

CLUB 80

Clubi nfo
der

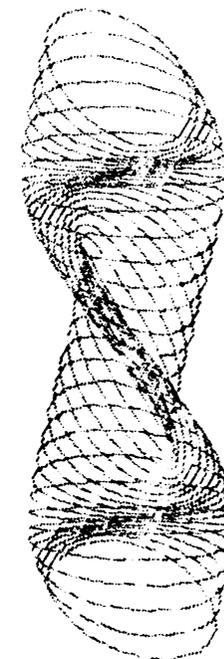
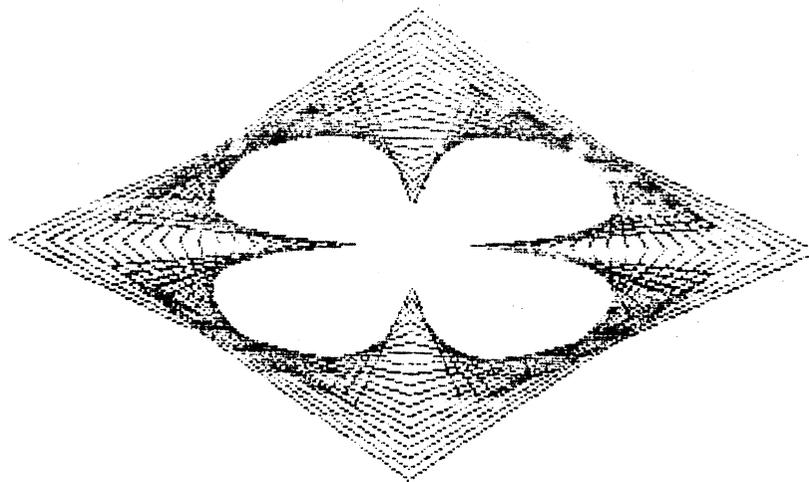
TANDY -

GENIE -

und KOMTEK -

ANWENDER

11. AUSGABE



KLEEBLATT, PENDEL
und WELTKARTE sind HRG-Produkte



Kontaktadresse : CLUB 80 / Günther WAGNER / Gartenstraße 4 / 8201 Neubulern

Tel.: 08935/3351 (18 - 20 Uhr)

Inhaltsverzeichnis

Seite:

Clubinternes

Der Vorstand informiert	01 - 03
An die "2. Hälften"	04

Software

nochmals INPUT	05 - 06
Files wie Sand am Meer	07 - 08
Adventure - Ecke	09 - 17
Programmierwettbewerb	19 - 20
Fehler bei BOOT II	21 - 22
Priazahl '39'	23 - 24
NEWDOS/80 + 10 (F,DR,LF,OUT)	25 - 30
Extended BASIC	31 - 32
NEWDOS/80 + 10 (I,P,DISK)	33 - 37

Hardware

Verbindend	39 - 40
ECB-Adapter für alle	41 - 46

Seite:

Börse

Wer hat was ???--Wer sucht was ???	47 - 48
--	---------

Sonstiges

BASIC-Vergleichstabelle	49 - 63
Murphy-Gesetz	65 - 66
Folgerungen aus Murphy	67 - 68

Programmiersbibliothek

Exklusiv	69
Das letzte Mal	69
Neue Programme	70 - 73
Unter neuer Verwaltung	74 - 75

Club - Bücherei

Kurzmitteilung	64
--------------------------	----

Die letzten Seiten

Impressum	75
Schluß	76
Adressenliste	letzte Seite
Kalender und Anleitung	am INFO-Ende

Liebe Mitglieder,

wir beginnen mit diesem 11. Clubinfo das Jahr 1986. Rückblickend auf das Jahr 1985 möchte ich sagen, daß es für uns Mitglieder des CLUB 80 ein sehr erfolgreiches war:

- unser erstes Clubtreffen war ein großer Erfolg; u.a. wurde dort auch die Satzung überarbeitet.
- der Jens übernahm das Clubinfo - er hat damit eindeutig die meiste Arbeit von allen Mitgliedern - er leistet hier eine hervorragende Arbeit. Die Übernahme durch ihn hat unserem Clubinfo wirklich gut getan.
- der Hartmut übernahm die Programmbibliothek. Er hat sich die große Arbeit gemacht, geschützte Programme auszusortieren. Er hat die Programmbibliothek wirklich gut verwaltet und trotzdem noch Zeit gefunden, sehr viele Beiträge zum Clubinfo beizusteuern.
- viele interessante Beiträge (große und kleine) konnten wir im Clubinfo lesen - gedankt sei hier allen Redakteuren, die sich die Zeit genommen haben oder sich getraut haben, etwas im Clubinfo zu veröffentlichen.

Ich möchte mich an dieser Stelle vor allem beim Jens und beim Hartmut bedanken - ohne deren Engagement und Idealismus wäre vieles nicht möglich gewesen im vergangenen Jahr.

Ich hoffe, daß wir auch 1987 auf ein gutes Jahr 1986 zurückblicken können - wir können das sicherlich, wenn wir alle zusammenhelfen, wenn jeder etwas zum Club beisteuert (außer dem Jahresbeitrag).

Aber nun zu den Finanzen.

Wir haben seit diesem Jahr drei Kassen; eine Hauptkasse und zwei Nebenkassen:

- Hauptkasse, wird von mir verwaltet
- Nebenkasse vom Jens Neueder für seine Ausgaben als Betreuer des Clubinfos
- Nebenkasse vom Hartmut Obermann als Betreuer der Programmbibliothek.

Der Kassier wird jeweils beim Clubtreffen entlastet. Beim Clubtreffen kann auch jedes Mitglied Einsicht in die 3 Kassenbücher nehmen. Deswegen sei hier nur eine sehr kurze Übersicht angegeben, wobei die drei Kassen zusammengefaßt sind:

ZUSAMMENSTELLUNG

Übertrag aus dem Jahr 1984		- 114.07	DM
Einnahmen:	Clubbeiträge und Aufnahmegebühren	+1615.00	DM
	Nachzahlung 10 DM	+ 400.00	DM
	Unkostenbeitrag für Clubinfos	+ 122.50	DM
	Sonstige Einnahmen	+ 70.00	DM
Ausgaben:	Briefmarken	- 245.20	DM
	Photokopien	- 524.70	DM
	Verbrauchsmaterialien, Sonstiges	- 185.03	DM
	Disketten, Kopien, Porto etc. (Hartmut)	- 142.55	DM
	Info 7-10 mit sämtlichen Nebenkosten (Ausgaben vom Jens Neueder)	- 544.65	DM

Somit schließen wir das Jahr mit einem Überschuß in Höhe von ca. 450 DM ab.

Dieser Überschuß wird laut Clubsatzung u.a. zum Ankauf von Büchern und Programmen verwendet. Der Hartmut hat von einem Verlag einen Sonderprospekt mit günstigen Preisen erhalten. Er hat mich auf die günstigen Preise der Bücher aufmerksam gemacht, und wir haben beschlossen, 3 Bücher zu bestellen. Ferner liegt ein Verkaufsangebot über Bücher von unserem ehemaligen Mitglied Mathias Hallup vor. Wahrscheinlich werde ich vom Mathias Hallup für den CLUB 80 zwischen 9 und 19 Bücher zu einem Höchstpreis von 250 DM abkaufen. Zum jetzigen Zeitpunkt kann ich hierzu noch nichts näheres mitteilen - also bitte bis zum nächsten Info warten. Auf alle Fälle werden die Bücher nur nach Rücksprache mit dem Jens und dem Hartmut angeschafft. Ich hoffe, daß wir (der Jens, der Hartmut und ich) im Sinne der Mehrheit der Mitglieder entscheiden.

Zu den Ausgaben ist noch etwas zu sagen: Durch den günstigen Vervielfältigungspreis, den wir über den Peter Spieß erhalten, konnten wir die Ausgaben erheblich vermindern. Eine weitere Einsparnis haben wir durch geringere Portokosten, da wir die gebundenen Clubinfos (die ab Nr. 7) als Büchersendung verschicken können. Nur so konnte überhaupt ein Überschuß in Höhe von etwa 450 DM entstehen. Ich möchte nicht versäumen, an dieser Stelle auch dem Peter Spieß meinen Dank für seine Arbeit auszusprechen.

Einige haben den Jahresbeitrag noch nicht bezahlt - bitte daran denken falls vergessen!

Mittlerweile ärgere ich mich auch über 4 Mitglieder, die bis jetzt noch nicht Ihre 10 DM Nachzahlung bezahlt haben. Dieses Quartett bekommt dieses Info erst nach Bezahlung der 10 DM.

CLUB TREFFEN

Auch hier möchte ich einiges anmerken. Bis Weihnachten sind bei mir erst 18 Anmeldungen eingetrudelt. Davon nehmen 6 Mitglieder am Treffen nicht teil. Es nehmen also bis jetzt 12 Mitglieder am Treffen teil. Von diesen 12 Mitgliedern werden 4 Mitglieder von der Frau bzw. Freundin begleitet.

Mir ist klar, daß viele zum jetzigen Zeitpunkt noch nicht sagen können, ob sie am Treffen teilnehmen können oder wollen. Dann schickt mir aber bitte trotzdem die Anmeldung zu und schreibt drauf, ob Ihr teilnehmt oder ob nicht und wann ev. Ihr näheres mitteilen könnt. Ich brauche unbedingt von den anderen 27 Mitgliedern, die bis jetzt noch nichts bezüglich Clubtreffen verlauten lassen haben, entsprechende Mitteilungen.

Auch die Mitglieder, die am Treffen nicht teilnehmen, müssen mir dies unbedingt mitteilen!

Ihr denkt hoffentlich alle daran, daß auf dem Clubtreffen ein neuer Vorstand zu wählen ist. Mit dem alten Vorstand (daß heißt mit meiner Person) bin ich selbst nicht zufrieden. Zu wenig pflege ich meiner Meinung nach die Kontakte mit den anderen Mitgliedern; viel zu selten denke ich an meine Aufgaben als Clubvorstand.

Warum? Ich habe wegen dem Studium (Diplomarbeit und Hauptdiplomprüfungen) viel zu wenig Zeit und außerdem nicht mehr so das Interesse am Amt des Vorstandes. Viel zu sehr stehen bei mir einfach andere Interessen (berufliche und private) im Vordergrund.

Ich stehe also mit Sicherheit bei der Neuwahl als Kandidat nicht mehr zur Verfügung. Ich hoffe, daß sich im Club Mitglieder befinden, die sich bereit erklären, das Amt des Vorstandes zu übernehmen. Sollte ein Mitglied bereit sein, für das Amt zu kandidieren und kann er am Clubtreffen nicht teilnehmen, so muß er seine Kandidatur unbedingt bei mir schriftlich melden, damit wir ihn bei der Neuwahl auch berücksichtigen können. Es ist hoffentlich jedem von Euch klar, daß wir auf dem Clubtreffen unbedingt einen neuen Vorstand wählen müssen - schön wäre es, wenn wir mehrere Kandidaten hätten. Wer Interesse am Amt des Vorstandes hat, sich aber über seine Aufgaben unklar ist, sollte sich bei mir melden.

Ich hoffe, daß ich jetzt nichts vergessen habe. Bis zum nächsten Mal

Euer

Günter Wagner

An alle computergeschädigten Ehefrauen und Freundinnen!!

Wie Ihr sicherlich wißt ist vom 11. bzw. 12. - 13.04.85 das Clubtreffen unserer Männer. Ich würde mich aus diesem Anlaß sehr freuen, wenn Ihr sie begleiten würdet. Gemeinsam könnten wir einiges unternehmen, z. B. einen Stadtbummel durch Limburg (das sehr schön ist), Wiesbaden oder Rüdesheim. Außerdem gibt es im Hotel ein Schwimmbad. Zusätzlich hat sich unser Clubmitglied Walter Piller bereiterklärt, seine Erfahrungen im Brotbacken an uns kostenlos weiterzugeben. Wer Interesse hat, melde sich bitte bei mir. Tel. 06124/3913.

Ihr seht, hier kann man vieles unternehmen. Vor allem können wir die Gelegenheit zu einem Gedankenaustausch nutzen.

Ich hoffe auf Euer zahlreiches Erscheinen und verbleibe mit freundlichen Grüßen

Jutta Obermann



Herr Doktor, hier ist meine Frau - sie bildet sich ein, ein Computer zu sein!

Vieles gibt's wie Sand am Meer -
zum Beispiel: Wenn vieles als Files auf fast gefüllten
Floppydisks wohlfeile Felder fand und viele Fehler
anfallen, weil vielleicht feinfühligere, frei formatierte
Informationen fehlen!

Jach kurze Luft holen: Ich meine, wenn man sich vor
lauter Filenamen, deren 8-11stellige Abkürzungen ihren
Inhalt mehr verschleiern als enthüllen, nicht sehr ausbeutet
- und wenn man sich in der sandreichen Menge Disks nicht
mehr zurechtfindet, weil es eben immer mehr, eben weil es
ein Meer zu speichern gibt -
dann muß irgendwas her, was hilft.

Das kann eine Datenbank sein. Muß nicht.
Günther gab uns in Heft 9, Seite 4, eine Anregung unter dem
Titel "Directory lesen" und legte die Bearbeitung eines
Unterprogramms aus BASIC vor, das sich den/die/das Directory
direkt aus DIR/SYS holt und unsere sorgfältig hingehaltenen
Variablen damit füllt.

Ferner hat er, auch nestlige "Leistungen" mitzugeben.
Deshalb scheue ich mich nicht, den bescheidenen Gebrauch,
den ich von seinem WP gemacht habe, indem ich meinen
"DISCAT" damit baue, hier als Anwendungsbeispiel zu
veröffentlichen.

Zeilen 1000-1130 stammen aus der Vorlage. Diese machen -
nach dem nötigen Vorspann - den Auszug aus dem/den/dem
Directory (ich stehe mit der "Antikultierung" der engl.
termini technical genau so auf dem Kriegsfuß wie Ihr und
gebe daher meist allen drei Antikultern das gleiche Recht).
Danach kann man wählen (warum "optional"?), ob man die
Filenamen schön stübentlich auf dem Bildschirm auszugeben
oder auf Papier gedruckt haben möchte.
Und im letzten Fall gibt's die Möglichkeit, dem Filenamen
einen hinreichend langen Kommentar zuzuordnen - das war's
schon!

Das Listing wurde übrigens mit dem "PAGE"-Befehl ausgedruckt.
Dieser hat seine Tücken, über die man eigentlich hier auch
einmal berichten sollte (vielleicht ein andermal). Hier
sowie: wer's abtippt, muß wissen, daß an jeder Zeilenende
ein "Delimitter" zu denken ist, zu Deutsch: ein
Syntax-Begrenzer.

Das ist nun allerdings beim editieren aus Bremen (1) und 2 in
Lamen sind selten), die Betonung liegt vielmehr auf der
zweiten Silbe. Man denke sich also am Ende jeder Zeile
entweder ein ":" - angenommen nach der jeweils letzten
Zeile vor der nächsten Zeilennummer - oder ein ";" -
welches von beiden, Aja, da muß man eben aufpassen; und
deshalb finde ich "FILE" zwar "schön", aber nicht sehr
praktisch... Was meint Ihr?

Bis demnächst! Euer Klausot... →

```

10 CLS
   CLEAR 2500
   DIM A$(200)
   PRINT " Directory lesen und verarbeiten (einschl. unsichtbarer Files)
15 REM      >>>> (C) Klausot Mühlenbein * Dezember 1985 <<<<<
20 PRINT STRING$(63,"*")
30 PRINT
   INPUT "Welches Laufwerk ";DR
40 GOSUB 1000
45 PRINT
   D$ = ""
   INPUT "Directory-Ausgabe auf Monitor oder auf Drucker <M oder ENTER>";D$

P:
46 IF D$ = "M" OR D$ = "m"
   THEN GOTO 50
   ELSE INPUT "Mit Kommentaren <J oder ENTER> ";J$
       GOTO 1500
50 CLS
   FOR I = 0 TO J STEP 45
   FOR K = 0 TO 44 STEP 3
   FOR L = 0 TO 2
55 IF A$(I + K) = ""
   THEN END
60 PRINT USING "##";I + K + L + 1;
65 PRINT ") "A$(I + K + L);"      ";
70 NEXT L
   PRINT
   NEXT K
80 INPUT
   NEXT I
   END
1000 ON ERROR GOTO 1130
1010 DR$ = "DIR/SYS:" + RIGHT$(STR$(DR),1)
1020 OPEN "R",1,DR$
1030 FOR I = 3 TO LDF (1)
1040 GET 1,I
1050 FOR II = 0 TO 7
1060 FIELD 1
      (II * 32) AS D$
      1 AS A$
      4 AS D$
      8 AS FS$
      3 AS FE$
1070 IF NOT (CVI (A$ + CHR$(0)) AND 208) = 16
   THEN GOTO 1100
1080 A$(J) = FS$ + " " + FE$
1090 J = J + 1
1100 NEXT II,I
1110 CLOSE
1120 RETURN
1130 RESUME NEXT
1500 CLS
   FOR I = 0 TO J - 1
1520 IF J$ = "J" OR J$ = "j"
   THEN PRINT "KOMMENTAR FÜR "A$(I);
       PRINT "      (Nur bis Ende Bildschirmzeile)
1525 INPUT " ";K$
1530 LPRINT I + 1;" "A$(I); TAB(20);K$
   NEXT
   END

```

Noch einmal: INPUT

Wie Ihr ja inzwischen gemerkt habt, stehe ich mit allem Assembler-Kram auf Kriegsfuß. Da tat die "Entwanzung" des DOS-Königs Arnulf schon not und ich bin auch sehr froh darüber, daß er meinen Artikel mit seinem Wissen richtiggestellt hat. Aber damit die Überwacher des einzig echten und richtigen Assembler-Codes auch weiterhin was zu tun haben, habe ich mich wieder in diesen undurchdringlichen Dschungel begeben und mir in meinem stillen Kämmerlein eine neue Frechheit ausgedacht. Wie Ihr alle Eurem Gedächtnis entnehmen könnt, gibt es in der Clubbibliothek zwei Verbesserungen des INPUT-Befehls. Eine stammt (ich muß es zu meiner Schande gestehen) von mir, die andere ist mir noch nicht vor die Füße gefallen. Ich hoffe also, daß meine dritte Lösung nicht mit der zweiten übereinstimmt. Wenn doch: Sorry! Isvinitje würden die Russen sagen (leider kann meine Textverarbeitung noch kein kyrillisch). Also: es gibt beim INPUT meines Erachtens zwei Probleme. Erstens kann man die Eingabe nicht auf eine bestimmte Länge beschränken. Zweitens löscht es die folgende Zeile. Nun gibt es im ROM eine Routine, die INPUT mit benutzt. Sie fängt bei 05D9H an. Sie erwartet zwei Eingangsparameter: Im Register B muß die maximal erlaubte Länge stehen; im Doppelregister HL die Anfangsadresse des Buffers, in den die Eingabe geschrieben werden soll. Außerdem erscheint sie noch auf dem Bildschirm.

Nun schreiben wir uns ein kleines Assembler-Programm mit vier Befehlen:

```
LD      B,Länge
LD      HL,Buffer
CALL   05D9H
RET
```

Für die Länge setzen wir Beliebiges von 1 bis 255 ein. Der Buffer ist schon ein größeres Problem, aber nicht unüberwindlich. Wir müssen nur wissen, was wir haben wollen, nämlich einen String mit der Eingabe, und die hervorragende Übersetzung von "Basic Faster and Better" unseres Generaldirektors. Da finden wir ein Superfeld mit diversen Erläuterungen, die Ihr da nachschlagen könnt. Das Endergebnis sieht so aus:

```
10 DEFINT A-Z 'alles in INTEGER
20 DIM US(4) 'unser Superfeld
30 A$=STRING$(10,32) 'statt '10' beliebige andere
40 'Länge, die größer als die Eingabe ist
50 X=0
60 PRINT$128,; 'Wo soll die Eingabe beginnen?
70 U(0)=1030:U(1)=8448
80 U(2)=CVI(CHR$(PEEK(VARPTR(A$)+1))+CHR$(PEEK(VARPTR(A$)+2)))
90 U(3)=-9779:U(4)=-14075
100 DEFUSRO=VARPTR(U(0)): X=USRO(0)
110 IF INSTR(A$,CHR$(13))>0 THEN A$=LEFT$(A$,INSTR(A$,CHR$(13))-1): PRINT A$
120 END 'RETURN ODER AUSWERTUNG VON A$
```

Bei diesem Beispiel ist die Eingabe auf 4 Zeichen beschränkt. Weiterentwickeln müßt Ihr die Sache schon selbst.

Einige Dinge sind zu beachten:

-erstmal "Basic Faster and Better" ansehen!

-A\$=" " geht nicht, weil die Zeile nach der ersten Eingabe ungefähr so aussieht: A\$="123"

Hier spielt das CR eine Rolle, das als letzter Buchstabe in A\$ auftaucht. Bei der nächsten Eingabe wäre das Ergebnis falsch, wenn weniger Buchstaben eingegeben würden, denn das "123" bleibt stehen. Also: A\$=STRING\$(10,32).

-die Eingabe kann auch durch Break abgebrochen oder mit einigen Kontrollcodes verzerrt werden. Vorsicht ist geboten, wenn kein CR in A\$ zu finden ist. Dann ist an der Sache was faul.

-wenn nur CR eingegeben wird, sollte die Anwenderroutine dies berücksichtigen. Bei obiger Programmierung bleibt nur A\$="" übrig. Keine sehr aussagefähige Eingabe.

Die Vorteile dieser Eingabe habt Ihr natürlich alle schon erfaßt und somit brauche ich mir keine ausdenken. Auf jeden Fall haben wir die Genugtuung, dem alten ROM wieder ein Schnippchen geschlagen zu haben. Ist doch schon was!

Fröhliches Tippen und Ausprobieren wünscht
Gerald Schröder

*Anmerkung:
Bei dieser Lösung wird die folgende Zeile noch gelöscht.
Um dies zu verhindern, müßte ein Filter in die Ausgabe 0033H
konstruiert werden. Wenn Ihr vorher einen Bildschirmeditor
(SEDIT etc.) ladet, wird die folgende Zeile nicht gelöscht!*

Adven-80

An Advanced Adventure Development System

For those not familiar with Adventure, it is a game which could simply not have existed before the age of the computer. Imagine reading a story tale where you follow the hero (or heroine) in a fantasy world where anything can happen. The difference between Adventure and a simple story book is that you can be the hero and be instrumental in affecting the outcome of the story, rather than being simply a passive reader.

More specifically, in most Adventures, you are in control of an "entity" which acts as your eyes and ears throughout the story. You can look, listen, manipulate objects, collect treasures, move in whatever direction you want, in fact do anything you want within the confines of a limited universe maintained by the computer.

Conceivably, some day, 3-dimensional tactile dream-worlds will become possible as technology progresses, perhaps coupling computer technology with some form of video disk system; but for the present, Adventure and similar games provide us with the first simple examples of this exciting concept.

This article is probably the first to be published of its kind, in that it provides a complete package for creating Adventures. The code presented here is a *completely general-purpose adventure "interpreter"* together with a BASIC program, designed to allow you, the reader, to create a database for your own Adventures with only the vaguest knowledge of assembly language programming. The code is not intended to be clever; experienced programmers can no doubt quite drastically improve upon the techniques used; however, it does work quite well and I would appreciate any feedback on possible improvements.

In Brief

ADVEN-80 is a program wherein the user resides in one of many LOCATIONS. A LOCATION may have the property of being LIGHT or DARK, and usually is associated with several alternative EXITS. The player moves from one LOCATION to another, and in any LOCATION he may find an OBJECT or OBJECTS. Objects have DESCRIPTIONS, and may optionally have DETAILED DESCRIPTIONS as well. Unless specifically mentioned an object may be TAKEN or DROPPED or LOOKED-AT or otherwise manipulated within the confines of the facilities of the MATCHW table. Each object may only be in one location at any give time (or may be worn or carried as applicable). The object of the game is usually to obtain some form of TREASURE or achieve a SCORE or so many points.

Of special interest is the LINE PARSER, whose job it is to take in whatever the player enters on the keyboard, and attempt to make some sense out of it. In simplified form, the parser in ADVEN-80 takes in a line of user entry, and looks at each word in turn to see if that is a known word. If so, a token entry (somewhat like the one-byte tokens used in some BASICs) is made in a separate buffer; if not, no entry is made, and the next word is checked. When the entire line has been

checked over, the token buffer is scanned for words such as "THEN" and "AND" and if found, the line is split, leaving all code after the "AND" for the next time round, in lieu of the player's next entry. This allows the use of multiple statement lines.

Generally, user input breaks down into simple two word sequences after irrelevant words have been discarded. A sentence such as "PLEASE GET THE LAMP AND LIGHT IT" breaks down into two operations, "GET LAMP" and "LIGHT LAMP." The program treats "IT" as the last object referred to. A table is scanned for such word sequences and appropriate action is taken, scores can be maintained, and the program has special facilities to accommodate scoring.

Adventure-type games for microcomputers have been around for some time and several Adventures have been made available for the more popular systems' although to date, very few have had anywhere near the capabilities to handle user input as had the original "BIG SYSTEMS" Adventure which many of the major Polytechnic computers have access to.

The Original Adventure for these systems was written in Fortran and required considerably more memory than that available on present-day microcomputers, and the program itself was inflexible (i.e., the creation of a new game would require the re-writing of the Fortran program as the database was an integral part of the overall structure).

ADVEN-80 is a new Micro-Adventure system, written in machine language. Consideration has been given as to obtaining the maximum usage out of the limited memory capacity of currently available small computers.

This Adventure system, written for the Z80 microprocessor, is a full-feature, machine-code Adventure utilizing the powerful functions of the above. It is partly because of this that the game runs at very high speed, with a minimum memory requirement (a simple Adventure could be written in well under 16K).

Because of the speed of the Z80, it was possible to use an extremely simple database layout, and the use of an intermediate translator accounts for high readability of the source listing. In reading the following, an elemental knowledge of assembly language programming in Z80 microprocessor is considered useful but not essential.

Tables

OBJLOC The format of this table, containing <object no./location> pairs is simple. The pairs are contiguous with a zero to end of the table. Several location numbers have special meanings and these are listed below.

Loc. 255 Object is non-existent. Cannot be found.
Loc. 254 Object is currently being carried.
Loc. 253 Object is currently being worn.

Finally, location 252 is special in that any object currently in this location is one of a pair of objects which together form a TWIN-STATE object. A typical example of this is a LAMP. The lamp may be LIT or UNLIT and hence it is necessary to have two versions of the same object.

This method is probably the simplest way of implementing a twin-state object, and in the table these two forms should always be concurrent. In the example of the lamp, the two states could be, say, LAMP and ULAMP. If the word LAMP is made accessible, then this must appear FIRST in the table. This is very important, first in ascending order; i.e.:

```
OBJLOC /* Table name. These are mandatory. */
LAMP 45 /* object 34, a lit lamp, is in location 45 */
ULAMP 252 /* object 35, an unlit lamp, is hidden from view. */
ENDTABLE /* end of table */
```

As you can see from the above table, if the scanner begins at the start of the table, and then finds a match with object LAMP, in order to blow out the lamp, the two LOCATION entries in the table are simply swapped over.

Object ULAMP was equated only and *must never* be found by the user-input encoding section, although CONDITIONS such as OC or OP can access both versions.

If a user says GET LAMP, and there are no conditions, the system will look for LAMP. If this is at location 252 then the pointer will be incremented automatically to point at the ULAMP entry. Hence the user will be given whichever version of the object is *not* at location 252. (Assuming that the other version is PRESENT or being carried or whatever.)

Inventories simply look at the location numbers and if a match is found, then a description is printed. The same applies during the initial description, where objects contained within the current location are described.

OBJDES This table containing <object no./MSG no.> pairs is closely related to the above OBJLOC table. During "I can see..." type routines, the table OBJDES is scanned in order to find objects contained within a given location, be it the current location or those reserved for objects being carried or worn...

When the object is found, the number of the object is used to scan through the left side of OBJDES, and when an object match is found, the description number, which refers to an entry in the UMESS table, is used to produce a description of the required object. The OBJDES table need be in no particular order, and no ZERO terminator is required as we are not searching for a non-existent object. In the sample database the ENDTABLE macro is shown, but this is just for neatness, it is not essential here.

OBPROP This table describes the things that can be done to an object, assuming you wish to limit that object to certain actions. I.e., you would not expect the player to be able to TAKE THE STATUE if the statue weighs a ton...

```
OBPROP
STATUE LOOK,LISTEN,FEEL
DOOR LOCK,UNLOCK,LOOK
ENDTABLE
```

Objects not in the table are considered to have universal attributes (within the limitations of the MATCHW table).

OBDSK This table finds use for those objects you would like to enable a close inspection of. Any object not appearing in the table will be described as ordinary looking.

```
OBJDSK
LAMP 'Looks like an old miner's lamp to me.'
ENDTABLE
```

TABKEY This table is used by the line parser. As the user input line is parsed, 4-byte word sections are put into a temporary buffer. (Words shorter than 4 characters long are left-justified and padded with spaces.) TABKEY is used to replace any known words with a single byte code. Unknown words are scrapped.

TABKEY is composed of the following. The first byte is KEY number. Following that are groups of 4 bytes with the last 4-byte group having the 8th bit set on the last character. If any of the 4-byte groups match that in the buffer, then the KEY byte is put into a buffer for later use. To signify the end of the table, ZERO is used as a KEY byte.

