

# CLUB 80

CLUBINFO  
DER

TANDY E  
GENIE E  
UND KOMTEK E  
ANWENDER E

8. AUSGABE

ICH WEISS, ICH WEISS... ANDERE  
KATZEN SITZEN AUCH VON MORGENS  
BIS ABENDS VOR MAUSELÖCHERN....



...TROTZDEM — IRGENDWIE MACH  
ICH MIR SORGEN....



Kontaktadresse : CLUB 80 / Günther WÄGNER / Gartenstraße 4 / 8201 Neubuam

Tel.: 08935/3361 (18 - 20 Uhr)

# Inhaltsverzeichnis

Seite:

## Clubinternes

Der Vorstand informiert . . . . .	01
Diskussion . . . . .	21 - 22

## Software

Das Abtipplisting . . . . .	02
LPRINT alles einfacher . . . . .	03 - 04
COBOL eine Einführung . . . . .	05 - 11
Tips und Tricks . . . . .	12 - 14
BASIC - Dolmetscher gesucht . . . . .	15 - 18
Tongenerator . . . . .	19 - 20

## Hardware

Shift - Trick . . . . .	23 - 26
ECB - Bus System . . . . .	24
Taktumschaltung für VIDEO-GENIE . . . . .	25 - 26
Super-Tape (nochmals) . . . . .	27
Tests . . . . .	28

Seite:

## Börse

Wer hat was ???--Wer sucht was ??? . . . . .	36
Fragen, Fragen, Fragekasten . . . . .	36

## Sonstiges

Kurzgeschichten . . . . .	37
---------------------------	----

## Programmbibliothek

Neue Programme . . . . .	29 - 34
Exklusiv . . . . .	35

## Die letzten Seiten

Clubmitgliederadressen . . . . .	38
Impressum . . . . .	39
Schluß . . . . .	40
Neue CLUBSATZUNG . . . . .	am INFO-Ende

Hallo liebe Club-Freunde,

dieses Mal darf ich mich kurz fassen. Ich bin gerade vom Urlaub zurück gekommen und noch nicht so ganz drin im Geschäft.

Zunächst das wichtigste - das Ergebnis unserer Abstimmung:  
\*\*\*\*\*

Abgegeben wurden bis zum 01.07.85 insgesamt 25 Abstimmzettel.

Punkt 1: Annahme der neuen Clubsatzung  
23 Ja-Stimmen  
2 Nein-Stimmen  
Hiermit ist die neue Clubsatzung gültig.

Punkt 2: Nachzahlung von 10 DM für das laufende Jahr  
23 Ja-Stimmen  
2 Nein-Stimmen  
Hiermit muß Jedes Mitglied 10 DM nachzahlen.

Punkt 3: Erhöhung des Jahresbeitrages auf 50 DM  
24 Ja-Stimmen  
1 Nein-Stimme  
Hiermit beträgt der Jahresbeitrag 50 DM.

Ich bin froh, daß sich die Mitglieder zu der neuen Clubsatzung so eindeutig bekannt haben. Verständlich ist, daß insbesondere die Beitragserhöhung und die Nachzahlung unsere Schüler und Studenten besonders trifft; daher überrascht es mich, das hier doch nur 1 bzw. 2 Nein-Stimmen abgegeben wurden.

Wir sind also jetzt vom finanziellen her in der Lage, weiterhin gute Clubarbeit zu leisten. Es kann weiterhin dicke Clubinfos geben. Aber dicke Clubinfos gibt es nur bei entsprechender Mitarbeit der Mitglieder - und dieses Mal hat der Jens nicht gerade viel erhalten. Hoffentlich schleicht sich hier jetzt keine Trägheit ein. Greift zur Feder, hackt in die Tasten - der Jens braucht Euer Beiträge um ein gutes Info rausschicken zu können. Und vergeßt den Hartmut nicht mit der Programm-Bibliothek.

Noch kurz zur Nachzahlung der 10 DM. Von vielen Mitgliedern habe ich diese Ja schon beim Clubtreffen erhalten. Viele haben auch auf das 7. Info hin gleich die 10 DM bezahlt. Alle anderen Mitglieder bezahlen bitte die 10 DM so bald wie möglich. Ich würde mich freuen, wenn bis Ende Juli alle Mitglieder die 10 DM bezahlt hätten.

So - das soll's dieses Mal von meiner Seite gewesen sein.  
Bis zum nächsten Mal

Euer

*Günther Wagner*

## Das Labyrinth und seine Maus

- das Programmlisting zum Abtippen -

```
1 CLS
2 ONERRORGOTO26
3 FORI=0TO122:SET(I,0):SET(I,45):NEXTI
4 FORI=0TO45:SET(0,I):SET(123,I):NEXTI
5 FORX1=0TO122STEP3:IFPOINT(X1,45)THENX=X1:Y=45:GOSUB11
6 NEXTX1
7 FORY1=0TO45STEP3:FORX1=0TO122STEP3
8 IFPOINT(X1,Y1)THENX=X1:Y=Y1:GOSUB11
9 NEXTX1:NEXTY1
10 GOTO90:'ab 100 startet ihr Maus-Programm!!!
11 ' Ab gesetztem Punkt, Wand ziehen
12 Z=1
13 R=RND(4)
14 Z=Z+1:IFZ=10THENZ=1:RETURN
15 IFR=1THENIFPOINT(X-3,Y)=-1THENGOTO13
16 IFR=2THENIFPOINT(X+3,Y)=-1THENGOTO13
17 IFR=3THENIFPOINT(X,Y-3)=-1THENGOTO13
18 IFR=4THENIFPOINT(X,Y+3)=-1THENGOTO13
19 FORI=1TO3
20 IFR=1THENX=X-1
21 IFR=2THENX=X+1
22 IFR=3THENY=Y-1
23 IFR=4THENY=Y+1
24 SET(X,Y)
25 NEXTI:Q=RND(3):IFQ=1THENGOTO15ELSEGOTO13
26 RESUME13
90 RESET(0,2):RESET(123,43)
100 CLEAR:DEFINTA,B:A=1:B=2:POKE16406,82:SET(0,2):SET(A,B):RESET
(0,2):IFPOINT(A,B+1)103
101 FORB=B+1TO47:RESET(A,B-1):SET(A,B):IFB=43IFPOINT(A-1,B)RESET
(A,B):B=B+1:GOTO103ELSE120ELSEIFNOTPOINT(A-1,B)107ELSEIFPOINT(A,
B+1)103ELSENEXT
103 FORA=A+1TO127:RESET(A-1,B):SET(A,B):IFA=121THEN115ELSEIFNOTP
OINT(A,B+1)IFB=41THEN118ELSE101ELSEIFPOINT(A+1,B)105ELSENEXT
105 FORB=B+1TO6STEP-1:RESET(A,B+1):SET(A,B):IFNOTPOINT(A+1,B)103
ELSEIFPOINT(A,B-1)107ELSENEXT
107 FORA=A+1TO6STEP-1:RESET(A+1,B):SET(A,B):IFNOTPOINT(A,B-1)125
ELSEIFPOINT(A-1,B)101ELSENEXT
115 RESET(A,B):A=A+1:SET(A,B):IFA=123POKE16406,227:END
116 IFB=44RESET(A,B):B=B-1:SET(A,B):GOTO115ELSEFORB=B+1TO43:IFP
OINT(A+1,B)RESET(A,B-1):SET(A,B):NEXTELSERESET(A,B-1):SET(A,B)80
TO115
118 IFPOINT(A+2,43)=0THENRESET(A,B):B=B+3:SET(A,B):GOTO103ELSERE
SET(A,B):A=A+12IFPOINT(A+1,B)=0THEN103ELSE105
120 IFPOINT(A-2,B-1)=0THEN107ELSEIFPOINT(A+2,44)RESET(A,B):A=A+1
:GOTO105ELSERESET(A,B):B=B+1:GOTO103
125 IFPOINT(A,B-4)IFPOINT(A-1,B-2)IFNOTPOINT(A-2,B)RESET(A,B):A=
A-1:GOTO107ELSEIFB=43RESET(A,B):B=44:GOTO103ELSEIFPOINT(A-2,B)RE
SET(A,B):A=A-1:GOTO101
126 GOTO105
```

Unser Assembler-Freak Arnulf Sopp hat sich im Clubinfo Nr. 6 ausgiebig über das Thema "Ausgabe der Steuercodes 0, 10 und 12 an den Drucker" ausgelassen. Mit seinem in diesem Beitrag abgedruckten Assemblerlisting versucht er dem Microsoft-BASIC beizubringen, alle Steuerzeichen ungefiltert zum Drucker durchzulassen. Leider verbiegt er dabei den Interpreter jedoch so, daß danach kein (normales) LIST mehr möglich ist. Hier nun eine einfache, leicht durchschaubare und realisierbare Möglichkeit den gleichen Effekt zu erreichen und dabei die LIST-Funktion völlig unberührt lassen.

Die Stelle, an der das unten stehende Programm in den BASIC-Interpreter eingreift, ist der sogenannte DCB (DeviceControlBlock). Dieser beinhaltet, neben der Angabe des DCB-Typs, die Treiberadresse und weitere Informationen. Der Drucker-DCB nimmt die Speicherstellen 4025h - 402Bh (16421d - 16427d) ein und sieht folgendermaßen aus:

```
4025 06          DCB-Typ
4026 BD 05      Treiberadresse
402B 43          Zeilen pro Seite
4029 ??        Zeilenzähler
402A 00          unbenutzt
402B 50 52      PR (Hinweis: Druckertreiber!)
```

Da der DCB, wie ein kurzer Blick in die Memorymap zeigt, im RAM-Bereich liegt, kann er sehr leicht geändert werden. So kann man z.B. die Adressen 4026h und 4027h so verändern, daß sie auf einen selbstgeschriebenen Druckertreiber zeigen. Dieser übernimmt es dann, die Zeichen zum Drucker zu senden. Der neue Druckertreiber ist bis auf die Grundfunktionen abgemagert.

```
100 LD    A,C      ;Zeichen..
110 PUSH AF        ; ....Retten
120 CALL 05D1      ;Druckerstatus abfragen
130 JR   NZ,#-3    ;wenn nicht READY, weiter abfragen
140 POP  AF        ;Zeichen zurückholen
150 LD    (37E8),A ;Zeichen an Drucker ausgeben
160 RET          ;zurück zum aufrufenden Programm
                ;(hier zum BASIC-Interpreter)
```

Ihn gilt es jetzt in den Speicher zu bringen. Dazu benutzt man am besten ein BASIC-Programm wie es hier abgedruckt ist. Der Treiber wird auf dem Platz einer Dummyvariablen gepackt und der DCB so geändert, daß er auf die entsprechende RAM-Adresse zeigt. Die REM's im Programm sollten zur Erklärung ausreichen.

```
10 LN$="ABCDEFGHJKLM": ' Freihalten von 12 Bytes durch LN$
20 V=VARPTR(LN$):ADDR=PEEK(V+1)+PEEK(V+2)*256: ' Ermitteln der Adresse von
LN$ und abspeichern in ADDR
30 FORI=0TO11:READ C:POKE ADDR+I,C:NEXT I: ' Programm in LN$ packen
40 DATA 121,245,205,209,5,32,251,241,50,232,55,201: ' Druckertreiber als
dezimale Opcodes
50 POKE 16422,PEEK(V+1): POKE 16423,PEEK(V+2): ' Ändern des DCB
55 ' Die Zeilen 60 und 70 dienen nur zur Demonstration!!!
60 LPRINTCHR$(27)"K"CHR$(0)CHR$(1);
70 FORI=0TO255:LPRINTCHR$(I);NEXT
80 POKE 16422,141:POKE 16423,5: ' Zurücksetzen des DCB auf die "alte"
Treiberadresse
```

Die letzte Zeile sollte man übrigens nicht vergessen, da ansonsten nach dem Löschen des Programms (NEW) der DCB auf eine Adresse zeigt, an der alles möglich steht, nur kein Druckertreiber und damit der Absturz unvermeidbar ist.

Zuletzt möchte ich noch an mein RESTORE n - Patch erinnern, daß im Info Nr. 5 erschienen ist. Natürlich könnte man auch den neuen Druckertreiber auf die dort beschriebene Art ins BASIC einpatchen. Wer dazu Lust hat, kanns ja mal versuchen.

Kurtmann Overmann

Quellenangaben:  
ROM-Listing von L. Röckrath  
"Alle Steuercodes verfügbar"  
von R. Metzger (ac 12/63)

"Nach dem Hochmut kommt der Fall" besagt ein altes Sprichwort und wie recht der Volksmund mit dieser Behauptung hat, möchte ich euch heute kurz schildern.

Sicher habe ihr sowohl den Artikel "LPRINT alles" von Arnulf (Info Nr. 6) als auch meinen Beitrag "LPRINT alles einfacher (Teil 1)" gelesen. Wie euch vielleicht aufgefallen ist, war ich Assembler-Greenhorn ziemlich stolz darauf, einem so ausgefuchsten Assembler-Freak wie dem Arnulf mal zeigen zu können, daß man nicht immer um 10 Ecken herum im Interpreter herummurgeln muß, um ein relativ einfaches Ziel zu erreichen. Dies habe ich ihm bei einem Treffen des Bremerhavener Computerclubs natürlich auch gleich auf die Nase gebunden. Ich erklärte dem Arnulf meine Lösung und erntete erst ungläubiges Erstaunen und sodann den Beweis, daß diese Lösung nur eine Teillösung des Problems darstellt. Ich war baff!

Der BASIC-Interpreter versaubert das Zeichen 0Ah (10d) doch tatsächlich schon, bevor es überhaupt zum Druckertreiber kommt! Das hat zur Folge, daß mit meiner Lösung alle Zeichen außer 10d bzw 0Ah ausgegeben werden können. Jetzt hieß es beschämt zum Rückzug anzutreten und mich beim Arnulf entschuldigen, was ich hiermit nochmal offiziell tue!

Der Arnulf war aber noch so nett, mir noch einen Lösungstip mitzugeben. Er wies mich darauf hin, daß der BASIC-Interpreter die Zeichen (0,10 und 12) nicht einfach nur unwandelt, sondern sie zuvor auf dem Stack (Stapelspeicher) ablegt und man sie sich von dort wieder unbeschadet herausholen könnte. Vielen Dank Arnulf!!! Auf dieser Aussage habe ich eine (beweisbar) voll funktionstfähige Lösung entwickelt, die ich euch kurz schildern möchte.

Auch bei dieser Lösung wird der DCB verbogen. Und zwar auf folgendes kurze Assemblerprogramm:

```
100 LD    (04018h),SP ;Wert des Stackpointers zwischenspeichern
200 LD    IX,(04018h) ;Wert des Stackpointers in das IX-Register laden
300 LD    A,(IX+11h)  ;zu druckendes Zeichen in den Accu laden
400 PUSH AF          ;Zeichen retten
500 CALL 05D1h       ;Druckerstatus abfragen
600 JR   NZ,#-3     ;wenn Printer nicht ready, weiter abfragen
700 POP  AF          ;Zeichen zurückholen
800 LD    (037E8h),A ;Zeichen an Drucker ausgeben
900 RET          ;zurück zum aufrufenden Programm
```

In den Zeilen 100 und 200 wird als erstes festgestellt, auf welche Speicheradresse der Stackpointer zur Zeit zeigt. In Zeile 300 wird der Inhalt der Speicherstelle, in der die zu druckenden Zeichen zwischengelagert werden, in den Accu geladen. Die Zeilen 400 - 900 sind praktisch die gleichen wie im vorhergehenden Programm.

