Diese Maßnahme mochte der anscheinend weise Staatsanwalt jedoch nicht gegen bisher unbescholtene Bürger wegen des geringfügigen Verdachts einer eventuellen unerlaubten Weitergabe von Raubkopien einleiten, zumal selbst bei begründetem Verdacht eine Wiederholung auszuschließen sei. Holger Koch appellierte denn auch weniger juristisch, vielmehr pädagogisch an die Kanzlei, den Versuch zu unterlassen, mit dem Mittel der Abmahnung Jugendliche zu kriminalisieren. Rolf habe sich nicht rechtswidrig verhalten. Basta.

Bei Redaktionsschluss war noch offen, ob der Appell an die Vernunft die erwünschte Wirkung gehabt hat.

test rät: Besonders Eltern, die ihren Kindern einen Heimcomputer unter den Tannenbaum gestellt haben, sollten vorbeugend aufklären: Gegen den Tausch von Computerprogrammen im Freundeskreis ist so lange nichts einzuwenden, wie die Gewähr besteht, daß keine Kopien hergestellt werden. Bei der Weitergabe an Fremde, die dann ohne Wissen des Eigentümers Raubkopien ziehen, macht sich

der Ausleiher mit strafbar. Die Frage, ob Computerprogramme überhaupt als urheberrechtlich schützenswerte Werke anzusehen sind, hat der Bundesgerichtshof in einem Urteil vom 9. Mai 1985 grundsätzlich bejaht (Az.: I ZR 52/83). Bei rechtswidriger Weitergabe von Programmen haben Freaks also mit Staatsanwalt, Klagen, Strafen und eben auch mit Abmahnschreiben zu rechnen.

Aus aktuellem Anlaß bekamen wir einen Brief von Herrn Adelman durch einen befreundeten CLUB zugesandt.

Wir möchten Euch diesen Brief ausschnittsweise darlegen. Gleichzeitig solltet Ihr Euch über dieses Thema einmal Gedanken machen. Eine Diskussionsrunde über diese Thematik wäre bei unserem kommenden Clubtreffen sicher sehr interessant.

// Kopieren ist nur in ganz wenigen Fällen verboten. Hätten wir in der Rechtspflege seriöse Zustände, würde die Behörde, bevor sie durchsucht, erst folgende Fragen prüfen:

Liegt ein Strafantrag vor ?

Diese Anträge fehlen meist überhaupt. Die Behörden durchsuchen ohne sie. Finden sie was, fragen sie bei allen möglichen Firmen an, ob eine einen Strafantrag stellen will. So kommen dann etwa 2 Strafanträge zusammen, aber 100 Programme sind erst mal beschlagnahmt worden.

Und genau hier läge der springende Punkt einer seriösen Rechtspflege; denn die Strafantragssteller haben oft gar kein Strafantragsrecht. Wenn sie ein Vertriebsrecht haben, können sie selbst nicht Strafanträge stellen. Richtig wäre es, wenn vor der Polizeirazzia geprüft würde, ob es überhaupt einen interessierten Urheberrechtshaber gibt.

Ist ein echtes geistiges Eigentum gestohlen worden ?

Selbst wenn es einen Erfinder des kopierten Spiels gibt, ist noch lange kein Urheberrecht verletzt. Die meisten Spiele haben gar keinen Schutzwert. Es ist gar kein "geistiges Eigentum". Es sind meist die selben Abläufe : Abschießen irgendwelcher beweglicher Ziele. Ob es eine Seeschlacht, ein Luftangriff oder eine Panzerschlacht ist, all das Zeug hat keinen Schutzwert. Wären die Strafverfolgungsbehörden seriös, würden sie vom Strafantragsteller ein Gutachten über den Schutzwert des Programms verlangen, bevor sie auf den Grundrechten anderer herumtrampeln.