When the sentence has been completely scanned, a buffer called CODBUF will contain the single byte KEY replacements to all the words in the user input that were matched successfully. A ZERO will indicate the end of this buffer. Hence the sequence N "N NORTH" will allow user inputs of "N" or "GO NORTH" to be placed in CODBUF as the single byte representing the key N.

A special byte which may end up in CODBUF is THEN. This is currently set at 250 decimal and is used for batching of commands, so that the user may say, for example "GET THE AXE THEN LOOK AT IT." The IT being taken care of automatically by a section of code that converts occurrences of IT into the last mentioned object.

KEYSETS are actually entered in macro form; i.e.:

```
BEGIN 'BEGISTAR'
```

Another example of this is LOOK "LOOKPERUEXAM" where LOOK is the word that will be equated, and will match LOOK, PERUSE and EXAMINE.

MATCHW This table is the one that looks at user input after it has been encoded into a byte sequence. The table is scanned until a pair of bytes corresponding to those in the CODBUF area are found. If there is no match then a default message is printed to indicate that the program found no sensible matches. Otherwise, a set of CONDITIONS is checked, and if all of these are TRUE, then a set of actions is taken.

The format is as follows:

```
<2 byte key><cond><255><actions><255><either ZERO or more keys>
```

Where cond = either no entry or any amount of conditions and arguments.

Where actions = any amount of actions.

Obviously from the above, 255 is used as a delimiter and so must not occur within either the conditions or the actions. However, 255 may be used as one of the two KEY bytes to tell the program to match up with any entry, i.e.:

```
<23,255> If CODBUF contains 23 followed by anything then the set will be accepted.
```

```
<34,0> In this case, CODBUF should contain 34 ONLY.
```

Again, simple equates are used to make life easy. Typical table entries are shown below, together with an explanation of each.

```
MATCH GET,ANY
TEST OH,INP
ACTION TKE,INP,OK
```

In the above example, if the user enters, say, GET LAMP then the system will test to see if the object is in the current location (OH) and if so, will allow the taking of the LAMP and will then print an OK message and exit the table, ready to more user input.

```
MATCH LOOK,NULL
ACTION :A,EX
```

If and only if the user enters the single word LOOK, the system will unconditionally Look Around, then EXIT the

by Peter D. Scargill

Peter D. Scargill, 24 Percy Park, Tynemouth, Tyne & Wear, Great Britain.

HEFT
11
January
1985

10

```

MATCH ANY,ANY
ACTION PRM,37,EX

```

This entry should be put at the end of the MATCHW table. This will match ANY user input, and print message 37 from the MSG table, which would be, for example, "I don't know what you mean."

```

MATCH ON,LAMP
TEST OC,MATCH,OC,ULAMP
ACTION SWP,LAMP,SWP,MATCH,OK

```

On the previous page we have a more complex example. If the user enters ON LAMP or LIGHT THE LAMP, etc., and then only if the MATCH and the ULAMP (unit lamp) are being carried, then swap the MATCH for its USED partner, and swap the ULAMP with its LIT partner.

Notice that no attempt was made to execute the instruction SWP,ULAMP. This would have been disastrous as the ULAMP is the SECOND entry in the table, and you must only SWAP the FIRST entry (i.e., the main entry), as the command SWP FRED looks for the entry FRED in the OBJLOC table, and swaps its location with that of the next entry in the table. That is why the user-accessible version of the object must be entered into the table before the alternative form.

Messages

MSG This is a general-purpose message table, containing messages such as "I can not see anything" and "ok.". These messages include directives and all general statements. The format is extremely simple:

```

0 message sequence 0 message sequence 0

```

The sequence is that of a zero byte followed by ASCII text until the next zero. The zero then acts as both delimiter and counter. As the maximum message number is known (i.e., there is no search), an end-of-table delimiter is not required, other than the zero at the end of the last message.

NOTE that the first 20 messages must *not* be altered in position as they are used by the system in some cases automatically. There is a limit of 255 messages, but this is far more than will normally be required.

In macro form, simply enter the message number and the message, i.e.:

```

45 'Hi there, this is a message, isn't it . %'

```

Note that the standard ' symbol cannot be used *within* text and must be replaced with the alternative ' form. You will have noticed the use of one other special character, the percentage sign. This is used as a CRLF and hence must not be used for its original purpose (see below).

UMESS This table contains descriptions of objects and all general messages that are specific to a particular adventure.

```

7 'This is another message'

```

PLEASE NOTE: To make assembly listings more easily readable, CRLF sequence will be sent to the screen on encountering a "%" percentage sign in the text. This of course also saves a little space if you're tight with your bytes. Normally, CRLF's are not required in bulk text as the system automatically inserts these depending on the width of the screen.

It must also be noted here that the use of the character ">" anywhere in a sentence will not produce any output but

will serve to destroy any pending BATCHING and so this may be used freely within error messages and any message which requires the user's immediate attention.

A final point concerning messages. It is possible to have conditional sections within messages (or indeed complete conditional messages). Flags 23 to 49 inclusive have been reserved for this purpose. These may be tested for being TRUE or FALSE only, in the following manner. Anywhere within a string, the game writer may insert parentheses as follows:

```

'This is a bit of {Auntidy } text {B as you can {C easily} see}.'

```

To explain. If flags 23 (i.e., A), 24 (i.e., B), and 25 (i.e., C), are all ZFRO then the output of the message would be "This is a bit of text."; however if one were to set flag 25 then the output would be "This is a bit of untidy text."

If flag 24 was set, then you would get "This is a bit of text as you can see." It is obvious that if flag 25 were also set then the word "easily" would be inserted. Note, however, that this would not apply if flag 24 were reset, whereby all text inside the matching set of parentheses would be ignored.

Only the letters A-Z inclusive and the "/" character may be used after the opening parentheses (A-Z corresponding to flags 24-49) and the maximum NESTING depth is 10, although as many NON-NESTED sets as desired may be put into any section of text. The character "/" may be used *before* any of the alpha chars A-Z to indicate that the flag has *not* been set, i.e., is ZERO. The system flag that indicates that the "/" char is in use is only set and reset on OPENING braces so do not try any fancy multi-level conditional text while using this form.

In this version, two special flags have been reserved. Flags 21 and 22 corresponding to @ and ? are automatically set by the program after every user entry to be always opposite; i.e., if flag 21 is "1" then flag 22 will be "0" and vice-versa, but *randomly* so. Hence it is possible to write the following default message.

```

44 '{@} simply don't understand you.} {?What ???}'

```

Whenever this message is called, there is a 50-50 chance of either "What ????" or "I simply don't understand you." being output by the program.

These conditional facilities find their main use in circumstances where you want the program to differentiate between LOOK AT THE GLOBE and, say, LOOK AT THE GLOBE CAREFULLY. Obviously you can have umpteen levels of description.

The facilities may be used anywhere that text is found, either in place descriptions or in object descriptions, or even in default replies. A good use here is to allow for either random or deliberate variations in, say, the default "CANT HELP YOU" message, etc. etc.

Descriptions

LOC Place descriptions begin with zero, a number 255 to 252 to indicate one of the following:

- 255 Normal place, full exit description
- 254 Normal place, no exit description
- 253 Dark place, full exit description (if lit by lamp)
- 252 Dark place, no exit description

In high level form the following words have the same effect as the numbers above:

NORMAL, NOEXIT, DARK, DARKNO
 MESSAGE / KEY / LOCATION bytes, 255 to indicate end of directions, and then textual description, zero being the text terminator.

I.e.:
 0 254 02 01 02 255 <TEXT> 0 253 03 32 12 255
 <TEXT> 0

In macro form, a typical entry may be as follows:

```

1 NOEXIT,02,N,2
'I am in a small room with no furniture. There
'appears to be nobody around at the moment.'

```

Again, no end of table required other than the mandatory zero, (or ENDTABLE) as we are not looking for locations which do not exist. In the above example, the first text contains only one sequence of exits, (not printed) MSG2 being the description, i.e., NORTH. This corresponds to a user KEY code of 01, and in this case, if a match for the key is found, the player will be transferred to location number 03.

In the second example, MSG3 prints out SOUTH; if the user typed in GO SOUTH, or SOUTH, or S then this would be converted to KEY 03 and the user would then be transferred to location 12. Meanwhile, the present location description is printed out, and any attempt to LOOK at anything will produce the response "I can't see anything" as the 253 at the start of the sequence dictates that this is a dark location. (Of course, if the user has a LAMP, then that is a different matter. See description of LAMP.)

A special location number is 249 (RTN). When a transfer

Flag Table

This is a 255-byte table numbered 1 to 255, zero not being used, as zero is used as a delimiter. Certain flags have special uses.

1	Dark location indicator . . . If ZERO then dark.
2	Number of items carried.
3	Maximum permissible items to be carried.
4	Counted up once every turn.
5	" "
6	" "
7	Counted down once every turn if not zero.
8	" "
9	" "
10	Counted down if dark, lamp ON, every turn.
11	" "
12	" "
13	Present score.
14	Present location.
15	Store FF or FE (Internal).
16	1 and 10 OR'd together. ZERO means can't see.
17	Lamp on/off. Zero when ON, 1 when OFF.
18	Number of items WORN.
19	Number of items HERE (present location).
20	Store of last location number (automatic).
21-22	Inclusive. For random text control (automatic).
23-48	Inclusive. Text controls A-Z.

Flags 49 to 250 are available to the user.



Achtung:
 Zu diesem Artikel gehört noch ein 13-seitiges Programmlisting.
 (Listing besteht aus Maschinensprache und BASIC-Teilen)
 Wer sich dafür interessiert melde sich bitte bei der Redaktion. Die "Paperware"
 wird dann umgehend zugesandt.
 Wer schon die entsprechende Software in seiner Programmbibliothek hat, kann das
 Programm ja den anderen interessierten Clubmitgliedern zu Verfügung stellen.
 Vielleicht findet sich auch jemand, der das Listing gerne eintippt.
 Bitte meldet Euch, wenn Ihr dazu etwas beizutragen habt.



request to this location is selected, the program will treat this in a special manner by returning the user to the previous location. Hence, you may enter a LARGE BOX and a RTN exit will bring you out, no matter where the box is situated. This avoids the need for an awful lot of location testing.

The macro version of this table makes life very simple as can be seen in the above example. No delimiters are required, and the final description is quite readable.

Assembly Notes

To use these programs you will need access to the following utilities. A Z80 assembler (Zilog mnemonics) with conditional assembly. (If your assembler does not support the CONMSG pseudo-op and/or conditional assembly, miss out the messages and only select the sections of code that you require for your application.)

The BASIC used for the converter is Microsoft Version

Conditions/Actions	
Here are the conditions and actions possible together with explanations for each. Those shown below are all available to the Adventure writer and there is no limit to the amount of conditions and/or actions that you may specify in any set.	
Conditions	
CL xx	Current location is xx.
NCL xx	Current location is NOT xx.
OP xx	Object xx is in present location or is worn or carried (xx=00 means obtain object number from user input).
OC xx	Object is being carried.
OW xx	Object is being worn.
OWC xx	Object is worn or carried.
OH xx	Object here but not carried or worn.
ONP xx	Object is not present.
ONC xx	Object is not being carried.
ONW xx	Object is not being worn.
NWC xx	Object is not worn and not carried.
ONH xx	Object not here, but MAY be worn or carried.
OE xx	Object exists at present.
ONE xx	Object does not exist at present.
@L xx yy	Object xx is at location yy.
N@L xx yy	Object is not at location yy.
RND xx	Random chance 0-254... Where 254 is dead cert.
TF xx yy	Test flag No. xx for value yy.
NTF xx yy	Test flag No. xx for NOT value yy.
GE xx yy	Flag xx greater than or equal to value yy.
LT xx yy	Flag xx less than value yy.
C2 xx	Compare 2nd word with xx.
C3 xx	Compare 3rd word with xx.
NC2 xx	True if 2nd word is NOT = to xx.
NC3 xx	True if 3rd word is NOT = to xx.

Actions

INV	Print an inventory of carried and worn objects.
TKE xx	Get object to carry it.
WR xx	Get object and wear it.
DR xx	Drop the object.
PR xx	Print message number xx from UMESS table.
PRM xx	Print message number xx from MESS table.
DL xx	Describe location.

SF xx yy	Set flag xx to value yy.
DSCOB xx	Describe the object.
SWP xx	Swap object xx with object xx+1.
MV xx	Move to location xx.
OK	Say 'OK' and exit table.
QT	Quit.
INF xx	Increment flag xx.
DEF xx	Decrement flag xx.
EX	Exit table.
SC	Print current score.
PU xx yy	Put object xx into location yy.
CR xx	Create object xx (Place here).
DES xx	Destroy object xx.
FLIP xx	Flip flag xx true to Zero and Vice Versa.
LA xx	Look around at items.
SMA xx	Set message A to UMESS number xx.
SMB xx	Set message B to UMESS number xx.
WT xx	Wait for xx seconds.
PF xx	Print the value of flag No. xx.
ATF xx yy	Add yy to the value of flag xx. If > 255 then set to 255. Make yy=0.
SFF xx yy	Subtract yy from flag xx. If makes xx < 0 then set xx=0. Make yy=0.
GRUN xx	Print description, directions etc. & exit.

The above conditions and actions represent all of those implemented in the present version of ADVEN-80.

Flag

This table is simply a collection of pairs of flag numbers and their corresponding initial data values. Entries must be contiguous.

35 22 /* means flag 35 is set to value 22 */
Certain flags are controlled automatically by the program and may be viewed only.

TITL

This final message is that which will be presented at the start of any game. It is comprised of ASCII text, with a ZERO delimiter at the end.

CHR\$(any single ASCII number you like, or several to clear screen.)
'any title message'
'blah blah'
ENDTABLE

5. Peculiarities of this BASIC are few, the most important one is the INSTR. This returns a value of the 1st occurrence of one string or character within another string:

A=INSTR(QS,"R") where QS="ARE", A=2

Only sequential disk files are used, read line by line from the source file into the destination file in a single pass. The program is designed to allow reasonably free format of the source file.

ADVEN-80 System Overall View

ADVEN-80 was a development of an original BASIC system⁶ which relied heavily on disk storage. The BASIC program itself was suggested by a small program called GROW³ (for those interested, the BASIC used was Microsoft Disk Extended Version 5.1 and the disks, Micropolis Metafloppy).

Anyone who wants to attempt to write their own Adventures in BASIC would be advised to start by reading Greg Hassett's article⁸ and for those who want to start in machine code (any processor) and are capable of working from generalized macros, a good starting point would be Ken Reed's program⁴.

Due mainly to the desire to re-do the system from scratch in Assembler, and with help from someone already familiar with the design of Adventure systems⁴, ADVEN-80 was developed. The present version has several outstanding features, firstly the ability to handle multiple commands (i.e., GET LAMP THEN LIGHT IT). This example shows another feature, the system assumes that when you say "IT" you mean the item mentioned last.

ADVEN-80 is fairly flexible when it comes to entering English, (depending on the database). For example, you may enter GO SOUTH or simply S. You may say GET LAMP or TAKE THAT LAMP. If, for example, the player were to say RUB THE LAMP, a particular effect may occur. If it is necessary to immediately repeat such an action, he may either re-enter RUB THE LAMP, or if preferred, he may simply enter AGAIN. The idea for this was taken from a magazine article on ZORK⁷. On long inputs this can save a small amount of key pressing and so was considered to be a worthwhile addition to the system.

Due to a potentially fairly massive vocabulary, the system can make sense out of many sentences, but of course this must not be confused with real intelligence (as anyone actually creating the database could not fail to notice!!). So if you try to confuse the program, then you probably will succeed, so be sensible... The present version was written under the M80 macro pre-processor⁵ and assembled by ASMB, the Cromemco macro-assembler. The listing shown here has already been run through M80 and is suitable for most Z80 assemblers. (The macro pre-pass accounts for the excess labels.)

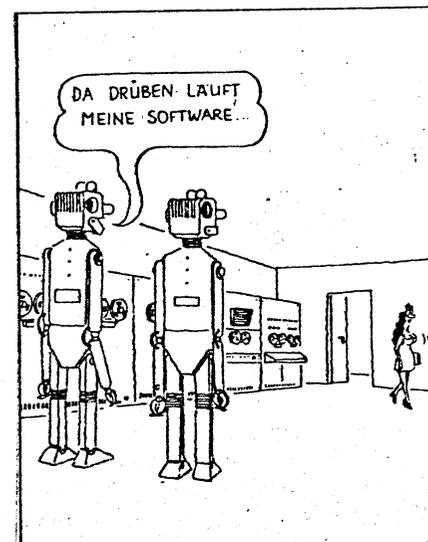
Readers who wish to use the programs in this article to create their own Adventures may find database construction made considerably easier by starting off with a very large piece of paper, drawing out all the locations as large "molecules." Connect them together, then make a list of all the interconnections, followed by all the items in the game, and where they are going to be at the start of the game. Finally, work through the game location by location, armed with as many sketches as possible, preferably with a friend known to be a whizz at finding faults in other people's programs... really,

you won't get it right first time; you'll probably be able to work through a game by yourself with no problems, then along comes young Johnny, totally unaware that you're not supposed to try EATING the AXE. Hey, presto, another slight bug in your database discovered! Spend a lot of time thinking out the game, and twice as much time debugging it, but most of all, enjoy yourself. **DDJ**

(LISTING ON PAGE 14)

References

- Scott Adams. Adventures available for TRS-80, Sorcerer, Apple, etc.
- Mostek Z80 manual.
- "Grow" by Jeff Levinsky. *Creative Computing* magazine.
- "Adventure II" by K. Reed. *Practical Computing* (Aug. 1980).
- M80. *A Structured Machine-Oriented Microcomputer Language*. By L. R. B. Pedroso. AD/A-010 055.
- "Fantasia" by P. Scargill. A BASIC Adventure system. *Creative Computing* (1981).
- "How To Fit a Large Program Into a Small Machine" by Marc S. Blank and S. W. Galley. *Creative Computing* (July 1980).
- "How To Write an Adventure" by Gregg Hassett. *Creative Computing* (July 1980).



DAS PIRATE-ADVENTURE

Als ich erwachte, fand ich mich in einem Londoner Apartment wieder, auf einem Teppich, der an den Boden genagelt war (die Angst vor Diebstahl führt schon zu den komischsten Präventivmaßnahmen). Neben mir lag ein Gegenstand, der mir das Rätsel meiner Anwesenheit in einer mir unbekanntem Wohnung scheinbar aufklärte: eine Flasche Rum. Ich war schon immer anfällig für diese Freude des Lebens gewesen und hatte ihren Genuß oft genug mit einem Gedächtnisschwund bezahlen müssen. Doch etwas machte mich stutzig: die Flasche war noch voll! Sollte mich nur vom Geruch des Alkohols eine (wein)selige Bewußtlosigkeit ergriffen haben? Unmöglich, so wenig vermag ich nun auch nicht!

Da fiel mein Blick auf ein Schild: "Bring die SCHATZE hierher und sage: SCORE". Nun war mir alles klar. Ich hatte wieder eine Mission zu erfüllen und war zu diesem Zweck durch Raum (und vielleicht auch Zeit) an diesen Platz gebracht worden. Sofort untersuchte ich in routinierter Weise die Umgebung, denn jeder Hinweis konnte mir helfen, möglichst schnell wieder mein Zuhause zu erreichen und die Lorbeeren meines Erfolges zu genießen. Doch bis dahin war es noch ein langer Weg. Als erstes fielen mir ein Paar Segeltuchschuhe auf, die sogleich gegen meine eigenen Schuhe eingetauscht wurden, und eine Tüte mit Crackers in die Hände. Ich widerstand der Versuchung, mit Crackers und Rum ein Picknick zu veranstalten, denn diese Gegenstände mochten mir später vielleicht noch von Nutzen sein.

Der einzige Weg aus diesem Zimmer führte über eine Treppe, die ich vorsichtig erklimmte, immer auf der Hut vor möglichen Gefahren.

Am Ende der Treppe befand sich ein Alkoven. Einen möglichen Fluchtweg bot ein offenes Fenster, doch als ich herauskletterte, mußte ich feststellen, daß sich unter meinen Füßen nur ein schmaler Sims und zwischen mir und dem Boden mehrere Dutzend Meter Luft befanden. Ich bemühte mich, meine Augen nicht nach unten zu richten, denn ich bin von Natur aus nicht schwindelfrei wie mein Adventure-Kollege James Bond, und kletterte vorsichtig wieder in das Zimmer. Aber in dem Zimmer stand noch ein vollgestellter Bücherschrank. Neugierig nahm ich ein Buch heraus und mit einem Quietschen schwang die Bücherwand beiseite, um einen Geheimgang freizulegen. Ohne das Buch weiter zu beachten, betrat ich den Gang.

Durch den dunklen Gang gelangte man auf einen muffigen Dachboden. Hier fand ich eine Fackel und die Tasche eines Piraten, in welcher sich eine Packung Streichhölzer befanden. Zurück in den Alkoven interessierte mich das Buch. Es war schon ziemlich zerfallen, doch der Titel ließ sich noch mit einiger Mühe entziffern: "Die Schatzinsel". Ein Satz war noch leserlich: "Long John Silver ließ zwei Wertgegenstände auf der Schatzinsel zurück." Nun war meine Mission klar: zuerst mußten die Schatzinsel und die beiden Wertgegenstände gefunden werden. Doch wie sollte man mitten in London eine Insel finden, mal abgesehen von der Verkehrsinsel am Piccadilly Circus?

Am Fenster sah ich mir das Buch noch einmal an und mir lief ein Schauer den Rücken hinunter: auf dem Vorsatzblatt war von zittriger Hand ein Wort mit Blut geschrieben: "YOHO". Leise murmelte ich das Wort vor mich hin und alles fing an, sich um mich zu drehen. Plötzlich stand ich vor dem Fenster. Verwirrt sagte ich noch einmal "YOHO!" und der Himmel wechselte von grau zu blau, von kaltem Londoner Nebel zu

blendender Sonne. Ich war in Erise. Durch die Segeltuchschuhe spürte ich Sand und an mein Ohr drang ein leises Rauschen: Meeresrauschen! Ein Teil dessen, was ich für den Himmel gehalten hatte, war in Wirklichkeit blaues Meerwasser. Vor mir stand ein Schild: "Willkommen auf der Pirateninsel, passen Sie auf die Tide auf!" Aus dem Sand ragten Kiel und Mast eines kleinen Bootes. Scheinbar waren schon vor mir andere Adventurer auf dieser Insel gestrandet und hatten das Schild als Warnung für ihre Nachfolger aufgestellt.

Im seichten Wasser einer Lagune fand ich einen rostigen Anker. Mangels Schwimmerfahrung mußte ich mich jedoch vom offenen Meer fernhalten. Somit blieb nur das Landesinnere, wo ich auf eine Wiese mit einer Grashütte stieß. In der Hütte residierte ein verschlagener dreinschauender Pirat. Um ihn zu besänftigen, bot ich ihm die Buddel Rum an, die er ohne Umschweife nahm und glucksend lachend damit verschwand. Zurück blieben eine Schatzkiste und ein Papagei, der mir viele nützliche Ratschläge geben konnte (wie mir vorher ein Zettel, der in dem Buch lag, schon prophezeit hatte). Allerdings entwickelte der Papagei einen ganz schön kräftigen Appetit und verbrauchte ziemlich viele Cracker.

Mit der Schatztruhe konnte ich vorerst nicht viel anfangen, denn sie war verschlossen. Also zog ich weiter ins Inselinnere und gelangte an einen mit Höhlen übersäten Berg. Im Höhlensystem war eine verschlossene Tür, die von Krokodilen bewacht wurde, Endstation. Eine helfende Stimme riet mir, Futter für die Krokos zu finden, doch der Papagei war ihnen zu mager und somit war ich gezwungen, die Höhlen unverrichteter Dinge wieder zu verlassen.

Ein Pfad führte zum Gipfel des Berges. Dort konnte ich mich durch eine Felsspalte zwängen, wobei ich das große Buch schweren Herzens zurücklassen mußte. In einer großen Höhle, die durch meine Fackel erhellt wurde, fanden sich ein Satz Segel, Bretter, und ein Werkzeugschuppen, in dem ich sofort herumstöberte. So kamen noch eine Schaufel, ein Hammer und ein Schwimmflügel zum Vorschein. Aus der Höhle ließen sich nur die Schwimmflügel und den Hammer mitnehmen, denn alle anderen Sachen waren für den Spalt zu sperrig und der einzige andere Ausgang, eine Tür am Ende eines Ganges, war verschlossen. Wieder an der frischen Luft, löschte ich die Fackel, damit sie nicht so schnell ausbrannte.

Gedanklich hatte ich schon die Verbindung von der Tür in der Höhle zu der von den Krokodilen bewachten hergestellt. Es galt nun, den Schlüssel und das richtige Futter zu finden. Mit den Schwimmflügel konnte ich mich endlich auf den Ozean wagen, aber der Fisch, den ich dort fing, entwischte wieder, sobald ich mich dem Land näherte. Ein Behälter mußte her! Doch der einzige erreichbare Behälter war die Schatztruhe, und die war verschlossen.

Entmutigt an den Strand zurückgekehrt, murmelte ich wieder "YOHO" und stand kurz darauf auf dem Sims hoch über dem nebelverhangenen London. Nach einer Klettereinlage durch das Fenster und die Treppe herunter, kam mir der Gedankenblitz: mit dem Hammer ließen sich die Nägel lösen und der Teppich verschieben. Tatsächlich fand sich darunter ein Schlüsselbund, mit dem man die Truhe öffnen konnte. In der Truhe befanden sich zwei Dinge: ein Plan, wie ein Schiff zu bauen war, und eine Karte, die den Weg zur Schatzinsel beschrieb, mit der seltsamen Aufschrift: "30 Schritte, dann graben!"

Die leere Truhe sollte mir beim Fischfang helfen, doch sie leistete mir ebenso schlechte Dienste wie meine Hände. Also kehrte ich erst an den Strand und dann in die Londoner Wohnung zurück. Einem lauten Geräusch folgend, gelangte ich in die staubige Dachkammer und fand dort den schlafenden Piraten mit der nun leeren Rumflasche. Diese ließ ich mitgehen und kehrte in die Fischfanggründe zurück. Nun endlich gelang es: erst das

Wasser und dann den Fisch in die Flasche und schon war es geschafft. Die Krokos freuten sich sehr über die Mahlzeit und verschwanden sofort. Die Tür ließ sich mit dem Schlüssel öffnen und über diesen Weg standen mir die Segel, das Holz und die Schaufel zur Verfügung. In der Lagune mußte der Anker ausgegraben werden, und nun konnte ich endlich alle Utensilien am Strand zusammentragen. Nach ausgiebigem Studium des Plans war es mir möglich, das Schiff zusammenzuzimmern. Doch es fehlte mir noch ein Kapitän!

Der befand sich schlafend in der Dachkammer, wie schon erwähnt. Er mußte kräftig geschüttelt werden, doch schließlich erwachte er und wir verabredeten uns am Schiff. Dort wartete er auch schon auf mich. Mein Buch gefiel ihm gar nicht und so ließ ich es widerwillig zurück. Wir erwarteten gemeinsam die Flut und setzten dann die Segel.

Schon nach einem Tag konnten wir an der Schatzinsel ankern. Am Strand probierte ich den Rat der Karte aus: 30 Schritte gehen und graben. Es kam eine Kiste Rum zum Vorschein. Der Pirat verschwand gleich mit einer Flasche.

Auf einem Feld mußte sich die Formel zum zweiten Mal bewähren und bescherte mir eine Holzkiste. In einem Kloster kam mir der Papagei zuhilfe: er verscheuchte die hochgiftigen Mamabas, kurz bevor sie mir den tödlichen Biß verpassen konnten. So konnte ich einen der beiden Schätze mitnehmen: DUBLONEN!

Der zweite Schatz lag vermutlich in der Holzschachtel. Doch um dies zu überprüfen, mußte ich erst zurücksegeln. Mein Kapitän befand sich in der gewohnten Verfassung: betrunken und schlafend. Er war schnell wach zu bekommen und stimmte mir zu, sofort die Segel zu setzen.

Wieder am Strand angelangt, ließ sich die Holzkiste mit dem Hammer und einiger Gewaltanwendung öffnen und der zweite Schatz kam zum Vorschein: die rote und die blaue Mauritius. überglücklich sprang ich in die Luft, nahm das Buch und kehrte in das Londoner Apartment zurück.

Dort bestätigte sich meine Vermutung: alle Schätze waren in meinem Besitz. Nun war es mir vergönnt, den Heimweg anzutreten und zuhause in Ruhe auf den nächsten Auftrag zu warten.





W E T T B E W E R B

Auch im neuen Jahr mal wieder ein Versuch alle Mitglieder unseres CLUB's anzusprechen. Das sich auch einjeder beteilige an diesem Wettbewerb.

--- Programmierwettbewerb ---

Bei diesem Wettbewerb geht es um folgendes:

Gesucht wird ein ZEHNZEILER IN BASIC

Als Bedingungen sind einzuhalten:

Das Programm muß in BASIC geschrieben sein und darf nicht mehr als 10 Programmzeilen lang sein. Die Funktion, die das Programm ausführen soll bleibt Euch überlassen. Es liegt also an Euch, was das Programm für die anderen CLUB-Mitglieder zu bieten hat. Die Anzahl der Programme, die pro CLUB-Mitglied eingesandt werden darf, ist nicht begrenzt. Wichtig ist noch, daß das Programm "auf Eurem Mist gewachsen ist" --auf gut deutsch: eine kreative Eigenproduktion Eurerseits.