Auch dieses kleine Maschinenprogramm bringt man am besten in der schon bekannten Weise in einem BASIC-Programm unter (oder patcht es direkt in den Disk-BASIC-Interpreter). Hier das kurze BASIC-Programm mit eingebautem Demo.

```
10 LN"1234567890123456789212": V=VARPTR(LN$)
20 ADDR=PEEK(V+1)+PEEK(V+2)*256
30 FORI=0TO21: READ C: POKE ADDR+I,C: NEXT I
40 DATA 237,115,24,64,221,42,24,64,221,126,17,245,205,209,05,32,
251,241,50,232,55,201
50 POKE 16422,PEEK(V+1): POKE 16423,PEEK(V+2)
60 LPRINTCHR$(27)"K"CHR$(0)CHR$(1);
70 FORI=0TO255:LPRINTCHR$(I);NEXT
80 POKE 16422,141: POKE 16423,5
```

Wie ihr hoffentlich alle seht, ist es, mit freundlicher Hilfe von Clubkameraden, selbst einem Assembler-Greenhorn möglich Probleme zu lösen und damit zu Erfolgserlebnissen zu kommen (wenn auch spät und über Umwege). Ich hoffe, daß mancher von euch dieses Beispiel (den Hochmut könnt ihr ja weglassen) folgen wird und so ein bißchen Freude an der Assembler-programmierung findet. Viel Spaß dabei, Euer Kurtmann Overmann

Wie im letzten Teil kurz erwaeht, folgt hier nun die dritte Division, die DATA DIVISION.

Hier werden allen Variablen Datenfelder zugewiesen, der Aufbau wird festgelegt und weiterhin der Typ (alpha oder numerisch) und schliesslich eine eventuelle Druck-Aufbereitung.

Was das nun alles im einzelnen bedeutet, wird im folgenden erlaeutert.

```
1-----67A---B----- usw... 72
DATA DIVISION.
*****
*
FILE SECTION.
*
FD DATUM
  BLOCK CONTAINS 1 RECORD
  RECORD CONTAINS 10 CHARACTERS
  LABEL RECORD IS STANDARD
  DATA RECORD IS DATUM-SATZ.
01 DATUM-SATZ.
  02 TAG          PIC X(10).
```

Was bedeutet das nun ?? - An dieser Stelle verweise ich auf den Teil 2 der Cobol-Einfuehrung. Dort wird in der ENVIRONMENT DIVISION innerhalb der ersten SELECT-Anweisung der physischen Datei 'HPTDATUM/DAT:1' der logische Dateiname 'DATUM' zugeordnet. Weiterhin wird festgelegt, dass es sich um eine Relative-Datei mit direktem Zugriff ueber die Satznummer (Variable:SATZ) handelt.

Ueber den Aufbau des Datensatzes wurde alleerdings nichts gesagt. Dies geschieht nun in der DATA-DIVISION (s.o).

Der vermeintlich grosse Aufwand fuer die Datei und Datenbeschreibung tauscht. Bei umfangreichen Dateien mit Saetzen bis zu einem halben Kb macht sich das Verfahren positiv bemerkbar.

Aber zurueck zur DATA DIVISION. Unter der FILE SECTION werden zuerst alle Dateien mit 'ihren' Datensatzen versorgt.

FD (File Description) N a m e	(Zuordnung Datensatz-Datei)
BLOCK CONTAINS	(Blockungs-Faktor)
RECORD CONTAINS xx CHARACTERS	(Anzahl Bytes per Record)
LABEL RECORD IS	(bei Dateien: STANDARD beim Drucker OMITTED)
DATA RECORD IS	(Name des Datensatzes)
01 DATUM-SATZ	(Beschreibung Datensatz)
02 TAG          PIC X(10)	(Datenformat)

Gehen wir zunaechst noch auf eine zweite Datei-Beschreibung ein: auf die 'STAMMDAT/DAT:1', die in der SELECT Anweisung den logischen Satz 'DATEN' zugeordnet bekommen hat.

Bei dieser Datei handelt es sich um eine Stammdaten-Datei fuer eine Adressverwaltung, dementsprechend gross ist der Datensatz.

```
FD DATEN
  BLOCK CONTAINS 10 RECDRDS      (nur Dokumentation)
  RECORD CONTAINS 274 CHARACTERS (Satzlaenge)
  LABEL RECORD IS STANDARD      (Label f. Dat-Header)
  DATA RECORD IS STAMM-SATZ.   (Name Datensatz)
01 STAMM-SATZ.
02 E-NUMMER          PIC X(8).
02 E-NAME           PIC X(30).
02 E-VDRN           PIC X(20).
02 E-STR            PIC X(25).
02 E-PLZ            PIC X(6).
02 E-ORT            PIC X(25).
02 E-TEL            PIC X(15).
02 E-GEB            PIC X(10).
02 E-TXT1           PIC X(45).
02 E-TXT2           PIC X(45).
02 E-TXT3           PIC X(45).
```

So, das waere die FD-Anweisung fuer die Stammdaten-Datei. Allen Variablen sind Datentyp und Datenlaenge innerhalb des Datensatzes zugeordnet worden. Fragen sind aber sicherlich noch einige offen: Was bedeutet die 'komische' Durchnummerierung der einzelnen Variablen ? oder: Was hat es mit dem PIC und dem X dahinter auf sich ??

Ein klein wenig Geduld noch, die Erklarung folgt in der WORKING-STORAGE SECTION.

Halten wir also noch einmal fest: Die DATA DIVISION ordnet allen Variablen, egal ob reine Arbeitsvariablen, wie Merker, Zaehler usw., oder Variablen die zu einer Datei gehoeren, den Datentyp (alpha, numerisch, alphanumerisch) und die Laenge zu. Die Variablen-Namen koennen dabei bis zu 32 Stellen lang sein. Unterschieden werden die Arbeitsvariablen und Datensatz-Variablen bei der Zuordnung der SECTION's.

Arbeits-Variablen 'stehen unter' der WORKING-STORAGE SECTION; Variablen zur Datei stehen unter der FILE SECTION, jeweils getrennt unter den einzelnen FD-Anweisungen.

(Der Vollstaendigkeit halber sei noch erwaeht, dass innerhalb der FILE SECTION gleiche Variablen-Namen verwendet werden duerfen. So kann es unter der FD DATUM z.B. einen TAG geben, ebenfalls unter der FD DATEN. Beide Variablen koennen unterschiedliche Informationen enthalten. Die Unterscheidung findet innerhalb der Programm-Anweisung statt.)

7

Kommen wir nun zur WORKING-STORAGE SECTION. Hier werden noch einige Dinge auftauchen, die zunächst Verwirrung auslösen können - aber trotz allem: immer daran denken, dass, egal wie es aussieht, lediglich Variablen-Namen, Länge und Datentyp zugeordnet werden - sonst nichts.

Ein Hinweis noch: alle hier angeführten Beispiele stammen aus einem Stammdaten-Verwaltungs Paket, genannt 'ADRESSEN TEXT', von mir verfasst und geschrieben. Eine Programm-Beschreibung und das Source Listing können von mir angefordert werden. Die Quellen sind als Listing oder als ASCII-Files auf Diskette zu erhalten. Mir ist es lieber, die Quellen auf Disk zu verschicken, ca. 60 Kb, der Ausdruck dauert nämlich ewig. Die Programmbeschreibung gibt es als Kopien. (Bei Source-Anforderung: bitte Disk mitschicken !)

So, nun aber zum Kern zurück:

```
1-----67A-----B-----usw... 72
      WORKING-STORAGE SECTION.
      77 SATZ      PIC 9 VALUE 1.
      77 WAHL     PIC 9 VALUE 0.
      77 MERKER   PIC 9 VALUE 0.
      77 ZEILE    PIC 99.
      77 DP-WAHL  PIC 9 VALUE 0.
      77 C-1      PIC X.
      77 C-2      PIC X.
      77 E-DAT    PIC X(10).
      77 STRICH   PIC X(63).
      77 LEERZEILE PIC X(63).
```

Es wurden also 10 Variablen definiert, die im Programm als reine Arbeitsfelder Verwendung finden.

Nun aber die Erklärungen:

PIC heisst eigentlich PICTURE (Bild) und bedeutet, dass in die jeweilige Variable ein bestimmtes 'Bild' gelegt wird, aus dem sich der Variablentyp, die Länge und der Aufbau der Variablen ersehen lassen.

PIC X: eine Variable, der PIC X zugeordnet wird, hat eine Länge von einem Byte (ein X=ein Byte).

PIC XX: Variable hat eine Länge von zwei Byte.

PIC XXX: drei Byte

PIC XXXX: vier Byte

PIC XXXXX: fünf Byte, aber jetzt wird es langsam unübersichtlich

Man hilft sich mit folgendem Trick:

PIC X(5):fünf Byte

PIC X(9):neun Byte

PIC X(1):ein Byte (geht natürlich auch)

8

Das X hinter der PIC Anweisung sagt aber auch etwas über den Datentyp aus: X bedeutet alphanumerisch. In diesem Zusammenhang bedeutet 9 hinter der PIC Anweisung: numerisch und A hinter der PIC Anweisung alphabetisch.

Also wird mit z.B.

```
77 TRALLA      PIC X(10)
```

die Variable 'TRALLA' mit 10 Stellen, alphanumerisch definiert.

So weit, so gut: Nun kommen die Zahlen vor den Variablen, mal ist sie 01, mal 02 und dann 77. Ganz einfach, die Variablen haben eine gewisse Priorität, einfach ausgedrückt: Variablen mit kleiner Nummer (01) haben hohe und Variablen mit niedriger Nummer (z.B. 77) niedrige Priorität.

Aber das geht jetzt nicht stur von 01 bis 99 oder so:

Über 50 gibt es nur noch die Stufen: 66, 77, 88. Die 66-er und 88-er lassen wir an dieser Stelle weg - bleiben also nur noch die 77-er. (66, 88 dienen z.B. für LINKAGE und andere feine Sachen, mit denen sich der erfahrene Programmierer auseinandersetzen kann)

Die Nummern vor den Variablen heissen auch nicht 'Nummern', sondern Stufen - also, die Stufe 77 kennzeichnet Arbeitsfelder, allgemein.

Was heisst nun allgemeine Arbeitsfelder ?? Das sind Variablen, von denen jeweils eine Var eine eigene unabhängige Variable darstellt, z.B. Zähler.

Nun gibt es aber auch z.B. folgendes:

```
10 CHEF-VARIABLE.
  20 ABTEILUNGSLEITER-1      PIC X(10).
  20 ABTEILUNGSLEITER-2      PIC X(10).
  30 ANGESTELLTER-1          PIC X(10).
```

Wie schon an den (willkürlich) gewählten Variablen-Namen kann man, wie auch an den Stufen, eine Abhängigkeit und Rangordnung feststellen. Man kann nun innerhalb des Programms jede Variable einzeln ansprechen und Daten zuordnen.

Also z.B.

```
MOVE "HEINZ" TO ANGESTELLTER-1.
MOVE "HERMANN" TO ABTEILUNGSLEITER-1.
```

Mit der übergeordneten Variablen (CHEF-VARIABLE) kann man sich die ganze Angelegenheit aber sehr stark vereinfachen:

```
1-----1-----2-----3
HEINZ  01-----01-----0      (Mit MOVE aufbereiteter
HERMANN  HERMANN  GUENTHER      Datensatz)
```

Wenn ich nun den Befehl gebe: MOVE Datensatz TO CHEFVARIABLE, so habe ich mit einem einzigen Befehl drei Namen zu drei Variablen zugeordnet.

Und genau nach dem gleichen Verfahren wird auch bei den niedrigeren Stufen verfahren.

Wenn ich die Variable mit der Stufe 01 anspreche, so kann ich alle ihr untergeordneten Variablen mit einem Befehl mit Daten fuellen oder aber mehrere Variablen mit einem Befehl von einem Datenbereich in einen anderen MOVEn.

Die Stufen von 01 bis 49 sind also dazu gedacht, um Datenfelder, die voneinander abhaengig sind, mit einem einzigen Befehl bequem handhaben zu koennen. Wird die Stufe 01 angesprochen, so kann man ueber die Stufen 02 ... 49 verfuegen; entsprechendes gilt natuelich auch wenn z.B. die Stufe 10 angesprochen wird. (Verfuegung ueber 11 ... 49).

Siehe dazu auch im Beispiel:  
 01 STAMM-SATZ.  
 02 E-NUMMER PIC X(8).  
 usw.

Das Einruecken der Stufen geschieht willkuehrlich und dient nur zur besseren Dokumentation. Man kann auch alle Stufen an der Position 8 'beginnen' lassen.

Zurueck zur PIC-Anweisung:  
 Der Aufbau duerfte nun klar sein. Also noch einmal: Das X steht fuer eine alphanumerische Variable, das A fuer eine reine alphabetische Variable und schliesslich die 9 fuer eine numerische Variable. Die Anzahl der Xen, der As oder der 9en schliesslich gibt die Laenge der Variable an.

Es gibt aber gerade bei den numerische Variablen noch viele Besonderheiten, von denen hier aber nur die wichtigsten angesprochen werden sollen:

Ganz allgemein: es gibt nur Integer-Variablen ! (Das stimmt natuerlich nicht, aber es dient zur besseren Darstellung).

Zahl	Feld-Aufbereitung	Feld-Inhalt
10	PIC 9(2)	!1!0!
1234	PIC 9(4)	!1!2!3!4!
1234	PIC 9(6)	! ! !1!2!3!4!
12.34	PIC 99V99	!1!2!3!4!
-2.34	PIC S9V99	!2!3!4!
.12.34	PIC 99.99	!1!2!.!3!4!
12.34	PIC 999.99	!0!1!2!.!3!4!
12.34	PIC Z9.99	! !1!2!.!3!4!
-3.2	PIC --9.99	! !-!3!.!2!0!

Und nun zur Erklarung:

Mit Feld-Inhalt ist die Darstellung der Variablen-Inhalte im Ram gemeint. Die ersten beiden Beispiele sind klar. Das dritte Beispiel zeigt die Ausrichtung der Daten im Feld, wenn die Daten der Variable kleiner sind als das entsprechende Datenfeld.

Beispiel vier und fuenf sind etwas neues. Es handelt sich hier um Datenfelder, die fuer Dezimalzahlen aufbereitet sind - Das 'V' gibt dabei die Position des Kommas an, das Komma selbst wird in dem Datenfeld nicht angezeigt.

Das Beispiel fuenf unterscheidet sich von dem vierten Beispiel dahingehend, dass statt der ersten '9' ein 'S' bei der Aufbereitung verwendet wurde. Dieses 'S' bedeutet nichts anderes als 'Sign', also Vorzeichen. Man sollte Rechenfelder immer mit diesem S ausstatten - nur dann besteht die Moeglichkeit auch mit negativen Zahlen zu rechnen.