Tatsächlich scheitern die meisten Strafverfahren dann beim Staatsanwalt, der alt aussieht, wenn er die Urheberrechtsqualität der Programme beweisen soll. Die Verfahren werden alle wieder eingestellt, meist ohne Buße und Folgen. Nur die ganzen Programme waren monatelang als Asservate unter Verschluss. //

// Das bedeutet, daß sich der User vor drohendem Unrecht durch die Rechtspflege schützen muß. Die Gerichte schaden mehr als sie nützen. Man verfährt so

Getauscht werden keine Kopien. Wer ein Programm hat, das ein anderer auch sehen soll, schickt das Original. Wenn der Partner eine Kopie fertigt, ist dies seine eigene Sache.

Im Hause behält man keine Originale, nur Arbeitskopien. Beschlagnahmt die Polizei eine Kopie, ist hieraus nicht zu beweisen, ob es eine Arbeitskopie oder eine "geräubte" ist.

Hierzu genort auch, sich total passiv gegenüber den Verfolgern zu verhalten. Man macht keine Angaben zur Sache. Wer eine Version vorträgt, schließt selbst 19 andere denkbare Versionen aus. Er torpediert seinen eigenen Schutz aus dem "im Zweifel für den Angeklagten". Man macht nur Angaben zur Person, keine zum Einkommen oder sonst was. Zu den Angaben zur Person gehören nur Name, Vorname, Alter, Wohnung und Beruf. Welcher Narr wollte behaupten, über ein geringes Einkommen zu verfügen, und dann sieht man Spiele im theoretischen Wert von 10.000 bei ihm.

Die Vertriebsfirmen wissen, daß das Urhebergesetz sie nicht schützt. Deswegen lassen sie heute Verträge unterschreiben, daß das erworbene Programm nur für eine bestimmte Anlage genutzt werden darf. Diese Verträge liegen den Kassetten oft bei. Nur kann kein Verkäufer solche Verträge durchsetzen. Die Feinde der Kopierer sitzen längst am kürzeren Hebel. Verurteilt werden nur die Schwätzer, die selbst glauben Unrecht getan zu haben.

Wer durchsucht wird, hat die Ehrenpflicht, seine Partner zu informieren. Die Hausdurchsuchung kommt dann wie die Tollwut nach einer Art Incubationszeit von bis zu 9 Monaten. Man hat also Zeit, alles zu ordnen. Man kann Spuren verwischen, indem man die Anschriften von Polizeibeamten unter die Listen der Tauschpartner aufnimmt, Anschriften mit Paßwörtern deckt und einen Ordner voller Rechnungen von Second-hand-shops hat, aus denen sich der Erwerb von Computerhardware und kompletten Spielesammlungen ohne Titel ergibt. //

CLUBBÜCHEREI

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

Ruedegeer 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 (als o mit Lern-Effekt).

Nr. 0010: BASIC: Matrix-Operationen

Busch Rudolf --- Franzis-Verlag GmbH, M nchen

Eine Software-Sammlung mit vielen nuetzlichen Programmen in Kursform (als o 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 Remardt --- 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 Roeckrath --- ?

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 --- IJE 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 betr. mit Mikrocomp.)

verschiedene --- MC, Franzis-Verlag, M nchen

Grundlagen, Schaltungstechnik, DF -Programme

HEFT
12
März
1986
88

Nr. 0017: GENIE DATA (Heft 1/83 bis Heft 3/85)

--- Heinz Hueben Verlag

Eine Sammlung von 13 Computerheften.

Nr. 0018: Basic Disk I/O Faster And Better & Other Mysteries

Lewis Rosenfelder --- IJG Inc., 1953 West, USA

Ein sehr gutes (englisches) Buch, das viele Tips und Tricks zum DOS, zur DISK, zu Files etc. zeigt (mit Programm-Diskette zum Buch).

Nr. 0019: Basic Faster And Better & Other Mysteries

Lewis Rosenfelder --- IJG Inc., 1953 West, USA

Ein sehr gutes (englisches) Buch, das viele Tips, Tricks und Routinen in BASIC bietet.

Nr. 0020: The Custom TRS-80 & Other Mysteries

Dennis Bathory Kitz --- IJG Inc., 1953 West, USA

Ein sehr gutes (englisches) Buch, das sich mit Soft- und Hardware für unseren Computer beschäftigt.

Nr. 0021: Microsoft Basic Decoded & Other Mysteries

James Farvour --- IJG Inc., 1953 West, USA

Ein kommentiertes Level II - BASIC in englischer Sprache.