Dieses Programm, daß Ihr an die Redaktion (natürlich mit ein paar erklärenden Worten) schickt, wird im jeweiligen folgenden INFO den anderen CLUB-Mitgliedern vorgestellt. Sie dürfen und sollen ausprobieren. Zum Jahresende 86 werden dann Stimmzettel an alle Mitglieder versandt und es wird gewählt werden.

Gewählt wird dann DAS PROGRAMM DER CLUB80'ER 1986.

Da für das Programm kein Thema gestellt ist, hat also jeder von Euch seine Chance mitzuwirken. Vielleicht hat man die richtige Idee und fällt unter die Ersten. Ich wünsche es Jedem.

Ich hoffe, Ihr lasst mich mit dem WETTBEWERB nicht im Regen stehen. Werft die FLOPPY's an und setzt die CPU unter Spannung. Ich rechne mit Euren Programmen.

Erfolgreiches programmieren wünscht Euch

Eure Redaktion.

„Geschichte schreiben ist eine Art, sich das Vergangene vom Hals zu schaffen.“

Johann Wolfgang von Goethe,
deutscher Dichter (1749-1832)

„Es stimmt, daß Geld nicht glücklich macht. Allerdings meint man damit das Geld der anderen.“

George Bernard Shaw,
anglo-irischer Dramatiker und
Satiriker (1856-1950)

„Gemessen an den bevorstehenden Flugaufgaben ist der Mensch eine Fehlkonstruktion.“

Edward T. Connolly,
amerikanischer Testpilot

„Man fällt nicht über seine Fehler. Man fällt immer über seine Feinde, die diese Fehler ausnutzen.“

Kurt Tucholsky,
deutscher Schriftsteller (1890-1935)

„Geister erinnern mich an den Männerpruch über Frauen: Du kannst mit ihnen nicht leben, und du kannst nicht ohne sie leben.“

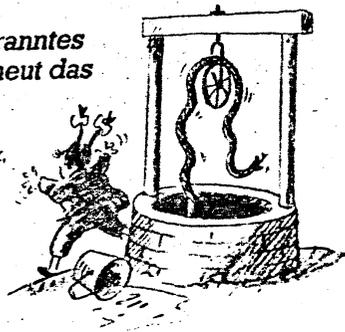
Eugene O'Neill,
amerikanischer Dramatiker
(1888-1953)

Chinesisch: Yangtòu mài gouròu.
Einen Schafkopf aufhängen und Hundefleisch verkaufen.
Eine schlimme Sache beschönigen.

Yi zhao bèi she yao, san nián pà jing shéng.
Einmal von einer Schlange gebissen, drei Jahre Angst vor dem Brunnen-seil.

Ein gebranntes Kind scheut das Feuer.

Rudolf Haselberger, Gröbenzell



Fehler im NEWDOS-Befehl 'BOOT'

Da das Basteln im DOS in den letzten Club-Info's sehr viel Raum einnahm, will auch ich hier einen kleinen Beitrag dazu liefern.

Wie schon aus der Überschrift hervorgeht, geht es um den 'BOOT'-Befehl des NEWDOS-80.

Vor einigen Wochen wurde ich durch einen Bekannten darauf aufmerksam gemacht, daß in seinem NEWDOS der 'BOOT'-Befehl nicht richtig arbeitet. Ich war ziemlich erstaunt, da ich erst einige Tage vorher mit diesem Befehl gearbeitet hatte.

Bei genaueren Nachforschungen ergab sich aber, daß ich mit GDOS und nicht mit NEWDOS gearbeitet hatte. Also wurde das gleiche mit NEWDOS ausprobiert, und siehe da, es ging nicht. Nach Eingabe des 'BOOT'-Befehls hängte sich der Rechner auf oder er landete im Debugger, was ja nun auch nicht unbedingt erwünscht war. Im GDOS funktionierte der 'BOOT'-Befehl allerdings einwandfrei. Also auf gings - "Hinein ins DOS".

Mit Hilfe von Grossers "DOS Buch" war dann sehr schnell die SYS9/SYS als zuständige DOS-Datei ermittelt.

Also wurde als erstes einmal die SYS9/SYS vom GDOS auf das NEWDOS kopiert und der 'BOOT'-Befehl nun nochmals getestet. Und siehe da - es ging. Also mußte das Problem in der SYS9/SYS des NEWDOS liegen.

Nachdem das klar war, kam der Disassembler zum Einsatz. Nach einigen Vergleichen der für den 'BOOT'-Befehl zuständigen Programmteilen kam ich dann zu folgenden Änderungen:

SYS9/SYS des NEWDOS-80

rel. Byte 00CA
Inhalt: 3E FE 32 EC 37 AF C3 D0 51

rel. Byte 04E4
Inhalt: 32 E1 37 61 02 00 CD 60 00 00
00 F3 AF C3 66 00

Nach dem letzten geänderten Byte sollten die Bytes 02 02 stehen.

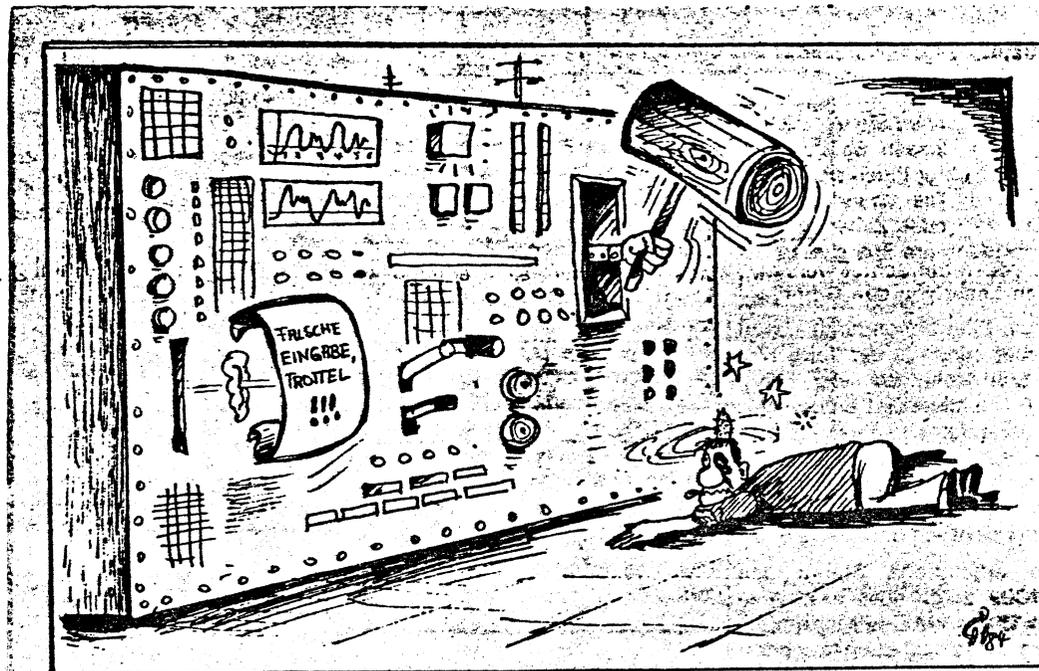
Die alten Inhalte der zu ändernden Bereiche habe ich nicht mit angegeben weil mir im Augenblick leider kein Drucker zur Verfügung steht.

Die relativen Bytes sind mit 'DDE' angegeben. Wer mit Superzap ändern will, muß im relativen Sektor 00, relatives Byte CA und im relativen Sektor 04, relatives Byte E4 die Änderungen vornehmen.

Nach diesen Änderungen arbeitet der 'BOOT'-Befehl auf meinem System einwandfrei.

So, nun aber genug der Theorie, Rechner einschalten und dann haut in die Tasten Leute, das die Bit's und Byte's nur so dampfen.

Bernd Retzlaff



Was ist mit der Primzahl '37' los ?

```
10 CLS:PRINT" Bildung von Quersummen des 1x1 von 37":PRINT
20 DEFINTF,I,M,Q:DEFSTRS:DEFDBLZ
30 INPUT"Niedrigster und höchster Multiplikator <M1, M2> ";M1,M2
40 FORJ=M1TOM2:Z=J*37:ZZ=Z
50 Z$=STR$(Z):FORI=1TOLEN(Z$):S=MID$(Z$,I,1):Q=Q+VAL(S):NEXT
60 IFQ>9Z=Q:Q=0:GOTO50
70 PRINT"Die kleinste Quersumme des Faktors "J" ist ="Q
80 PRINT"Die kleinste Quersumme des Produktes"J" * 37 ist ="Q:Q=0:P
RINT
90 Y$="":Y$=INKEY$:IFY$="",90
100 NEXT:END
```

Hier ist das Programm zum Durchspielen der Quersummen des "Kleinen Einmaleins" mit der ominösen Zahl '37'; blättert nochmal zurück zu meiner "Vorstellung"! Ich hatte dort behauptet, daß jedes Produkt der Zahl 37 mit einem beliebigen Faktor die Eigenschaft habe, daß seine Quersumme gleich der Quersumme des betreffenden Faktors sei. Dazu eine Erläuterung des von mir hier verwendeten Begriffs "Quersumme": Natürlich ist dies zunächst die Summe der Ziffern, aus denen sich eine Zahl zusammensetzt. Wenn diese Summe aber selbst mehrstellig ist, kann man von ihr wieder die Quersumme bilden und so fort, bis man schließlich zu einer einstelligen Zahl kommt. (Diese kann nie Null sein; warum nicht?) Dies nenne ich "die kleinste Quersumme".

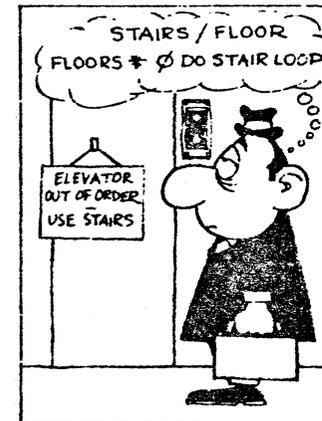
Multipliziert man 37 mit einem beliebigen Faktor ("Multiplikator"), so hat die kleinste Quersumme des Produktes denselben Wert wie die Quersumme des Multiplikators - das war meine tief-sinnige Behauptung.

Gewiß, nichts weiter als eine Zahlenspielerlei - keine Mathematik! Außer der "Entdeckerfreude", die dieses Spiel fußhaltet, zeigt das Programmchen dem Anfänger (und der ist immer ein Spieler, sonst würde er wohl nicht mit der Computerei anfangen) erstens, wie sich jede beliebig lange Zahl auf dem Umweg über ihre Stringumwandlung elegant in ihre Einzelteile auflösen läßt, und zweitens, wie eine Rekursion aufgebaut werden kann, um schließlich zum erstrebten Ziel zu gelangen. Das gleiche hätte sich auch rein numerisch ohne Verwendung von Strings erreichen lassen - aber sehr viel langwieriger mit einigen umständlichen Formeln.

Wirklich: ein winziges Beispiel nur. Aber klein fängt man an...

> Euer KaJot <

PS: 'fuß-' (wo doch?) kann natürlich auch durch 'bein-' ersetzt werden



NEWDOS/80 + 10

Version 2.4
26.10.85

10/12/85 12:00:02

DOS>

Extended NewDOS 80

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

Heute: F(orm), DR, LF und OUT

Im letzten Info wurde der "Einbau" der ID-Funktion ins NewDOS80 beschrieben. Außerdem wurde gezeigt, wie man DDE und LWT in unter NewDOS startbare /CMD-Programme umwandelt. Heute nun will ich die Änderungen beschreiben, die nötig sind, um die DOS-Befehle F(orm), DR, LF und OUT im NewDOS zu installieren.

Zuvor jedoch ein paar Bemerkungen zu den Quellen, aus denen die entsprechenden Routinen stammen:

Funktion Quelle

F	SYS28/SYS aus GDOS 2.2 (dort als FORM im Library!)
DR	SYS28/SYS aus GDOS 2.2
LF	SYS28/SYS aus GDOS 2.2
OUT	SYS29/SYS aus HDOS (Autor Arnulf Sopp, veröffentlicht im Info des Bremerhavener Computerclubs)

Da bei meiner Ur-NewDOS-Version die Systemfile 29 als BASIC-Overlay (LINE und NAME) benutzt wird, mußte OUT im SYS28/SYS Platz finden, was auch keinerlei Schwierigkeiten bereitet!

Die folgenden Zap's setzen voraus, daß die Änderung "Platz im SYS1" durchgeführt und SYS28/SYS an der richtigen Stelle auf der Systemdiskette eröffnet ist!

Besitzer von VideoGenie-Maschinen können die Zap's weglassen, die sich ausschließlich mit der "memory-mapped" (und damit TRS80-spezifischen) Druckersteuerung befassen. Diese Änderungen sind mit EPTRS80 gekennzeichnet!

Nun aber frohen Mutes an die Arbeit!*

Viel Spaß und Erfolg wünscht Euch Euer

Karlmut Obermann

* Auf alle Fälle nur an einem Backup arbeiten!!!

Zuallererst muß man sich natürlich SYS28/SYS von einer G- oder H-DOS -Diskette in das zwar schon eröffnete aber noch leere Systemfile auf seiner neuen Systemdiskette kopieren. Danach sind die folgenden Änderungen (Format: File,File relativ Sector, Sector relativ Byte, Änderung) vorzunehmen:

EPTRS -> alle Änderungen mit diesem Zeichen dienen der Druckersteuerung bei TRS 80-Rechnern und brauchen von VideoGenie-Besitzern nicht durchgeführt werden!

EPTRS
SYS28/SYS,0,4F
ändere DB FD F6 0F in CD 11 4F 00

Folgende Änderung behebt einen Fehler (falscher relativer Sprung!), und man sollte sie auch auf HDOS- und GDOS-Originaldisketten durchführen!!!
SYS28/SYS,0,6D
ändere E2 in E1

EPTRS
SYS28/SYS,2,1D
ändere 00 00 00 00 00 00 in 3A EB 37 F6 0F C9

EPTRS
SYS28/SYS,2,57
ändere D3 FD 00 in 32 EB 37

EPTRS
SYS28/SYS,2,87
ändere D3 FD 00 in 32 EB 37

EPTRS
SYS28/SYS,2,C1
ändere D3 FD 00 in 32 EB 37

EPTRS
SYS28/SYS,2,DF
ändere DB FD 00 in 3A EB 37

Die nächste Änderung betrifft den Druckercode für 6 bzw. 8 Zeilen pro Zeile. Dieser ist natürlich vom Drucker abhängig und wird hier für EPSON RX80 angegeben!

SYS28/SYS,4,6A
ändere 1B 41 03 00 1B 42 03 00 in 1B 32 0D 00 1B 30 0D 00
Die alten Codes gelten, soweit ich weiß, für ITOH-Drucker!

Damit es nicht nur bei den recht trockenen Zaps bleibt, hier eine kurze Erklärung der Assembleranweisungen:

wo früher
D3 FD = OUT (FD),A * Inhalt des Accus auf Port
* FD ausgeben

00 = NOP * keine Operation
stand, haben wir
32 EB 37 = LD (37EB),A * Inhalt des Accus in Speicher-
* zelle 37EB laden

eingetragen. Wir haben aus einer portorientierten Ein/Ausgabe eine speicherorientierte Ein/Ausgabe gemacht (wie sie der TRS-80 verwendet!). Entsprechend wurde die Befehlsfolge IN A,(FD) [DB FD] in LD A,(37EB) [3A EB 37] geändert!

Wer mehr wissen möchte, kann sich die Zaps ja mal mit dem Disassembler anschauen!

HEFT

11

Januar

1985

26

Die bis hierher durchgeführten Änderungen betreffen nur die Befehle DR, F und LF. Der OUT-Befehl kommt etwas später an die Reihe. Zuvor will ich noch kurz auf die Parametertabelle des F(orm)-Befehls eingehen.

Aus dem GDOS-Handbuch:
FORM, parameterkette

FORM erlaubt die Ausgabe von besonders definierten Zeichenketten an den Drucker. Da die Möglichkeiten der Drucker recht unterschiedlich sind, wurde wie folgt verfahren.

Die Parameter A bis P bezeichnen 16 Zeichenketten von jeweils 15 Zeichen und einem Endzeichen, die vom Anwender selbst definiert werden können. Das Endzeichen muß entweder 03h (kein Wagenrücklauf) oder ODh (Wagenrücklauf) sein.

Ein Punkt gibt ebenfalls ODh an den Drucker aus.

X erlaubt die Ausgabe einer bis zu 255, Y einer bis zu 123 Zeichen langen, vom Anwender definierten Zeichenkette.

Die Zeichenketten für die Parameter A-P können mit DDE in SYS28/SYS Sector 1 relativ, für X in B00T/SYS (GDOS- oder HDOS/SYS) Sector 4 relativ und für Y in SYS28/SYS Sector 3 relativ geändert werden!

Hier nun ein Hardcopy des relativen Sectors 1 von SYS28/SYS. Wie man sieht, sind die Parameter A-E belegt, während F-P noch frei sind. Die in A-E eingetragenen Werte gelten für EPSON RX80 - Drucker und bedeuten:

- A = Drucker neu initialisieren
- B = Schmalschrift, linker Rand 10, rechter Rand 122
- C = Schmalschrift
- D = linker Rand 10
- E = rechter Rand 122

```

000100: 1B40 0D20 2020 2020 2020 2020 2020 2020 2020 .5.
000110: 1B0F 1B6C 0A1B 517A 0D20 2020 2020 2020 2020 ...1..Qz.
000120: 1B0F 0D20 2020 2020 2020 2020 2020 2020 2020 ...
000130: 1B6C 0A0D 2020 2020 2020 2020 2020 2020 2020 .1..
000140: 1B51 7A0D 2020 2020 2020 2020 2020 2020 2020 .Qz.
000150: 4646 4646 2C46 4646 462C 4646 4646 2C03 FFFF,FFFF,FFFF,
000160: 4747 4747 2C47 4747 472C 4747 4747 2C03 GGGG,GGGG,GGGG,
000170: 4B4B 4B4B 2C4B 4B4B 4B2C 4B4B 4B4B 2C03 HHHH,HHHH,HHHH,
000180: 4949 4949 2C49 4949 492C 4949 4949 2C03 IIII,IIII,IIII,
000190: 4A4A 4A4A 2C4A 4A4A 4A2C 4A4A 4A4A 2C03 JJJJ,JJJJ,JJJJ,
0001A0: 4B4B 4B4B 2C4B 4B4B 4B2C 4B4B 4B4B 2C03 KKKK,KKKK,KKKK,
0001B0: 4C4C 4C4C 2C4C 4C4C 4C2C 4C4C 4C4C 2C03 LLLL,LLLL,LLLL,
0001C0: 4D4D 4D4D 2C4D 4D4D 4D2C 4D4D 4D4D 2C03 MMMM,MMMM,MMMM,
0001D0: 4E4E 4E4E 2C4E 4E4E 4E2C 4E4E 4E4E 2C03 NNNN,NNNN,NNNN,
0001E0: 4F4F 4F4F 2C4F 4F4F 4F2C 4F4F 4F4F 2C03 OOOO,OOOO,OOOO,
0001F0: 5050 5050 2C50 5050 502C 5050 5050 2C03 PPPP,PPPP,FPPP,

```

Natürlich kann und soll sich jeder Anwender für seinen Drucker seine eigenen Codes zusammenstricken. Die hier aufgeführten Funktionen dienen nur der Demonstration!

Nun aber zu dem ersten wirklich neuen Befehl, zu

OUT,port#,xx,yy- - - !

Wie schon zu Beginn dieses Artikels gesagt, entsprang der Befehl "OUT,port#,xx,yy,..." den genialen Gehirnwindungen unseres allseits bekannten(!) und beliebten(!?) Clubmitglieds Arnulf Sopp, von dem böse Zungen behaupten, er würde sogar Telefonnummern hexadezimal angeben (was eine reine Verleumdung ist)!

Aber Scherz beiseite, diese Funktion ist wirklich nützlich und gehört deshalb in "unser" NewDOS80 +10. Mit ihr kann man z.B. den HRG-Bildschirm abschalten (OUT,00,00) oder, falls man ein VideoGenie besitzt, Codes an den Drucker ausgeben. Wie die Routine funktioniert, durchschaut Ihr (hoffentlich) anhand der Kommentare im Listing unseres Assemblergroßmeisters!

Ich möchte hier nur kurz auf zwei Unterschiede zum Original (erschienen im Info des Bremerhavener Computerclubs) hinweisen:

1. Die ORG-Adresse (Zeile 100) mußte geändert werden, da die Routine bei uns ja in einem anderen SYS-File Platz finden muß.
2. In Zeile 250 mußte aus CALL 4454h ein CALL 4CD5h gemacht werden. Die Speicherzellen 4454h-4456h, die im GDOS 2.2 ein JP 4CD5 (C3 D5 4C) enthalten sind im NewDOS mit NOP's (00 00 00) gefüllt. Durch diese Änderung tritt keinerlei Funktionsänderung ein. Warum und wozu im GDOS 2.2 der Anspungpunkt geändert wurde, ist mir nicht bekannt!

Im nebenstehenden Sectordump des rel. Sector 4 von SYS28 sind die neu einzutragenden Codes unterstrichen. Vorher waren dort überall NOP's (00) eingetragen. Zusätzlich muß im rel. Sector 0 von SYS28/SYS, rel. Byte 1A noch die Einsprungadresse der Routine abgelegt werden. Dort wird 7351 statt 4B4D eingetragen!

Nun brauchen wir nur noch die neuen Befehle ins Library einzutragen. Dazu hängen wir an unsere Befehlstabelle folgende Codes an:

```

44 52 82 FE 00 (für DR)
46 88 FE 00 (für F)
4C 46 81 FE 00 (für LF)
4F 55 54 87 FE 00 (für OUT)

```

Das war schon alles. Eigentlich nur eine Sache von ein paar Minuten, und schon hat man vier neue DOS-Befehle zur Verfügung!

Beim nächsten mal kommen dann die Befehle P(ort), I(nfo) und DISK dran. Bis dahin, viel Spaß beim Zappen, Euer

Karstaut Obermann

```

00010 *****
00011 ;*
00012 ;* OUT, port#,xx,yy,....
00013 ;* (C) '84 by Arnulf Sopp
00014 ;*
00015 *****
5173 00100 ORB 5173M ;Beginn der Routine
5173 CD7E51 00110 OUT CALL GETVAL ;Hexzahl einlesen
5176 4F 00120 LD C,A ;Portnummer in C laden
5177 CD7E51 00130 LOOP CALL GETVAL ;Hexzahl einlesen
517A ED79 00140 OUT (C),A ;auf Port ausgeben
517C 1BF9 00150 JR LOOP ;bis zum bitteren Ende
517E CDBB51 00160 GETVAL CALL GETCHR ;eine Hexzahl einlesen
5181 17 00170 RLA ;Stellenwert korrigieren
5182 17 00180 RLA ;d. h. in linkes Nibble
5183 17 00190 RLA ;schieben
5184 17 00200 RLA ;(4 Bits nach links)
5185 57 00210 LD D,A ;linke Hexziffer merken
5186 CDBB51 00220 CALL GETCHR ;naechste Hexziffer einlesen
5189 B2 00230 OR D ;LSM mit MSM vereinigen
518A C9 00240 RET ;jetzt der korrekte Wert in Accu
518B CDD54C 00250 GETCHR CALL 4CDSH ;trennzeichen und CR erkennen
518E 2B17 00260 JR Z,EXIT ;ende bei NEW LINE
5190 7E 00270 LD A,(HL) ;Hexziffer laden
5191 FE30 00280 CP '0 ;Dezimalziffer?
5193 3B16 00290 JR C,ERROR ;falls ASCII < Dezimalziffer
5195 FE3A 00300 CP ;> ASCII '9' ?
5197 380A 00310 JR C,RETURN ;falls korrekte Dezimalziffer
5199 FE41 00320 CP 'A' ;Hexziffer > ASCII '9' ?
519B 380E 00330 JR C,ERROR ;falls < ASCII "A"
519D FE47 00340 CP ;> ASCII "F" ?
519F 300A 00350 JR NC,ERROR ;falls falsche Eingabe
51A1 D607 00360 SUB 7 ;falls Alpha-Hexziffer
51A3 E60F 00370 RETURN AND OFH ;ASCII nach binar umwandeln
51A5 23 00380 INC HL ;naechste Bildschirmstelle
51A6 C9 00390 RET ;verlaedigt
51A7 F1 00400 EXIT PDP AF ;Stack korrigieren
51A8 F1 00410 PDP AF ;(2. CALL-Ebene)
51A9 AF 00420 XOR A ;Z-Flag fuer "kein Fehler"
51AA C9 00430 RET ;ins Betriebssystem
51AB F1 00440 ERROR PDP AF ;Stack korr. (s.o.)
51AC F1 00450 PDP AF
51AD 3E2F 00460 LD A,2FH ;Fehlercode "falsche Parameter"
51AF B7 00470 OR A ;Z-Flag ruecksetzen
51B0 C9 00480 RET ;ins Betriebssystem
002F 00490 END ;wohilverdient

```

```

00000 Fehler
33448 Zeichen verfügbar

ERROR 51AB 00440 00290 00330 00350
EXIT 51A7 00400 00260
GETCHR 518B 00250 00160 00220
GETVAL 517E 00160 00110 00130
LOOP 5177 00130 00150
OUT 5173 00110
RETURN 51A3 00370 00310

```

```

000400: 01FA F050 03E5 7EFE 41D2 3E51 0620 211D ...P..B.A.>Q. !.
000410: 51BE 2B0C 2310 FAE1 FE0D C83E 34C3 0944 Q[.(#. ....>4..D
000420: 7B21 EC4F 0707 1600 5F19 CD6A 44E1 231B x!.D.....jD.#.
000430: D23F 3E3D 3C3B 3A39 3B37 3635 3433 3231 .?>=<=:987654321
000440: 302F 2E2D 2C2B 2A29 2B27 2625 2423 2221 0/.-,+)(%&X$#!
000450: 2000 FE51 3023 D640 1110 0047 21E8 4D19 ..QO#.5...G!.M.
000460: 10FD C316 5121 0042 1BF8 1B32 0D00 1B30 ....Q!.B...2...0
000470: 0D00 1B07 0300 1B0A 03FE 5B28 EBFE 5921 .....X(!..Y!
000480: 7050 2BDE E118 94CD 7E51 4FCD 7E51 ED79 pP(...).BQO.BQ.y
000490: 1BF9 CDBB 5117 1717 1757 CDBB 51B2 C9CD ...QO...W.QO...
0004A0: D54C 2B17 7EFE 303B 16FE 3A3B 0AFE 413B .L(.B.OB...:B..AB
0004B0: 0EFE 4730 0AD6 07E6 0F23 C9F1 F1AF C9F1 ..GO....#...#.
0004C0: F13E 2FB7 C900 0000 0000 0000 0000 0000 .>/[.....
0004D0: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0004E0: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0004F0: 4D30 3220 2E2E 2E2E 2E2E 2E2E 0202 004D M02 .....M

```

Hier eine Kurzübersicht über die besprochenen, neuen Befehle!

DR <,\$,#><text>
DR ermöglicht die direkte Ausgabe von Text und Sonderzeichen an den Drucker. Der Text kann max. 76 Zeichen lang sein und wird in Grossbuchstaben umgewandelt.
\$ = Escape (= ASCII 27)
= Umwandlung des folgenden Zeichens in einen Controlcode (z.B.: N=16)

F,parameterkette
F entspricht dem GDOS-Befehl FORM. Er erlaubt die Ausgabe besonders definierter Zeichenketten an den Drucker.
Die Parameter A-P bezeichnen 16 Zeichenketten zu je 15 Zeichen. X bezeichnet eine 255, Y eine 128 Zeichen lange Zeichenkette. Die Zeichenketten koennen vom Anwender selbst definiert werden. A-P stehen im Sector 2, Y im Sector 4 des SYS28/SYS. X steht im BOOT/SYS Sector 4.

LF <,& ,6>
LF,8 setzt die Zeilenzahl fuer den Drucker auf 8 / Zoll
LF,6 setzt die Zeilenzahl fuer den Drucker auf 6 / Zoll
Dabei wird der interne Zeilenzaehler jeweils auf 0 gesetzt.
LF ohne Parameter aktiviert das Unterprogramm zur Positionierung des Druckers.
Pfeil oben = Zeilenvorschub bis Seitenende
Pfeil unten = Zeilenvorschub (Taste festhalten)
Pfeil rechts = Zeilenvorschub
Pfeil links = negativer Zeilenvorschub (je nach Drucker!!)

OUT,##,xx,yy,..
= Nummer des angesprochenen Ports
xx,yy,.. = Werte, die an den Port ausgegeben werden sollen
Die Werte muessen hexadezimal angegeben werden. Die Anzahl der Bytes, die auf einmal abgesendet werden kann, wird von der Groesse des DOS-Eingabebuffers bestimmt (norm. 80 Zeichen+CR).

ID,#<,A>
Mit ID werden die PDRIVE-Parameter unbekannter Disketten erkannt und auf Diskette geschrieben.
Wird ,A angehaengt werden die erkannten Parameter sofort aktiv.

Noch ein Zap!
Wenn man versucht, ein Programm auf einer Diskette zu löschen, welche schreibgeschützt ist, so bekommt man die Fehlermeldung "Directory Write Error". Die gleiche Meldung bekommt man auch wenn man versucht, ein File auf eine schreibgeschützte Diskette zu kopieren.