Die nun folgenden vier Beispiele unterscheiden sich von dem bisher gesagten ganz erheblich:

Waren bisher alle behandelten Darstellungsformen nur zum Rechnen gedacht, so muss nun natuerlich noch die Druckaufbereitung behandelt werden.

Das ist nun etwas ganz neues fuer Basic-Anwender. Die Datenfelder zum Rechnen und Darstellen (egal ob auf dem Bildschirm oder Drucker) sind nicht identisch mit den Datenfeldern zum Rechnen. Das muetet vielleicht etwas seltsam an - ist jedtch eine ganz sinnvolle Einrichtung da man bei Cobol viele Moeglichkeiten zum Ablegen der numerischen Daten im Ram hat (z.B. gepackt).

Wenn alle Moeglichkeiten hier dargestellt werden sollten, so liesse sich zu diesem Thema ein eigenstaendiger Kurs schreiben.

Nun aber zu den Druckaufbereitungen:

Bei PIC 99.99 wird eine Dezimalzahl in der gewoehnten Weise angezeigt (man kann aber nicht mit ihr Rechnen). Damit die Zahl angezeigt werden kann, muss sie zunaechst erst einmal von dem Rechnfeld ins Druck-Feld geMOVED werden.

MOVE RE-Feld TO DRUCK-Feld.

Wenn das RE-Feld Beispiel 2 ist, dann steht im Druck-Feld: 12.34

Ist das RE-Feld Beispiel 5, so steht im Druck-Feld: 02.34

Aber wo ist das negative Vorzeichen und wieso steht vor der 2 eine 0 ??

Die Erklarung ist einfach:  
Fuer diese Darstellung ist die Druck-Aufbereitung nicht geeignet, man muss eine andere waehlen.

Zur Vor-Null-Unterdrueckung gibt es das 'Z' (Zero suppress) Wuerde man nun das Beispiel Nr. 5 in das Beispiel Nr. 8 (PIC Z79.99) moven, so wuerde die Vor-Null unterdrueckt werden. das Vorzeichen allerdings ist immer noch nicht beruecksichtigt.

Moven wir also das Rechenfeld (PIC S9V99) mit dem Inhalt -234 (es ist dazimal aufbereitet, (V)), in das Beispiel Nr. 10 (PIC --9.99).

Bei dieser Druckaufbereitung wuerde auf dem Bildschirm d'yer Inhalt der Variablen so angezeigt werden, wie wir es gewohnt sind.

Diese Art der Programmierung ist nur halb so umstaendlich wie sie jetzt erscheint. Wenn man alle Moeglichkeiten ausnutzt, hat man ein Vielfaches der vom Basic her gewohnten Darstellungsformen (PRINT USING ist dagegen ueberhaupt nichts).

Jetzt fehlen eigentlich nur noch die Befehle fuer den Programm-Ablauf (in der PROCEDURE DIVISION).

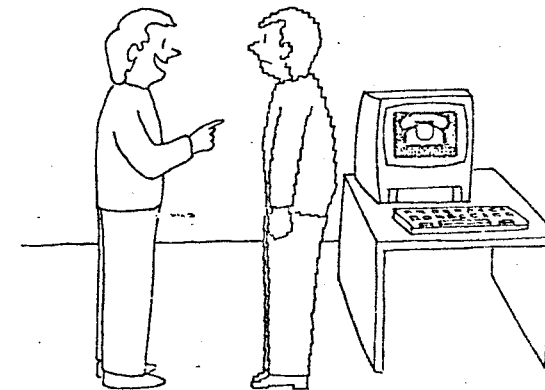
Man wird staunen, mit wie wenigen Befehlen man ein komfortables Programm schreiben kann. (Dafuer musste man ja schon einiges an Arbeit in der DATA DIVISION auf sich nehmen.

Die Eingabesteuerung und Ausgabemoeglichkeiten sind derart zahlreich, dass man angenehm ueberrascht sein wird (z.B. automatisches Carrige Return bei Erreichen des Endes bei Eingabefeldern oder automatisches Prompt in der Laenge des Eingabefeldes (das ist der Unterstrich, den man bei Eingabemasken haeufig findet, z.B. NAME: \_\_\_\_\_), korrigiert man etwas bei seiner Eingabe (Pfeil links), so werden die Promts, also der Unterstrich, wider automatisch gesetzt usw.)

Aber man muss es erst einmal ausprobieren....

Noch ein Hinweis in eigener Sache:  
Mich interessiert natuerlich brennend, ob mein kleiner Kurs verstaendlich genug ist, und vor alle Dingen, ob er ueberhaupt von Interesse fuer die Clubmitglieder ist.

Wenn also jemand Zeit und Musse hat, so moege er mir vielleicht ein paar Zeilen schreiben.



„Ich sehe schon ... sie gehen voll in Ihre Arbeit auf“

## Tips & Tricks -- Tips & Tricks -- Tips

Im 3. INFO wurde ein PEEK über die Druckerabfrage für den TRS-62 gebracht.

```
100 IF PEEK(14312) > 127 Then PRINT "Drucker anschalten !": GOTO 100
```

Im 6. INFO fragte Werner wie er den Befehl für seinen GENIE verwenden kann.

Da der Genie den Drucker über seinen User-Port ansteuert muß nun der Port abgefragt werden. Dies geschieht mit folgendem Syntax:

```
100 IF INP(253) > 127 Then PRINT "Drucker anschalten !": GOTO 100
```

Beim EPSON und dem GENIE wurden folgende Werte ermittelt:

INP(253) = 63 Drucker aus oder ok.

INP(253) = 159 off line

INP(253) = 223 Paperout

INP(253) = 255 Kein Druckerkabel angeschlossen.

--Noch kurz vor INFO-Schluß von Gerald und Hartmut per Telefon.--



**Zap im SUPERZAP!**

Einer der größten Mängel von SUPERZAP ist die Tatsache, daß man die Sectorinhalte nur hexadezimal modifizieren kann. Vor allem dort, wo längere Texte geändert werden sollen, ist dieser Mangel sehr störend! Dieses Manko ist nun abgestellt. Das unten abgedruckte Zap ermöglicht es, einen Sector nicht nur in HEX, sondern auch in ASCII zu ändern.  
 Die Umschaltung erfolgt durch die Taste <CLEAR>. Die Steuerung des Cursors geschieht weiterhin mit den Pfeiltasten, mit der Ausnahme, daß man, um eine Zeile nach oben zu springen, <SHIFT> + ↑ drücken muß.

```

0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 A88E8E8E8E8E8E8E8E8E
0010 0000 0000 0000 0000 0000 0000 0000 00F6 FE7E 8E8E8E8E8E8E8E8E8E
0020 2B02 FE1B CAF7 5CFE 1FC2 345D E521 3952 (B)A(0) "40"19R
0030 2274 5C21 5B03 22EF SDE1 C369 5C33 333E "t0!AC"00"i033>
0040 6C32 E15D 79E6 F04F 2600 2929 79E6 0FB5 "2"uy"0&5"y"01
0050 6F11 303C CDDE SDE1 CBC0 FE20 3B0C CB48 0000 "0"BL"41
0060 23A60 CB88 C2A6 5D1E 00C3 905C 4FCR 80FE 08CA "U"5"00"U"41
0070 AESC CBC0 FE1F C2AE SCE5 21BC 5D22 745C "5"2"00!"0"t0
0080 2105 622E F55D E1CB 801B AF00 0000 0000 !Eb."U"X"555555
0090 0000 0000 0000 0000 0000 0000 0000 0000 555555555555555555
0100 0000 0000 0000 0000 0000 0000 0000 0000 555555555555555555
0110 0000 0000 0000 0000 0000 0000 0000 0000 555555555555555555
0120 0000 0000 0000 0000 0000 0000 0000 0000 555555555555555555
0130 0000 0000 0000 0000 0000 0000 0000 0000 555555555555555555
0140 0000 0000 0000 0000 0000 0000 0000 0000 555555555555555555
0150 0000 0000 0000 0000 0000 0000 0000 0000 555555555555555555
0160 0000 0000 0000 0000 0000 0000 0000 0000 555555555555555555
0170 0000 0000 0000 0000 0000 0000 0000 0000 555555555555555555
0180 0000 0000 0000 0000 0000 0000 0000 0000 555555555555555555
0190 0000 0000 0000 0000 0000 0000 0000 0000 555555555555555555

```

Der obenstehende Sectordump stellt den Sector 0 von SUPERZAP/CMD dar. Dieser Sector enthält im Original nur Nullen. Die Änderungen sind also nicht schwer zu erkennen. Weiterhin müßen noch vier Bytes im Sector 9 geändert werden. Beginnend beim relativen Byte BEH werden folgende Änderungen vorgenommen:  
 C2 34 5D 7D in C2 1A 52 7D.

Viel Spaß beim Zappen, euer Rantmut Obermann.

Quelle: Clubinfo des TRS80-Computerclub München

**Noch zwei Tips zu SCRIPSIT:**

Änderung der Voreinstellung für die Anzahl Zeichen pro Zeile  
 =====  
 auf dem Bildschirm:  
 =====

```

0,A6 von 0E 3C FD = 60 Zeichen pro Zeile
nach 0E 3F FD = 63 Zeichen pro Zeile

```

Umdefinieren einiger SCRIPSIT-Kommando-Tasten  
 =====

Mit den folgenden Änderungen werden Kommando-Tasten bei Scripsit so geändert, das die Bezeichnung der Taste der Funktion des Kommandos entspricht (z.B. <CONTROL><I> statt <CONTROL><S> für Insert):

Bedeutung	Alt		Neu		Speicher
	Taste	Code	Taste	Code	
Wort	Z	1A	W	17	7A0D 79E4
Paragraph	C	03	P	10	79A5 7A0E 79E2
Zeile	X	18	Z	1A	79A4 79E1 568B
Blanks	F	06	B	02	79E0
Insert	S	13	I	09	79A1

Literatur: 80 Micro 3/83/18

Josef Konrad

Nochmal: Ausoabeumschaltung in BASIC  
 (diesmal ohne Fehler!?!)

Schon im letzten Info war das hier abgedruckte Programm veröffentlicht. Leider habe ich nicht beachtet, daß das von mir benutzte Textverarbeitungsprogramm das \$-Zeichen als Steuerzeichen benutzt und damit das BASIC-Programm verstümmelt und mit Fehlern spickt. Hier die korrigierte und (hoffentlich) fehlerfreie Version:

```

10 CLEAR 1000 : CLS
20 PRINT @ 0, "Ausgabe auf (D)rucker oder (B)ildschirm?"
30 A$=INKEY$: IF A$="D" OR A$="d" THEN 50
40 IF A$="B" OR A$="b" THEN 60 ELSE GOTO 30
50 CMD"ROUTE,DO,PR" : GOTO70
60 CMD"ROUTE,CLEAR"
70 FOR X=32 TO 191 : PRINT CHR$(X); " "; : NEXT : RUN

```

Rantmut Obermann

**Wenn die Affen es dahin bringen könnten, so Langeweile zu haben, so könnten sie Menschen werden.**  
 Johann Wolfgang von Goethe, deutscher Dichter (1749-1832)

Bei einem Wettlauf zwischen einer Schnecke und einem Kamel gewann zum Erstaunen aller die Schnecke. Sofort wurde dieses Ereignis von einer Sonderkommission untersucht. Sie stellte fest: Das Kamel hatte den Dienstweg eingeschlagen.

# Basic-Dolmetscher gezücht

Basic ist die Sprache der Computer. Aber es ist ein unglaubliches Sprachgewirr. Jeder Hersteller hat offenbar den Ehrgeiz, einen eigenen Dialekt zu schaffen. Deshalb ist es mühsam und gerade für den Anfänger unmöglich, ein fremdes Listing für seinen Computer zu übersetzen.

Der Mann aus Schleswig-Holstein sagt: »We snakt Platt!«. Der Niederbayer erklärt: »I red boarisch!«. Würden beide auf ihrem Dialekt beharren, könnten sie sich kaum miteinander unterhalten. Glücklicherweise verstehen beide Hochdeutsch. Damit ist das Gespräch gerettet. Hochdeutsch bildet hier also den Kern der Sprache, den beide »Volksstämme« verstehen. Ähnlich gelagert ist auch das Verhältnis bei Basic. Entwickelt wurde sie vor genau 20 Jahren von den Professoren John G. Kemeny und Thomas E. Kurtz. Am Dartmouth College wurde sie zuerst eingeführt und trat von dort aus ihren Siegeszug an. Heute gibt es über 250 Basic-Dialekte, die alle um den ursprünglichen Kern aufgebaut sind. Jeder Hersteller entwickelte für seinen Heimcomputer eine besonders geeignete Version. An einen Standard oder gar eine Norm brauchte sich keiner zu halten, das wurde natürlich ausgenutzt. Zum großen Ärger und Verdruss der Anwender. Sie sitzen nämlich vor ihrem Heimcomputer und haben nur ihr Basic zur Verfügung. Da gibt es keinen Umschalter für die unterschiedlichen Dialekte. Interessante Programme müssen also »zu Fuß« von einem »Slang« in den anderen übersetzt werden. Dabei kommt es natürlich zu Problemen. Denn nicht jeder Hersteller bringt seinem Heimcomputer alle Befehle der Sprache Basic bei.

Außerdem werden gleichartige Befehle anders genannt. Ein klassi-

sches Beispiel dafür ist der Befehl zum Löschen des Bildschirms. In der »Urform«, die auf den meisten Heimcomputern eingebaut ist, lautet er »CLS«. Übersetzt heißt das Clear Screen – lösche den Bildschirm. Beim Apple II und allen Commodore-Computern zum Beispiel führt diese Eingabe zu einem Syntax Error. Das heißt soviel wie: »Den Befehl kenne ich nicht«. Der

Apple hätte hier gerne die Eingabe »Home«, bei den Commodore-Computern ist »Print Chr\$(147)« erwünscht und beim TI 99/4A muß mit »Call Clear« das entsprechende Maschinenprogramm aufgerufen werden. Diese Eingaben sind immer auf den Einsatz in Programmen ausgelegt. Zum direkten Löschen des Bildschirms haben die meisten Computer eine entsprechende Tastenfunktion eingebaut. Würde für die Commodore-Computer tatsächlich »Print Chr\$(147)« in einem Listing abgedruckt, gäbe es bei der Kon-

```
1 REM "DIESE VERSION GIBT EINEN
2 REM "BLINKENDEN TEXT BEI DEN
3 REM "COMMODORE- COMPUTERN AUS
4 REM "DIE STEUERCODES FUER ANDERE
5 REM "COMPUTER SIND DER TABELLE
6 REM "ZU ENTNEHMEN.
7 REM "MIT CHR$(145) WIRD DER
8 REM "CURSOR EINE ZEILE NACH
9 REM "OBEN BEWEGT
10 PRINT "TEXT ";PRINTCHR$(145);
15 REM "ZEITDAUER FUER TEXTDARSTELLUNG
20 FOR A = 1 TO 100: NEXT A
30 PRINT " ";PRINTCHR$(145);
35 REM "FUER DIESE ZEIT WIRD KEIN
36 REM "TEXT AUSGEGEBEN
40 FOR A = 1 TO 100: NEXT A
50 GOTO 10
```

Listing 1. Durch zwei For-To-Next-Schleifen läßt sich der Befehl »Flash« simulieren

```
10 REM KONTROLL- CODES
20 REM FUER COMMODORE
30 REM COMPUTER
40 PRINT "C": REM BILDSCHIRM
50 " REM LOESCHEN
60 PRINT "S": REM CURSOR IN
70 " REM DIE OBERE
80 " REM LINKE ECKE
90 PRINT "M": REM CURSOR RECHTS
91 PRINT "L": REM CURSOR LINKS
92 PRINT "D": REM CURSOR RUNTER
93 PRINT "U": REM CURSOR RAUF
```

vertierung in andere Dialekte keine Probleme. Hier werden aber besondere Controll-Codes eingesetzt (Bild 1).