Nr. 0022: TRSDOS 2.3 Decoded & Other Mysteries

James Farvour --- IJG Inc., 1953 West, USA

Ein kommentiertes TRSDOS 2.3 in englischer Sprache.

Nr. 0023: TRS-80 Disk & andere Geheimnisse

H.C. Pennington (G. Daubach) --- IJG Inc., 1953 West, USA

Das ins Deutsche übersetzte Buch beschreibt die Diskettenorg., die Zuordnung freier Speicherbereiche, die Behebung v. Diskettenfehlern etc.

Nr. 0024: TRS-80 Data File Programming (Model I/III)

Finkel & Brown ---

Ein Lehrbuch, wie man in BASIC Datenfiles programmiert und benutzt. Mit sehr vielen Beispielen (englischsprachig).

Nr. 0025: Programming Techniques for Level II BASIC

William Barden ---

Ein gutes Buch über schnelle Graphiken, Suchroutinen und Sortier Routinen im Level II BASIC (englischsprachig).

Nr. 0026: Advanced Level II BASIC

Don Imman, Albrecht, Zamora ---

Ein gutes Lehrbuch mit vielen Beispielen zu allen Themen der Computerei (englischsprachig).

Nr. 0027: Level II BASIC

Don Imman, Albrecht, Zamora ---

Ein gutes Lehrbuch für den Anfänger in BASIC (englischsprachig).

Nr. 0028: Programming In Style

Th. Dwyer, Margot Critchfield ---

Einführung in BASIC, strukturiertes Programmieren, Tips zur besseren Aufmachung der eigenen Programme, Tips für den Profi (englischsprachig)

Nr. 0029: The First Book Of 80-US

verschiedene --- Hofacker Verlag, Muenchen

Sammelband aus dem TRS-80 Users Journal.

Nr. 0030: The Rest Of 80

verschiedene --- Wayne Green Books, USA

Viele Beiträge aus der 80-Micro (Allgemeines, Graphik, Utilities).

Nr. 0031: Z-80-Applikationsbuch

Michael Klein --- Franzis Verlag, München

Eine Einführung in die Programmier- und Interfacetechniken des Mikroprozessors Z 80.

Nr. 0032: Z 80 Assembler Handbuch

--- Hofacker, München

Die Z 80 Befehle mit Kommentar und je einem Beispiel.

Nr. 0033: Programmierung des Z 80

Rodnar Zaks (Bernd Ploss) --- Sybex-Verlag, Duesseldorf

Als Lehr- und Lernmittel konzipiert ist dieses ins Deutsche bersetzte Buch ein umfassendes Nachschlagewerk und eine gruendliche Einfuehrung.

Nr. 0034: Daten - Disketten - Dateien

Hans-Joachim Sacht --- CHIP, Vogel-Verlag, Wuerzburg

Anregungen und Hinweise, wie Programme mit komfortabler Dateiverw. geschrieben werden koennen und was bei den versch. DOS zu beachten ist.

Nr. 0035: Anwenderprogramme fuer TRS-80 und Video Genie

M. Stuebs --- Hofacker, Muenchen

Viele (meist einfache) BASIC-Programme fuer den TRS-80.