Wenn das nicht gefällt, kann mit folgender Änderung Abhilfe schaffen:
SYS0/SYS,6,2D ändere 12 in 0F

Danach bekommt man in oben genannten Fällen die Fehlermeldung "Write Protected Diskette"

HEFT
11
Januar
1985

30

Karlmut Obermann

NEWDOS/80+10

Version 2.4
26.10.85

20/12/85 17:00:02

DOS>

Extended NewDOS 80

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

Hier: I(nfo), P(ort), und DISK

Eigentlich sollte dieser Teil der NewDOS-Modifikation ja erst im nächsten Clubinfo erscheinen. Durch eine schwere Mandelentzündung und die danach fällige Erholungspause (ich erhole mich nirgends besser als am Computer) bin ich jedoch in der Lage, die Fortsetzung schon in diesem Info zu bringen.

Quellenhinweis:

Die in dieser Folge vorgestellten Befehle sind der G-DOS Version 2.2 entnommen (dort SYS29/SYS). Die gleich mitgelieferte BASIC-Erweiterung entspringt meinem Artikel "BASIC-Patch für RESTORE n" und der Artikelserie "LPRINT alles - einfacher", die von Arnulf Sopp und meiner Wenigkeit gemeinsam verbrochen wurde.

Jetzt aber ohne weitere Vorrede ans Werk!

Nur ein Byte

ist im SYS29/SYS von G-DOS 2.2 zu ändern, um es im Extended NewDOS80 +10 als SYS27/SYS verwenden zu können!

Zuvor muß man natürlich SYS27/SYS an der richtigen Stelle auf der Systemdiskette eröffnet und SYS29/SYS von einer G-DOS-Systemdisk auf das noch leere SYS27/SYS kopiert haben. Dann braucht man nur noch im relativen Sector 0 von SYS27/SYS das relative Byte 5 von FF in FD zu ändern und im SYS1/SYS die Befehle I, P und DISK mit ihren Aufruf- und Sprungbytes einzutragen:

49 81 FD 00 (für I)
50 82 FD 00 (für P) und
44 49 53 4B 83 FD (für DISK)

Schon hat man diese Befehle verfügbar!

Im neuen G-DOS 2.4 (für VideoGenie IIIs) ist der I(nfo) Befehl erheblich erweitert. Wer ein G-DOS 2.4 sein eigen nennt, kann natürlich diese Version verwenden. Die Änderungen sind dann zwar etwas umfangreicher, lohnen sich aber auf alle Fälle! Wer eine solche G-DOS-Version verfügbar hat, kann sich wegen der Änderungen direkt an mich wenden.

Wie schon erwähnt, ist die DUKE-Möglichkeit nicht gerade die eleganteste Art, die Adresse des neuen Druckertreibers in den DCB (Device Control Block) zu bringen. Daß es auch anders geht, zeigen die folgenden Seiten. Weiterhin gibt diese Lösung ein Beispiel dafür, wie man einen DOS-Befehl in seiner Funktion erweitern kann.

Wir wissen, daß man mit dem I(NFO)-Befehl nicht nur eine Reihe von Informationen über das System auf dem Bildschirm sichtbar machen, sondern auch die Treiberadressen im Tastatur-, Monitor- und Drucker-DCB auf ihre Normalwerte zurücksetzen kann. Warum sollte man den I(nfo)-Befehl nicht auch dazu mißbrauchen, einen eigenen, speziellen Grafikdruckertreiber zu inialisieren? Die Befehlsfolge I,G wäre doch eigentlich genau das richtige dafür!

Bevor man dieses neue Argument in den I-Befehl implementieren kann, muß man sich das SYS27/SYS etwas genauer anschauen.

Kommentiertes Assemblerlisting der "Sprungtabelle" und des "Argumentauswerters" von SYS27/SYS (SYS29/SYS aus G-DOS!).

Die ausgelassenen Teile sind für die jetzige Betrachtung uninteressant!

4D00	FEFD	START	ORG	4D00H	
4D02	201B		CP	OFDH	'richtiges Systemfile?
4D04	0D		JR	NZ,ERRDR	'nein, dann Fehler anzeigen!
4D05	282F		DEC	C	'I-Befehl?
4D07	0D		JR	Z,INFO	'dann Info-Routine!
4D08	CA444F		DEC	C	'P-Befehl?
4D0B	0D		JP	Z,PROT	'dann Port-Routine!
4D0C	CAB84F		DEC	C	'DISK-Befehl?
			JP	Z,DISK	'dann DISK-Routine
4D1F	3E2A	ERROR	LD	A,2A	'Code für ILLEGAL DOS FUNKTION laden
4D21	B7		OR	A	'Z-Flag zurücksetzen
4D22	C9		RET		'ins DOS und Fehler anzeigen!
4D23	211645	TAST	LD	HL,4516	'normale Treiberadresse
4D26	2211640		LD	(4016),HL	'in Tastatur-DCB laden!
4D29	C9		RET		'zurück zu WEITER
4D2A	218D05	DRUCK	LD	HL,058D	'normale Treiberadresse
4D2D	222640		LD	(4026),HL	'in Drucker-DCB laden!
4D30	C9		RET		'zurück zu WEITER
4D36	CDD54C	INFO	CALL	4CD5	'Argumente vorhanden?
4D39	2836		JR	Z,ANZEI	'nein, dann Information anzeigen!
4D3B	00		NOP		
4D3C	00		NOP		
4D3D	E5		PUSH	HL	'HL retten, wird gebraucht!
4D3E	CD474D		CALL	ARGUM	'Welches Argument?
4D41	00	WEITER	NOP		
4D42	E1		POP	HL	'HL wiederholen!
4D43	23		INC	HL	'auf nächstes Zeichen
4D44	28F0		JR	Z,INFO	'Fehler? Nein, dann weiter mit der
4D46	C9		RET		'Argumentausw., sonst Fehleranzeige!
4D47	7E	ARGUM	LD	A,(HL)	'Zeichen, auf das HL zeigt in A
4D48	FE54		CP	'T'	'ein T ?
4D4A	28D7		JR	Z,TAST	'ja, dann Tastaturroutine
4D4C	FE4D		CP	'M'	'oder vielleicht ein M ?
4D4E	2814		JR	Z,MONIT	'dann Monitorroutine
4D50	FE44		CP	'D'	'vielleicht D ?
4D52	28D6		JR	Z,DRUCK	'dann Druckroutine
4D54	FE5A		CP	'Z'	'letzter Versuch! Ein Z ?
4D56	2806		JR	Z,ZEILE	'dann Zeilenroutine!
4D58	3E34		LD	A,34	'nein, dann Fehlercode für ILLEGAL
4D5A	B7		OR	A	'KEYWORD OR SEPERATOR OR TERMINATOR
4D5B	C30944		JP	4409	'laden und Fehler anzeigen!

HEFT

11

Januar

1985

34

35

Gehen wir einmal davon aus, es wurde der DOS-Befehl I,D eingegeben. Das DOS hat mit Hilfe des Aufrufbytes SYS27 geladen und bei 4D00h gestartet. Da sich das Aufrufbyte (in diesem Fall FDh) noch im Accu befindet, wird die erste Hürde, die Überprüfung, ob ein Illegal DOS Funktion - Fehler vorliegt, erfolgreich genommen. Danach wird das C-Register einmal DECREMENTIERT (Inhalt von C - 1) und abgefragt, ob sein Inhalt jetzt gleich Null ist! Wäre dies nicht der Fall, würde solange C DECREMENTIERT, bis entweder die richtige Routine gefunden ist, oder eine Fehlermeldung ausgegeben wird! Da aber beim Aufruf von SYS27 mit dem Befehl I das C-Register mit 81h geladen und das höherwertige Nibble während des RST 28 auf 0 gesetzt wird, ist nach einmaligem DECREMENTIEREN die Bedingung für den Befehl JR,Z (springe relativ, wenn das Zero-Flag gesetzt ist) erfüllt! Es geht also bei der Routine mit dem Label INFO weiter.

Hier wird nun als erstes eine Routine im DOS aufgerufen, die feststellt, ob irgendwelche Argumente hinter dem Befehl folgen. Ist dies nicht der Fall, wird nur die Information über die verschiedenen Treiber, Himem usw. ausgegeben. Folgt dem Befehl ein Argument (Zero-Flag nicht gesetzt), so wird dieses im Argumentauswerter (4D47h-4D5Bh) auf seine Richtigkeit überprüft. Ist es ein nicht erlaubtes Argument, wird ein Illegal Keyword or Seperator or Terminator - Fehler angezeigt, ansonsten wird die entsprechende Routine angesprungen. In unserem Beispiel würde also die Routine mit dem Label DRUCK aufgerufen. Nachdem diese die normale Treiberadresse in den Drucker-DCB eingetragen hat, wird abgefragt, ob noch ein weiteres Argument folgt. Ist dies nicht der Fall, werden jetzt die Informationen angezeigt, ansonsten geht es nochmal in den Argumentauswerter.

Wie ihr sicher schon selbst festgestellt habt, ist der Platz, an dem wir mit unserer Erweiterung ansetzen müssen, der Auswerter für die möglichen Argumente des I-Befehls. Wir müssen nichts weiter tun, als nach

```
CP 'Z' 'Argument ein Z ?
JR Z,ZEILE 'dann zur Routine ZEILE springen!
durch Einfügen von
CP 'G' 'Argument ein G ?
```

```
JP Z,GRAFIK 'dann zur Routine GRAFIK springen!
auch ein ,G als gültiges Argument abzufragen und zu einer
entsprechenden Routine zu springen, dann können wir mit I,G
unsere neuen Treiber initialisieren.
```

Aber ganz so einfach ist es leider doch nicht! Wir haben nämlich keinen Platz zwischen der Abfrage von ,Z und der Anzeige eines Illegal Keyword ... - Fehlers. Wir müssen uns also damit behelfen, nach der Abfrage von ,Z an einen Platz zu springen, wo wir genügend Raum zur Verfügung haben, um unser Vorhaben zu realisieren!

Wir ersetzen aus diesem Grund die Befehle LD A,34 und OR A (4D58-4D5A) durch NOP's (No Operation = keine Operation) und den darauffolgenden JP 4409h durch einen JP 4DD0h. Dort haben wir genug Platz, um die Abfrage von ,G und die eventuelle Anzeige eines Fehlers auszuführen!

```
4DC E1810 JR 4DE0 'die früheren NOP's überspringen!!!
4DD0 FE47 NEU CP 'G' 'Ein G ?
4DD2 CA1184F JP Z,GRAFIK 'Ja, dann die neue Routine aufrufen!
4DD5 3E34 LD A,34 'Wenn nicht,
4DD7 B7 OR A 'dann Fehleranzeige
4DD8 C30944 JP 4409 'wie gehabt!
```

36

Die eigentliche Routine zur Initialisierung der neuen Druckertreibers folgt dann bei 4F18h.

```
4F18 118866 GRAFIK LD DE,M6688 'Die Adresse des neuen Treibers in DE
4F1B 1A LD A,(DE) 'erstes Byte des neuen Treibers in A
4F1C FE21 CP 21 'dieses Byte = 21 ?
4F1E 2005 JR NZ,FEHL 'nein, dann Fehler
4F20 ED532640 LD (4026),DE 'ja, dann neue Adresse in DCB!
4F24 C9 RET 'zurück zu WEITER!
4F25 3E08 LD A,08 'DEVICE NOT AVAILBLE
4F27 B7 OR A 'Fehler
4F28 C9 RET 'anzeigen!
```

Zuerst wird überprüft, ob das richtige BASIC, sprich das mit unserer Änderung, geladen ist. Ist dies nicht der Fall, wird ein DEVICE NOT AVAILBLE - Fehler angezeigt. Wenn diese Fehlermeldung nicht gefällt, kann sie selbstverständlich ändern (einfach einen anderen Fehlercode eingeben, z.B. 23h = MEMORY FAULT), ich finde sie am sinnvollsten! Ist Extended-BASIC geladen, wird die Adresse der neuen Druckerroutine in den Drucker-DCB eingetragen und zum Argumentauswerter zurückgesprungen.

Das war wirklich nicht besonders schwierig! Oder? Ihr könnt Euch ja mal selbst an einem ähnlichen Problem versuchen, das praktisch auf jeden zukommen wird, der diese Änderung des I-Befehls nutzt. Jedesmal wenn man zwischen dem Grafik- und dem normalen Druckertreiber wechselt, wird die komplette Information über Treiber, Himem, Zeichensatz usw. angezeigt und zerstört damit den Bildschirmaufbau des aufrufenden Programms. Eine Erweiterung des I-Befehls um das Argument ,N = keine Anzeige der Information, wäre also durchaus sinnvoll!

Noch zwei Hinweise zum Schluß:

1. Der Platz, an dem unsere Erweiterung des Argumentauswerters steht (4DD0h-4DD6h) ist normalerweise mit NOP's gefüllt, die zwischen zwei Programmteilen liegen und ohne irgendeine Funktion zu erfüllen durchlaufen werden. Da wir dort jetzt unsere Routine abgelegt haben, muß ein Befehl zum Überspringen dieses Bereichs vorangestellt werden!
2. Solange man in BASIC ist, kann der Grafikdruckertreiber immer initialisiert bleiben. Nur bevor man ins DOS geht, sollte der normale Druckertreiber mit CMD"I,D (oder mit CMD"S=I,G) wieder eingestellt werden. Ansonsten wird man beim ersten Versuch, etwas vom DOS aus auszudrucken (z.B.: DIR P), sein blaues Wunder erleben!

Die Änderungen, die im SYS27/SYS nötig sind, stehen direkt unter denen für BASIC/CMD. Wie ihr dem Hexdump sicher schon anseht, ist für den Einbau von ,N noch genügend Platz vorhanden!

Ich hoffe, diese Beschreibung war denen, die sie von Anfang an gelesen haben, nicht zu langatmig und/oder unverständlich! Vielleicht regt es sogar manchen dazu an, selbst einmal ein bisschen etwas in dieser Richtung zu unternehmen? Ich wünsche mir, es wäre so!

Selbstverständlich wird das nicht die letzte Änderung sein, die ich, angeregt durch meinen "Vorturner" Arnulf Sopp, im NewDOS tätige. Zur Zeit wird daran gearbeitet, die ID-Funktion auch auf dem Model 3/4 lauffähig zu machen. Weiterhin stört es mich, daß man einmal falsch eingegebene DOS-Befehle nicht korrigieren kann, sondern sie komplett neu eintippen muß! Und da ich (hoffentlich) bald über einen Mapper und damit über 96k RAM verfügen werde, wird sich in dieser Richtung auch bald mal was tun!

Bis dahin wünscht euch viel Erfolg, Euer *Karlmut Obermann!!!*

Bei der in diesem Clubinfo beschriebenen Installation des I(nfo)-Befehls können die Attribute <,H>,<,NP> und <,N> nicht benutzt werden!

I <,T><,D><,M><,Z><,H><,G><,NP><,N>

I entspricht dem GDOS-Befehl INFO.

Ohne Parameter gibt I eine Uebersicht der Teiberadressen fuer Drucker, Tastatur und Monitor, den Beginn des reservierten Speicherplatzes, eine vollstaendige Auflistung aller derzeit verfuegbaren Bildschirmzeichen und die Eingangsdaten aller aktiven Ports.

Durch das Anhaengen von T fuer Tastatur, D fuer Drucker, M fuer Monitor und Z fuer die Zeilenzahl je Seite kann man die Treiber in ihren Initialisierungszustand versetzen.

Durch das Anhaengen von H wird HIMEM auf FFFFh gesetzt. Durch das Anhaengen von G wird ein spezieller Grafikdrucker-treiber initialisiert (nur in BASIC mit entspr. Modifikation!). Mit N wird jegliche, mit NP nur die Ausgabe der Ports unterdrueckt.

P entspricht dem GDOS-Befehl PORT.

P listet alle Ports, deren INPut-Signale nicht hochgesetzt sind, mit den entsprechenden Eingabewerten auf.

DISK #=#

Mit dem Befehl DISK koennen die PDRIVE-Parameter fuer fremde Disketten im Speicher eingestellt werden, ohne sie auf der Systemdiskette zu speichern.

Dabei gibt # die Laufwerksnummer und \$ einen Buchstaben zwischen a/A und y/Y an. Der Buchstabe bezeichnet das gewuenschte Diskettenformat.

z.B.: DISK 1=P

Die den Buchstaben entsprechenden Formate entnehmen Sie bitte dem GDOS - Handbuch.

Zap im Superzap!

Wer in sein Superzap die Aenderung aus Info Nr. 8 Seite 13 eingebaut hat, kann mittels der 14 Byte langen, zusaetzlichen Aenderungen noch eine Unschoenheit von Superzap ausräumen. Nachdem die Aenderungen durchgefuehrt sind, kann man den Modify-Modus durch druecken von BREAK verlassen, ohne ein Byte auf der Diskette geaendert zu haben! Das kann manchmal ganz nuetzlich sein, vor allem dann, wenn man in einen falschen Sektor geraten ist oder sich vertippt hat.

Karntaut Obermann

DRV	00	0100	0052	0000	0000	0000	0000	0000	0000	A55R55555555555555
O	10	0000	0000	0000	0000	0000	0000	00F6	FE7B	555555555555555555
OH	20	2802	FE1B	CAF7	5CFE	1FC2	B752	E521	3952	(B-A 56 4 TRI 19R
	30	2274	5C21	5B03	22EF	5DE1	C369	5C33	333E	"to!AC" 5U 1633)
DRS	40	8C32	E15D	79E6	F66F	2600	2929	79E6	0FB5	2"Uye a&5))v.3L
B10	50	6F11	303C	CDDE	5DE1	CBC0	FE20	380C	CE48	000K 2 7 8 BL 7
32AH60		CB88	C2A6	5D1E	00C3	905C	4FCB	80FE	08CA	7. 7U 5 60 7 11
	70	AESC	CBC0	FE1F	C2AE	5CE5	218C	5D22	745C	5 7 7 5 1 1 0 7 6
	80	2105	622E	F55D	E1CB	8018	AF3A	4038	FE04	Feb. 7U 4 X8: 98 0
	90	CAC5	54C3	345D	0000	0000	0000	0000	0000	ET 40555555555555
A0	0000	0000	0000	0000	0000	0000	0000	0000	0000	555555555555555555
B0	0000	0000	0000	0000	0000	0000	0000	0000	0000	555555555555555555
FFS	C0	0000	0000	0000	0000	0000	0000	0000	0000	555555555555555555



Verbindend

Der ECB-Bus

Johannes Assenbaum

Vor acht Jahren von Mitarbeitern der Firma Kontron in die Welt gesetzt, hat der ECB-Bus recht schnell große Verbreitung gefunden. Heute ist er zumindest hierzulande der Bus schlechthin für Z80-Rechner in Steckkartenbauweise. Erstaunlicherweise findet man nur ganz selten einmal eine Beschreibung dieses Busses, die über die Pinbelegung der Steckverbinder hinausgeht. Und dabei hat der real existierende ECB-Bus durchaus auch ein paar handfeste Macken.

Als 1977 bei Kontron die ersten ECB, in vollem Wortlaut Einfach-EuropafORMAT-Computer-Baugruppen, geschaffen wurden, ist bestimmt niemandem in den Sinn gekommen, daß der ECB-Bus ein paar Jahre später De-Facto-Standard für modulare Z80-Systeme sein würde. Damals war die Z80-Bausteinfamilie ganz neu, und es gab noch keine Europakartensysteme mit diesen leistungsfähigen ICs.

Durch die hohe Integrationsdichte der neuen Bausteine konnte man das ECB-Konzept äußerst flexibel auslegen. Ausgehend von einer CPU-Karte, die bereits ein vollständiges Kleinstsystem enthält, kann man durch simples Hinzufügen von Speicher- oder Ein-/Ausgabe-Erweiterungskarten fast beliebig komplexe Rechner aufbauen. Die Grenzen des Wachstums liegen hauptsächlich im mechanischen Bereich: Da alle Speicher und Ports vollständig dekodiert und die Bussignale doppelt gepuffert sind (Treiber auf der CPU-Platine und auf den Erweiterungskarten), ist eher der Kartenträger (zumeist ein 19-Zoll-Rack) voll, als daß der Adreßraum ausgeschöpft oder der Bus überlastet wäre. Das war jedenfalls in den späten 70ern Stand der Technik.

Vielseitig

Die Flexibilität war entscheidend für den Erfolg der ECB. In der Industrie, dem zunächst angepeilten Kundenkreis, gibt es kaum Standardanwendungen. Die ECB boten nun die Möglichkeit, aus standardisierten Baugruppen schnell eine passende Hardware zusammenzustellen, ohne einen unsinnigen Überhang an nicht gebrauchten Ports oder Speichern beziehungsweise Neuentwicklungen

in Kauf nehmen zu müssen, weil es die gewünschte Konfiguration nicht gab. Das große Interesse am ECB-System und die ständig steigende Nachfrage vor allem nach spezialisierten Portkarten führte dazu, daß auch andere Firmen in diesen Markt einstiegen. Und Anfang der 80er Jahre entdeckten dann die Hobbyisten die Z80-Bausteine und den ECB-Bus für sich, der daraufhin einen wahren Boom erlebte.

Diese Entwicklung brachte aber gewisse Probleme mit sich: Kontron hatte nicht von vornherein alle 64 Pins der verwendeten VG-Steckverbinder irgendwelchen Signalen zugeordnet. Einige Leitungen wurden für spätere Erweiterungen reserviert, seinerzeit dachte man an den Ausbau zu einem 16-Bit-System mit der Z8000-CPU. Doch schon ziemlich bald zeigte sich, daß ein 16-Bit-ECB-Bus gegen die Übermacht der vorhandenen 8-Bit-Karten keine Chance mehr hätte. Man entschloß sich, den 16-Bit-Plan aufzugeben und die so gewonnenen Busleitungen für realistischere Zwecke zu verwenden. Für zusätzliche Adreßleitungen zum Beispiel, zumal im Zuge ständig steigender Integrationsdichte und fallender IC-Preise Hauptspeicher über 64 KByte keine Utopie mehr darstellten.

Nun kamen aber mehrere Firmen etwa gleichzeitig auf diese Idee. Und da keine stark genug war, von den anderen als Vorreiter anerkannt zu werden, gibt es heute mehrere ECB-Bus-Belegungen und damit ein paar grundsätzliche Inkompatibilitäten zwischen den Platinen verschiedener Hersteller.

Das betrifft in erster Linie die zusätzlichen Adressierungssignale für Speicher über 64 KByte, aber auch einige

Hilfssignale wie zum Beispiel die Prioritätssteuerung für DMA-Einheiten (BAI/BAO). Dabei wäre die unterschiedliche Bus-

Unmöglich

belegung noch das kleinere Übel, so etwas ist bei Bedarf mit Messer, Lötkolben und ein paar Zentimetern Schaldraht schnell behoben. Beim Umschalten der Speicherbänke kommen jedoch unterschiedliche Auswahlverfahren zum Einsatz. So liegen bei Kontron acht dekodierte Bank-Select-Signale auf dem ECB-Bus, mit denen man bis zu acht 64-KByte-Bänke ansteuern kann. Andere Hersteller haben den Adreßbus um einige Bits erweitert, die erst auf den Speicherkarten dekodiert werden. Hauptsächlich verwendet man vier weitere Adreßsignale, womit insgesamt ein Megabyte für Speicher zur Verfügung steht. Anwender des zweiten Verfahrens sind unter anderem auch Janich & Klaas sowie der Elektronikladen Giesler & Danne (ELZET 80), deren Busse bei Hobbyisten weit häufiger anzutreffen sind als der Kontron-Bus.

Leider blieb es nicht bei diesen wenigen kritischen Stellen, die im wesentlichen nur bei Speichererweiterungen zum Tragen kommen. Besonders im Hobby-Bereich erschienen Rechnerkarten auf dem Markt, bei denen die ursprüngliche Busdefinition etwas eigenwillig ausgelegt war. Beispielsweise wurden Stromversorgungsleitungen umgelegt oder die Puffer weggelassen. Gelegentlich übernahm man den ECB-Bus auch für Systeme mit einer 'Nicht-Z80-CPU' (8085, 8086/88 und seit neuerem 68008), was fast jedesmal weitere Inkompatibilitäten mit sich brachte, da dann ein anderes Timing auf dem Bus herrscht. C't-Leser, die schon länger dabei sind, kennen so einen Fall: den c't 86 beziehungsweise dessen 'jüngeren Bruder', den c't 68 ECB.

Vorsichtig

Wer sich heute eine der vielen ECB-Bus-kompatiblen Rechnerkarten kaufen will, muß also schon genau wissen, was tatsächlich auf seinem Bus los ist. Das Prinzip 'Auspacken - Einschalten - Geh' ist nur dann angebracht, wenn man sich der Kompatibilität wirklich sicher

	a	c
+5V	1	+5V
D5	2	D5
D4	3	D7
D3	4	D2
D4	5	A8
A2	6	A3
A4	7	A1
A5	8	A8
A6	9	A7
/MAIT	10	#
/BUSRD	11	#
+12V	12	#
+12V	13	#
-5V	14	D1
-5V	15	-12V
2<phi>	16	-1E0
	17	A11
A14	18	A18
+15V	19	#
/M1	20	/M1
	21	/INT
	22	/M2
	23	#
UCHDS	24	/RD
	25	/HALT
	26	/PWRL
/IORD	27	A12
/RFSH	28	A15
A13	29	<phi>
A9	30	/HRED
/BUSAK	31	/RESET
Gnd	32	Gnd

Tabelle 1. Die ECB-Busbelegung, soweit sie einheitlich gehandhabt wird, die umstrittenen Pins („#“) siehe Tabelle 2. Mit „-“ und „+“ markierte Pins sind beim c't 86/68 ECB nicht beziehungsweise anderweitig verwendet (siehe c't 11/85, Seite 63).

sein kann; blindes Vertrauen auf diesbezügliche Werbeaussagen führt in der Regel zu einem 'Geh' nicht', unter Umständen sogar zum Tode etlicher ICs, wenn nämlich die ±12V-Versorgung an die falschen Pins zu liegen kommt.

Als Hilfestellung zeigen die Tabellen die original Kontron-ECB-Busbelegung, die der beiden anderen genannten Firmen sowie die unserer Rechnerprojekte c't 86/68 ECB und c't 80 (PROF-80). Bitte beachten Sie, daß auch die Richtung der Signale von Bedeutung sein kann. Dazu ein letztes Beispiel aus der Praxis: Es gibt Systeme, bei denen die Signale RESET und FWRL gleichberechtigt sind (beides Ausgänge der CPU-Karte). Laut ECB-Busdefinition ist RESET jedoch ein Eingang, über den vom Bus aus ein System-Reset ausgelöst werden kann. Kombiniert man nun eine solche CPU-Karte mit 'normalen' ECB-kompatiblen Baugruppen, gelangt eine Reset-Anforderung auf dem Bus nicht mehr zur CPU. Umgekehrt kann es auch passieren, daß der Reset-Impuls von der CPU-Karte zur Peripherie (PWRL) nicht durchkommt, weil deren Reset-Eingang am anderen Pin liegt und die Anforderung nicht über den Bus gelaufen ist. □

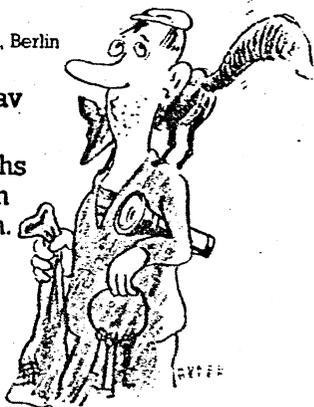
Dänisch: En fjer bliver der i hons. Aus einer Feder werden fünf Hühner. Aus einer Mücke einen Elefanten machen.

Dorte Salomon, Berlin

Have en rav bag øret.

Einen Fuchs hinter dem Ohr haben.

Etwas im Schilde führen.



Englisch: To get on the gravy train. Auf den Bratensofenzug aufspringen. Absahnen.

To paint the town red. Die Stadt rot anmalen. Auf die Pauke hauen.

Cavin Specht, Regensburg

Pin	Bedeutung Kontron	J & K	ELZET80	c't 80	c't 84/88 ECB
19c	MSB0 (A16)	A16	---	reserv.	PCLK <phi>/2
12a	BA1	BA1	A18	A18	A18
12c	MSB1 (A17)	A17	A19	A19	A19
13c	MSB2 (A18)	A18	---	reserv.	-12V
14a	MSB3 (A19)	A19	-12V	-12V	CLOCK <phi>
15a	-12V	-5V	---	reserv.	-5V
17a	BA0	BA0	A17	A17	A17
19c	MSB5 (A21)	---	A16	A16	A16
21a	MSB6 (A22)	---	---	reserv.	reserv.
22a	MSB7 (A23)	/DMARDY	/BOOT	---	reserv.
23a	RET10R	/PF	/BA1	BA1	(-)
23c	MSB4 (A20)	/SEL	A1D BLNK	---	reserv.
24a	---	UCHDS	UCHDS	UBAT	SU-BAT (-)
25a	n<phi>	---	/BA0	/BA0	BA0 (-)
26a	/WRITEN	/DESLECT	---	reserv.	reserv.

Tabelle 2. Die zunächst undefinierten Leitungen des ECB-Bus und was aus ihnen geworden ist. Man beachte, daß Kontron die Pins 15a und 24a nachträglich umbelegt hat. Die Markierungen „-“ in der Spalte c't 86/68 ECB haben dieselbe Bedeutung wie in Tabelle 1.