Auch bei der Benennung von Basic-Befehlen kommt es zu Unterschieden. Wird zum Beispiel in einem Programm für den Sharp-Computer MZ-700 der Befehl »SPC (b)« aufgelistet, so wird eine Zeichenkette erzeugt, die, dem Wert der Variablen »b« entsprechend, aus Leerzeichen besteht. Bei anderen Computern wiederum wird mit diesem Befehl bei einer Ausgabe mit »Print« die durch »b« angegebene Anzahl von Zeichen in einer Zeile übersprungen.

Zusätzlich erschwert wird die Konvertierung durch irreführende Tastenbeschriftungen. Soll zum Beispiel der Colour Genie die Anweisung »Clear« ausführen, so ist damit

Bild 1. So stellen Commodore-Computer Control-Codes dar

```
1 REM "MIT DIESEM PROGRAMM
2 REM "WIRD DIE PRINT USING
3 REM "FUNKTION SIMULIERT
4 REM
10 INPUT "WIEVIELE STELLEN ?";Z
20 INPUT "ZAHL";A
30 GOSUB 9000
35 PRINT "AUS";A; " WURDE ";ZF$
50 GOTO 10
60 REM "MIT DIESEM BEISPIEL
70 REM "WIRD IN DIE ROUTINE
80 REM "AB ZEILE 9000 GESPRUNGEN
9000 F$=STR$(A)
9010 FOR X= 1 TO LEN(F$)
9020 IF MID$(F$,X,1)="#" THEN 9050
9030 NEXT X: ZF$=F$
9040 RETURN
9050 F$=F$+"0000000000000000"
9055 ZF$=LEFT$(F$,X+Z)
9060 RETURN
```

gemeint daß er alle Variablen löscher soll. Auf der Tastatur dieses Computers findet sich aber auch eine Taste mit der Beschriftung »Clear«. Durch einen Druck auf diese Taste wird der Bildschirm gelöscht. Die Befehle »Print« und »Input«, so erstaunlich es klingt, sind noch bei allen Basic-Dialekten gleich. Durch den Befehl »Print« erfolgt die Ausgabe einer Leerzeile, »Print a« gibt den Inhalt der Variablen »a« aus und »Print a\$« führt zur Ausgabe der Textvariablen »a\$«. Besondere Aufmerksamkeit wird durch blinkende Texte erregt. Hierfür stellt der Apple II-Computer und

Listing 2. Die Funktion »Print Using« mit einem Unterprogramm simuliert

```
1 REM "AUCH MIT DER 'GET'ABFRAGE
2 REM "KOENNEN ZEICHENKETTEN
3 REM "EINGEGEBEN WERDEN
4 REM "DIESES PROGRAMM VERLAENGERT
5 REM "DIE TEXTVARIABLE 'B'
6 REM "SOLANGE, BIS 'RETURN'
7 REM "EINGEGEBEN WIRD.
10 GET A$
20 IF A$="" THEN 10
25 IF A$=CHR$(13) THEN 60
30 B$= B$+A$
40 PRINTB$
50 GOTO 10
60 END
```

Listing 3. Eingabe einer Zeichenkette mit der »Get«-Abfrage

der Sinclair Spectrum die Funktion »Flash« zur Verfügung. Andere Computer können diese Funktion mit einem kleinen Unterprogramm simulieren (Listing 1). Beim Ausdruck von Daten erleichtert eine formatierte Ausgabe den Überblick. In vielen Basic-Dialekten findet man hierzu den Befehl »Print Using«. Mit einer Anordnung von Doppelkreuzen »\*« wird vor der Ausgabe ein Format festgelegt. So führt zum Beispiel die Anweisung: Print Using \*\*\*\*\*A dazu, daß die Variable »A« immer mit drei Nachkommastellen ausgedruckt wird. Diese Ausgabeform läßt sich bei den meisten Heimcomputern durch eine geeignete Unterroutine simulieren (Listing 2). Mit der »Input«-Anweisung werden während eines

```

1 REM "DEF TEXTVARIABLEN "A$
2 REM "WIRD DER TEXT "1000"
3 REM "ZUGEGORDNET. DIE "VAL"
4 REM "FUNKTION WANDELT DARFUS
5 REM "DIE NUMERISCHE VARIABLE
6 REM "MIT DEM WERT "1000"
10 A$ = "1000"
20 PRINT A$;" TEXT "
30 B = VAL(A$)
40 PRINT B;" ZAHL "
50 C = B+234
60 PRINT C;" VARIABLE + ZAHL "
70 REM "SO LASSEN SICH TEXTE
71 REM "MATHEMATISCH BEEINFLUSSEN
72 REM "DIE FUNKTION "STR$"
73 REM "WANDELT DIE ZAHL
74 REM "WIEDER IN EINE TEXT-
75 REM "VARIABLE UM.
80 A$=STR$(C)
90 PRINTA$;" TEXT "

```

**Listing 4.**  
**Mit Textvariablen rechnen**

Programmlauf Variablen eingegeben. Dabei wartet das Programm solange, bis die Eingabe durch ein »Return« abgeschlossen wird. Für viele Aufgaben ist diese Lösung in-diskutabel, zum Beispiel wenn ein Programm durch Tastendrücke gesteuert wird. Auch wenn keine Taste gedrückt ist, sollen Berechnungen durchgeführt werden. Die Hauptanwendung liegt hier bei Spielen. Dafür bietet sich die »Get« oder »Inkey« Funktion an. Aber nicht jede »Get«-Abfrage arbeitet gleich. Das Basic des Apple II unterbricht das laufende Programm auch bei dieser Abfrage und arbeitet erst nach einem Tastendruck weiter (ohne »Return«). Bei den Commodore-Computern wird jedoch, wie bei allen »Inkey«-Befehlen, das Programm nicht zur Dateneingabe angehalten. Soll hier eine Entscheidung in Abhängigkeit eines Tastendrucks getroffen werden, muß in den nachfolgenden Programmzeilen gegebenenfalls ein Rücksprung zur »Get«- oder »Inkey«-Abfrage erfolgen. Beide Möglichkeiten haben gemeinsam, daß sie immer nur ein Zeichen verarbeiten. Zur Eingabe einer Zeichenkette kann ein entsprechendes Programm dienen (Listing 3). Zur Verarbeitung von Zeichenketten stellen die meisten Heimcomputer eine große Befehlsauswahl zur Verfügung. Die Befehle Val(A\$) und Str\$(A) sind in fast allen Dialekten enthalten. Außer bei den Computern von Atari. Die kennen diese Befehle überhaupt nicht. Dabei handelt es sich um sehr praktische Befehle, die eine Beeinflussung von

```

1 REM DER BEFEHL "STRING$"
2 REM KANN SIMULIERT
3 REM WERDEN. DAZU
4 REM DIESE KLEINE
5 REM BEISPIEL
10 L = 20
15 REM IN "L" IST DIE
16 REM LAENGE DER ZEICHEN-
17 REM KETTE ENTHALTEN
20 A$ = "*"
25 REM A$ ENTHAELT DAS
26 REM AUSZUGEBENDE ZEICHEN
30 FOR S = 1 TO L
40 PRINTA$;
50 NEXT S
60 REM WO IN ANDEREN
70 REM PROGRAMMEN DER
80 REM "STRING$" BEFEHL
90 REM AUFTAUCHT, KANN
100 REM ER DURCH DIESE
110 REM ZEILEN ERSETZT
120 REM WERDEN.

```

Zeichenketten nach mathematischen Regeln erlauben (Listing 4). Mit Val(A\$) wird die Textvariable A\$ in eine numerische Variable umgewandelt. Diese kann nun durch mathematische Ausdrücke verändert werden. Danach kann durch den Befehl Str\$(A) dieser Ausdruck wieder in eine Textvariable zurückverwandelt werden. Enthält die Textvariable A\$ außer Zahlen auch Buchstaben, so übernimmt der Befehl Val(A\$) nur die linksbündigen Zahlen. Zum Beispiel wird aus A\$="1234DER56" die Zahl »1234«.

Die nachfolgenden Ziffern werden ignoriert.

Stößt man in einem Listing auf den Befehl »String\$(10,\*)«, so handelt es sich dabei um eine Funktion, die einen String erzeugt, der zehnmal das Zeichen »\*« enthält. Diese Funktion kann relativ leicht simuliert werden (Listing 5). Ebenso leicht läßt sich der Befehl »Swap« ersetzen. Dabei handelt es sich um den Austausch von zwei Variablen. Ist zum Beispiel in der Variablen »A« der Wert »5«, und in der Variablen »B« der Wert »8« gespeichert, so ist nach Ausführung des Befehls »Swap (A,B)« der Inhalt von »A« und »B« getauscht. Um diese Funktion in den jeweiligen Basic-Dialekt umzusetzen, bietet sich der Austausch über eine Hilfsvariable an (Listing 6).

Bei allen Programmbeispielen sollte auf das richtige Einsetzen der

```

1 REM DIE "SWAP" FUNKTION
2 REM TAUSCHT
3 REM ZWEI VARIABLEN
10 INPUT " ZAHL 1 ";A
20 INPUT " ZAHL 2 ";B
30 C=A:A=B:B=C
40 PRINT A,B
50 REM DIESE BEFEHLSFOLGE
60 REM SIMULIERT DIESEN
70 REM BEFEHL

```

**Listing 6. Der Wert einer Variablen kann auch ohne »Swap« getauscht werden**

**Listing 5. »String\$« — für viele Computer ein Fremdwort**

Variablen geachtet werden. Solche Variablennamen, die im Programm bereits benutzt werden, können natürlich nicht für eine Befehlsimulation eingesetzt werden. Beim Ersatz von Befehlen durch Befehlsfolgen verlangt man sich natürlich die Arbeitsgeschwindigkeit.

In dieser Ausgabe befaßen wir uns mit Befehlen, die Textein- und -ausgabe sowie die Verarbeitung von Zeichenketten ausführen. Der nächste Teil behandelt die unterschiedlichen Grafikmöglichkeiten.

siehe Übersicht auf Seite 70

(15)

	Computer	Apple II	Atari 800/800 XL	Commodore VC 20, C 64, 3000er, 4000er, 8000er	Sharp MZ700	Texas Instruments TI 99-4A	Sinclair ZX81	Sinclair Spectrum	Lace Colour Genie	Laser 210 SIC V2 23C	Yanco TRS 6 Level II
Funktionen											
Bildschirm löschen	Call — 93f oder Home	Print CHR\$(125)	Print CHR\$(147)	Print CHR\$(198)	Call Clear	CLS	CLS	CLS	CLS	CLS	CLS
Umschaltung auf inverse Darstellung	Inverse	Atari-Taste	Print CHR\$(16)	—	—	—	Invers 1	—	—	—	—
Umschaltung auf normale Darstellung	Normal	Atari-Taste	Print CHR\$(146)	—	—	—	Invers 0	—	—	—	—
Cursor nach rechts bewegen	Print CHR\$(21)	Print "Esc Control —"	Print CHR\$(29)	Print CHR\$(195)	—	—	—	Print CHR\$(25)	Print CHR\$(8)	Print CHR\$(25)	—
Cursor nach links bewegen	Print CHR\$(8)	Print "Esc Control —"	Print CHR\$(157)	Print CHR\$(196)	—	—	—	Print CHR\$(24)	Print CHR\$(5)	Print CHR\$(24)	—
Cursor nach oben bewegen	Call-998	Print "Esc Control ↑"	Print CHR\$(145)	Print CHR\$(194)	—	—	—	Print CHR\$(27)	Print CHR\$(27)	Print CHR\$(27)	—
Cursor nach unten bewegen	Print CHR\$(10)	Print "Esc Control ↓"	Print CHR\$(17)	Print CHR\$(193)	—	—	—	Print CHR\$(26)	Print CHR\$(10)	Print CHR\$(26)	—
Text blinkend darstellen	Flash	siehe Listing 1	siehe Listing 1	siehe Listing 1	—	—	Flash	siehe Listing 1	siehe Listing 1	siehe Listing 1	—
Ausgabe von Variablen	Print	Print	Print	Print	Print	Print	Print	Print	Print	Print	Print
Ausgabe von Variablen an einer bestimmten Bildschirmposition	HDABx, VTABy, Print	Position (Spalte, Zeile), Print	Print "Cursorsteuercode," Variable	Cursor (Spalte, Zeile), Print	Display At (x,y), Variable (nur Extended-Basic)	Print at (X, Y), Variable	Print at (x,y), Variable	Print @	Print @	Print @	Print @
Ausgabe von Variablen an einer bestimmten Zeilenposition	Print TAB	—	Print TAB	Print TAB	Print TAB	Print TAB	Print TAB	Print TAB	Print TAB	Print TAB	Print TAB
Formatierte Textausgabe	siehe Listing 2	—	siehe Listing 2	siehe Listing 2	Print Using (nur Extended-Basic)	—	—	Print Using	Print Using	Print Using	—
Eingabe von Variablen während des Programmlaufs	Input	Input	Input	Input	Input	Input	Input	Input	Input	Input	Input
Tastaturabfrage (im Programm)	Get	XIO 7	Get	Get	Call Key	Inkey	Inkey	Inkey	Inkey	Inkey	Inkey
Den ASCII-Wert eines Zeichens ermitteln	ASC(A\$)	ASC(A\$)	ASC(A\$)	ASC(A\$)	ASC(A\$)	Code (A\$)	Code (A\$)	ASC(A\$)	ASC(A\$)	ASC(A\$)	ASC(A\$)
Ab welchem Punkt a ist B\$ in der Textvariablen C\$ enthalten	—	—	—	—	Pos (C\$,B\$,a)	—	—	—	—	—	Instr (C\$,B\$,a)
Länge einer Zeichenkette ermitteln	Len (A\$)	Len (A\$)	Len (A\$)	Len (A\$)	Len (A\$)	Len (A\$)	Len (A\$)	Len (A\$)	Len (A\$)	Len (A\$)	Len (A\$)
Eine Zahl in einen String umwandeln	Str\$(A)	Str\$(A)	Str\$(A)	Str\$(A)	Str\$(A)	Str\$(A)	Str\$(A)	Str\$(A)	Str\$(A)	Str\$(A)	Str\$(A)
Einen String in eine Zahl umwandeln	Val(A\$)	Val(A\$)	Val(A\$)	Val(A\$)	Val(A\$)	Val(A\$)	Val(A\$)	Val(A\$)	Val(A\$)	Val(A\$)	Val(A\$)
Aus A\$ die ersten "B"-Zeichen von links entnehmen	Left \$(A\$,B)	Print * \$(A\$,B)	Left \$(A\$,B)	Left \$(A\$,B)	Seg \$(A\$,B)	—	—	Left \$(A\$,B)	Left \$(A\$,B)	Left \$(A\$,B)	—
Aus A\$ einen Teil-String vom "B"-ten Zeichen bis zum "C"-ten Zeichen entnehmen	Mid \$(A\$,B,C)	Print * \$(A\$,B,C)	Mid \$(A\$,B,C)	Mid \$(A\$,B,C)	Seg \$(A\$,B,C)	Teilstringabfrage über A\$(B to C)	—	Mid \$(A\$,B,C)	Mid \$(A\$,B,C)	Mid \$(A\$,B,C)	—
Aus A\$ die letzten "B"-Zeichen von rechts entnehmen	Right \$(A\$,B)	Print * \$(A\$,B)	Right \$(A\$,B)	Right \$(A\$,B)	Seg\$(A\$, Len(A\$)-B)	—	—	Right \$(A\$,B)	Right \$(A\$,B)	Right \$(A\$,B)	—
Erzeugt "B"-lange Zeichenkette mit dem ersten Zeichen aus A\$	siehe Listing 5	String \$(B,A\$)	siehe Listing 5	siehe Listing 5	RPT \$(A\$,B)	siehe Listing 5	siehe Listing 5	String \$(A\$,B)	siehe Listing 5	String \$(A\$,B)	—
Erzeugt "B"-lange Zeichenkette mit Leerzeichen	—	siehe Listing 5	—	SPC(B)	RPT \$( " ",B)	—	—	String \$(B, " ")	—	Space \$(B)	—
In einer Zeile werden "B"-Zeichen übersprungen	SPC(B)	String \$(B, " ")	SPC(B)	—	—	—	—	—	—	—	—
Löschen aller Variablen	CLR	CLR	CLR	CLR	—	Clear	Clear	Clear	Clear	Clear	Clear
Inhalt zweier Variablen tauschen	siehe Listing 6	siehe Listing 6	siehe Listing 6	siehe Listing 6	siehe Listing 6	siehe Listing 6	siehe Listing 6	Swap (A,B)	siehe Listing 6	Swap (A,B)	—