Nr. 0036: Programmieren mit TRS-80

M. Stuebs --- Hofacker, Muenchen

Eine Einfuehrung in den TRS-80 und viele (meist einfache) BASIC-Programme

Nr. 0037: Die grosse BASIC Referenztabelle der 51 Dialekte

Wolf-Detlev Luther --- Luther-Verlag, Sprendlingen

Nr. 0038: Textsystem in BASIC und Z80-Assembler

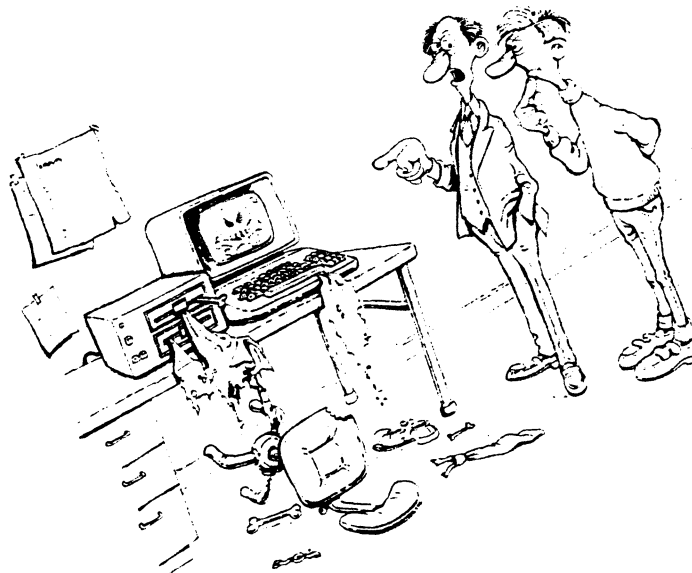
Gert Scheller --- Luther-Verlag, Sprendlingen

Das Buch beschreibt ein fertiges Textsystem fuer den TRS-80, das in BASIC und Z80 modular aufgebaut ist, und leicht erweiterbar ist.

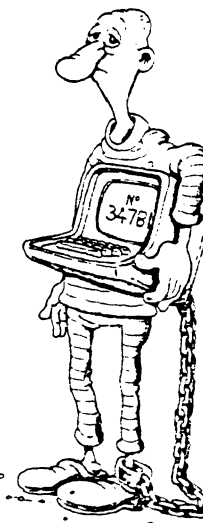
Nr. 0039: Textsystem in BASIC und Z80-Assembler - Band 2

Wolfraem Weber --- Luther-Verlag, Sprendlingen

Der Band 2 beschreibt Erweiterungen zum Programm aus Band 1.



Versuche ja nicht, dem da beim Schlüsselwort was vorzumachen...



Vorstand

Kontaktadresse für
Clubangelegenheiten
Clubbücherei / Fundgrube
Clubkasse

Günther WAGNER
Gartenstraße 4
8201 Neubeuern
Tel.: 08935 / 3361
< 18 - 20 Uhr >

Programmbibliothek

Kontaktadresse

Klaus-Jürgen Mühlenbein
Am Mönchgarten 28
6948 Weinheim
Tel.: 06201 / 55052

Redaktion

Kontaktadresse

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

Adventure-Ecke

Kontaktadresse

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

Hardware

Kontaktadresse

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

Redakteure

dieser Ausgabe

Ulrich Böckling * Gerald Dreyer
Klaus-J. Mühlenbein * Jens Neueder
Hartmut Obermann * Gerald Schröder
Arnulf Sopp * Günther Wagner
sowie Artikel aus: Computer Persönlich, 88 Micro,
Test und MC

Bankverbindung des CLUB 88

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.

Hallo Club-88er,

am Ende unseres 12. INFO darf ich die Gelegenheit wieder nutzen,
ein paar abschließende Worte zu finden:

Zuerst möchte ich nochmals auf die Programmkurzanleitungen zu
sprechen kommen. Es hat sich ja aus Eurer Sicht lange nichts dazu
getan. Aber es dauert halt doch etwas, bis das Material gesichtet
und sortiert ist. Weiterhin ist für überarbeitete oder andere
Merke -als in INFO 9 vorgestellt- in der Kurzanleitungsbibliothek
noch Platz. Ihr braucht sie mir nur zuzusenden.

Des weiteren findet im April (11.04.-13.04.86) unser 2.
Clubtreffen statt. Ich hoffe, daß Ihr alle die Möglichkeit
findet, daran teilzunehmen. Es ist doch meist sehr interessant,
sich einmal persönlich kennen zu lernen. Mit Sicherheit hat auch
der Eine oder Andere ein Diskussionsthema parat, über das man in
versammelter Runde einmal sprechen sollte. Weiterhin geht es ja
auch um die Wahl eines neuen Vorstandes und der Entlastung des
vorherigen. Auch die "Computerei" wird an diesem Wochenende
sicher nicht zu kurz kommen. Hier findet ...