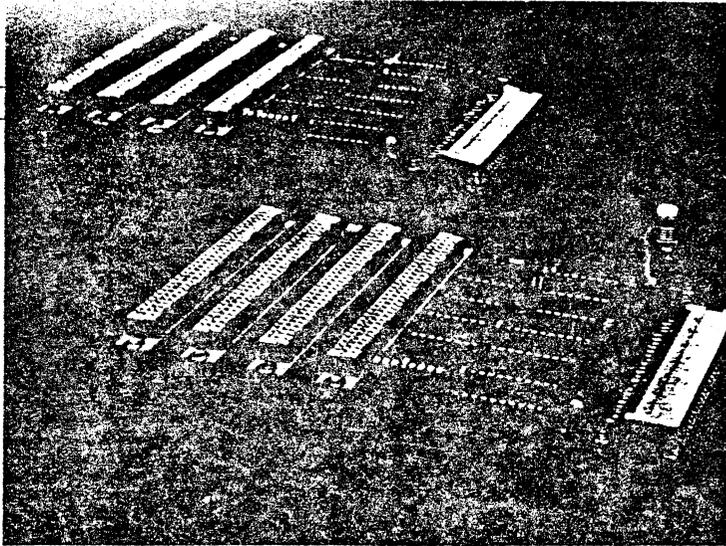
Signal	Pin	Richtung (Fan in/Out)	Bedeutung	A12	27c	A	(9 TTL)
A8	5c	A	(9 TTL) Adreßbus	A12	27a	A	(9 TTL)
A2	6a	A	(9 TTL) der CPU	A14	18a	A	(9 TTL)
A3	6c	A	(9 TTL) der CPU	A15	28c	A	(9 TTL)
A4	7a	A	(9 TTL) der CPU	D8	2c	E/A	(Fan Out)
A5	8a	A	(9 TTL) der CPU	D1	14c	E/A	25mA
A6	9a	A	(9 TTL) (hoch-ohmig bei BUSAK)	D2	4c	E/A	bei Low
A7	9c	A	(9 TTL) (hoch-ohmig bei BUSAK)	D3	4a	E/A	
A8	8c	A	(9 TTL) (hoch-ohmig bei BUSAK)	D4	5a	E/A	
A9	30a	A	(9 TTL) (hoch-ohmig bei BUSAK)	D5	2a	E/A	(Fan In)
A10	18c	A	(9 TTL) (hoch-ohmig bei BUSAK)	D6	3a	E/A	0,25mA
A11	17c	A	(9 TTL) (hoch-ohmig bei BUSAK)	D7	3c	E/A	bei Low
				/MRO	30c	A	(9 TTL) CPU-Steuer

Tabelle 3. Eine Kurzübersicht über die Bedeutung und Richtung (aus der 'Sicht' der CPU-Karte) der Signale auf dem ECB-Bus. Die Angaben zur Treiberleistung (Fan Out) sind Mindestwerte. Nicht genannte Signale sind nur in dem betreffenden System von Belang.

/IORD	27a	A	(9 TTL)	Ausgänge
/RD	24c	A	(9 TTL)	'hohehmig
/M2	22c	A	(9 TTL)	bei BUSAK
/M1	20a	A	(9 TTL)	(Tri-State-Treiber nur
/RFSH	28a	A	(9 TTL)	bei mehr als
/HALT	25c	A	(9 TTL)	einer CPU im System)
/MAIT	10a	E	(Pullup)	CPU-Steuer-
/INT	21c	E	4k7	eingänge
/M1	20c	E		
/RESET	31c	E	(plus 1 TTL-Eingang)	
IE1	11c	E	(Pullup 4k7)	Interrupt-
1E0	16c	A	(1 TTL)	Daisy-Chain
/PWRL	26c	A	(9 TTL)	Peripherie-
(PHI)	29c	A	(keine Angaben)	Reset
2<PHI>	16a	A		CPU-Takt
				Taktsignal
				doppelter Frequenz
/BUSRD	11a	E	(Pullup 4k7)	DMA-Steuer-
/BUSAK	31a	A	(9 TTL)	Signale
+5V	1a,c			Stromver-
Gnd	32a,c			sorgungen
+12V	12a			
-5V	13a			(-12V bei Kontron)
-12V	14a			ELZET80/c't88
+15V (+12V)	19a			Für U,24
-15V (-12V)	15c			und Analoge
UCHDS/UBAT	24a			Backup-Aku
				(Spannung unterschiedlich,
				bei Kontron nicht mehr vorgesehen)
n<PHI>	A			Weitere Signale:
BA1	E			noch ein Takt
BA0	A			(nur Kontron)
				Daisy-Chain
				für Bus
A16...19	A	(9 TTL)		zusätzliche
				Adressen
MSB6...7	A	(9 TTL)		Kontron: Bank-
oder				select- oder
A16...23	A	(9 TTL)		Adreßsignale

Andreas Stiller

Schluß mit der Spezial-Hardware! Es lebe die Kompatibilität. So wie man beispielsweise mit SuperTape durch ein einheitliches Kassettenverfahren das byzantinische Formatwirrwarr verschiedener Rechner beseitigt, ist es auch sinnvoll, sich für alle Z80-Rechner auf einen Anschluß-Standard für die Peripherie zu einigen. Hierfür bietet sich der bei professionellen Rechnern weitverbreitete ECB-Bus an, den mit Hilfe eines Adapters nun auch die Schneiderfamilie 'an-treiben' kann.



Schneider findet Anschluß

ECB-Adapter für alle CPCs

Was der ECB-Bus eigentlich ist und welche Bedingungen er erfüllen muß, war uns in einem eigenen Artikel an anderer Stelle in diesem Heft wert. Für dieses Bus-System hat c't nämlich bereits etliche Hardware-Projekte vorgestellt:

- Intelligente Floppy-Karte, die eigenständig CP/M fährt
- Eprommer
- I/O-Karte mit CTC, SIO und 2 x PIO
- EPAC-80 (der auch als I/O-Karte zu gebrauchen ist)
- 1-MByte-RAM-Karte
- Bus-Monitor
- Floppy-Karte (vom c't86) mit WD 2797
- I/O-Karte (vom c't186) mit 8153A, 8255A, 8251A, mit SuperTape-Kassettschnittstelle

Diese umfangreiche Palette von Peripherie-Karten ist nun auch für die Schneiders verfügbar, wobei die RAM-Karte allerdings noch ein wenig Hardware-Unterstützung benötigt. Für den Eprommer gibt es eine umfangreiche Software (unter CP/M), die zur Zeit auf die etwas beschränkte TPA (freier Speicher) und den 16-Bit-I/O

der kleinen CPCs angepaßt wird.

Hürden

Allzuleist haben es dabei die Amstrad-Entwickler nicht gemacht, den CPCs die ECB-Welt in ihrer Vielfalt zu erschließen. Für Schwierigkeiten sorgen nicht nur die unüblichen - und nicht ausdekodierten - 16-Bit-Ports sowie die ungewöhnliche Interrupt-Struktur (siehe Schneiders Kern, Teil 1, c't 8/85), sondern auch die Unterschiede innerhalb der inzwischen auf drei Mitglieder angewachsenen CPC-Familie.

Zweckmäßigerweise bietet der Adapter auch weiterhin einen Standard-Platinenschluß für die Schneider-Peripherie. Das ist besonders für den CPC 464 wichtig, da sich dadurch problemlos und ohne Umlötaktion eine Floppy-Station (von Schneider, Vortex etc.) anschließen läßt. Um auch hier etwas 'Leistungsreserve' zu bekommen, erhalten auch die Signale für den Platinenschluß genau wie die ECB-Bus-Signale über die vier Bustreiber (LS244 und

LS245) eine kräftige Verstärkung.

Beim CPC 664 und CPC 6128 befinden sich Floppy-Controler und Floppy-ROM ja bereits mit im Rechnergehäuse, also auf der anderen Seite des Adapters, was eine etwas aufwendigere Umschaltung der Datenrichtung (des bidirektionalen LS245) notwendig macht.

$$16/2 = 256$$

Mit 16 Bit lassen sich theoretisch 65536 Portadressen ansprechen. Da aber wohl kaum jemand einen so riesigen Peripherie-Bedarf hat, haben sich die CPC-Entwickler auf maximal zehn Geräte beschränkt (vier interne und sechs externe). Das Adreßbit A10 entscheidet dabei, ob es sich um ein externes oder ein im Rechner bereits eingebautes internes 'Gerät' (Gate-Array, Video-Chip, Centronics-Ausgang, 8255-PIO) handeln soll. A13 dient zusammen mit den Datenbits zur Selektion eventuell vorhandener Expansions-ROMs. Die verbleibenden vier Bits (A9, A8, A1, A0) können beliebig zur Steuerung des durch die übrigen Bits ausgewählten Geräts eingesetzt werden.

Für den ECB-Adapter ist insbesondere das Adreß-Bit A10 (interner/externer Port) von Bedeutung. Um auf einen ECB-Port zugreifen zu können, muß A10 low sein (und A13 sowie A9 auf high). Nur dann wird nämlich das Signal IORQ über den Adreßdekoder LS138 (IC9,

Ausgang Y5) an den ECB-Bus weitergereicht.

Die Adapterschaltung ist so ausgelegt, daß sie auch eine korrekte Interrupt-Behandlung für den ECB-Bus ermöglicht. Dazu ist es nötig, daß die Peripherie die Antwort auf einen Interrupt von der CPU erhält, das sogenannte Interrupt Acknowledge (INTACK), das aus der Signalkombination M1 und IORQ gleichzeitig auf low - besteht. Diese Kombination wird von einem OR-Gatter in IC10 'erkannt', und das erzeugte INTACK-Signal sorgt dafür, daß unabhängig von irgendwelchen Portadressen das IORQ-Signal auf den ECB-Bus kommt (über ein UND-Gatter von IC6).

Auf der Software-Seite ist für einen Zugriff auf einen 8-Bit-ECB-Port also in bester Schneider-Manier weiterhin eine 16 Bit Portadresse nötig. Hierbei müssen die oberen fünf Adreßbits (A11 bis A15) gesetzt sein, da sich sonst vielleicht gleichzeitig der Video-Controller angesprochen fühlt oder der Drucker losrattert.

Auch A9 muß bei einem Portzugriff high sein, da die Kombination A9=0, A10=0, A13=1 als Software-Reset für die ECB-Peripherie interpretiert wird. Folglich sind die Adressen &FAxx und &FBxx für die ECB-Ports xx und &F8xx und &F9xx für den Software-Reset ausserorden. Die Adreßlage für den Reset ist so gewählt, daß sie der Adresse des System-Resets (&F8FF) entspricht, den die Schneider-Firmware beim Einschalten oder nach

CTRL-SHIFT-ESC ausgibt. Für einen ordnungsgemäßen Reset der üblichen Z80-Bausteine muß das Reset-Signal einige Taktzyklen auf low liegen, so beispielsweise bei Mostek:

SIO: ein Takt,
PIO: zwei Takte,
CTC: drei Takte

Normalerweise ist die Dauer eines I/O-Zugriffs mit etwa zweieinhalb Takten dafür etwas knapp. Beim Schneider aber helfen die zusätzlichen WAIT-Zyklen kräftig mit (siehe 'Vertrackte Warte-Takte', c't 9/85), so daß der Reset-Puls bei OUT (C), A drei zusätzliche Takte erhält und nun lang genug ist.

Weitere Adressen sind &DExx und &DFxx, die für die ROM-Selektion verantwortlich sind. Das dazu erzeugte Signal EX-TROM (Ausgang Y6 von IC9) gelangt allerdings nicht auf den ECB-BUS, es steht aber an der Steckleiste X2 zur Verfügung.

Für Spezialzwecke kann es manchmal nützlich sein, ein Strobe-Signal, also einen kurzen Puls, an einem Portausgang zu haben. So beispielsweise für ein DATA-STROBE-Signal an einem Druckerport. Hiermit ließe sich also auch das berühmte-berühmte achte Bit der verstümmelten Centronics-Schnittstelle restaurieren.

Solche Strobe-Signale kann man den noch unbenutzten Ausgängen des Adreßdekoders (IC9) entnehmen, von denen eins (Y0, Adresse &D9xx) auch auf die Steckleiste X2 geführt ist.

Die Länge dieser Strobe-Signale beträgt ebenso wie beim Software-Reset etwa 5,5 Taktzyklen, also knapp 1,5 Mikrosekunden.

Durchgeschleift

Genau wie auch beim ECB-Bus erfährt das IORQ-Signal für den Platinenschluß der Schneider-Peripherie eine Sonderbehandlung. Das wäre zwar nicht unbedingt nötig, ermöglicht aber, viele Portadressen zu retten, die sonst durch die sehr unvollständig dekodierten Peripherie-Geräte verloren gingen. Beispielsweise belegt die Floppy-Station des CPC 464 bei einem externen Portzugriff (&FAxx oder &FBxx) sämtliche Adressen zwischen xx=&00 und xx=&7F, so daß für den

ECB-Bus nur noch die Portadressen xx=&80 bis xx=&FF verbleiben. Beim CPC 664 und CPC 6128 kann man allerdings diese Einschränkung der Adreßfreiheit nur verhindern, wenn man einen Eingriff in den Rechner vornimmt, um hier die Manipulation am IORQ-Signal zwischen CPU und Floppy-Station vorzunehmen. Das sollte man aber tunlichst lassen und sich lieber mit den noch freien 128 Portadressen zufriedengeben. Die einstellbare Dekodier-Logik (J3 und IC8) sorgt dann dafür, daß nicht noch zu viele weitere Portadressen durch Anschluß eines Modems oder anderer nur teilweise dekodierter Peripherie-Geräte verlorengehen.

Es gibt vier Wege für das IORQ-Signal, um an den Platinenschluß zu gelangen: Software-Reset, ROM-Select, Interrupt-Acknowledge und Zugriff auf einen externen Port, wenn die Adreßbits A2 bis A7 die durch S1.1 bis S1.6 einstellbare Bedingung erfüllen. Das Vierfach-UND (IC5a) führt diese Wege zusammen.

S1 kann man wahlweise mit Lötbrücken, Stiftleisten und Jumpern oder einem Achtfach-DIL-Schalter bestücken. Ist ein Schalter von S1.1 bis S1.6 auf 'on', so muß das zugehörige Adreßbit high sein, damit die Dekodierbedingung erfüllt ist. S1.8 ist auf 'on' zu stellen, falls der Rechner eine eingebaute Floppy-Station hat, also beim 664 und beim 6128. Beim CPC 464 ist die zugehörige Schalterstellung von S1.8 'off'.

Die übliche Schalterstellung für den CPC 464 mit Floppy-Station ist dann:

S1.8 off	S1.4 on
S1.7 on	S1.3 on
S1.6 off	S1.2 on
S1.5 on	S1.1 on

Die Floppy-Station nimmt dann im 'Ram der Portadressen' nur noch den Bereich von xx=&7C bis xx=&7F in Anspruch. Alle anderen Adressen stehen dem ECB-Bus zur freien Verfügung.

Beim CPC 664 oder 6128 ist die übliche Schalterstellung bei Anschluß einer seriellen Schnittstelle:

S1.8 on	S1.4 on
S1.7 on	S1.3 on
S1.6 on	S1.2 on
S1.5 off	S1.1 on

Die für den ECB-Bus freibleibenden 8-Bit-Adressen sind demnach xx=&80 bis &FF, wovon die serielle Schnittstelle nur die Adressen &DC bis &DF belegt.

Richtungswechsel

Da die Daten in zwei Richtungen 'fließen' können - von der Peripherie zur CPU und umgekehrt -, hat die Adapterschaltung dafür zu sorgen, den bidirektionalen Datentreiber (LS245, IC3) immer rechtzeitig umzuschalten. In der Ruhestellung ist der Treiber auf Schreiben (aus der Sicht der CPU) gestellt. Die Peripherie akzeptiert die anstehenden Daten aber nur, falls gleichzeitig auch das WR-Signal aktiviert ist.

Unter drei Voraussetzungen soll der Treiber nun von außen ankommende Daten auf den CPU-Datenbus legen.

- Lesen von externem ROM oder RAM
- Lesen von einem externen Port
- beim Interrupt Acknowledge

Die Schwierigkeit besteht nun darin zu erkennen, ob es sich nun um einen internen oder externen Zugriff handelt. Bei externem ROM hat die Peripherie mitzuhelfen: Wenn sich irgendwo ein externes ROM angesprochen fühlt (zum Beispiel in der externen Floppy-Station), muß dieses Gerät ein besonderes Signal (ROMDIS, aktive high) liefern, das das interne ROM ausblendet. Ebenso ist es bei externem RAM mit dem Signal RAMDIS. Die Umschaltlogik wertet nun diese Signale aus (über IC10c, IC11d, IC10a, IC5b), wobei die Dioden D1 und D2 dafür sorgen, daß eventuell vom Rechner herrührende Ausblendsignale (bei eingebaute Floppy-Station) unberücksichtigt bleiben. In diesem Fall bleibt der Treiber-Baustein daher weiterhin auf Schreiben stehen, so daß er nicht die vom internen Floppy-ROM kommenden Daten überblendet (der LS245 ist nämlich weitaus stärker).

Bei den Ports ist die Sache scheinbar einfacher, da ja wie schon beschrieben das Adreßbit A10 über intern/extern entscheidet. Nun wird aber die Floppy-Station grundsätzlich als externes Gerät angesehen, egal ob sie eingebaut ist oder

nicht. Daher muß man über den Schalter S1.8 der Umschaltlogik die 'Lage' der Floppy-Station mitteilen. Bei S1.8 auf 'on' (eingebaute Station) interpretiert sie alle Portadressen xx=&7F als intern und schaltet beim Lesezugriff den LS245 auf Schreiben (IC11a, IC12c, IC10c, IC5b).

Mit IM 2

Ein besonderes Augenmerk verdient die Behandlung der Interrupts. Viele ECB-Bus-Karten haben Interrupt-Quellen 'on board', die korrekt bedient werden wollen.

Der Schneider läuft normalerweise im Interrupt-Mode 1, das heißt, daß die CPU bei einem Interrupt immer das Programm ab &38 ausführt. Da aber die CPCs auch über periodische, interne Interrupts verfügen (300mal pro Sekunde), ergibt sich das Problem, wie man zwischen externen und internen Interrupts unterscheiden kann.

Wie schon in der Serie über des Schneiders Kern ausgeführt, haben sich die CPC-Entwickler hier etwas Besonderes einfallen lassen. Die Interrupt-Routine prüft nämlich nach einiger Zeit nach, ob die Interrupt-Anforderung immer noch besteht. Falls nein, interpretiert sie den Interrupt als intern, im anderen Fall vermutet sie eine externe Quelle und führt dann die User-Interrupt-Routine (Sprungvektor steht in &3B) aus. Diese Behandlungsroutine muß dann gezielt durch einen OUT-Befehl die unterbrechende Einheit zur Rücknahme der Interrupt-Anforderung veranlassen.

Diese Art der Interrupt-Behandlung entspricht allerdings nicht den Bedürfnissen der üblichen Z80-Peripherie-Bausteine, wie man sie meistens auf den ECB-Karten vorfindet. Man hat jedoch mehrere Möglichkeiten zur Abhilfe:

- Man ersetzt die Interrupt-Routine (ab &B939 beim CPC 464 und &B941 bei CPC 664) durch eine eigene, die auf eine andere Art die Herkunft des Interrupts ermittelt, zum Beispiel, indem sie die in Frage kommenden Geräte abfragt, ob sie einer Behandlung bedürfen (Polling).

- Die Adapterschaltung bietet die Möglichkeit, (Steckbrücke J2 gesetzt) mittels eines Monoflops (IC7a) die Interrupt-

HEFT

11

Januar

1985

42

...einige Mikrosekunden zu verlängern, so daß das Betriebssystem den externen Interrupt richtig erkennt. Die Zeitkonstante des Monoflops sollte dabei normalerweise etwa 30 Mikrosekunden betragen, was sich durch entsprechende Wahl des Zeitglieds R1/CI erreichen läßt. Da $\tau \approx 0.7 \times R \times C$ ist, käme beispielsweise $R1 = 3k9$ und $C1 = 10 nF$ in Frage.

Die weitaus komfortabelste Möglichkeit bietet der Interrupt-Mode 2, bei dem das auslösende Gerät einen Interrupt-Vektor liefert. Der Adapter unterstützt diesen mächtigen Modus, indem er dafür sorgt, daß der Interrupt-Vektor richtig eingelesen wird, und er setzt bei einem internen Interrupt diesen Vektor mit Sicherheit auf &FF. Der dadurch adressierte Tabellenplatz (Interrupt-Register I: high Byte, Interrupt-Vektor: low Byte), der hier ausnahmsweise ungerade ist, muß die Adresse der normalen Interrupt-Routine beinhalten. Wenn also beispielsweise die Interrupt-Tabelle ab Adresse &B600 stehen soll (im Sound-Puffer beim CPC 464), so muß man vor dem Umschalten auf IM 2 die Interrupt-Adresse eintragen:

&B6FF: 38
&B700: 00

Zwei Dinge sind beim Arbeiten mit Interrupts noch zu beachten. Zum einen müssen Tabelle und Behandlungs-Routinen im umgebankten RAM stehen, zum anderen ist einzuplanen, daß sich die Reaktionszeit auf einen Interrupt manchmal etwas hinziehen kann. Wenn der Rechner sich nämlich gerade mit der internen Interrupt-Behandlung abmüht, sind externe Interrupts erst einmal auf Eis gelegt. Ohne Eingriff in den Rechner läßt sich die interne Interrupt-Quelle leider nicht abstellen. Wenn Zeitnot besteht, kann man aber eine eventuelle Verzögerungszeit durch den internen Interrupt gewaltig abkürzen, indem man die Behandlungsroutine ersetzt durch:

EI
RETI

Komm zurück

Wenn sich der Rechner mal wieder jenseits von Gut und Böse im 'Nirvana' befindet und sich

auch mittels CTRL-SHIFT-ESC nicht zur Rückkehr bewegen läßt, bleibt einem nur noch ein Ausweg: der Reset. Das Aus- und Einschalten des Rechners ist jedoch dafür nicht die beste Lösung, da unter Umständen noch versehentlich eingelegte Disketten Schaden nehmen könnten. Außerdem hat man so mit Sicherheit keine Chance mehr, wertvollen Speicherinhalt zu retten.

Der Adapter bietet deshalb auch eine kleine Reset-Logik, die über den Taster T1 und den Monoflop IC7b einen kurzen Reset-Puls auf die BUSRESET-Leitung legt.

Es muß allerdings die Software mithelfen, daß bei der nun folgenden Initialisierung nicht doch noch der gesamte Speicherinhalt gelöscht wird, ein unangenehmer Effekt, den nicht das Kern- sondern das BASIC-ROM zu verantworten hat. Falls jedoch ein anderes Hintergrund-ROM die Kontrolle übernimmt, vielleicht sogar mit einem Monitor-Programm, kann man diesen Löscheffekt verhindern.

Den Reset-Vektor im RAM ab &0000 auf eine eigene Routine zu verbiegen, bringt nichts ein, da zumindest der verlagereigene CPC 464 beim Reset scheinbar immer das ROM auswählt. Andersfalls spielte man soost va banque, ob gerade das RAM (Sekt) oder das parallele Kern-ROM (Selters) selektiert ist.

Optionen

Wer weder die Interrupts noch den speichererhaltenden Reset benötigt, kann auf die beiden Monoflops (IC7) und die Zeitglieder R1, C1, R2, C2 auch verzichten. Um dann mittels T1 einen einfachen Reset auslösen zu können, ist eine Verbindung

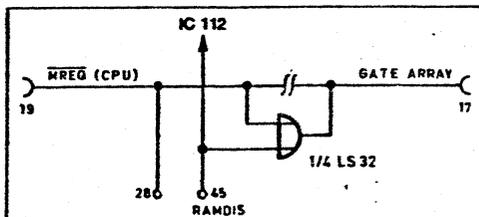


Bild 1. Wenn man dieses Gatter (huckepack über den vorhandenen 74 LS 32) in den CPC einbaut, kann man problemlos externes RAM anschließen.

zwischen Pin 9 und Pin 12 von IC7 zu ziehen.

Noch mehr sparen läßt sich, wenn man gar keine Schneider-Peripherie am durchgeschleiften Bus anschließen möchte oder wenn man nur wenige Portadressen auf dem ECB-Bus benötigt. Dann können IC5, RN2 und der DIL-Schalter J3 (wahlweise Jumperfeld) entfallen. Lediglich zwei Verbindungen sind statt dessen nötig: Pin 9 von IC11 an Pin 8 von IC7 und Pin 1 von IC11 entweder an plus (CPC 464) oder an Pin 8 von J3 (CPC 664 oder 6128).

Auch die Dioden D4 und D5 samt Steckbrücken J3 und J4 sind 'optional'. Über sie kann man die Ausblendsignale für RAM und ROM vom ECB-Bus empfangen. Um problemlos mit RAM-Erweiterungen arbeiten zu können, sollte man aber einen Eingriff in den Rechner wagen und das Beschreiben des internen RAMs bei Zugriff auf externe Speicher verhindern (siehe Bild 1).

Die vorgestellte Lötbrücke LI ist für Spezialisten gedacht, die signalgesteuert den Schneider vom ECB-Bus abkoppeln wollen. So kann man entscheiden, ob dann der Takt mit abgehängt wird (vorgegestellt) oder weiterhin als Bus Clock auf dem Bus verbleibt. Auch das Schicksal der Signale RFSH, M1 und HALT hängt von der Einstellung ab. Normalerweise sind sie ebenso wie die anderen CPU-Signale abhängig vom Abkoppelsignal (Pin 12 von X2).

Wer all diese Möglichkeiten nutzen möchte, kann dafür eine Pfostenstifteleiste samt Steckbrücke einsetzen und die Voreinstellung aufretten.

Schließlich bleibt noch die Steckbrücke J1, die darüber entscheidet, woher der Adapter

und der Bus den 'Soft' bekommen. Da das Netzteil im Schneider-Monitor noch einige Reserven hat, reicht die Stromversorgung für den ECB-Betrieb meist noch aus (J1 gesetzt). Im anderen Fall muß wohl oder übel ein zusätzliches Netzteil (5V) her, das man an die vorgesehenen Lötstifte anschließen kann (J1 offen). R6 sorgt dann dafür, daß die Treiber so lange hochohmig sind, bis auch der Rechner eingeschaltet ist.

Auf die Plätze...

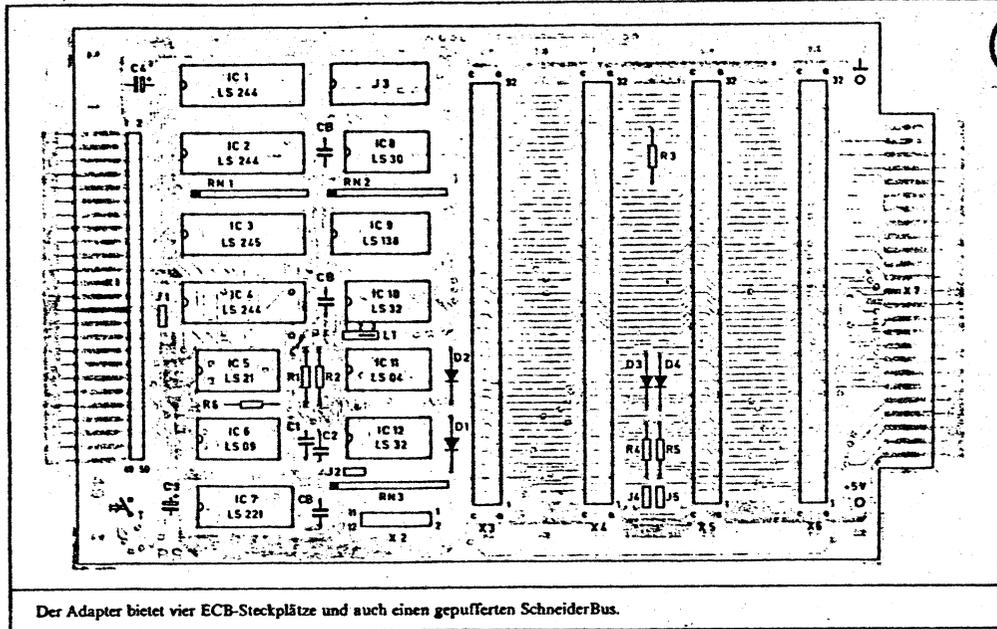
Je nach CPC-Typ variiert der Anschluß an den Rechner ein wenig. Beim CPC 464 und 664 hat man die Wahl, ob man den Adapter auf 'Beinchen' stellt (30 mm bei 464, 10 mm beim 664) und mit einem Platinenstecker direkt anschließt oder ob man lieber ein Stück Flachbandkabel dazwischen hat. Dann erfolgt am besten der Anschluß am Adapter über die Pfostenstifteleiste. Letzteres gilt auch für den 6128, der an der Rechnerseite einen 50poligen 'Centronics'-Stecker erfordert. Beim Direktanschluß des CPC 464 ist noch zu beachten, daß der Platinenstecker leicht nach unten abzuwinkeln ist, damit er auf die schief eingebaute Platine richtig paßt.

...Fertig, los!

Wie üblich sollte man nach dem Zusammenbau die Platine noch einmal sorgfältig auf kalte Lötstellen, Kurzschlüsse und andere Unschönheiten überprüfen. Als nächstes sollte man wenn möglich die Stromaufnahme der Platine messen, die etwa 120 mA verschluckt. Nun ist es Zeit, die Platine (mit der richtigen DIL-Schalterstellung) an den Rechner anzuschließen - natürlich im ausgeschalteten Zustand.