\* Mit x wird der Anfangswert und mit y der Endwert angegeben, der den Teilstring definiert.

Übersicht der Basic-Dialekte

# Tongenerator

Wenn sich ein Programm dem Benutzer bemerkbar machen soll, kann dieses auch auf dem akustischen Weg geschehen.

Eine Tonmeldung bietet sich an z. B. bei einer falschen Eingabe, nach einer längeren Rechnung, oder als Effekt bei Spielen.

Bei dem TRS 80 kann man Töne erzeugen, indem man die Spannung am AUX-Stecker steuert.

Das Maschinenprogramm gibt auf den Ausgabeport 255 so schnell zwei verschiedene Werte des Akkumulators aus, daß eine Schwingung simuliert wird. Die Warteschleifen zwischen den Ausgaben von A beeinflussen die Anzahl der Schwingungen pro Sekun-

de, also die Frequenz des Tones. Je länger die Warteschleifen sind, desto tiefer ist der Ton. Und je häufiger man die Schwingungen wiederholt, desto länger wird der Ton gehalten.

Wenn man über einen Macro-assembler mit einigen arithmetischen Erweiterungen verfügt, kann man das Maschinenprogramm unverändert benutzen. Sonst lasse man die Zeilen 140 bis 190 weg und ersetze die Macroaufrufe von Zeile 450 bis 650 durch die angegebenen DEFEB-Befehle.

Die Zahlenangaben im Programm sind alle dezimal! Einem NOTE X-Befehl entsprechen zwei DEFEB-Befehle,

und zwar bestimmt der erste DEFEB-Befehl die Tonhöhe und der zweite die Tonlänge.

Im BASIC-Programm ist das Maschinenprogramm in der Zeile 10 gespeichert und wird in Zeile 20 geladen. Die Tondaten sind in der Zeichenkettenvariablen A\$ gespeichert. Man kann in einem BASIC-Programm mehrere Tonfolgen benutzen, indem man einfach den Wert von A\$ ändert u. das Unterprogramm ab Zeile 90 aufruft. Die Tonfolge wird dann mit IUSR (0), PRINT USR (0) o.ä. über den AUX-Stecker ausgegeben.

```

10 DATA 243,6,21,221,33,40,125,80,221,102,0,221,110,1,62,2,93
    ,68,61,211,255,16,254,68,60,211,255,16,254,29,32,241
    ,221,35,221,35,66,16,224,201
20 FOR I=32000 TO 32039: READ J: POKE I,J: NEXT I
30 A$="AZBZCZDZEZFGZHIZJZKZLZMZNZPZQZRZSZTZUZVZWZXZYZZ"
40 GOSUB 90
50 DEFUSR=32000 'ODER POKE 16526,0 UND POKE 16527,125
60 I=USR(0)
70 END
90 REM DAS FOLGENDE UNTERPROGRAMM INITIALISIERT DEN
    WERT VON A$ ALS DATEN FUER DIE TONROUTINE
90 POKE 32002,LEN(A$)/2
100 I=VARPTR(A$)
110 POKE 32005,PEEK(I+1)
120 POKE 32006,PEEK(I+2)
130 RETURN
    
```

DATEN	DEFB	76
	DEFB	247
	DEFB	80
	DEFB	238
	DEFB	85
	DEFB	229
	DEFB	90
	DEFB	220
	DEFB	95
	DEFB	211
	DEFB	102

DEFB	202
DEFB	107
DEFB	193
DEFB	114
DEFB	184
DEFB	121
DEFB	175
DEFB	129
DEFB	166
DEFB	136

DEFB	157
DEFB	144
DEFB	148
DEFB	153
DEFB	139
DEFB	162
DEFB	130
DEFB	172
DEFB	121
DEFB	183
DEFB	112
DEFB	194
DEFB	103
DEFB	206
DEFB	94
DEFB	219
DEFB	85
DEFB	232
DEFB	76
DEFB	246
DEFB	67

DEND

```

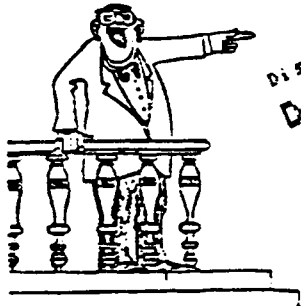
00100 ; #####
00110 ; ## TONGENERATOR ##
00120 ; #####
00130 ;
00140 NOTE MACRO #PAR1
00150 TON1 DEFL 5*#PAR1*#PAR1*#PAR1
00160 TON2 DEFL 8*#PAR1*#PAR1+447*#PAR1+7145
00170 DEFEB TON2/100+TON1/1000
00180 DEFEB -9*#PAR1
00190 ENDM
00200 ORG 32000
00210 STAR1 DI ; INTERRUPTS SIOEREN DIE IOENE!
00220 ZAHL DEFL DEND-DATEN
00230 LD B,ZAHL/2+1 ; ANZAHL DER IOENE
00240 LD IX,DATEN
00250 WEITER LD D,B
00260 LD H,(IX) ; TONHOEHE
00270 LD L,(IX+1); TONLAENGE
00280 LD A,Z
00290 LD E,L
00300 WIEDER LD B,H
00310 DEC A
00320 OUT (255),A
00330 WARTE1 DJNZ WARTE1
00340 LD B,H
00350 INC A
00360 OUT (255),A
00370 WARTE2 DJNZ WARTE2
00380 DEC E
00390 JR NZ,WIEDER ; TON HALTEN
00400 INC IX
00410 INC IX
00420 LD B,D
00430 DJNZ WEITER
00440 REI
00450 DATEN NOTE 1 00560 NOTE 12
00460 NOTE 2 00570 NOTE 13
00470 NOTE 3 00580 NOTE 14
00480 NOTE 4 00590 NOTE 15
00490 NOTE 5 00600 NOTE 16
00500 NOTE 6 00610 NOTE 17
00510 NOTE 7 00620 NOTE 18
00520 NOTE 8 00630 NOTE 19
00530 NOTE 9 00640 NOTE 20
00540 NOTE 10 00650 DEND NOTE 21
00550 NOTE 11 00670 END START
    
```

Das Schönste am Berühmtsein ist: Wenn man Leute langweilt, glauben sie, es ist ihre Schuld.

Henry Kissinger,  
ehemaliger amerikanischer Außenminister (geboren 1923)

Das absolut Negative an den schnellen Flugzeugen ist, daß es jetzt keine entfernten Verwandten mehr gibt.

Autor unbekannt



Diskussion  
Diskussion  
Diskussion  
Diskussion  
Schreibt Eure Meinung zu diesen Themen !!

#### INHALTSVERZEICHNIS VON ZEITSCHRIFTEN

Sicher hat jeder schon mal Informationen zu einem gewissen Thema über seinen Rechner (ich z.B. suche z.Zt. Informationen über die Programmiersprachen 'C' und 'LISP' für Modell I) oder ganz allgemein zu einem Thema gesucht.

Da nun innerhalb des Clubs bei den einzelnen Mitgliedern sicherlich viele verschiedene Zeitschriften vorhanden sind, möchte ich folgende Anregung machen:

Man könnte doch die Inhaltsverzeichnisse von Zeitschriften austauschen, und zwar entweder als Kopie oder über ein geeignetes Dateiprogramm.

Ich habe z.B. die Inhaltsverzeichnisse von verschiedenen (z.T. unvollständigen) Jahrgängen der Zeitschriften MC, ELEKTOR, 80 MICRO und 80 US als UNIDAT80-Datei.

Der Austausch über Kopien ist zunächst mal der einfachere Weg, hat allerdings ab einem gewissen Umfang dann bei der Suche nach Artikeln Nachteile (man muß viel suchen, übersieht leicht etwas).

Beim Austausch über ein Dateiprogramm ergibt sich das Problem der Auswahl eines geeigneten Dateiprogramms (UNIDAT80 ist z.B. nur für SD, SUPER ist bei der Tastaturabfrage recht langsam und hat eine feste Satzlänge, selbstgeschriebene Programme erfordern einen ziemlich hohen Programmieraufwand, da sehr schnell einige tausend Datensätze zusammenkommen (bei mir z.Zt. über 2000) und das Programm dadurch ziemlich schnell sein muß, auch muß es wirklich ausgereift sein, um nicht nach tagelanger Dateneingabe feststellen zu müssen, daß das Programm dieses oder jenes nicht leistet), während die Eingabe eines Inhaltsverzeichnisses bei einem guten Programm recht schnell erledigt ist (je nach Umfang 1-2 Tage).

Wer Interesse oder Anregungen zu diesem Vorschlag hat, soll sich doch bei mir melden oder über das nächste Info seine Meinung darlegen.

Josef Konrad

#### Kurzerklärungen für Programme

Sicher habt Ihr das Problem auch schon gehabt: Mitten in der Anwendung eines Programmes wisst Ihr den entsprechenden Befehl nicht. Man muß in der Anleitung nachsehen. Aber wo steht's jetzt?

Bis nun das Richtige gefunden ist gibt es eine ganz schöne Sucherei, Blättereier und Leserei. Und diese zusätzliche unnütze Nebenbeschäftigung kostet uns Zeit. Zeit, die uns vom reinen Vergnügen der Computerei abhält.

Viele Programme haben umfangreiche Erklärungen über ihre Funktion oder ihre Bedienung. Märe es nicht gut eine einseitige, maximal zweiseitige Programmanleitung zu erstellen. Ein Blick genügt, und man findet das Gewünschte auf Anhieb. Gleichzeitig könnte von den meist englischen Anleitungen eine Deutschversion geschrieben werden. Ein weiterer nützlicher Nebeneffekt besteht darin, daß auf diesen Eigenproduktionen von Anleitungen kein Copyright mehr besteht. Und somit von uns -im Club untereinander - getauscht oder vervielfältigt werden kann. Es ist auch günstiger und weniger zeitaufwendig eine Kurzversion zu haben, als hinzustehen und sich ein mehrseitiges Werk zu vervielfältigen. Ganz davon zu schweigen, daß dies sowieso verboten ist.

In unserer Clubgemeinschaft haben sicher einige von Euch solche Kurzerklärungen, die dem einen oder anderen die Bedienung bzw. Anwendung eines Programmes erleichtern hilft. Ich glaube, daß wir in dieser Beziehung zusammenarbeiten sollten. Bevor sich jeder selbst eine solche Hilfe erstellt, -die einer von Euch vielleicht schon hat-.

Nun mein Vorschlag, man könnte die Kurzanleitungen für die Clubmitglieder mit in die Programm-Bibliothek als Textfile aufnehmen, und/oder auf Karton gedruckt mit dem INFO für die Interessenten mitschicken. Ich würde mich bei eventuellem Interesse bereiterklären, die Anfertigung der kartonierten Hilfen, von Euren Textfile, zu übernehmen.

Was meint Ihr dazu

# Shift-TRICK

Programme, um den Kleinschriftmodus bei den Genie-Computern Genie I, Genie II und Colour Genie umzudrehen, gibt es ja eigentlich genug. Warum noch mal das gleiche Thema? Erstens, um einmal zu verdeutlichen, warum und wie es eigentlich kommt, daß die Genie-Computer in der Grundversion auf einfachen Tastendruck groß, bei Shift und Taste erst kleinschreiben und zweitens, um einige Hilfen zu geben, dieses leidige Softwareproblem anzugehen.

Der Leser hat richtig verstanden. Es handelt sich nicht etwa um ein Problem der Hardware (das könnte man ja noch verstehen!), sondern um die Software im ROM (dem Betriebssystem des Computers). Nimmt man sich nämlich ein ROM-Listing zur Hand, in dem das Betriebssystem ausführlich aufgelistet und erklärt ist, so können sich einem manchmal die Haare sträuben, wie gewisse Betriebsroutinen angelegt sind.

Es scheint, daß einige Softwarelösungen mit Absicht so geschrieben wurden, daß dem sehr ernstesten Anwender (z. B. Textverarbeiter) später nichts anderes übrig bleibt, als auf die Erweiterungen für den Computer zurückzugreifen, um kleine, aber manchmal doch sehr störende Eigenheiten des Systems wieder auszubügeln. Dies gilt auch in gewissem Maße für die Hardware eines Computersystems, wie beispielsweise mein Artikel in der GENIE DATA Nr. 4 zeigt. Mit einem wirklich minimalen Aufwand wird hier der Einbau von vier

weiteren Tasten in die Tastatur beschrieben, so daß sich die deutschen Umlaute direkt erreichen lassen.