.. der Anfänger geballtes Computerwissen, das er dann mit
seinen gezielten Fragen durcheinanderbringen darf -

.. der alte Hase einen anderen ganz ausgefuchsten Kollegen,
der zum gleichen Problem natürlich eine andere Lösung aus seiner
Trickkiste zieht.

Ihr seht also, es lohnt sich sicher für jeden von Euch, an
dem Clubtreffen teilzunehmen. Auf alle Fälle liegt jedem Clubinfo
ein persönliches Kärtchen bei, das Euch als Teilnehmer des
Clubtreffens ausweist und gleichzeitig als Namenskärtchen dienen
soll.

Zum INFO-Abschluß dieses 12. INFO's wünsche ich Euch allen
"Happy Hacking", viel Freude an der neuesten INFO und ein
gelungenes Clubtreffen.

Bis zum nächsten INFO -
oder noch früher in Holzhausen a. d. Heide verbleibe ich

Euer

J. Neueder

CLUB 80 Mitgliederadressenliste

Name	Vorname	Straße	Plz	Stadt	Telefon
Alber	Herbert	Alemannenstr. 20	7732	Niedereschach	07721 /7102
Albers	Herbert	Zum Duwelshopen 14	2117	Westedt	
Beckhausen	Wolfgang	Vuerfelser-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	Buchentalweg 8	4939	Steinheim	05233 /4320
Fröhlich	Burghard	Brandströstraße 11	4350	Recklinghausen	02361 /63416
Grajewski	Werner	Zedernweg 29	4220	Dinslaken	02134 /54573
Held	Manfred	Stirnerstr. 22	8835	Pleinfeld	09144 /6563
Hermann	Klaus	Gartenstr. 22	7401	Pliezhausen	07127 /70024
Hill	Peter	Eckstraße 36	6750	Kaiserslautern 31	0631 /54782
Hummel	Anton	Schubertstr. 2	7612	Haslach	07832 /8289
Jablotschkin	Rainer	Thiekamp 29	4780	Lippstadt 8	
Kasper	Dieter	Zeppelinstr. 9	8952	Marktobendorf	08342 /1630
Koenig	Hans J.	Hebbelstr. 25	2000	Pinneberg	04101 /209444
Konrad	Josef	Anzengruberstraße 35	8038	Gröbenzell	08142 /8494
Kuhn	Eckehard	Im Dorf 14	7443	Frickenhausen 1	07022 /45417
May	Holger	Marienstr. 9	5768	Sundern 2	02935 /1668
Misioch	Waldemar	Adenauerring 25	8505	Röthenbach	0911 /506051
Mühlenbein	Klaus-Jürgen	Am Mönchgarten 28	6940	Weinheim -Lü.	06201 /55052
Müller	Kurt	Soltaustraße 24a	2050	Hamburg 80	040 /7246083
Neueder	Jens	Panoramastr. 21	7178	Michelbach/Bilz	0791 /42077 (dienstl.44-667)
Obermann	Hartmut	Schwalbacher Str. 6	6209	Heidenrod 1	06124 /3913
Perschbach	Patrick	Waldstr. 52	5000	Koeln 91	0221 /872118
Piller	Walter	Rohnenstraße 8	CH-8835	Feusisberg	01 /7847418
Preuss	Lothar	Lautshof 13	2940	Wilhelmshaven	04421 /84247 (dienstl. 804-1)
Raggan	Hans	Backnanger Weg 36	7146	Tamm	07141 /603611
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	0231 /593883
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
Volz	Oliver	Dusestraße 13	7000	Stuttgart 80	0711 /731285
Wagner	Alexander	Theresienstr. 21c	8224	Chieming	08664 /1500
Wagner	Guenther	Gartenstr. 4	8201	Neubeuern	08035 /3361
Weiß	Dieter	Bürglestraße 3	7209	Wehingen	07426 /7194
Wucherer	Jürgen	Brauneggerstraße 14	7750	Konstanz	07531 /29145
Zwickel	Walter	Lengfelden 123	A-5101	Bergheim	0043662/51130

Neuzugänge :

Waldemar Misioch
 Peter Hill
 Kurt Müller

Stand: März 1986

Bitte überprüft Eure Anschrift und meldet eventuelle Adressenänderungen an die Redaktion.