Als erstes kann man die Funktion des Reset-Tasters testen, dessen Wirkung deutlich sichtbar sein müßte. Falls sich nichts tut, haben Sie vielleicht die Steckbrücke J1 für die Versorgungsspannung vergessen.

Für das weitere Testen aller Busfunktionen und zur eventuellen Fehlersuche wäre der ECB-Bus-Monitor aus c't 9/85 ideal. Falls Sie selbst sich noch keinen zusammengebaut haben: vielleicht könnte ein Freund (oder auch eine Freundin?) Ih-



Der Adapter bietet vier ECB-Steckplätze und auch einen gepufferten Schneider-Bus.

nen für ein paar Stunden dieses praktische Werkzeug überlassen.

Zum Testen stellen Sie den Monitor auf &FB80, WR und IORQ. Wenn Sie nun

OUT (&FB80), &FF

ausgeben, müßten alle Datenlämpchen leuchten. Sie sollten an diese Adresse mehrere verschiedene Werte ausgeben, um die Datenleitung auf eventuelle Kurzschlüsse zu überprüfen. Mit

OUT (&F8FF), 0

müßten alle Lämpchen wieder verlöschen (Software-Reset).

Wenn Sie den Monitor auf &BDD3, M1 und RD stellen, Wait zulassen und dann einen Reset auslösen, passiert auf dem Bildschirm erstmal gar nichts - nur das Wait-Lämpchen leuchtet auf. Nun läßt sich mit dem Wait-Taster auf der Monitor-Karte Zeichen für Zeichen der Initialisierungsmeldung durchschreiten.

Wenn soweit alles läuft, können Sie davon ausgehen, daß der Adapter funktionsbereit ist und sich nun mit der Eprommer-, I/O-, Floppy- oder Speicherkarte beschäftigen. □

Stückliste Schneider ECB-Adapter

ICs	Fassungen
IC1,2,4 74LS244 Treiber	4 x 20pol
IC3 74LS245 Bi-Treiber	2 x 16pol (+1 für DIL-Schalter)
IC5 74LS21 2 x 4 UND	6 x 14pol
IC6 74LS09 4 x 2 UND o.C.	Sonstiges
IC7 74LS221 2 x MMF	Bei Direktanschluß (CPC 464 und CPC 664):
IC8 74LS30 1 x 8 UND	1 x Platinenstecker 2 x 25pol 1/10" mit Lötstiften
IC9 74LS138 Decoder	6 x Abstandsröhrchen 30 mm (464) oder 10 mm (664) mit passenden M3-Schrauben und Muttern
IC10,12 74LS32 4 x 2 OR	Anschluß über Kabel
IC11 74LS04 6 x INV	1 x Platinenstecker 2 x 25pol 1/10" mit Lötanschlüssen oder in Schneid-Klemmtechnik (CPC 464 und 664) oder in Schneid-Klemmtechnik (CPC 6128)
Dioden	1 x Pfostenstifteleiste 2 x 25pol 1/10"
D1,2,3,4 1N4148	1 x Pfostenbuchsleiste 2 x 25pol mit Lötanschlüssen oder in Schneid-Klemmtechnik
Kondensatoren	5 bis 20 cm Flachkabel 50pol
C1 10nF	4 x VG-Buchsleisten 64pol, gerade
C2 10nF	1 x Pfostenstifteleiste 2 x 6pol
C3 1µF/10V Tantal	1 x Pfostenstifteleiste 2 x 8pol oder
C4 47µF/10V Tantal	1 x 8fach-DIL-Schalter
3 x CB 100 nF	5 x Pfostenstifteleiste 2 x 25pol maximal 12 Steckbrücken
Widerstände	1 x Resettaster
R1 3k9	Für Spezialzwecke:
R2 10k	1 x Pfostenstifteleiste 2 x 3pol mit einer Steckbrücke
R3 330R	
R4,5 2k2	
R6 1k	
RN1 8 x 4,7k	
RN2 8 x 10k	
RN3 8 x 1k	

Jiddisch: Dus helft wie a toiten bankes.
Das hilft wie Schröpfköpfe bei einem Toten.
Das hilft gar nichts.

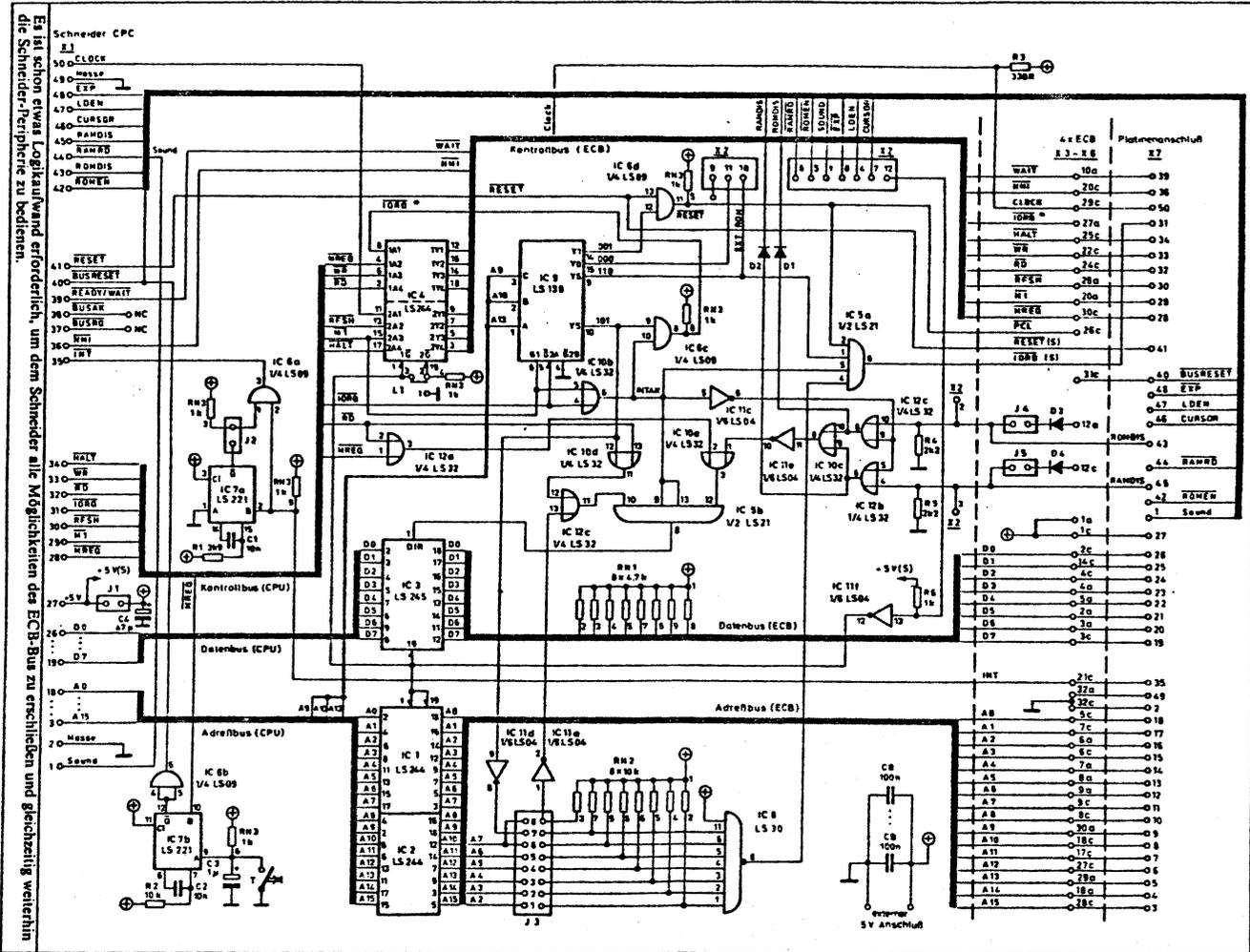
Persisch: Henvañe sir-e baqal-e kasi gosañan.
Jemandem eine Wassermelone unter die Achsel stecken.
Jemandem schmeicheln.

Manche Leute werden hauptsächlich deshalb für gebildet gehalten, weil ihnen das wenige, was sie wissen, im richtigen Augenblick einfällt.

(Friedrich Julius Stahl)

Spanisch: Hay moros en la costa. Es sind Mauren (algerische Seeräuber) an der Küste!
Vorsicht, Feind hört mit!

Ungarisch: A hazug ember könnyebb utólémi, mint a sánta kutyát. Ein Lügner ist schneller zu erreichen als ein hinkender Hund.
Lügen haben kurze Beine.



Es ist schon etwas Logikwissen erforderlich, um dem Schneider alle Möglichkeiten des ECB-Bus zu erschließen und gleichzeitig weiterhin die Schneider-Terminologie zu bedienen.

Jedesmal, wenn ein Mensch lacht, fügt er seinem Leben ein paar Tage hinzu.

Curzio Malaparte
(Pseudonym von Kurt Erich Suckert),
italienischer Schriftsteller (1898–1957)

Die Menschen müssen begreifen, daß sie das gefährlichste Ungeziefer sind, das je die Erde verwüftet hat.

WER HAT WAS -- WER SUCHT WAS --

Verkaufe:

Computers: KOMTEK 1S, int. 32 K-Ram (Aufrüstung Rübiger E-Prom)
 Drucker: Epson RX-80
 Monitor: Zenith Data System 12" Grün mit Video-Eingang
 Recorder: ISP Cassette Tape Recorder TRC 5. 2 St.
 IC: Ram's 4116 25 St.
 Software: Diverse-Programme, MON 5, RTTY-Prog, Tebast, Verstärker
 Unterla.: Schaltpläne, ROM-Listings,

Preis: 1400,- DM alles zusammen. Versand möglich.

Von 8 - 16.30 Uhr bin ich unter der Nummer 089/6 11 40 11 und nach 16.30 Uhr unter 089/6 11 59 71 zu erreichen.

im voraus besten Dank

H. B. B.

Wer hat groessere oder auch kleinere nuetzliche Programme die mir als Physikstudent 1.Sem. helfen koennten. geschrieben? (Themenbereiche Physik I. Mechanik. Waerme. Akustik. Hoehere Mathematik. z.B. Integral- Diff-rechnung. Vektorrechnung. komplexe Zahlen. usw. usw.) Oder wer kann an leicht umschreibbare Basic-Listings solcher Programme anderer Computersysteme kommen? Also schaut mal in Eure Sammlungen und fragt Freunde und wenn Ihr was gefunden habt. schreibt an Holger Mav. Danke.

Da ich nun ein TRS 80 Md1 I/II besitze möchte ich mein Md1 III verschreiben. (Md1 III, 2x40TRK SS DD LWke) Hat Jemand Interesse?

Ich habe neueddings einen Druckertreiber für Md1 III/II der es bewerkstelligt, daß die Umlaute (aus Basic) gedruckt werden - sowohl in NEUDOS wie auch TRSDOS. Alliddings auf dem Epsondrucker. Wer Interesse hat soll er mir sagen.

Ich suche z.Zt 80-Micro Ausgaben: Dez 83, Jan 84 und Jan 85 und Feb. 85. Kann mir jemand helfen.

Peter Stevens

Systembefehle				
Wort:	AUTO Z,A	BLOAD Datel, Adresse	BSAVE Datel, Adr., Länge	BYE
Bedeutung:	Generiert automatische Zellennummern, beginnend mit Zelle Z im Abstand A	Lädt eine im Binär- (Maschinen-code-) format abgesp. Datel ab angegeb. Adr. in Speicher	Speichert RAM-Inhalt ab »Adr.« im angegebenen Laufwerk unter Namen »Datel«	Rückkehr von BASIC ins Betriebssystem
System:				
ACORN	AUTO			
ALPHATRONIC	AUTO			
APPLE	AUTO	BLOAD ¹⁾	BSAVE	
ATARI				DOS
COLOUR GENIE	Entsprechende Befehle sind bei diesem Rechner nicht vorhanden			
COMMODORE 8096/C-64/VC20	Entsprechende Befehle sind bei diesem Rechner nicht vorhanden			
SIMONS BASIC	AUTO			COLD
EX-BASIC	AUTO			BASIC
DRAGON		CLOAD	CSAVEM	
EPSON HX-20	AUTO	LOADM	SAVEM	MON ⁴⁾
IBM PC	AUTO	BLOAD	BSAVE	SYSTEM
LASER 310	Entsprechende Befehle sind bei diesem Rechner nicht vorhanden			
LASER 3000		BLOAD	BSAVE	
MSX-BASIC	AUTO	BLOAD ¹⁾²⁾		
ORIC	Entsprechende Befehle sind bei diesem Rechner nicht vorhanden			
SCHNEIDER CPC	AUTO			CPM ³⁾
SHARP MZ-Serie	AUTO			
SHARP PC-1500 ³⁾	Entsprechende Befehle sind bei diesem Rechner nicht vorhanden			
SIRIUS	AUTO			
SPECTRAVIDEO	AUTO	BLOAD	BSAVE	
TANDY C.C.		CLOAD, DLOAD ³⁾	CSAVEM	
TANDY TRS 80	AUTO			
TI-99/4A	NUMBER			BYE
-EXT-BASIC	NUMBER			BYE
ZX-81 ZX-SPECTRUM	Entsprechende Befehle sind bei diesem Rechner nicht vorhanden			

Systembefehle				
Wort:	CALL	CHAIN Programm ³⁾	CLOAD Programm ³⁾	COMMON A,B, ³⁾
Bedeutung:	Ruft Maschinen-Prog. auf u. speichert es ab Adr. A, wobei Parameter P ₁ übergeb. werden	Wirkt wie LOAD Programm + RUN. Siehe auch COMMON	Lädt ein Programm von Kassette	Die angegeb. Variablen (o. alle b. ALL) werden b. CHAIN-Befehl ins aufgerufene Prog. überg.
System:				
ACORN	CALL [Name A] P ₁ ,P ₂ ..	CHAIN		
ALPHATRONIC	CALL A (P ₁ ,P ₂ ...)	CHAIN		COMMON
APPLE	CALL A	CHAIN		
ATARI			CLOAD	
COLOUR GENIE	CALL A		CLOAD	
COMMODORE 8096/C-64/VC20	SYS A		LOAD Programm	
SIMONS BASIC			LOAD Programm	
EX-BASIC	CALL (P ₁ ,P ₂ ...)		LOAD Programm	
DRAGON			CLOAD	
EPSON HX-20	EXECA oder USR A(P)			
IBM PC	CALL A (P ₁ ,P ₂ ...)	CHAIN		COMMON
LASER 310	USR (P)		CLOAD ³⁾	
LASER 3000	CALL Name			
MSX-BASIC	CALL Name (P ₁ ,P ₂ ...)		CLOAD	
ORIC	CALL A		CLOAD	
SCHNEIDER CPC	CALL A, P ₁ , P ₂ ...	CHAIN		
SHARP MZ-Serie	Entsprechende Befehle sind bei diesem Rechner nicht vorhanden			
SHARP PC-1500		CHAIN	CLOAD	
SIRIUS	CALL A (P ₁ ,P ₂ ...)	CHAIN		COMMON
SPECTRAVIDEO			CLOAD	
TANDY C.C.	EXECA oder USR A		CLOAD	
TANDY TRS 80	CALL A (P ₁ ,P ₂ ...)	CHAIN		COMMON
TI-99/4A	Entsprechende Befehle sind bei diesem Rechner nicht vorhanden			
-EXT-BASIC	CALL LOAD (Name, Adr., Param.)			
ZX-81 ZX-SPECTRUM	Entsprechende Befehle sind bei diesem Rechner nicht vorhanden			

HEFT
11
Januar
1985

Systembefehle (II)

51

Wort:	CONT	COPY	CSAVE Programm	DELETE A-B
Bedeutung:	Programmfortsetzung nach einem STOP oder BREAK	Druckt den Bildschirminhalt aus (Hardcopy)	Speichert ein Programm auf Kassette (falls eigener Befehl)	Löscht die Programmzeilen A bis B
System:				
ACORN	CONT			DELETE
ALPHATRONIC	CONT			DELETE
APPLE	CONT ⁷			DEL
ATARI	CONT		CSAVE	
COLOUR GENIE	CONT		CSAVE	
COMMODORE 8096/C-64/YC20	CONT		SAVE	
SIMONS BASIC	CONT	COPY	SAVE	
EX-BASIC	CONT	HRDCOPY	SAVE	DEL
DRAGON	CONT		CSAVE	DEL
EPSON HX-20	CONT			DELETE
IBM	CONT	PrSc-Taste drücken		DELETE
LASER 310	CONT		CSAVE	
LASER 3000	CONT			DEL
MSX-BASIC	CONT		CSAVE	
ORIC	CONT			DELETE
SCHNEIDER CPC	CONT			DELETE
SHARP MZ-Serie	CONT			DELETE
SHARP PC-1500	CONT		CSAVE	
SIRIUS	CONT			DELETE
SPECTRAVIDEO	CONT		CSAVE	DELETE
TANDY C.C.	CONT		CSAVE	DEL ⁸
TANDY TRS 80	CONT			DELETE
TI-99/4A	CONTINUE			
-EXT-BASIC	CONTINUE			
ZX-81 ZX-SPECTRUM	CONT	COPY		

Systembefehle (III)

52

Wort:	EDIT N	LIST A-B	LLIST A-B	LOAD Programm
Bedeutung:	Zelle N wird zum Bearbeiten bereitgestellt. ⁹	Gibt die Programmzeilen A bis B auf dem Bildschirm aus.	Gibt die Programmzeilen A bis B auf dem Drucker aus.	Lädt ein Programm von Diskette oder Kassette in den Arbeitsspeicher.
System:				
ACORN		LIST A, B ¹⁰	FX 5, 1: LIST A, B	LOAD Programm
ALPHATRONIC	EDIT N	LIST A-B	LLIST A-B	LOAD Programm
APPLE	EDIT N	LIST A, B	PR # 1: LIST A, B	LOAD Programm
ATARI		LIST A, B	LIST "P": A, B	LOAD Programm
COLOUR GENIE	EDIT N	LIST A-B	LLIST A-B	LOAD Programm
COMMODORE 8096/C-64/YC20		LIST A-B	OPEN 4.4: CMD4: LIST A-B: PRINT # 4: CLOSE 4	LOAD Programm (Kassette) LOAD Programm, 8 (Diskette)
SIMONS BASIC		LIST A-B	OPEN 4.4: CMD4: LIST A-B: PRINT # 4: CLOSE 4	LOAD
EX-BASIC		LIST A-B	OPEN 4.4: CMD4: LIST A-B: PRINT # 4: CLOSE 4	LOAD*
DRAGON	EDIT N	LIST A-B	LLIST A-B	¹⁴
EPSON HX-20		LIST A-B	LLIST A-B	LOAD Programm, R ¹³
IBM		LIST A-B	LLIST A-B	LOAD Programm, R ¹³
LASER 310		LIST A-B	LLIST A-B	¹⁴
LASER 3000		LIST A-B	PR # 1: LIST A-B	LOAD Programm
MSX-BASIC		LIST A-B	LLIST A-B	LOAD Programm
ORIC	EDIT N	LIST A-B	LLIST A-B	¹⁴
SCHNEIDER CPC	EDIT N	LIST A-B	LIST A-B, # 8	LOAD Programm
SHARP MZ-Serie		LIST A-B	LIST/P A-B	LOAD Programm
SHARP PC-1500	EDIT N	LIST ¹⁵	LLIST A-B	¹⁴
SIRIUS		LIST A-B	LLIST A-B	LOAD Programm, R ¹³
SPECTRAVIDEO		LIST A-B	LLIST A-B	LOAD Programm, R ¹³
TANDY C.C.	EDIT N	LIST A-B	LLIST A-B	LOAD Programm, R ¹³
TANDY TRS 80	EDIT N	LIST A-B	LLIST A-B	LOAD Programm, R ¹³
TI-99/4A	EDIT N	LIST A-B	LIST "P": A-B	OLD Programm
-EXT-BASIC	EDIT N	LIST A-B	LIST "P": A-B	OLD Programm
ZX-81 ZX-SPECTRUM		LIST A-B ¹²	LLIST A-B	LOAD Programm

Systembefehle (IV)

Wort:	MERGE Programm	NEW	RENUM neu, alt, Abstand	RUN Programm
Bedeutung:	Lädt ein Programm in den Arbeitsspeicher und „mischt“ es mit dem vorhandenen. Gleiche Zeilennummern werden vom eingeladenen Programm überschrieben.	Löscht den Arbeitsspeicher und alle Dauerbefehle wie z. B. TRON.	Neumerrierung eines Programms. Beginn der Nummerierung bei Zeile alt, Beginn der neuen Nummerierung bei neu, mit Zeilenabstand Abstand.	1) RUN Z startet ein vorhandenes Programm ab Zeile Z 2) RUN Programm wirkt wie LOAD + RUN
System:				
ACORN		Dieser Befehl ist bei allen Rechnern gleich. Er kann beim Acorn und bei Simons BASIC mit OLD rückgängig gemacht werden.	RENUMBER	nur RUN
ALPHATRONIC	MERGE		RENUM	1 2
APPLE				1 2
ATARI				1 2
COLOUR GEME			RENUM	1
COMMODEORE 8096/C-64/VC20	POKE 43, PEEK (45)-2 POKE 44, PEEK (46) LOAD Programm POKE 43,1:POKE 44,8			1 2
SIMONS BASIC	MERGE		RENUMBER	1 2
EX-BASIC	MERGE		RENUM	1 2
DRAGON			RENUM	1
EPSON HX-20	MERGE		RENUM	1 2
IBM PC	MERGE	RENUM	1 2	
LASER 310			1	
LASER 3000			1	
MSX-BASIC	MERGE	RENUM	1	
ORIC			1	
SCHNEIDER CPC	MERGE	RENUM	1 2	
SHARP MZ-Serie	MERGE	RENUM	1	
SHARP PC-1500	MERGE		1*	
SIRIUS	MERGE	RENUM	1 2	
SPECTRAVIDEO	MERGE	RENUM	1 2	
TANDY C.C.		RENUM*	nur RUN	
TANDY TRS 80	MERGE		1 2	
TI-99/4A		RESEQUENCE, neu, Abstand	1	
EXT-BASIC	MERGE	RESEQUENCE, neu, Abstand	1 2	
ZX-81 ZX-SPECTRUM	MERGE		1	

Heute: Dateiverwaltung/Sequentielle Dateien

Wort:	zum Lesen öffnen	zum Schreiben öffnen	zum Anfügen öffnen	schließen
Bedeutung:	Die Datei namens DAT (der Name ist in DS gespeichert) wird zum Lesen eröffnet.	Die Datei namens DAT wird zum Schreiben eröffnet.	Eine Datei wird so eröffnet, daß am Dateiende Daten angefügt werden können (Erweitern).	Der Kanal zur Datei wird wieder geschlossen.
System:				
ACORN	X=OPENIN DS	X=OPENOUT DS		CLOSE#X
ALPHATRONIC	OPEN DS FOR INPUT AS #1	OPEN DS FOR OUTPUT AS #1	OPEN DS FOR APPEND AS #1	CLOSE 1
APPLE	PRINT CHR\$(4);"OPEN";DS PRINT CHR\$(4);"READ";DS	PRINT CHR\$(4);"OPEN";DS PRINT CHR\$(4);"WRITE";DS	PRINT CHR\$(4);"APPEND";DS	PRINT CHR\$(4);"CLOSE";DS
ATARI	OPEN#1,4,0,"D:";DS	OPEN#1,8,0,"D:";DS	OPEN#1,9,0,"D:";DS	CLOSE#1
COLOUR GEME	OPEN"1",1,DS	OPEN"0",1,DS		
COMMODEORE 8096/C-64/VC20	OPEN 1,1,0,DS (Kassette) OPEN 1,8,2,DS;"S,R" (Disk)	OPEN 1,1,1,DS (Kassette) OPEN 1,8,2,DS;"S,W" (Disk)	OPEN1,8,2,DS;"S,A"	CLOSE1
SIMONS BASIC				
EX-BASIC				
DRAGON	automatisch mit FREAD	automatisch mit FWRITE, vorher Zeiger auf Null setzen	automatisch durch FWRITE	CLOSE1
EPSON HX-20	OPEN"1",#-1,DS	OPEN"0",#-1,DS		CLOSE#1
IBM PC	OPEN DS FOR INPUT AS 1	OPEN DS FOR OUTPUT AS 1	OPEN DS FOR APPEND AS 1	CLOSE#1
LASER 310	nicht nötig	nicht nötig	nicht möglich	nicht nötig
LASER 3000	PRINT CHR\$(4);"OPEN";DS; PRINT CHR\$(4);"READ";DS	PRINT CHR\$(4);"OPEN";DS; PRINT CHR\$(4);"WRITE";DS	PRINT CHR\$(4);"APPEND";DS	PRINT CHR\$(4);"CLOSE";DS
MSX-BASIC	OPEN DS FOR INPUT AS 1	OPEN DS FOR OUTPUT AS 1	OPEN DS FOR APPEND AS 1	CLOSE #1
ORIC	nicht nötig	nicht nötig	nicht möglich	nicht nötig
SCHNEIDER CPC	OPENIN DS	OPENOUT DS	nicht möglich	CLOSEIN bzw. CLOSEOUT
SHARP MZ-Serie	ROPEN"DAT"	WOPEN"DAT"	nicht möglich	CLOSE
SHARP PC-1500	nicht nötig	nicht nötig	nicht möglich	nicht nötig
SIRIUS	OPEN"1",1,DS	OPEN"0",1,DS		CLOSE#1 bzw. RESET (für alle Dateien gleichzeitig)
SPECTRAVIDEO	OPEN DS FOR INPUT AS 1	OPEN DS FOR OUTPUT AS 1	OPEN DS FOR APPEND AS 1 (nur RANDOM-Dateien)	CLOSE#1
TANDY C.C.	OPEN"1",#-1,DS	OPEN"0",#-1,DS	nicht möglich	CLOSE#-1
TANDY TRS 80	OPEN"1",1,DS	OPEN"0",1,DS	OPEN"E",1,DS	CLOSE 1
TI-99/4A	OPEN#1:DS,SEQUENTIAL, DISPLAY,INPUT,VARIABLE	OPEN#1:DS,SEQUENTIAL, DISPLAY,OUTPUT,VARIABLE	OPEN#1:DS,SEQUENTIAL, DISPLAY,APPEND,VARIABLE	CLOSE#1
EXT-BASIC				
ZX-81 ZX-SPECTRUM	OPEN#1;"m";1;"DAT"	OPEN#1;"m";1;"DAT"	nicht möglich	CLOSE#1 oder CLEAR#

HEFT
II
Januar
1985

54

Sequentielle Dateien

Wort	Variablen lesen	ein Zeichen lesen	ganze Zeile lesen	Variablen schreiben
Bedeutung:	Liest die Texte und Zahlen aus der Datei ein und überschreibt sie auf die Variablen AS,A...	Liest ein Zeichen aus der Datei (auch Steuerzeichen) und überschreibt es auf XS.	Eine Textzeile wird ohne Berücks. von Trennzeichen bis z. nächst. CR gelesen.	Die Variablenwerte AS,A... werden auf die Datei geschrieben
System:				
ACORN	INPUT #X,AS,A...	BGET #X,XS		PRINT #X,AS,A...
ALPHATRONIC	INPUT #1,AS,A...	XS=INPUT\$(1, #1)	LINE INPUT #1,XS	PRINT #1,AS,A... ¹²
APPLE	INPUT AS,A...	GET XS		PRINT AS;CHR\$(13);A...
ATARI	INPUT #1,AS,A...	GET #1,X ⁹		PRINT #1,AS,A...
COLOR GENIE	keine Angaben			keine Angaben
COMMODORE 8086/C-64/VIC20	INPUT #1,AS,A...	GET #1,XS		PRINT #1,AS,A...
SIMONS BASIC				
EX-BASIC				
DRAGON	FREAD DS;AS,A...		FLREAD DS;XS	FWRITE DS;AS,A...
EPSON IX-20	INPUT #1,AS,A...	XS=INPUT\$(1, #1)	LINE INPUT #1,XS	PRINT #1,AS,A... ¹²
EPSON	INPUT #1,AS,A...	XS=INPUT\$(1, #1)	LINE INPUT #1,XS	PRINT #1,AS,A... ¹²
LASER 310	INPUT # "DAT",AS,A...			PRINT # "DAT",AS,A...
LASER 3000	INPUT AS,A...		INPUT XS	PRINT AS;CHR\$(13);A...
MSX-BASIC	INPUT #1,AS,A...	XS=INPUT\$(1, #1)	LINE INPUT #1,XS	PRINT #1,AS,A...
ORIC	RECAL AS, "DAT",S ¹⁸			STORE AS, "DAT",S ¹⁸
SCHNEIDER CPC	INPUT #9,AS,A...		LINE INPUT #9,XS	PRINT #9,AS,A... ¹²
SHARP MZ-Serie	INPUT /AS,A...			PRINT /AS,A...
SHARP PC-1500	INPUT # "DAT",AS,A ^(*) ... ¹³			PRINT # "DAT",AS,A ^(*) ... ¹³
SIRIUS	INPUT #1,AS,A...	XS=INPUT\$(1, #1)	LINE INPUT #1,XS	PRINT #1,AS,A... ¹²
SPECTRAVIDEO	INPUT #1,AS,A...		LINE INPUT #1,XS	PRINT #1,AS,A... ¹²
TANDY C...	INPUT #1,AS,A...			PRINT #1,AS,A...
TANDY TRS 80	INPUT #1,AS,A...	XS=INPUT\$(1, 1)	LINE INPUT #1,XS	PRINT #1,AS,A... ¹²
TI-99/4A	INPUT #1,AS,A...			PRINT #1,AS,A...
EX-BASIC			INPUT #1,XS	wie TI-BASIC ¹²
ZX-81				
ZX-SPECTRUM	INPUT #1,AS,A...	XS=INKEY\$ #1		PRINT #1,AS,A...