Doch nun zurück zu unserem eigentlichen Thema. Für die, die über kein ROM-Listing von einem der drei Genie-Computer verfügen, möchte ich grob auf die Tastaturabfrage bei den oben genannten Systemen eingehen. Das Prinzip ist für alle drei Systeme gleich. Dies ist nicht verwunderlich, da der Genie I und der Genie II direkte Nachfahren (Kopien) des guten alten TRS 80 Computers sind. Dieser Computer hat auch in Deutschland ein Stück Microcomputer-Geschichte mitgeschrieben und zählt eigentlich längst zu den Veteranen der Microcomputer. Dank seines guten Konzepts (vor allem seiner Software) erlebte er in den Genie-Computern eine Renaissance, die sein Betriebssystem fast unverändert übernahmen und lediglich eine bessere Hardware darumbauten. Leider wurden allerdings auch viele Schwächen des Betriebssystems mit übernommen.

Eine Weiterentwicklung des Betriebssystems stellt das ROM des Colour Genie dar. Bei Vergleich der ROM's des Colour Genie und des Genie I oder II ergibt sich, daß das Colour Genie fast die gleichen Adressen der einzelnen Routinen benutzt wie seine „Brüder“. Besonders auffallend ist dies bei den Befehlen Set und Reset. Sie befinden sich bei allen drei Systemen im ROM an fast der gleichen Stelle. Während bei Genie I und II der Befehl

für das Setzen, bzw. Rücksetzen, eines Grafikpunktes verantwortlich ist, bewirken sie beim Colour Genie das Setzen, bzw. Rücksetzen eines Bits in einer bestimmten Adresse.

Auch das Prinzip und der Aufbau der Tastaturabfrage ist bei allen drei Systemen fast identisch. Der jeweilige Computer trennt die Tasten nach Buchstaben, Ziffern und Sonderzeichen und nach Kontrolltasten. In jeweils gesonderten Teilprogrammen wird der zugehörige ASCII-Code errechnet und einem Register der CPU (Zentraleinheit = Mikroprozessor) zur weiteren Verarbeitung übergeben. So steht der ASCII-Code der jeweiligen Taste am Ende im sog. Akku (Register A der CPU).

Bei der Behandlung von Buchstaben wird es jetzt endlich für unser Ausgangsproblem interessant.

Hier erfolgt nämlich eine Abfrage, ob zu dem gedrückten Buchstaben auch die SHIFT-Taste betätigt wurde. Und siehe da, bei gedrückter SHIFT-Taste wird der sog. „Lower Case“ erzeugt, was nichts anderes bedeutet, als daß der ASCII-Code eines Kleinbuchstaben erzeugt wird. Bei nicht betätigter SHIFT-Taste hingegen wird Großschrift erzeugt.

Hier fragt man sich zuerst, warum dies so sein muß. Kann man denn diese Routine nicht so anlegen, daß bei SHIFT Großschrift erzeugt wird?

Man kann! Nämlich durch die Abänderung eines Maschinensprachebefehls. Hierdurch wird die oben erklärte Abfra-

## Hardware - Arbeitskreis ECB - BUS System

### Zwischenbericht:

Im letzten Clubinfo berichtete der Initiator der Hardwaregruppe Walter Zwickel über das geplante ECB - Bussystem. Ich bin heute so frei, in Stichpunkten über die Fortschritte der Entwicklung zu berichten.

1. Die Schaltung der Basis-Flatine (Entwicklung: Walter Zwickel) ist fertig und in der praktischen Erprobung! Die Schaltunterlagen wurden an alle Mitglieder der Hardwaregruppe verschickt!
2. Die Erstellung des Platinenlayouts für die Basis-Platine hat der Manfred Held übernommen. Wie er berichtet, geht es damit gut voran!
3. Das Netzteil für die ECB-Buserweiterung erstellt der Ulrich Böckling. Auch hier sind beachtliche Fortschritte zu vermelden!
4. Es ist zu erwarten, daß sowohl die Platine für die Basis-Schaltung, als auch das Netzteil bis zum Erscheinen des nächsten Clubinfos fertig sind. In diesem Info wird dann die ganze Sache eingehend vorgestellt und besprochen!

Übrigens, bei der Hardwaregruppe des **CLUB 80** kann jedes Mitglied mitmachen, egal wieviel Hardwareerfahrung es mitbringt!!! Die Kontaktperson ist Walter Zwickel.

Karsten Quernmann

ge umgedreht, sodaß der „Lower Case“ jetzt ohne SHIFT-Taste erzeugt wird.

Aber wie kann man ein doch unveränderliches ROM abändern?

Hier sind die Besitzer eines EPROM-Programmierers gut dran. Denn EPROM's (löschbare Festspeicher) lassen sich nach Belieben ändern und anstelle der ROM's in den Genie Computer einsetzen. Der EPROM-Typ für alle drei behandelten Computer ist ein 2532-EPROM.

Es handelt sich um einen 4 KByte Speicher, von dem sich im Colour Genie vier, im Genie I und II jeweils drei befinden.

Für die Programmierung geht man am besten folgendermaßen vor:

Die entsprechende Stelle, die geändert werden muß, befindet sich bei allen drei Genies im ersten ROM.

h., die Abfrage wurde einfach umgekehrt. Kleine Veränderung - große Wirkung!

Am besten schreibt man zu Beginn der Prozedur in einen freien Speicherplatz des RAM (Benutzerspeicher) den Wert 38H bzw. 56 dez. mittels Poke-Befehl. Günstig ist es dabei, wenn diese Speicherstelle vorher geschützt ist, damit dieser Wert nicht durch irgendwelche Umstände verlorengeht. Dann wird das ROM 1 bis zu der Adresse in ein EPROM 2532 kopiert, die vor dem zu ändernden Byte steht. Die nächste EPROM-Adresse wird mit dem Wert 38H aus der RAM-Speicherzelle programmiert. Der Rest des EPROM's wird wieder mit dem entsprechenden Teil aus dem ursprünglichen ROM programmiert, sodaß anschließend im EPROM eine genaue Kopie der ersten 4 KByte des ROM's steht, mit

nerlei Einfluß, da das Betriebssystem der Genie-Computer intern alle Kleinbuchstaben, die nicht in Anführungszeichen stehen, in Großbuchstaben umwandelt. Um dies zu Veranschaulichen, geben Sie im Direktmodus folgendes ein:

>10 goto 100  
Wichtig! Das „goto“ muß klein geschrieben sein.

Nun listen Sie das Miniprogramm! Es erscheint nun: 10 GOTO 100  
Wie man sieht, ist das „goto“ jetzt groß geschrieben. Colour-Genie-Besitzer müssen sich allerdings daran gewöhnen, daß auch die Funktions-tasten, die auch als Buchstaben behandelt werden, umgedreht sind.

Übrigens finden sich im Betriebssystem des Colour-Genie an einigen Stellen Freiräume, in die eventuell kleine eigene Routinen Platz

## Synchronisierte Taktumschaltung für VideoGenie

Liebe Clubfreunde,

Im Clubinfo Nr. 5 hat Gerald Dreyer eine Schaltung veröffentlicht, mit der man die Taktfrequenz des VideoGenie verdoppeln kann. Nachteil dieser Schaltung ist die Tatsache, daß bei der Umschaltung zwischen einfacher und doppelter Taktfrequenz in den meisten Fällen der Rechner hängenbleibt. Außerdem kann man die Taktfrequenz nur mit einem Schalter, also hardwaremäßig, jedoch nicht softwaremäßig umschalten. Um diese Nachteile zu beseitigen, hat Manfred Held die unten abgebildete Schaltung entworfen.

Die eigentliche Umschaltung der Taktfrequenz bewerkstelligen zwei NAND-Gatter (x2, rechts neben Z38). Die Umschaltung kann durch einen Schalter (linke obere Ecke) oder über den Port FEh (254d) (Decodierlogik linke untere Ecke) erfolgen. Um einen Systemabsturz während der Umschaltung zu verhindern, wird durch Z14 sichergestellt, daß der Umschaltvorgang erst erfolgt, wenn die CPU weder Schreib- noch Leseoperationen durchführt.

Durch die erweiterten Möglichkeiten der Schaltung ist natürlich auch der Bauteileaufwand gestiegen. Es werden folgende Teile benötigt:

- IC x1 = 74LS00 = vierfach NAND-Gatter mit je 2 Eingängen
- IC x2 = 74LS03 = vierfach NAND-Gatter mit offenem Kollektor-Ausgang
- IC x3 = 74LS04 = sechsfach Inverter
- IC x4 = 74LS30 = NAND-Gatter mit 8 Eingängen
- IC x5 = 74LS74 = zweifach Speicher-Flip-Flop, positiv flankengetriggert
- Alle IC's je 1 mal. Weiterhin:
- 2\* 1N4148 = Siliziumdiode
- 3\* 4700  $\Omega$  Widerstand
- 1\* 10 k $\Omega$  Widerstand
- 1\* 10  $\mu$ F/16V Kondensator

Viel Spaß beim Basteln, wünscht euch *Karlswald Obergmann*

*Eine Operatuse aus Wien  
dacht' beim Arbeiten stets nur an „Ihn“:  
Ihr Computer, voll Frust,  
verlor deshalb die Lust -  
er war auch nur ein Mensch, wie es schien.*

- Bei Genie I, II auf Adresse 0418H
- Beim Colour Genie auf Adresse 044CH

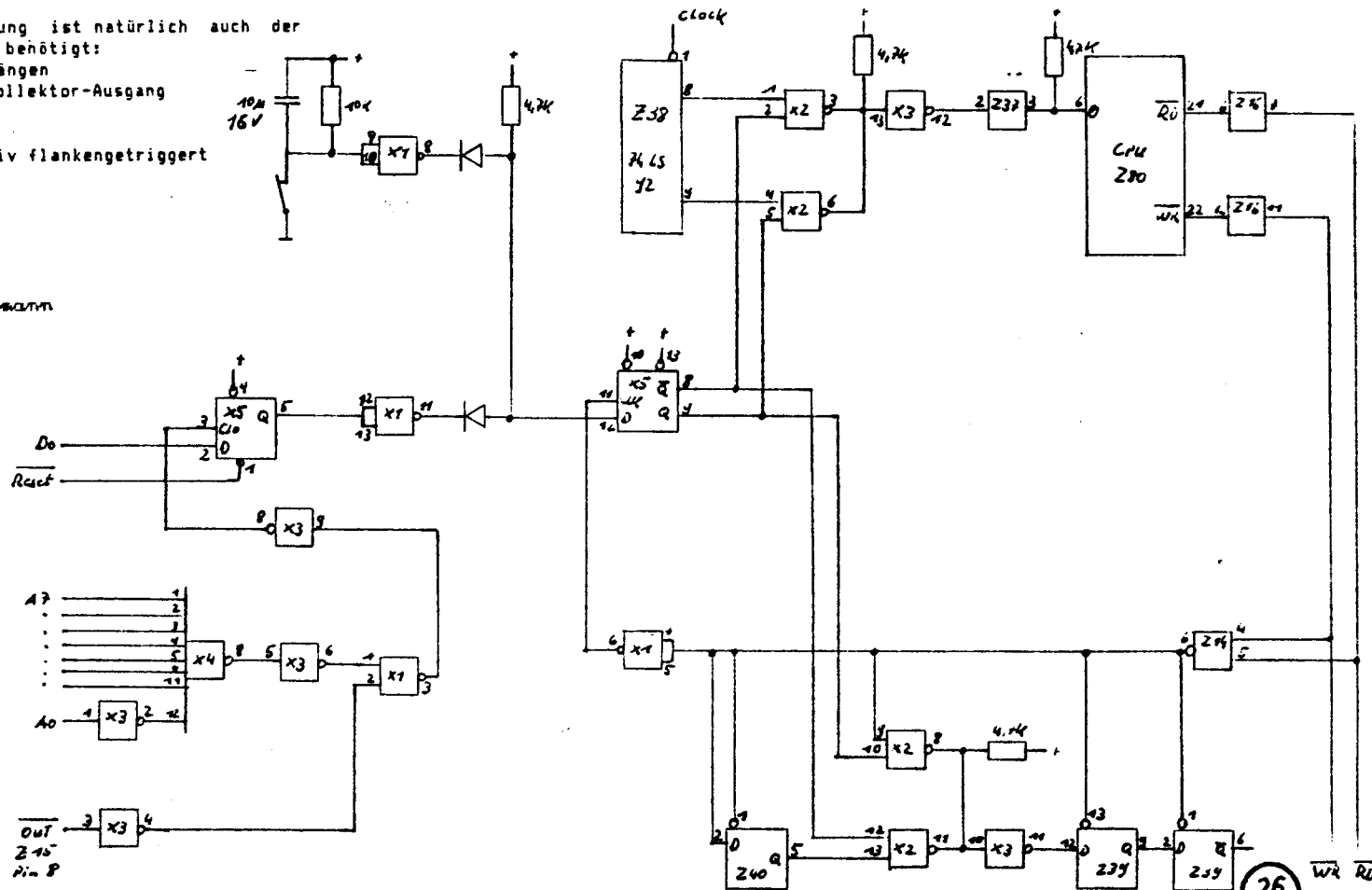
Auf dieser Adresse befindet sich der Wert 30H bzw. 48 dez. Er verzweigt das Programm, wenn keine SHIFT-Taste gedrückt ist. Dieser Wert, übrigens auf allen drei Systemen gleich, muß in 38H bzw. 56 dez. umgewandelt werden. Jetzt verzweigt das Programm bei gedrückter SHIFT-Taste, d.

Ausnahme der einen Adresse. Das EPROM sollte erst dann für das jeweilige ROM in das Gerät eingesetzt werden, wenn alle Adresseninhalte überprüft worden sind. Die meisten EPROM-Programmierer gestatten dies. Nach dem Einbau kann wie bei einer normalen Schreibmaschine mit der SHIFT-Taste groß- ansonsten klein geschrieben werden. Auf die Erkennung von Basic-Befehlen hat diese Umkehrung kei-

finden könnten. Weiterhin könnte man bestimmte Adressen des Betriebssystems abändern, um bestimmte Effekte nach dem Einschalten des Gerätes zur Verfügung zu haben.

Sie sehen, ein Blick in ein ROM-Listing lohnt sich.

Ich hoffe, ich konnte einige Anregungen bieten, und zeigen, daß auch in der Grundversion eines Computers mit oftmals geringen Mitteln einiges erreicht werden kann.



## Die Software schmilzt

Beim Begießen der Blumen im Raum mit der Datenverarbeitungsanlage hat es einige Wasserflecken im Teppichboden gegeben. Es ist Hochsommer und sehr warm. Eine neue Mitarbeiterin – mit den Feinheiten der EDV wenig vertraut – betritt den Raum, sieht die Flecken im Teppichboden, erleichtert und schreit: »Die Software schmilzt!«

### Thema: SUPERTAPE

Das im Info Nr. 7 (Seite 44-50) vorgestellte Bandaufzeichnungsverfahren SUPERTAPE ist sicher für einige Mitglieder des **CLUB 80** so interessant, daß sie sich entschlossen haben, das Programm abzutippen und die Platine nachzubauen. Dazu zwei Tipps:

1. Das Programm wird vom Heise-Verlag, der auch die Zeitschrift **C't** herausgibt für 7,- DM auf Kassette verschickt! Man kann sich also die Mühe des Eintippens und Fehlersuchens sparen. Wer zudem noch etwas Zeit hat, kann ja warten bis das Programm in der Clubbibliothek auftaucht (was sicher nicht allzu lange dauert).
2. Die Platine für die benötigte Zusatzschaltung wird für unter 20,- DM von verschiedenen Inserenten in der **C't** angeboten. Die Schaltung kann also auch von den in der Hardware nicht so erfahrenen Clubmitgliedern aufgebaut werden.