Sequentielle Dateien

RANDOM-Dateien

Wort	Datei-Ende	Katalog	Datei löschen	Datei öffnen
Bedeutung:	Prüft auf Datei-Ende und gibt als Funktion den Wert "wahr" (= -1 oder +1) bzw. "falsch".	Druckt den Katalog aller registrierten Dateien aus.	Löscht die Datei namens DAT.	Eine Random-Datei mit einer Satzlänge von 50 Byte wird eröffnet.
System:				
ACORN	EOF #X	CAT	"DELETE" DAT	
ALPHATRONIC	EOF(1)	FILES	KILL "DAT"	OPEN "R", #1,DS,50
APPLE		CATALOG	DELETE DAT	PRINT CHR\$(4);"OPEN";DS;"L50"
ATARI	STATUS #1,A (A=136)	über DOS-Menü	über DOS-Menü	
COLOUR GENIE				
COMMODORE 8086/C-64/VIC20	ST (ST=64)	LOAD "\$",8,LIST	OPEN 1,8,15:PRINT #1,"S:DAT"	OPEN 1,8,2,DS;"L"+CHR\$(50)
SIMONS BASIC		DIR "\$"	DISK "S:DAT"	
EX-BASIC			JS:DAT	
DRAGON	EOF("DAT")	DIR	KILL "DAT"	CREATE DS,50
EPSON IX-20	EOF(1)	FILES	KILL "DAT"	OPEN "R", #1,DS
IBAI	EOF(1)	FILES	KILL "DAT"	OPEN "R", #1,DS,50
LASER 310		nicht möglich	nicht nötig	
LASER 3000		CATALOG	DELETE DAT	PRINT CHR\$(4);"OPEN";DS;"L50"
MSX-BASIC		Betriebssystem	Betriebssystem	
ORIC		nicht möglich	nicht nötig	
SCHNEIDER CPC		CAT	nicht nötig	
SHARP MZ-Serie		nicht möglich	nicht nötig	
SHARP PC-1500		nicht möglich	nicht nötig	
SIRIUS	EOF(1)	FILES	KILL "DAT"	OPEN "R", #1,DS,50
SPECTRAVIDEO	EOF(1)	FILES	KILL "DAT"	OPEN DS AS 1 ¹⁸
TANDY C...	EOF(-1)	nicht möglich	nicht nötig	
TANDY TRS 80	EOF(1)	SYSTEM "DIR"	KILL "DAT"	OPEN "D", 1,DS,50
TI-99/4A	EOF(1)	nicht möglich	DELETE DS	OPEN #1,DS,RELATIVE, DISPLAY,INPUT,FXED,50
EX-BASIC				
ZX-81				
ZX-SPECTRUM		CAT 1	ERASE "m";"DAT"	

RANDOM-Dateien				
Wort	Struktur festlegen	positionieren	Lesen	Schreiben
Bedeutung:	Der Satz wird eingeteilt in Felder, die Strings vorgegebener Länge aufnehmen	Positioniert den Dateizeiger auf Satz Nr. S (Byte BY)	Liest von Satz S Daten in AS bzw. die FIELD-Variablen	Die angegebenen oder die FIELD-Variablen werden in Sektor S geschrieben
System:				
ACORN		PTR#X-S	wie bei sequentiellen Dateien	
ALPHATRONIC	FIELD 1,12 AS AS, 4 AS BS,...	autom. bei Lesen/Schreiben	GET#1,S	PUT#1,S
APPLE	automatisch	autom. bei Lesen/Schreiben	PRINT CHR\$(4);"READ"; DS;"R";S	PRINT CHR\$(4);"WRITE";DS; ;"R";S;PRINT AS,A...
ATARI				
COLOUR GENIE				
COMMODORE 8096/PC-64/VC20	automatisch	OPEN2,8,15,"P"+CHR\$(2)+10 CHR\$(LO)+CHR\$(HI)+CHR\$(BY)	wie bei sequentiellen Dateien	wie bei sequentiellen Dateien
SHON'S BASIC				
EX-BASIC				
DRAGON	autom. beim Lesen/Schreiben	autom. bei Lesen/Schreiben	FREAD DS, FROM S, FOR L:AS	FWRITE DS FROM S, FOR L:AS,A...
EPSON (PC-20) DEFIL 100,ST	FIELD 1,12 AS AS, 4 AS BS,...	autom. bei Lesen/Schreiben	GET#1,S GET%S,AS,A... ¹⁰	PUT#1,S PUT%S,AS,A... ¹⁰
IBM PC	FIELD 1,12 AS AS, 4 AS BS,...	autom. bei Lesen/Schreiben	GET#1,S	PUT#1,S
LASER 110				
LASER 3000	1 Satz wird jeweils durch ein RETURN begrenzt	PRINT CHR\$(4);"POSITION"; DS;"R";S	PRINT CHR\$(4);"READ";DS; ;"R";S;"B";Z ¹⁰	PRINT CHR\$(4);"WRITE";DS; ;"R";S;"B";Z ¹⁰
MSX-BASIC				
ORIC				
SCHNEIDER CPC				
SHARP MZ-Serie				
SHARP PC-1500				
SIRIUS	FIELD 1,12 AS AS, 4 AS BS,...	autom. bei Lesen/Schreiben	GET#1,S	PUT#1,S
SPECTRAVIDEO	FIELD 1,12 AS AS, 4 AS BS,...	autom. bei Lesen/Schreiben	GET#1,S	PUT#1,S
TANDY				
TANDY TRS 80	FIELD 1,12 AS AS, 4 AS BS,...	autom. bei Lesen/Schreiben	GET 1,S	PUT 1,S
TI-99/4A		RESTORE#1,RECS	INPUT#1,RECS:AS,A...	PRINT#1,RECS:AS,A... ¹⁰
EXT-BASIC				
ZX-81				
ZX-SPECTRUM				

Sequentielle Dateien				
Wort	Besondere Schreibweise	Datei-Ende	Datei-Länge	Datei-Anzahl festlegen
Bedeutung:	(1) schreibt ein Zeichen in die Datei. (2) umgibt Strings mit "" und trennt sie durch Kommas.	Prüft auf Dateiende und gibt als Funktion den Wert "wahr" (= -1 oder +1) bzw. "falsch".	Diese Funktion gibt die Datellänge in Byte an.	Damit kann die Anzahl gleichzeitig offener Dateien erweitert werden.
System:				
ACORN	BPUT#X,XS(1)	EOF#X	EXT#X	
ALPHATRONIC	WRITE#1,AS,BS(2)	EOF(1)		
APPLE				MAXFILES=N
ATARI	PUT#1,A ¹⁰ (1)	STATUS#1,A(A=136)		
COLOUR GENIE				
COMMODORE 8096/PC-64/VC20		ST(ST=64)		
SHON'S BASIC				
EX-BASIC				
DRAGON		EOF("DAT")	LOF("DAT")	
EPSON (PC-20)		EOF(1)	LOF(1)	FILNUM=N
IBM PC	WRITE#1,AS,BS,...(2)	EOF(1)	LOF(1)	BASIC/F:N (b. Starten d. Systems)
LASER 110				
LASER 3000				
MSX-BASIC				MAXFILES=N
ORIC				
SCHNEIDER CPC	WRITE#9,AS,BS,...(2)			
SHARP MZ-Serie				
SHARP PC-1500				
SIRIUS	WRITE#1,AS,BS,...(2)	EOF(1)	LOF(1)	BASIC/F:N (b. Starten d. Systems)
SPECTRAVIDEO	WRITE#1,AS,BS,...(2)	EOF(1)	LOF(1)	MAXFILES=N
TANDY		EOF(-1)		
TANDY TRS 80	WRITE#1,AS,BS,...(2)	EOF(1)		
TI-99/4A		EOF(1)		CALL FILES(N)
EXT-BASIC				
ZX-81				
ZX-SPECTRUM				

HEFT
11
Januar
1985
58

Stringverarbeitung

Wort:	String(feld) definieren	String deklarieren	String-Länge	Teilstring suchen
Bedeutung:	Ein zweidim. Stringfeld wird definiert (Sonderbedeutungen bei Atari und Sinclair).	Die Variablen, die mit A bis Z beginnen, werden als Strings definiert. ⁹	Bestimmt die Länge eines Strings (=ganze Zahl zwischen 0 und – meistens – 255).	Sucht den Such-String SS im String Z\$ ab Pos. P. Resultat 0 = nicht gefunden.
System:				
ACORN	DIM A\$(5,10)	-	LEN(Z\$)	INSTR(Z\$,S\$,P)
ALPHATRONIC	DIM A\$(5,10)	DEFSTRA,Z	LEN(Z\$)	INSTR(P,Z\$,S\$)
APPLE	DIM A\$(5,10)	-	LEN(Z\$)	-
ATARI	DIM A\$(10) ⁸	-	LEN(Z\$)	-
COLOUR GENIE	DIM A\$(5,10)	DEFSTRA,Z	LEN(Z\$)	INSTR(P,Z\$,S\$) ⁸
COMMODORE 8096/7-84/VC20	DIM A\$(5,10)	-	LEN(Z\$)	-
SIMONS BASIC	DIM A\$(5,10)	-	LEN(Z\$)	PLACE(Z\$,S\$)
EX-BASIC	DIM A\$(5,10)	-	LEN(Z\$)	INSTR(Z\$,S\$,P)
DRAGON	DIM A\$(5,10)	-	LEN(Z\$)	INSTR(Z\$,S\$,P)
EPSON FX-20	DIM A\$(5,10)	DEFSTRA,Z	LEN(Z\$)	INSTR(P,Z\$,S\$)
IBM PC	DIM A\$(5,10)	DEFSTRA,Z	LEN(Z\$)	INSTR(P,Z\$,S\$)
LASER 310	DIM A\$(5,10)	-	LEN(Z\$)	-
LASER 3000	DIM A\$(5,10)	-	LEN(Z\$)	-
MSX-BASIC	DIM A\$(5,10)	DEFSTRA,Z	LEN(Z\$)	INSTR(P,Z\$,S\$)
ORIC	DIM A\$(5,10)	-	LEN(Z\$)	-
SCHNEIDER CP	DIM A\$(5,10)	-	LEN(Z\$)	-
SHARP MZ-Series	DIM A\$(5,10)	-	LEN(Z\$)	-
SHARP PC-1500	-	-	LEN(Z\$)	-
SIRIUS	DIM A\$(5,10)	DEFSTRA,Z	LEN(Z\$)	INSTR(P,Z\$,S\$)
SPECTRAVIDEO	DIM A\$(5,10)	DEFSTRA,Z	LEN(Z\$)	-
TANDY C.C.	DIM A\$(5,10)	-	LEN(Z\$)	-
TANDY TRS 80	DIM A\$(5,10)	DEFSTRA,Z	LEN(Z\$)	INSTR(P,Z\$,S\$)
TI-99/4A	DIM A\$(5,10)	-	LEN(Z\$)	POS(Z\$,S\$,P)
XT-BASIC	DIM A\$(5,10)	-	LEN(Z\$)	POS(Z\$,S\$,P)
ZX-81	-	-	-	-
ZX-SPECTRUM	DIM A\$(5,10) ⁸	-	LEN(Z\$)	-

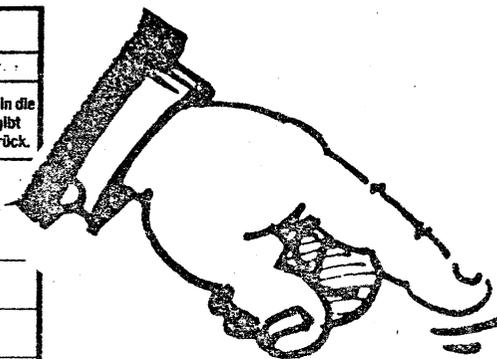
Stringverarbeitung

Wort:	Teilstring links	Teilstring rechts	Teilstring Mitte	Strings verknüpfen
Bedeutung:	Kopiert aus dem String Z\$ die ersten zehn Zeichen heraus.	Kopiert aus dem String Z\$ die letzten zehn Zeichen heraus.	Kopiert aus dem String Z\$ fünf Zeichen ab Position 3 (also die Zeichen 3 bis 7).	Verkettet zwei Strings miteinander.
System:				
ACORN	LEFT\$(Z\$,10)	RIGHT\$(Z\$,10)	MID\$(Z\$,3,5)	AS + BS
ALPHATRONIC	LEFT\$(Z\$,10)	RIGHT\$(Z\$,10)	MID\$(Z\$,3,5) ⁸	AS + BS
APPLE	LEFT\$(Z\$,10)	RIGHT\$(Z\$,10)	MID\$(Z\$,3,5)	AS + BS
ATARI	Z\$(1,10)	Z\$(LEN(Z\$)-9,LEN(Z\$))	Z\$(3,7)	AS(LEN(AS)+1)-BS
COLOUR GENIE	LEFT\$(Z\$,10)	RIGHT\$(Z\$,10)	MID\$(Z\$,3,5)	AS + BS
COMMODORE 8096/7-84/VC20	LEFT\$(Z\$,10)	RIGHT\$(Z\$,10)	MID\$(Z\$,3,5)	AS + BS
SIMONS BASIC	LEFT\$(Z\$,10)	RIGHT\$(Z\$,10)	MID\$(Z\$,3,5)	AS + BS
EX-BASIC	LEFT\$(Z\$,10)	RIGHT\$(Z\$,10)	MID\$(Z\$,3,5)	AS + BS
DRAGON	LEFT\$(Z\$,10)	RIGHT\$(Z\$,10)	MID\$(Z\$,3,5)	AS + BS
EPSON FX-20	LEFT\$(Z\$,10)	RIGHT\$(Z\$,10)	MID\$(Z\$,3,5) ⁸	AS + BS
IBM PC	LEFT\$(Z\$,10)	RIGHT\$(Z\$,10)	MID\$(Z\$,3,5) ⁸	AS + BS
LASER 310	LEFT\$(Z\$,10)	RIGHT\$(Z\$,10)	MID\$(Z\$,3,5)	AS + BS
LASER 3000	LEFT\$(Z\$,10)	RIGHT\$(Z\$,10)	MID\$(Z\$,3,5)	AS + BS
MSX-BASIC	LEFT\$(Z\$,10)	RIGHT\$(Z\$,10)	MID\$(Z\$,3,5) ⁸	AS + BS
ORIC	LEFT\$(Z\$,10)	RIGHT\$(Z\$,10)	MID\$(Z\$,3,5)	AS + BS
SCHNEIDER CP	LEFT\$(Z\$,10)	RIGHT\$(Z\$,10)	MID\$(Z\$,3,5)	AS + BS
SHARP MZ-Series	LEFT\$(Z\$,10)	RIGHT\$(Z\$,10)	MID\$(Z\$,3,5)	AS + BS
SHARP PC-1500	LEFT\$(Z\$,10)	RIGHT\$(Z\$,10)	MID\$(Z\$,3,5)	AS + BS
SIRIUS	LEFT\$(Z\$,10)	RIGHT\$(Z\$,10)	MID\$(Z\$,3,5)	AS + BS
SPECTRAVIDEO	LEFT\$(Z\$,10)	RIGHT\$(Z\$,10)	MID\$(Z\$,3,5)	AS + BS
TANDY C.C.	LEFT\$(Z\$,10)	RIGHT\$(Z\$,10)	MID\$(Z\$,3,5)	AS + BS
TANDY TRS 80	LEFT\$(Z\$,10)	RIGHT\$(Z\$,10)	MID\$(Z\$,3,5)	AS + BS
TI-99/4A	SEGS(Z\$,1,10)	-	SEGS(Z\$,3,5)	AS & BS
XT-BASIC	SEGS(Z\$,1,10)	-	SEGS(Z\$,3,5)	AS & BS
ZX-81	-	-	-	-
ZX-SPECTRUM	Z\$(1 TO 10)	-	Z\$(3 TO 7)	-

Stringverarbeitung					Stringverarbeitung II				
Wort	Zeichen in Code	Code in Zeichen	String in Zahl	Zahl in String	Wort	Zeichen vervielfachen	Leerstring erzeugen	Strings löschen	Stringspeicher angeben
Bedeutung:	Verwandelt das erste Zeichen des Strings Z\$ in den entsprechenden ASCII-Code.	Verwandelt die Zahl Z in das dem ASCII-Code entsprechende Zeichen.	Verwandelt den String Z\$ in eine Zahl, sofern und soweit dies möglich ist.	Verwandelt die Zahl Z in einen String mit Vorzeichen.	Bedeutung:	Erzeugt einen String, der N-mal die Zeichenkette in Anführungszeichen enthält.	Erzeugt einen String aus N Leerzeichen.	Löscht alle String-Variablen und schafft N freie Speicherplätze (in Bytes).	Gibt an, wieviele Bytes im Arbeitsspeicher für Strings noch frei sind.
System:					System:				
ACORN	ASC(Z\$)	CHR\$(Z)	VAL(Z\$)	STR\$(Z)	ACORN	STRINGS(N,"-0-")		CLEAR N	
ALPHA CHIC	ASC(Z\$)	CHR\$(Z)	VAL(Z\$)	STR\$(Z)	ALPHATHRONIC	STRINGS(N,"-0-")	SPACES(N)	CLEAR N	FRE(X\$)#
APPLE	ASC(Z\$)	CHR\$(Z)	VAL(Z\$)	STR\$(Z)	APPLE			CLEAR N	
ATARI	ASC(Z\$)	CHR\$(Z)	VAL(Z\$)	STR\$(Z)	ATARI			CLR	
COLOUR GENIE	ASC(Z\$)	CHR\$(Z)	VAL(Z\$)	STR\$(Z)	COLOUR GENIE	STRINGS(N,"-0-")		CLEAR N	
COMMODORE 8096/C-64/VC20	ASC(Z\$)	CHR\$(Z)	VAL(Z\$)	STR\$(Z)	COMMODORE 8096/C-64/VC20			CLR	
EX-BASIC	ASC(Z\$)	CHR\$(Z)	VAL(Z\$)	STR\$(Z)	SIMONS BASIC	DUP("-0-",N)		CLR	
EX-BASIC	ASC(Z\$)	CHR\$(Z)	VAL(Z\$)	STR\$(Z)	EX-BASIC	STRINGS(N,"-0-")		CLR	
DRAGON	ASC(Z\$)	CHR\$(Z)	VAL(Z\$)	STR\$(Z)	DRAGON	STRINGS(N,"-0-")		CLEAR N	FRE\$(#)
EPSON HX-20	ASC(Z\$)	CHR\$(Z)	VAL(Z\$)	STR\$(Z)	EPSON HX-20	STRINGS(N,"-0-")	SPACES(N)	CLEAR N	FRE(") #
IBM PC	ASC(Z\$)	CHR\$(Z)	VAL(Z\$)	STR\$(Z)	IBM PC	STRINGS(N,"-0-")	SPACES(N)	CLEAR N	FRE(") #
LASER 310	ASC(Z\$)	CHR\$(Z)	VAL(Z\$)	STR\$(Z)	LASER 310	Entsprechende Befehle sind bei diesem Rechner nicht vorhanden			
LASER 3000	ASC(Z\$)	CHR\$(Z)	VAL(Z\$)	STR\$(Z)	LASER 3000			CLR	FRE(AS)#
MSX-BASIC	ASC(Z\$)	CHR\$(Z)	VAL(Z\$)	STR\$(Z)	MSX-BASIC	STRINGS(N,"-0-")	SPACES(N)	CLEAR N	FRE(") #
ORIC	ASC(Z\$)	CHR\$(Z)	VAL(Z\$)	STR\$(Z)	ORIC			CLEAR N	FRE(") #
SCHNEIDER CPC	ASC(Z\$)	CHR\$(Z)	VAL(Z\$)	STR\$(Z)	SCHNEIDER CPC	STRINGS(N,"-0-")	SPACES(N)	CLEAR N	FRE(") #
SHARP MZ-Serie	ASC(Z\$)	CHR\$(Z)	VAL(Z\$)	STR\$(Z)	SHARP MZ-Serie			CLR	
SHARP PC-1500	ASC(Z\$)	CHR\$(Z)	VAL(Z\$)	STR\$(Z)	SHARP PC-1500			CLEAR N	
SIRIUS	ASC(Z\$)	CHR\$(Z)	VAL(Z\$)	STR\$(Z)	SIRIUS	STRINGS(N,"-0-")	SPACES(N)	CLEAR N	FRE(") #
SPECTRAVIDEO	ASC(Z\$)	CHR\$(Z)	VAL(Z\$)	STR\$(Z)	SPECTRAVIDEO	STRINGS(N,"-0-")	SPACES(N)	CLEAR N	FRE(") #
TANDY C.C.	ASC(Z\$)	CHR\$(Z)	VAL(Z\$)	STR\$(Z)	TANDY C.C.			CLEAR N	
TANDY TRS 80	ASC(Z\$)	CHR\$(Z)	VAL(Z\$)	STR\$(Z)	TANDY TRS 80	STRINGS(N,"-0-")	SPACES(N)	CLEAR N	FRE(") #
TI-99/4A	ASC(Z\$)	CHR\$(Z)	VAL(Z\$)	STR\$(Z)	TI-99/4A	Entsprechende Befehle sind bei diesem Rechner nicht vorhanden			
EX-BASIC	ASC(Z\$)	CHR\$(Z)	VAL(Z\$)	STR\$(Z)	EX-BASIC	RPT("-0-",N)			
ZX-81 ZX-SPECTRUM	CODE Z\$	CHR\$(Z)	VAL(Z\$)	STR\$(Z)	ZX-81 ZX-SPECTRUM			CLEAR N	

Stringverarbeitung II

Wort:	Spezialumwandlungen 1	Spezialumwandlungen 2	Dezimal in Hex	Dezimal in Oct
Bedeutung:	Verwandelt spezielle Strings in Zahlen (ganz, einfach genau, doppelt genau).	Verwandelt Zahlen (ganz, einfach genau, doppelt genau) in Strings.	Wandelt eine Dezimalzahl in die entspr. Hexadezimalzahl um u. gibt Ergebn. als String zurück.	Wandelt eine Dezimalzahl in die entspr. Oktalzahl um und gibt das Ergebnis als String zurück.
System:				
ACORN	Entsprechende Befehle sind bei diesem Rechner nicht vorhanden			
ALPHATRONIC	CV(Z\$), CVS(Z\$), CVD(Z\$)	MKS(Z%), MKSS(Z), MKDS(Z#)	HEX\$(X)	OCT\$(X)
APPLE	Entsprechende Befehle sind bei diesem Rechner nicht vorhanden			
ATARI	Entsprechende Befehle sind bei diesem Rechner nicht vorhanden			
COLOUR GENIE	Entsprechende Befehle sind bei diesem Rechner nicht vorhanden			
COMMODORE 8096/C-64/VC20	Entsprechende Befehle sind bei diesem Rechner nicht vorhanden			
SIMONS BASIC	Entsprechende Befehle sind bei diesem Rechner nicht vorhanden			
EX-BASIC	Entsprechende Befehle sind bei diesem Rechner nicht vorhanden			
DRAGON			HEX\$(X)	
EPSON HX-20	CV(Z\$), CVS(Z\$), CVD(Z\$)	MKS(Z%), MKSS(Z), MKDS(Z#)	HEX\$(X)	OCT\$(X)
IBM PC	CV(Z\$), CVS(Z\$), CVD(Z\$)	MKS(Z%), MKSS(Z), MKDS(Z#)	HEX\$(X)	OCT\$(X)
LASER 310	Entsprechende Befehle sind bei diesem Rechner nicht vorhanden			
LASER 3000	Entsprechende Befehle sind bei diesem Rechner nicht vorhanden			
MSX-BASIC			HEX\$(X)	OCT\$(X)*
ORIC			HEX\$(X)	
SCHNEIDER CPC			HEX\$(X)*	
SHARP MZ-Serie	Entsprechende Befehle sind bei diesem Rechner nicht vorhanden			
SHARP PC-1500	Entsprechende Befehle sind bei diesem Rechner nicht vorhanden			
SIRIUS	CV(Z\$), CVS(Z\$), CVD(Z\$)	MKS(Z%), MKSS(Z), MKDS(Z#)	HEX\$(X)	OCT\$(X)
SPECTRAVIDEO	CV(Z\$), CVS(Z\$), CVD(Z\$)	MKS(Z%), MKSS(Z), MKDS(Z#)	HEX\$(X)	OCT\$(X)*
TANDY C.C.	Entsprechende Befehle sind bei diesem Rechner nicht vorhanden			
TANDY TRS 80	CV(Z\$), CVS(Z\$), CVD(Z\$)	MKS(Z%), MKSS(Z), MKDS(Z#)	HEX\$(X)	OCT\$(X)
TI-99/4A	Entsprechende Befehle sind bei diesem Rechner nicht vorhanden			
-EXT-BASIC	Entsprechende Befehle sind bei diesem Rechner nicht vorhanden			
ZX-81 ZX-SPECTRUM	Entsprechende Befehle sind bei diesem Rechner nicht vorhanden			



CLUB-Bücherei
Gerald Dreyer hat mehrere Ausgaben der
GENIE-DATA
gestiftet. Die Hefte sind in der CLUB-
Bibliothek und können ausgeliehen werden.

Ein Computer läßt sich nicht täuschen

Auf der New Yorker Weltausstellung steht ein Computer, der jedem Besucher – auf Knopfdruck – Namen und Herkunft sagt.

Auch der Indianer Tom To-peeh aus Kansas zahlt seinen Dollar, drückt den Knopf, und der Computer sagt: »DU HEISST TOM TO-PEEH, BIST INDIANER, KOMMST AUS KANSAS, DEIN LETZTER ZUG GEHT UM 18 UHR 30.«

Tom To-peeh hält das für einen Schwindeltrick und beschließt, den Computer hereinzuliegen. Nach langem Suchen findet er einen Neger-Schuhputzer, der bereit ist, ihm seine Uniform und seinen Putzkasten zu borgen; er schmiert Gesicht und Hände mit Schuhcreme schwarz, zieht die Uniform an und tritt, den Putzkasten in der Hand, wieder vor den Computer.

Er wirft seinen Dollar ein, drückt den Knopf – der Computer sagt: »DU HEISST TOM TO-PEEH, BIST INDIANER, KOMMST AUS KANSAS, DEIN LETZTER ZUG GING UM 18 UHR 30. – UND WENN DU DEN QUATSCH MIT DER VERKLEIDUNG NICHT GEMACHT HÄTTEST, HÄTTEST DU IHN NICHT VERPASST!«

DAS GESETZ VON MURPHY

JEDER VON UNS WIRD TAEGLICH - BEI DER ARBEIT, BEI DER AUSUEBUNG DES HOBBYS, WIE AUCH IM GANZ PRIVATEN BEREICH - MIT DEM GESETZ VON MURPHY KONFRONTIERT. DOCH BLEIBT DIES MEISTENS UNBEWUSST, WEIL DIESES GESETZ - OBWOHL ES WEITAUS UNIVERSALER ALS DAS OHMSCHE GESETZ IST - ZUMINDEST BEI UNS FAST UNBEKANNT BLIEB.

VON DER EXISTENZ DES GESETZES VON MURPHY WUSSTE ICH SEIT LAENGEREM, UND ICH HALTE ES FUER MEINE PFLICHT, DIESE ERKENNTNISSE HIERMIT WEITERZUGEBEN - ERKLAEREN SIE BEISPIELSWEISE ENDLICH, WARUM DER CHIEF AUSGERECHNET IMMER DANN DEN RAUM BETRIT, WENN MAN NACH STUNDENLANGER, INTENSIVER ARBEIT DEN ARBEITSPLATZ FUER DAS NAECHSTE PROJEKT AUFGERAUMT HAT UND EINEN AUGENBLICK LANG NACHDENKLICH AUS DEM FENSTER SCHAUT.

1. EDEL MURPHY

DER AMERIKANER EDEL MURPHY ENTDECKTE UND FORMULIERTE ALS ERSTER DIE "GESETZMAESSIGKEITEN DES STETIGEN AERGERNISSES" DIE TATSACHE, DASS IHM IN SPAETEREN JAHREN DER SICHERE PLATZ IN DER RUHMESHALLE GROSSER FORSCHER UND ENTDECKER VERSAGT BLIEB, KANN NUR ALS AUSWIRKUNG SEINES EIGENEN GESETZES ERKLAERT WERDEN, DESSEN ORIGINALFASSUNG LAUTET:

IF ANYTHING CAN GO WRONG, IT WILL

DIES KANN UEBERTRAGEN WERDEN:

FALLS IRGENDETWAS SCHIEFGEHEN KANN, DANN GEHT ES SCHIEF.

ODER IN EXAKTER MATHEMATISCHER FORM: $1 + 1 \rightarrow 2$
WORIN \rightarrow DAS MATHEMATISCHE SYMBOL FUER "KAUM JEMALS" IST.

ZUR ERLAEUTERUNG DIESES UNIVERSALEN GESETZES WERDEN IM FOLGENDEN EINIGE ANWENDUNGSBEISPIELE AUS DEM BEREICH DER ELEKTRONIK ANGEBOEN.