Einige Diskuser werden sich natürlich fragen, was man mit einem Tapeaufzeichnungsverfahren, sei es auch noch so schnell, soll. Die Antwort ist relativ einfach. Mit SUPERTAPE erhält man die Möglichkeit, Programme und Daten mit anderen (supertapefähigen) Computern auszutauschen! Weiterhin kann man sich damit vielleicht manche Backupdiskette sparen.

### Ein sicheres Plätzchen (für TRS80)!

In letzten Clubinfo wurde eine Schaltung veröffentlicht, mit der sich die Besitzer von VideoBene-Computern ein sicheres Plätzchen für Maschinenroutinen in ihrer Maschine sichern können. Die Schaltung ist für TRS80-User nur mit erheblichen Änderungen brauchbar und somit mit Vorsicht zu genießen. Wer sich trotzdem in seinen TRS80 etwas mehr RAM einbauen möchte, der möge noch etwas Geduld haben. Ich bin z.Z. dabei eine Schaltung aus einer älteren amerikanischen Zeitschrift (80 US) auszutesten, die speziell für TRS80-Computer 1920 Bytes zusätzlichen Speicher bringt. Die Schaltung und der zugehörige Artikel werden dann in nächsten Info veröffentlicht (wenn alles funktioniert!).

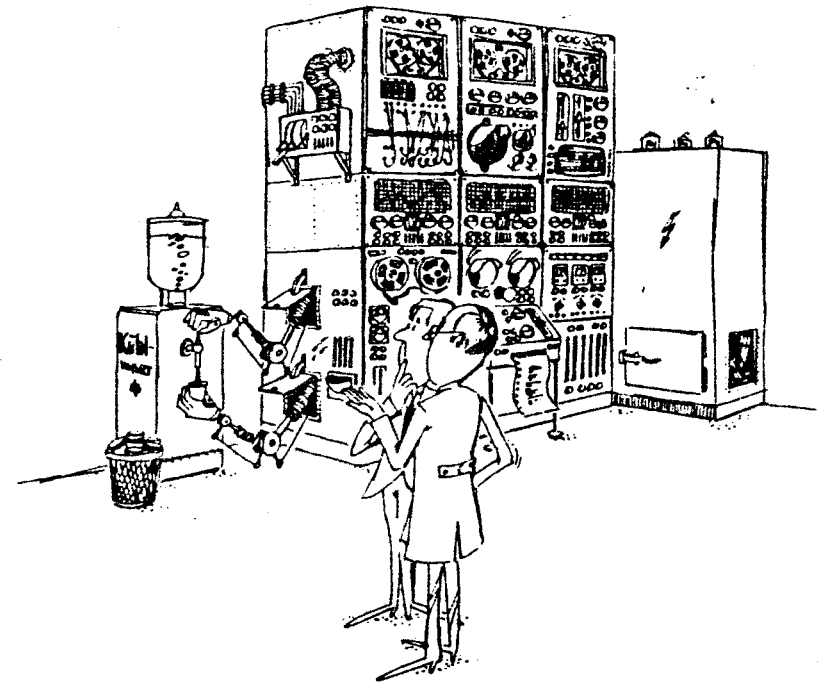
*Raimund Obermann*

Nicht wenige Computerfreunde schreckt die Erlernung der Maschinensprache bzw. Assembler deshalb ab, weil man dabei sehr viel mit verschiedenen Zahlensystemen umgehen muß. Die dauernde Umrechnung zwischen dezimaler, dualer, oktaler und hexadezimaler Darstellung von Zahlen macht die Arbeit mit dem Assembler oder Monitor nicht gerade leichter. Und selbst, wenn man garnicht allzu tief in die Maschine einsteigen will, bekommt man schon Probleme. Ein Beispiel: in einem BASIC-Programm wird eine Adresse angepökt (die Adresse ist natürlich dezimal angegeben). Um schnell festzustellen, was dieses PÖKE eigentlich macht, schaut man am besten ins rückrath'sche ROM-Listing. Hier sind aber alle Adressen hexadezimal angegeben (wie sich das gehört) und schon geht die Umrechnung los!

BASIC-Umrechnungsprogramme bieten kaum einen Ausweg aus dem Dilemma. Wer verläßt z.B. gern den Assembler und lädt ein BASIC-Programm, nur um eine Zahl umzurechnen? Eine wirkliche Lösung bieten eigentlich nur Taschenrechner mit integrierter Umrechnungsfunktion. Diese aber waren bis jetzt relativ teuer. Inzwischen jedoch hat die Firma CASICO mehrere technisch wissenschaftliche Taschenrechner zu einem durchaus erschwinglichen Preis herausgebracht (50 - 70 DM), die diese Funktion besitzen.

So hat mich mein kürzlich erworbener CASICO fx 115 nur 59,- DM gekostet. Er beherrscht nicht nur die Umrechnung aller benötigten Zahlensysteme sondern auch die Logikoperationen AND, OR und XOR sowie die Darstellung dualer Zahlen in sog. Zweierkomplement!

Meiner Meinung nach erleichtert ein solches Hilfsmittel den Einstieg in die Maschinensprache erheblich und hilft dem erfahrenen Assemblerprogrammierer erheblich Zeit zu sparen. Euer *Raimund Obermann*



»Donnerwetter! Das ist ja beinahe menschlich, nicht?«



EISENBAHN S 0126 A 00 /BAS --  
Maessiges Geschicklichkeitsspiel mit ebenso maessiger Grafik.

VIERGEWINNT D 0141 A 01 /BAS --  
Das Programm stellt das benoetigte Spielfeld zur Verfuegung und verwal-  
tet die Einwuerfe der Spieler, spielt jedoch nicht selbst!

VERLIES D 0142 A 01 /BAS --  
Adventure, bei dem man durch ein Verlies mit 10 Stockwerken irrt und  
dabei gegen Monster kaeampfen muss.

FUNKTIONSPLOTTER M 0143 A 02 /BAS --  
Dieses Programm plottet Funktionen hochaufloesend (400 oder 800 Punkte  
) auf EPSON MX/RX/FX - Drucker. Die Aufloesung ist sehr gut. Quelle:  
ELCOMP

ROBBER D 0150 A 01 /BAS --  
Ein mittelmuessiges Denk-/Geschicklichkeitsspiel mit maessiger Grafik.

KUNDESLIGATABELLE A 0158 A 01 /BAS --  
Bundesligatabelle fuer den Hausgebrauch in BASICCODE 2 geschrieben.  
Die Daten werden auf Kassette gespeichert (kann ungeschrieben werden).

PFLANZEN S 0159 A 01 /BAS --  
Dieses BASICCODE 2-Programm stellt eine grosse Menge Wissen ueber  
viele Garten -Blumen und -Gewuese zur Verfuegung.

NIMMSPIEL D 0161 A 02 /BAS --  
Ein sehr bekanntes Streichholzspielchen wird hier gegen den Computer  
gespielt. Er gewinnt (fast) immer, probiert's mal aus.

PROTON S 0162 B 01 /CMD --  
Dieses Programm realisiert eine Protokollfunktion fuer TRSDOS (aehn-  
lich ROUTE,DD,DD,PR) im NEWDOS80.

TASCHENRECHNER M 0268 A 01 /BAS --  
Ein Taschenrechner als BASIC-Einzeiler. Quelle: Computer Persoenlich

FOOND A 0269 A 03 /BAS --  
Aktienverwaltungs-Programm amerikanischer Machart. Quelle: Micro 80

DIGITALKURS S 0270 A 03 /BAS --  
Eine Einfuehrung in die Digitaltechnik in englischer Sprache. Sehr gut  
gemacht und vor allem fuer den Anfaenger interessant.

MCEDIT S 0271 A 03 /BAS --  
Ein besonders komfortabler Texteditor aus der MC 8/83

Funktionssucher M 0272 A 03 /BAS --  
Ein Programm, dass nach der Eingabe von Wertepaaren (z.B. aus einer  
Messreihe) die mathematische Funktion ermittelt, die diese Wertepaare  
unterliegen.

Wordchecker T 0273 A 03 /BAS --  
Dieses Programm dient zur Fehlersuche in ASCII-Texten.  
Quelle: Micro 80 8/84

Adressen-Text A 0274 A 50 /COB --  
Ein sehr komfortables Adressenverwaltungsprogramm, in CDRDL geschrie-  
ben. Das System enthaelt u.a. eine sehr gute Etikettendruckroutine.  
Autor: Harald Trapp Umfang: 3 Disketten.

H-DOS S 0275 B 99 /xxx --  
Ein von Arnulf Sopp sehr stark veraendertes und verbessertes 6-DOS.  
Features: Grafik-Hardcopy, Bildschirmsicherung, akustische Fehlersig-  
nale, Laufwerktest, DiskDatenEditor, CLS fuer HRG, uvm.!!!

Die beiden wohl wichtigsten und besten Programme bzw. Programmpakete, die sich in der Bibliothek des CLUB 80 befinden, sind das Adressenverwaltungsprogramm **ADRESSENT** und das Diskettenbetriebssystem **H-DOS**. Beide Programme verdienen es, hier noch einmal speziell erläutert und angepriesen zu werden.

1. Das **H-DOS** von Arnulf Sopp ist eine (erhebliche) Erweiterung des G-DOS der Firma TCS (Trommelschläger Computer Service). Der Arnulf hat nach einer Vereinbarung mit TCS das Recht, das G-DOS zu verändern und als sein **H-DOS** zu veräußern. Dies jedoch nur an Leute, die sich irgendwann einmal rechtmäßig ein G-DOS beschafft haben.

**H-DOS** besitzt gegenüber dem ursprünglichen G-DOS erhebliche Erweiterungen und damit Vorteile! Hier ein paar davon:

I. Die JKL-Funktion ist so geändert, daß Grafiken positiv oder negativ ausgedruckt werden können. Dabei wird gleichzeitig die Verzerrung, die normalerweise bei Bildschirmhardcopies auftritt, ausgeschaltet. Weiterhin ist JKL nun auch als DOS-Befehl implementiert!

II. Bildschirmspeicherung mit dem Dreitastenbefehl 345 oder dem Utilityprogramm GRA/CHD.

III. Akustische Signale bei DOS- und BASIC- Fehlern!

IV. Der DOS-Befehl LMT (Laufwerktest) ist, im Gegensatz zu einigen G-DOS-Versionen, wieder implementiert.

V. Der DOS-Befehl DDE (Diskdateneditor) ist erheblich erweitert. So kann man z.B. nicht nur an Dateisektoren sondern auch an allen Sektoren einer Diskette herumdoktern.

VI. CLS 6 löscht den Bildspeicher der hochauflösenden Grafik. Das gleiche passiert bei jedem Bootvorgang.

VII. Eine der wichtigsten Neuerungen ist der DOS-Befehl ID! Er ermöglicht die automatische Ermittlung der PDRIVE-Parameter unbekannter Disketten und erspart damit viel nervenraubende Sucherei bei fremden Disketten!

VIII. Weiterhin unterstützt das **H-DOS** voll die Hardwareerweiterung EG 64 MBA (Bankingbaustein zur vollen Nutzung der ab Baujahr 83 in alle Genie I und II eingebauten 64k-RAM) und gibt dem Benutzer gleich sinnvolle Anwendungen der gewonnenen 16k-RAM an die Hand (Spooler, Änderungen des BASIC-Interpreters usw.).

Alle Änderungen aufzuzählen würde den Rahmen dieser kurzen Vorstellung sprengen. Ich kann nur jedem G-DOS-Besitzer raten, mir eine G-DOS-Systemdiskette zu schicken und sich das **H-DOS** selbst einmal anzuschauen. Eine Anleitung für die Erweiterungen wird übrigens mitgeliefert!

2. Das Programmpaket **ADRESSENT** von unserem Cobol-Spezialisten Harald Trapp zeichnet sich vor allem durch die Verwendung sog. ISAM-Files zur Speicherung der Daten aus. Dadurch ermöglicht es einen sehr schnellen Zugriff auf einzelne Datensätze der Adressendatei, auch wenn diese schon eine erhebliche Größe erreicht hat. Neben den Funktionen "Anlegen, Ändern und Löschen" von Datensätzen, werden folgende Zusätze geboten:

1. Drucken der Stammdatenliste (komplett oder als Kurzliste)
2. Eingabe, Speicherung und Druck von Normetiketten
3. Druck von Etiketten aus der Adressen-Datei

Ein spezielles Programm dient der Sortierung der Stammdatei und beschleunigt somit nochmals das Auffinden einzelner Datensätze.

Das Programmpaket wird mit Anleitung geliefert. Wenn gewünscht, können auch die Cobol-Quelltexte mitgeliefert werden. Ein Cobol-Compiler ist nicht enthalten!!!

Wie ihr seht, haben wir hier wirklich zwei ausgezeichnete Programmpakete in unserer Bibliothek, die sich keines der Clubmitglieder entgehen lassen sollte. Ich möchte mich an dieser Stelle nochmals und im Namen aller Mitglieder des CLUB 80 bei den Autoren für ihre Arbeit bedanken!

*Karsten Obermann*

**Achtung:** Die beiden Programme können unter ihrem (normalen Namen bestellt werden!!!

# BÖRSE

WER HAT WAS -- WER SUCHT WAS --

Ich habe Interesse am EG 64 MBA. Kann es jedoch bei keiner mir bekannten Firma beziehen. Bei TCS konnte man mir nicht weiterhelfen. Wer weiß, wo man es noch beziehen kann?

*JÜRGEN NUCHERER*

FRAGEN, FRAGEN, FRAGEN ...

??? FRAGENKASTEN ???

Was bringt mir als Einsteiger der ECB-BUS der Hardware-Gruppe?

*WALTER PILLER*

Anmerkung der Redaktion

Hallo Walter, eine Antwort dazu möchte ich den Hardware-Profis überlassen. Näheres über den ECB-BUS ist ja sicher einen Infodartikel wert.

Mein GENIE II steht nur 2 Meter vom Radio entfernt. Leider ist vom Radio nur noch ein übles Rauschen zu vernehmen, wenn gleichzeitig der Computer läuft. Wer weiß Abhilfe?

*JÜRGEN NUCHERER*

CLUB 80 - Programmbibliothek : neue Programme

An dieser Stelle will ich euch eine ganze Menge neuer Programme vorstellen, die seit dem Erscheinen der letzten Liste in die Bibliothek aufgenommen wurden. Es sind vor allem Anwenderprogramme und Utilities in BASIC (auch kompiliert) und Maschinensprache. Die Programmbibliothek ist vor allem um zwei Programmpakete reicher geworden, die ich am Schluß dieser Liste genauer vorstellen werde. Ich danke allen Einsendern von Programmen für ihre Mitarbeit und hoffe, daß möglichst viele Mitglieder etwas für sie interessantes in der Bibliothek entdecken werden!

ERRORS                           S 0057 A 01 /BAS    --  
DN ERROR GOTO - Routinen fuer Diskfehler - Behandlung in BASIC-Programmen.

GRAFIK 3-D                       Z 0058 A 01 /BAS    --  
3-D Grafikdemo aus Micro 60 4/84.

Integer->Oktal                   M 0061 A 01 /CMD    --  
Umwandlung von Integerzahlen in Oktalzahlen.

JAHRESKALENDER                   S 0064 A 02 /BAS    --  
Das Programm druckt nach Angabe der Jahreszahl einen Kalender aus. Aus CP 6/82

OSTERN                            S 0065 A 01 /CMD    --  
Das Programm druckt das Datum fuer Ostern fuer die Jahre 1930 - 1990 aus.

Lohnberechnung                   M 0067 A 01 /BAS    --  
Das Programm errechnet Stunden-, Monats- und Jahreslohn aus verschiedenen moeglichen Angaben.

Lohntabelle                       M 0068 A 01 /BAS    --  
Das Programm erstellt eine Lohntabelle nach Eingabe der Wochenarbeitszeit unter Beruecksichtigung verschiedener Stundenloehne.

Pascal Dreieck                    M 0069 A 01 /CMD    --

Das Programm erstellt die rechte Haelfte des pascal'schen Dreiecks mit acht Reihen.

Roemische Ziffern                 M 0070 A 01 /CMD    --

Das Programm rechnet Integerzahlen in "roemische" Ziffern um.

Temperaturumrechnung             M 0071 A 01 /CMD    --

Das Programm rechnet Fahrenheit in Celsius um und umgekehrt.

Wochentag                         S 0074 A 01 /CMD    --

Das Programm gibt nach Eingabe des Datums den entsprechenden Wochentag aus.

SPOOLER                           S 0075 A 01 /CMD    --

Ein Druckerspooier aus der MC 6/82. Die Sourcecodefile wird mitgeliefert.

Titelgrafik                       S 0076 A 01 /BAS    --

Dieses Programm zeigt eine Unmenge von Moeglichkeiten, wie man seinem BASIC-Programm einen interessanten Vorspann geben kann. Die einzelnen Unterprogramme koennen uebernommen und leicht geaendert werden.

BALISTIK                          G 0078 A 00 /BAS    --

Ein mittelaessiges Geschicklichkeitsspiel.

STRICKMUSTERGRAFIK               Z 0079 A 02 /BAS    --

Dieses Programm laesst nach Eingabe zweier Parameter sehr schoene Grafikauster aus dem Bildschirm erscheinen.

GALTON                            S 0080 A 01 /BAS    --

Demonstration eines Nagelbrettes, auf dem Kugeln von oben nach unten fallen. Die Zahl der Kugeln kann vorgegeben werden, ebenso die Wahrscheinlichkeit, dass eine Kugel an einem Nagel nach rechts faellt.

SCHULDEN

B 0084 A 02 /BAS --

BASICCODE 2-Programm zur Berechnung von Ratenkrediten und sonstigen Schuldvertraegen.

DISKILL S 0095 A 01 /BAS --

Dieses Programm ueberschreibt alle freien Sektoren einer Diskette mit Nullen und loescht dabei die nach dem Killen von Dateien in der noch vorhandenen Daten.

EDITTEXT T 0096 C 03 /CMD --

Ein Texteditor fuer Model III aus Micro 80, mit Sourcecodefile.

Flowchartgenerator S 0097 A 02 /BAS --

Dieses Programm erstellt Flussdiagramme von als ASCII-Files abgespeicherten BASIC-Programmen. Aus Micro 80.

TAFY B 0098 A 01 /BAS --

Man muss mit moeglichst wenigen Versuchen ein Wort (5 Buchstaben) erraten. Die Worte stehen in DATA-Zeilen und koennen geaendert werden. Aus Micro 80.

DriveTimer S 0100 C 03 /CMD --

Mit diesem Programm koennen die Model III - Besitzer die Geschwindigkeit ihrer Laufwerke ueberpruefen. Aus Micro 80, mit Sourcecodefile.

MONDPHASEN S 0109 A 02 /BAS --

Dieses Programm ermittelt die Phase, in der sich der Mond zu einem angegebenen Zeitpunkt befindet. Quelle: Computer persoenlich 25/82 S. 37

FUENZEHNERSPIEL D 0110 A 04 /BAS --

Das Programm stellt das beliebte Spiel auf dem Bildschirm nach. Die Zahlen werden zufaellig verteilt. Auf Wunsch wird angegeben, ob und mit wievielen Zuegen das Problem loesbar ist.

ANGRIFF S 0112 A 00 /BAS --

Maessiges Geschicklichkeitsspiel.

PFERDERENNEN Z 0113 A 01 /BAS --

Wettspiel fuer mehrere Spieler. Maessige Grafik.

GUTHABEN B 0114 A 02 /BAS --

BASICCODE 2-Programm zur Errechnung von Sparguthaben und Rentensparvertraegen.

MAUMAU K 0115 A 02 /BAS --

Bekanntes Kartenspiel gegen den Computer.

MENUE S 0117 B 02 /BAS --

Direktoryorientiertes Menuprogramm in BASIC. Quelle: REST of 80.

BENCHMARKTEST S 0118 A 01 /BAS --

Eine Zusammenstellung von Benchmarktests, die sich vor allem zum Vergleich einzelner, auf dem TRS80/VG/Koatek lauffaehiger BASIC-Versionen und -Compiler eignen.

KURVENEN M 0120 A 03 /BAS --

Ein sehr gutes und umfangreiches Programm zur Kurvendiskussion fast aller Funktionen. Mit Schaubild, Ableitung (1. und 2.), Nullstellen, Extrema und Wendestellen. Autor: Herbert Alber

ABENTEUER D 0122 A 02 /BAS --

Adventure in deutscher Sprache aus Homecomputer 5/84.

TEXTEDITOR T 0123 A 03 /BAS --

Bildschirmorientierter Texteditor aus CHIP.

SUPERHIRN D 0124 A 02 /BAS --

MASTERMIND-Spiel. Der Computer gibt eine Zahl vor, die erraten werden muss.

## Teuer, teuer

Hochbetrieb im Rechenzentrum des Versandhauses. Ein Stoß Rechnungen ist zu schreiben.

Plötzlich bekommt die Anlage – schon ein älteres Modell – einen Tobsuchtsanfall. Erst stockt der Drucker, registriert unter wildem Geratter dieselbe Summe zweimal, frißt dann mit wahnwitziger Geschwindigkeit den Formularstapel in sich hinein und speit ihn mit Getöse wieder aus.

Während die roten Warnlampen aufblitzen und der Operator fassungslos auf den Rechnungswust blickt, sagt jemand hinter ihm: »Stehst Du, die Preise von heute können selbst eine Maschine rasend machen.«

## Bitte Ruhe

Ein Besucher betritt das Rechenzentrum. Überall im Raum herrscht betretenes Schweigen. Über dem Computer wird gerade ein Schild mit der Aufschrift »Bitte Ruhe« angebracht. Der Besucher fragt nach der Ursache dieses ungewöhnlichen Verhaltens. Man bedeutet ihm mit einer Finger-auf-den-Mund-Geste, leiser zu sprechen. Und der Chefprogrammierer flüstert völlig verwirrt:

»Der Computer antwortete auf die letzte Frage: BEI DEM LÄRM KANN KEIN MENSCH KLAR DENKEN!«

## Computer als Steuereintreiber

Der Computer des Aachener Finanzamtes schickte einem ahnungslosen Auto-Eigner 17 Mahnzettel ins Haus. 18 weitere Mahnungen lagen versandbereit, wurden jedoch von einem aufmerksamen Beamten zurückgehalten. Wären sie alle ans Ziel gekommen, hätte der überraschte Automobilist 10 056 DM rückständige Steuern nachzahlen sollen.

Des Irrtums Grund: Irgendwo im Zusammenspiel von Mensch und Mechanik war eine Sechs mit der Null verwechselt worden. Statt der Rückstände für 1967 versuchte der Computer die Steuern ab 1907 einzutreiben.

## CLLE 88 Mitgliederadressenliste

Name	Vorname	Straße	PLZ	Stadt	Telefon
Alber	Herbert	Alemannenstr. 20	7722	Niedereschach	07721 /7102
Baldes	Nans	Johann-Strauss-Str. 4	8825	Unterhaching	089 /6115175
Beckhausen	Wolfgang	Vuerfelser-Kaule 30	5860	Bergisch-Gladbach 1	02204 /62791
Boecker	Dieter	Lehweg 4	2930	Varrel 1	04451 /7640
Boeckling	Ulrich	Am Sonnenhang 11	5414	Vallendar	0261 /69522
Bozek	Hans Juergen	Gut Fachenfelde	2093	Stelle	
Buskowiak	Thomas	Eschersheimer Landstr. 257	6000	Frankfurt 1	069 /5601621
Dreyer	Gerald	Am Speiergarten 8	6200	Kiesbaden-Bierstadt	06121 /568218
Grajewski	Werner	Zedernweg 29	4220	Dinslaken	02134 /54573
Hallupp	Matthias	Stockumer Straße 405	4600	Dortmund 50	0231 /754413
Held	Manfred	Stinnerstr. 22	8835	Steinfeld	09144 /6563
Hermann	Klaus	Gartenstr. 22	7401	Pliezhausen	07127 /78024
Hummel	Anton	Schubertstr. 2	7612	Haslach	07832 /8209
Jablotschkin	Rainer	Thielkamp 29	4700	Lippstadt 8	
Kasper	Dieter	Zeppelinstr. 9	8952	Marktoberdorf	08242 /1630
Koenig	Hans J.	Heibelstr. 25	2000	Pinneberg	04181 /289444
Konrad	Josef	Anzengruberstraße 35	8038	Gröbenzell	08142 /8494
Kuhn	Eckehard	Im Dorf 14	7443	Frickenhausen 1	07822 /45417
Marx	Andreas	Mecklenburgring 48	6600	Saarbrücken	0681 /612983
May	Holger	Marienstr. 9	5768	Südern 2	02935 /1688
Neuder	Jens	Panoramastr. 21	7170	Michelbach/Bilz	0791 /42677 (dienstl. 44-450)
Obermann	Hartmut	Schwalbacher Str. 6	6209	Heidenrod 1	06124 /3913
Perschbach	Patrick	Waldstr. 52	5000	Koeln 91	0221 /872118
Piller	Walter	Ronenstrasse 8	CH-8835	Feusisberg	01 /7847418
Preuss	Lothar	Lautshof 13	2940	Wilhelmshaven	04421 /84247 (dienstl. 884-1)
Rank	Heinrich	Fruehlingstr. 2	8800	Fuerstenfeldbruck	08141 /3791
Retzlaff	Bernd	Kleiner Sand 96	2002	Uetersen	04122 /43551
Schaefer	Walter	Rathausstr. 4	8166	Miesbach	08925 /1631
Schneider	Manfred	Rheinkasseler Weg 11	5090	Koeln 71	0221 /787044
Schneue	Christian	Fliederweg 32	4000	Duesseldorf 31	0203 /748957
Schroeder	Gerald	Am Schuetzenplatz 14	2105	Seevetal 1	04105 /2662
Smerling	Frank	Tangstedter Str. 5	2000	Pinneberg	04101 /287284
Sopp	Arnulf	Wakenitzstr. 8	2400	Luebeck 1	0451 /791926
Spieß	Peter	Trugenhofenerstr. 27	8859	Rennertshofen	08434 /454
Stephan	Hans-Martin	Am Glasesch 9a (Postf. 1267)	4506	Hagen a.TW.	05401 /99585
Stevens	Peter	Postfach 6327	7800	Freiburg	0761 /35304
Trapp	Harald	Kranichstr. 46	4270	Dorsten 1	02362 /42497
Troesch	Eberhard	Altenessener Str. 414	4300	Essen 12	0201 /342324
Voigtlaender	Helm	Haselnussweg 30	6940	Wienheim	06281 /65241
Wagner	Alexander	Theresienstr. 21c	8224	Chieming	08664 /1500
Wagner	Guenther	Gartenstr. 4	8201	Neubeuern	08835 /3361
Wies	Jean-Claude	Harthweg 9	6609	Saarbrücken	0681 /582513
Wücherer	Jürgen	Brauneggerstraße 14	7750	Konstanz	07531 /29145
Zwickel	Walter	Lengfelden 122	A-5101	Bergheim	0843662/51130

Wegen Systemwechsel ausgetreten:

Sickmann Bernhard

Neuzugang:

Retzlaff Bernd

Vorstand

Kontaktadresse für  
Clubangelegenheiten  
Clubbücherei / Fundgrube  
Clubkasse

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

Programmbibliothek

Kontaktadresse

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

Redaktion

Kontaktadresse

Jens NEUEDER  
Panoramastraße 21  
7178 Michelbach /Bilz  
Tel.: 0791 /42877  
tagsüber 0791 /44-450

Adventure-Ecke

Kontaktadresse

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

Hardware

Kontaktadresse

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

Redakteure

dieser Ausgabe

Manfred Held \* Josef Konrad  
Jens Neueder \* Hartmut Obermann  
Arnulf Sopp \* Harald Trapp  
Günther Wagner \*  
sowie Artikel aus: Genie Data und  
Computer Persönlich

Bankverbindung des CLUB 80

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

Das INFO erscheint zweimonatlich.

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

# Schluß

Hallo Club-80er,

nun zum Schluß wieder einige Zeilen von der "Redaktion".

Zunächst einmal muß ich mich bei allen, die Probleme mit den Listings hatten, entschuldigen. Es wird nicht wieder vorkommen, daß solche Listings in der INFO erscheinen. Ich hatte es leider auch erst nach dem Druck bemerkt. Für diejenigen, die gerne eine Kopie vom Original haben möchten, bin ich gerne bereit -mit der nächsten INFO- einen Abzug mitzuschicken. Meldet Euch bitte.

Zu der vor Euch liegenden 6. Ausgabe unsres INFO muß ich leider mitteilen, daß ich wenig Informationen von Euch für das INFO bekommen habe. Ich hoffe, das dies nur in der beginnenden Ferienzeit oder endigenden Frühjahrs Müdigkeit seine Ursache hat. Ich würde mich freuen, wenn sich möglichst viele Mitglieder an der INFO beteiligen. Das INFO ist ja das Aushängeschild unseres Clubs, und sollte deshalb auch etwas darstellen. Da ich privat ziemlich eingespannt bin, ist es mir zur Zeit unmöglich, noch extra einige Artikel für das INFO zu schreiben, so gern ich das auch täte. Ich rechne also mit Eurer regen Teilnahme und mache mich auf Berge von Infomaterial gefaßt.

Ein weiterer Wunsch von mir wäre, daß Ihr bei der Erstellung Eurer Beiträge bitte darauf achtet, auf den DIN-A4-Seiten zwei oder drei Zeilen frei zu lassen. Ich brauche diesen Platz um einen Lochrand -auf der geleimten Seite der INFO- zum Einheften gewähren zu können.

Dies ist nun das 6. Clubinfo. Ich hoffe, daß es Euch genau so gut wie die 7. Ausgabe gefällt. Euren Briefen nach zu urteilen, hat es Euch gefallen. -- Danke --

Bis zum nächsten INFO

J. Neueder