2. DAS GESETZ VON MURPHY IN FORSCHUNG UND ENTWICKLUNG

- JE HARMLOSER EINE KONSTRUKTIONSAENDERUNG ERSCHEINT, DESTO WEITTRAGEN- DER SIND IHRE TATSAECHLICHEN FOLGEN.
- DIE NOTWENDIGKEIT FUER EINE GROESSERE KONSTRUKTIONSAENDERUNG WAECHST MIT DEM NAHERUECKEN DES FABRIKATIONSBEGINNS.
- KOPIERMASCHINEN ZERMANGELN IMMER DIE ORIGINALZEICHNUNGEN, VON DENEN NOCH NICHT EINMAL EINE KOPIE VORHANDEN IST.
- DIE WAHRSCHEINLICHKEIT, DASS EIN MASS IN EINER ZEICHNUNG VERGESSEN WIRD IST DIREKT PROPORTIONAL ZU SEINER WICHTIGKEIT.
- JEDER FEHLER DER SICH IN EINE BERECHNUNG EINSCHLEICHEN KANN WIRD DIES MIT SICHERHEIT TUN; UND ZWAR SO, DASS MAN VDELLIG VON VORNE BEGINNEN MUSS.
- IN JEDER BELIEBIGEN BERECHNUNG IST IMMER DIEJENIGE ZAHL DIE URSACHE VON FEHLERN, DIE AM OFFENSICHTLICHSTEN RICHTIG WAR.
- DEZIMALPUNKTE SIND IMMER AN DER FALSCHEN STELLE, VORZEICHEN VERTAUSCHT
- EINEN HINWEIS, DEN MAN IN EINEM BUCH, LEXIKON, ORDNER ODER AEHNLICHEM SUCHT, FINDET MAN AUF DER LETZTEN SEITE, WENN MAN VON VORNE ANFAENGT ZU SUCHEN, UND AUF DER ERSTEN SEITE, WENN MAN SCHLAUERWEISE HINTEN ANFAENGT.
- EINE TOTSICHERE SCHALTUNG FUNKTIONIERT NIE.
- EINE OSZILLATORSCHALTUNG IST DURCH NICHTS ZUM SCHWINGEN ZU BEWEGEN; EINE VERSTAERKERSCHALTUNG DAGEGEN SCHWINGT SOFORT.
- EIN DURCH EINE ULTRAFLINKE SICHERUNG GESCHUETZTER TRANSISTOR SCHUETZT DIE SICHERUNG, INDEM ER ZUERST "HOCHGEHT".
- EIN SEIT WOCHEN FUNKTIONIERENDER LABORAUFBAU VERSAGT PLOETZLICH, WENN DER CHIEF ODER EINEM KUNDEN VORGEFUEHRT WERDEN SOLL. (DIESER TEIL- ASPEKT DES GESETZES VON MURPHY IST ALS "VORFUEHREFFEKT" BEKANNT)

3. DAS GESETZ VON MURPHY IN FERTIGUNG UND SERVICE

- JE DRINGENDER MAN EIN BAUTEIL BENDETIGT, DESTO SCHWIERIGER LAESST ES SICH BESCHAFFEN.
- BENDETIGT MAN FUER EIN PROJEKT N BAUELEMENTE, DANN SIND N-1 AUF LAGER
- IST EIN BESTIMMTER KAPAZITAETSWEIT ERFORDERLICH, SO IST DIESER WERT NICHT VORRAETIG; AUSSERDEM LAESST ER SICH WEDER DURCH PARALLEL- NOCH DURCH SERIENSCHALTUNG VORHANDENER WERTE REALISIEREN.
- AUSTAUSCHTEILE SIND IN DER PRAXIS NICHT AUSTAUSCHBAR.
- EIN AUF RICHTIGE LAENGE ABGESCHNITTNER DRAHT ERWEISST SICH IMMER ALS ZU KURZ.
- BAUTEILE, DIE KEINESFALLS FALSCH EINGEBAUT WERDEN DUERFEN, UND DIE AUCH GARNICHT FALSCH EINGEBAUT WERDEN KOENNEN, SIND ES AM SCHLUSS DOCH
- AUSGERECHNET DAS ZERBRECHLICHSTE BAUTEIL LAESST MAN FALLEN.
- DIE AUSFALLWAHRSCHEINLICHKEIT EINES BAUTEILS ODER EINER BAUGRUPPE IST UM SO HOEHER, JE SCHWIERIGER DIE REPARATUR ODER EIN AUSTAUSCH DURCHGEFUEHRT WERDEN KANN.
- EIN WERKZEUG DAS VERSEHENTLICH IN EIN GERAET FAELT, LANDET UNWEIGERLICH AN DER STELLE, WO ES DEN GROESSTEN SCHADEN ANRICHTEN KANN. (DIESER SPEZIALFALL DES GESETZES VON MURPHY IST AUCH ALS "SELEKTIVE GRAVITATION" BEKANNT)
- WENN EIN GERAET KOMPLETT MONTIERT IST, FINDET MAN NOCH UEBRIGE BAUTEILE AUF DER WERKBANK.
- WENN MAN IN EINEM GERAET MIT AUSSETZFEHLERN EIN OFFENSICHTLICH DEFEKTES BAUTEIL ERSETZT HAT, SO TRITT DER FEHLER WIEDER AUF, SOBALD DAS GERAET IM EINSATZ IST.

4. DAS GESETZ VON MURPHY UND DIE DATENVERARBEITUNG

- TEILPROGRAMME, DIE EINZELN IN ORDNUNG SIND, FUNKTIONIEREN ZUSAMMEN GELADEN NICHT MEHR.
- JE EINFACHER EIN STRUKTOGRAMM, DESTO KOMPLIZIERTER DIE UEBERSETZUNG IN EINE PROGRAMMIERSPRACHE.
- EIN PROGRAMM VERHAELT SICH IMMER ANDERS ALS DAS STRUKTOGRAMM.
- DIE VERBINDUNG EINER DATENFERNUEBERTRAGUNG, BRICHT IMMER IN DIESEN MOMENT ZUSAMMEN, WENN DIE SYNCHRONISATION ZU ENDE IST, UND DIE ERSTEN DATEN KOMMEN.
- PROGRAMMLISTEN AUS ZEITSCHRIFTEN ODER FACHBUECHERN, HABEN GRUNDSAETZLICH FEHLER
- DER SCHREIBSCHUTZ WIRD IMMER BEI DER ORIGINAL DISKETTE VERGESSEN.
- PROGRAMME, DIE ERWORBEN WERDEN, LAUFEN ERST NACH GROESSEREN AENDERUNGEN, UND DANN NICHT SO, WIE DAS ORIGINAL.
- BEI EINER DATENBANK IST DAS SCHLUESSELWORT UMGEKEHRT PROPORTIONAL ZUR GEHEIMHALTUNG.
- DAS SCHLUESSELWORT EINER DATENBANK WIRD GRUNDSAETZLICH VERGESSEN
- KOPIERGESCHUETZTE PROGRAMME SIND GANZ EINFACH ZU KOPIEREN.

5. BEMERKUNGEN

ETLICHE DER AUFGEFUEHRTEN BEISPIELE, WIE VOR ALLEM DER "VOHRFUEHREFFEKT", SIND DEN MEISTEN VON UNS AUS DER BERUFSPRAXIS BESTENS VERTRAUT; IM UEBRIGEN SPRECHEN WIR AUCH VON DER "TUECKE DES OBJEKTS". DIES ALLES DEUTET DARAUF HIN, DASS DIE KENNNTNIS DES GESETZES, WELCHES EDEL MURPHY FORMULIERTE, IM DEUTSCHEN SPRACHBEREICH ZUMINDEST LATENT SCHON VORHANDEN WAR.

(MANFRED HELD)

Folgerungen aus Murphys Gesetz

Guttersons Gesetze

Jedes Programmierprojekt, das gut anfängt, endet schlecht.
Jedes Programmierprojekt, das schlecht anfängt, endet noch schlechter.

Kleinbrunners Folgerungen

Wenn eine Programmieraufgabe leicht aussieht, ist sie schwer.
Wenn eine Programmieraufgabe schwer aussieht, ist sie unlösbar.

Munbrights Gesetze

Jedes Programm ist es veraltet, wenn es läuft.
Jedes Programm kostet mehr und dauert länger.
Wenn ein Programm nützlich ist, muß es geändert werden.
Wenn ein Programm unnütz ist, muß es dokumentiert werden.
Jedes Programm ist ein bißchen länger, als der verfügbare Speicher es zuläßt.
Der Wert eines Programms ist umgekehrt proportional dem von ihm verbrauchten Papier.
Die Komplexität eines Programms wächst solange, bis sie die Fähigkeiten des Programmierers übersteigt, der es unterhalten soll.
Der unangenehmste Fehler wird erst entdeckt, wenn das Programm mindestens 6 Monate im Einsatz ist.
Maschinenunabhängigen Code gibt es nicht.
Zusätzliche Programmierer, die zur Einhaltung eines Termins eingesetzt werden, verzögern den Termin noch weiter.
Softwareprobleme sind umso größer, je weniger Zeit man zu ihrer Behebung zur Verfügung hat.

Favours Gesetz

Es ist immer noch ein Fehler vorhanden.

Brunks Gesetz

Wenn ein Listing einen Anfang hat, hat es auch ein Ende.

Zeppelmiers Folgerung

Die letzten vier Seiten eines kritischen Listings sind immer verloren gegangen.

Fenningtons Beobachtung

Die Wahrscheinlichkeit, daß ein Programm die gestellten Anforderungen erfüllt, ist umgekehrt proportional dem Vertrauen des Programmierers in seine eigenen Fähigkeiten.

Schrödingers Gesetze der Programmierung

Zwei Fehler in einem Programm heben sich beim Probelauf in ihrer Wirkung auf.
Im späteren Einsatz des Programms ergänzen sie sich, um den größtmöglichen Schaden anzurichten.

Schrödingers Erkenntnisse

I. Man konstruiere jeden Programmteil so kurz, daß nur ein Fehler auftauchen kann.
II. So kurze Programmteile gibt es in der Praxis nicht.
III. Die überprüften Bestandteile eines Programms laufen garantiert nicht zusammen.

Hergeröders Gesetz zur Kürzung eines Programms

Werden in einem Programm alle vermeintlich unwichtigen Zeilen gelöscht, läuft das Programm nicht mehr.

Ableitungen aus Hergeröders Gesetz

1. Unter den unwichtigen Zeilen war mindestens eine wichtige.
2. Es läßt sich unmöglich rekonstruieren, welche Zeile dies war.
3. Beim Versuch, das gekürzte Programm zum Laufen zu bringen, wird es bis zum vollkommenen Kollaps getrieben.
4. Kurz vor dem Zusammenbruch zerstört das fehlerhafte Programm seinen Urcode.
5. Der Programmierer kann sich beim erneuten Schreiben des Programms unmöglich an die wichtigsten Algorithmen erinnern.

Zwergmanns allgemeingültigen Naturgesetze

Jedes Unterprogramm strebt danach, zum Hauptprogramm zu werden.

Je länger ein Programmierer ohne Zwischenspeichern tippt, desto größer ist die Wahrscheinlichkeit für einen Stromausfall.

Mit jedem Zwischenspeichern bei seiner Arbeit steigt die Wahrscheinlichkeit für einen irreparablen Diskettenfehler. Jede Fehler-Erkennungs-Routine trifft früher oder später auf einen Fehler, den sie nicht kennt. Dieser Fehler veranlaßt die Routine, alles das zu machen, was ihre Erbauer verhindern wollten.

Klingers Daten-Gesetze

Jede Datenverwaltung oder -verarbeitung hat die Aufgabe, möglichst viele Daten im unpassendsten Augenblick auf verheerendste Weise zu zerstören.

1. Ableitung: Je umfassender die Zerstörung desto weniger Vorwarnungen gibt es.
2. Ableitung: Drei Ereignisse charakterisieren diesen Vorgang:
 - in letzter Zeit lief alles glatt und ohne Fehler
 - im Moment der Zerstörung meldet das System absolut nichts
 - nach einer alles umfassenden Zerstörung meldet sich das System als wäre nichts gewesen
 - nach einem kleineren Zwischenfall fängt irgendetwas an zu rauchen oder zu klingeln, um möglichst große Hektik zu verursachen

Erkenntnis: Je ruhiger ein System, desto unruhiger sollte der Betreiber sein.

Mareks Konfusionsregeln

Verläßt ein Programmierer während des Ausdrucks eines wichtigen Listings für fünf Minuten den Raum, so passieren in diesem Moment mit größter Wahrscheinlichkeit folgende Dinge:

- das Papier bleibt hängen oder
- der Papierstapel ist aufgebraucht oder
- das schon beschriebene Papier wird vom Drucker noch mal eingezogen ("Epsonsche Rekursion")
- das Farbband verhakt sich
- es wird ein falscher Steuercode zum Drucker gesandt
- das Drucker-Interface beschließt, seinen Geist aufzugeben

CLUB 80 - Programmbibliothek exklusiv

Liebe Clubfreunde!

Auch in diesem Info kann ich euch wieder eine Menge neuer Programme vorstellen. Außerdem möchte ich euch noch auf ein paar ganz besondere Leckerbissen in der Bibliothek hinweisen!

Schon seit einiger Zeit sind drei moderne Programmiersprachen in der Bibliothek für jedes Mitglied verfügbar. Es handelt sich dabei um Implementierungen von LISP, PLOT und LOGO, die alle in Micro80 - Heften abgedruckt waren. Natürlich beinhalten diese Minimalsysteme nicht alle möglichen Sprachelemente, aber um in eine Sprache einmal hineinzuschnuppern (z.B. bevor man sich einen teureren Interpreter oder Compiler zulegt) reicht es allemal!

Unser Clubmitglied Klaus Herman hat mir in letzter Zeit eine ganze Menge Utilities für das Model 4 von Tandy geschickt, die im letzten und in diesem Info in die Bibliothek aufgenommen wurden. Ich hoffe, daß die Programmsammlung damit auch für die Besitzer des Model 4 (und das sind inzwischen schon einige) interessanter wird.

Gerald Schröder, der sich ja schon ein paarmal durch überdurchschnittliche Beiträge zur Bibliothek (z.B. Übersetzung des Handbuchs zur SuperUtility+) ausgezeichnet hat, ist auch diesmal wieder mit zwei selbstgeschriebenen Programmen dabei. Zum einen hat er einen schon fast professionellen Grafikeditor geschrieben (BASIC compiliert), der durch seine Vielzahl an Möglichkeiten glänzt, zum andern hat er das Brettspiel RISIKO auf unseren Computer installiert. Die dazu benötigten Weltkarten hat er übrigens mit seinem selbstgeschriebenen Grafikeditor erstellt!

Noch jemand hat sich diesmal besonders hervorgetan. Walter Schäfer hat zwei selbstgeschriebene Denkspiele (NIMM und MASTERMIND) beigesteuert, die durch ihre gute Grafik auffallen.

Aber natürlich waren auch diesmal wieder viel mehr Mitglieder als die hier genannten, an dem rapiden Wachstum der Softwaresammlung schuld. Sie alle hier aufzuzählen erspare ich mir und begnüge mich damit, ihnen im Namen des Club's zu danken!

Das letzte Mal!

Mit 37 Neuzugängen und einigen speziellen Tips ist die Rubrik Programmbibliothek exklusiv heute wieder einmal recht groß geraten. Dieser Umstand zeigt, daß entgegen aller Abschaffungsversuche einzelner Mitglieder, die Softwaresammlung ihren festen Platz in CLUB 80 behauptet hat. Nicht umsonst ist sie im Laufe ihres Bestehens auf fast 300 Titel gewachsen! Dies ist nun das letzte Clubinfo, in welchem ich für die Programmbibliothek des CLUB 80 verantwortlich zeichne. Ich habe einen Nachfolger in diesem Amt gefunden und bin sicher, er wird die Bibliothek genauso gut, vielleicht sogar besser betreuen als ich!

Ich hoffe, ihr werdet ihm genauso viel Unterstützung zukommen lassen wie mir, damit die Programmbibliothek als eines der lebendigsten und wichtigsten Elemente des CLUB 80 noch lange weiterbesteht! Ich möchte mich noch einmal für eure Mitarbeit bedanken und meinem Nachfolger, Klaus Jürgen (Kajot) Mühlenbein viel Glück und Erfolg bei seiner Arbeit wünschen, Euer

Karsten Obermann

ADRESS-100

A 0229 A 01 /BAS

Adressenverwaltung für bis zu 100 Datensätze. Speicherung auf Disk!

ADRESS-CAS

A 0232 A 01 /BAS

Adressenverwaltungsprogramm mit recht komfortablen Möglichkeiten. Speicherung der Daten auf Kassette!

ADVENTURE

D 0234 A 02 /BAS

Wie der Name schon sagt, ein Adventure-Spiel. Worum es geht, weiß ich nicht (ich interessiere mich nicht für diese Art Spiele), vielleicht bekommt es jemand von euch heraus!? Englisch!

AUTORENNEN

G 0236 A 01 /CMD

Ein ziemlich einfaches Geschicklichkeitsspiel mit mittelmäßiger Grafik

Kalender

S 0237 A 02 /BAS

Noch ein Kalenderdruckprogramm, allerdings ein recht ansprechendes! Für alle Drucker geeignet!

Vergleich

S 0238 A 03 /BAS

Dieses Programm von Jens Neueder ermöglicht den Vergleich von Diskfiles mit Ausdruck der Unterschiede. (s. Clubinfo Nr. 10 Seite 43/44)

DATVAR

S 0240 B 03 /CMD

Dieses Hilfsprogramm listet auf Tastendruck, auch mitten in einem laufenden BASIC-Programm, alle gültigen Variablen und Felder sowie deren Inhalte. Der BASIC-Bildschirmaufbau wird dabei nicht beschädigt!!

BARGRAPH

M 0241 A 01 /BAS

Demoprogramm zur Erstellung horizontaler Balkendiagramme. Leicht zu ändern und in eigene Programme einzubauen!

EXECUTE

S 0243 B 02 /CMD

Menüprogramm zum Aufruf von Programmen mittels Tastendruck.

DECIPOKE

S 0244 A 01 /BAS --

Hilfsprogramm zur Umwandlung von Maschinenroutinen in DATA-Zeilen. Anleitung bzw. Artikel in englisch aus '80 US' kann geliefert werden!

Artillery

G 0245 A 01 /BAS --

Mit einem Artilleriegeschütz soll man einen unsichtbaren Feind treffen (Koordinaten bekannt). Dabei muß man sowohl die Geschößgeschwindigkeit als auch den Abschußwinkel angeben. Keine Grafik!!

CHECKBOOK

M 0246 A 01 /BAS

Dieses Programm hilft beim Verwalten des, durch Frau, Freundin und Computer sowieso schon ziemlich abgemagerten, Guthabens. Vielleicht wird es dadurch mehr!?

Vokabeltrainer

S 0247 A 02 /BAS

Praktisches Hilfsprogramm zum Einpauken von Vokabeln. Die Vokabeln können eingegeben, auf Disk gespeichert und wieder geladen werden. Danach wird die Abfrage mit Erfolgsquotenanzeige gestartet.

KATALOG4

A 0276 D 03 /BAS

Programmkatalog aus CHIP 5/83 S.129, geändert fuer TANDY Model4 von Klaus Hermann!

HUSTLE

G 0277 A 03 /BAS

Ein recht nettes Geschicklichkeitsspiel ähnlich dem bekannten WURMI.

Panzerschlacht

G 0278 A 02 /BAS

Spiel für zwei Personen. Jeder hat einen Panzer in einem Labyrinth und muß den Gegner ausschalten.

Mini-Rechner

M 0279 A 01 /BAS

Mini-Taschenrechner (+,-,*,/) als BASIC-Einzeiler programmiert. Quelle: Computer Persönlich

VAMPIR

G 0280 A 02 /BAS

Nettes Geschicklichkeitsspiel, bei dem es darum geht, sich als Vampir seine blutige Nahrung zu suchen.

Kurvendiskussion

M 0281 A 02 /BAS

Programm zur Berechnung von Nullstellen, Wendepunkten, Minima und Maxima usw. einer Funktion. Keine grafische Darstellung.

NIMM

D 0282 A 03 /BAS

Das bekannte Streichhölzchenspiel in einer sehr interessanten Version unseres Clubmitgliedes Walter Schäfer.

Mastermind

D 0283 A 03 /BAS

Noch ein bekanntes Spiel gut aufgemacht von unserem Mitglied Walter Schäfer!

Lagerverwaltung

A 0284 A 02 /BAS

Lagerverwaltung mit allen Funktionen, die man zur Verwaltung eines kleinen Lagers benötigt. Speicherung der Daten auf Kassette.

HRG-Demoprogramm

S 0285 B 01 /BAS

Die beiden Demoprogramme zum Thema 'HRG vom BASIC aus' aus dem Info 10 Seite 31/32. Autor: Jens Neueder

Finanzen

M 0286 A 03 /BAS

Ein bischen Finanzmathematik in ein recht ansprechendes Programm gepackt!

Magic Square

M 0287 A 01 /BAS

Programm zur Erstellung der geheimnisvollen 'magischen Quadrate'.

TAPE-IDENT

S 0288 B 03 /CMD

Identifiziert Tapefiles und gibt Art, Namen und ev. Anfangs-, End- und Startadresse an.

SCRABBLE

D 0289 A 02 /BAS

Ein Wortspiel für 1 bis 4 Spieler. Anleitung in deutsch im Spiel enthalten!

Ich grüße den CLUB 80!

Als neues Mitglied ist es meine Pflicht mich vorzustellen. Ich bin zwar ein neues Mitglied - aber nicht mehr ganz "neu". Meine Mitgliedschaft ist ca. 2 Wochen alt (s. Datum dieses INFO, falls es eins besitzt, wie ich hoffe...)

Ich bin also eins der "jüngsten" Mitglieder in Eurer Runde - und doch wiederum nicht - im Gegenteil! Das nennt man ein "Paradoxon"! So paradox wie eine Primzahl?

Jetzt dürft Ihr raten!
 Raten? Wozu hat man einen Computer? -
 Dann "computet" (rechnet) mal schön!

Ausgangsbasis ist also eine Primzahl. Eine Primzahl ist, wie Ihr wißt, unteilbar. Like me. Nur durch "1" und sich selbst. Die erste Primzahl ist die "1", die zweite die "2", die dritte die "3", die vierte die - nein, eben nicht die "4" - jetzt wird's interessant! - sondern die "5", und nun paßt auf:

die 5. Primzahl ist die "7",
 " 7. " " " " "13" und
 " 13. " " " " "37" !

Und die ist besonders interessant!

Das Interessante an ihr ist, daß die "kleinste Quersumme" ihres Produktes mit einer beliebigen Zahl immer wieder gleich der "kleinsten Quersumme" dieser Zahl selbst ist. Programmiert das mal! (Das Programm findet Ihr irgendwo in Heft)

Nun kommt's:

In ihrem Quadrat steckt mein Geburtsjahr - wieder als Quersumme natürlich und somit unsichtbar.

Multipliziert man sie mit meinem Geburtsmonat, so erhält man wieder meinen Geburtsmonat als Quersumme.

(Die Quersumme ihrer dritten Potenz ergibt übrigens wieder mein Geburtsjahr - das nur nebenbei...)

Nachdem Ihr auf diese Weise sowohl Jahr als auch Monat jenes denkwürdigen Datums errechnet habt, an dem ich das "Licht der Welt" leider nicht erblickte (denn es war Nacht), erhaltet Ihr den Rest leicht, wenn ich Euch noch verrate, daß diese Primzahl auch die Summe meiner Geburtsdaten, also gleich TT+MM+JJ ist * ganz einfach nach Herrn Riese, genannt A.R. (bitte nicht mit J.R. verwechseln, wie anderwärts schon geschehen...)

Habe ich mich damit genug vorgestellt? * Prima!

Nein, Irrtum: es liegt kein astrologischer Unterbau zugrunde (obwohl ich meine Skorpion-Pflichten hinreichend erfülle); es ist reine Zahlenmystik des Mittelalters (noch bewahrt von S) wenigleich ich auch dieses schon hinter mir weiß (wie Ihr nun wißt).

Sonst noch was? Ach so - was Ihr von meiner Mitgliedschaft habt? Was kann der Club von mir profitieren? * Laßt mal sehen: Ihr könnt mich alles fragen, egal was * ich habe nichts dagegen! * Nein, wieder Irrtum: nicht, daß ich alles wüßte (im Gegenteil!), vielmehr: ich wüßte (auch) gern alles! Wir machen das vielmehr so: Ihr fragt mich was (tel. oder briefl.) und ich schaue in der klugen Datei "PROGRAMME" nach. Diese habe ich nämlich seit 1.1.86 von Hartmut übernommen. Vorausgesetzt sie enthält die Antwort und Ihr habt mir so eine kleine dumme Zielscheibe zugeschickt, schieße ich Euch die Antwort drauf und mach sie schlau * allerdings darf sie nicht mehr als 41 Ringe haben, ab 42. treffe ich nichts mehr, und den 80. Innenring schon gar nicht! Und außerdem muß mir der hoffnungsvolle Einsender mitteilen, mit welchem Gewehr ich schießen soll: mit dem Karabiner oder der Schrotflinte; letztere hat zwei (!) Läufe und schießt d* = dank dessen doppelt dicht! (Einen Drilling besitze ich leider noch nicht; dafür aber drei Laufzwerge, häufig "Drraiws" genannt * hinten mit weichem "s".)

HEFT
 11
 Januar
 1985

74

THREEKEY

S 0290 B 02 /CMD

Dieses Programm ermöglicht es, verschiedene DOS- und BASIC-Funktionen mit einer Kombination aus SHIFT+Pfeil unten+entspr. Buchstaben aufzurufen.

ORBITENG

D 0291 A 04 /BAS

Das Programm aus Micro 80 5/82 ermöglicht es, die Landung auf verschiedenen Planeten (u.a. Mond, Mars, Titan) "lebensecht" zu simulieren. Schoene Grafik! eingeg. von Gerald Schroeder!

Sortierprogramm fuer Model 4

U 0292 D 03 /BAS

Sortieroutine (in Maschinenspr.) fuer BASIC-Felder, hier als BASIC-Lader! Fuer Model 4 aus 80MICRO Maerz 85,S.70; eingegeben von Klaus Herman

GRAFIK-EDITOR

S 0293 A 50 /CMD

Ein in BASIC geschriebener und dann compilierter Grafikeditor mit vielen Moeglichkeiten (u.a. Speichern, Laden, Umwandeln in BASIC-Zeilen, verzerren des Bildes uvam.). Autor: Gerald Schroeder!

Set Funktion Keys

U 0294 D 03 /CMD

Dieses Programm ermöglicht es, die Funktionstasten des Model 4 mit beliebigen Werten zu belegen. Quelle: Micro80 11-12/84, eingegeben von Klaus Herman

Menue-Helfer

U 0295 D 03 /BAS

Das Programm erstellt Hilfsmenues, die durch CTRL-A in den ersten sechs Zeilen des Bildschirms angezeigt werden. Achtung: nur fuer Model 4! Quelle: Micro80 7-10/85, eingegeben von Klaus Herman

HELP-File-Generator

U 0296 D 03 /BAS

Erstellt HELP-Files (aehn1. DOS/HELP), die mit HELP/CMD aufgerufen werden. Nach Eingabe eines Stichwortes erfolgt dann die Erklaerung! Quelle: Micro80 03/85, eingegeben von Klaus Herman

RISIKO

D 0297 A 05 /BAS

Dieses Programm wurde dem bekannten Brettspiel nachempfunden. Zur Zeit dient der Computer nur als Spielfeldverwalter, es soll jedoch eine Computergegner-Version folgen. Gute Grafiken! Autor: G. Schroeder!

TAUSCH

U 0298 B 03 /CMD

Dieses Programm ermöglicht das Arbeiten unter NewDOS (und seinen Abkoemmlingen) mit zwei Bildschirmen! Umschaltung mit SHIFT+BREAK. Quelle: GenieData Autor: Thomas Weinstein

Ich kann übrigens auch um die Ecke schießen. Wer's nicht glaubt, dem durchlöchre ich die Scheibe mal von hinten, das heißt, vorn und hinten abwechselnd. Das nennt sich dann "DS" (Drehricht-Scheibe, weil man sie nicht umzudrehn braucht, wenn sie vorne voll ist).

Ihr dürft natürlich auch schießen - wollt' sagen: schicken, was Ihr der Mit- und Nachwelt würdig befunden. Das staple ich in die Diskothek, auf daß sie wachse: MIRO = Much In * Random Out. **Trotz ID/CMD:**

Vorabklärung über PD's wird dankbar registriert!!!
Denn nicht immer klappen diese Tricks. Ich habe schon erlebt, daß mein TRS80 keine GENIE-Wurst verdaute, auch wenn sie in besagte Scheiben geschnitten war...

Und Ihr wollt doch alle, daß das mit dem Neuling klappt, oder?
Das walte Gott!
So spricht:
Kajott

PA für Anfänger und solche, die es werden wollen:

Meine Laufwerke sind nicht groß-, sondern nur 40-spurig;
eins - das System-Laufwerk - ist einseitig, die beiden andern hingegen sind vielseitig;
(leider mehr als zwei Seiten gibt's glaube ich noch nicht)
und alle drei sind besonders dicht, nämlich doppelt (jedenfalls wurde das von meinem Lieferanten behauptet.
Ich habe es noch nicht geprüft, weil an keine elektrisch aufgeladenen Geräte mit in die Badewanne nahen soll...)

* * * ALLES KLAR?? * * *

Impressum

Vorstand

Kontaktadresse für
Clubangelegenheiten
Clubbücherei / Fundgrube
Clubkasse

Günther WAGNER
Gartenstraße 4
8281 Neubuern
Tel.: 08835 /3361
< 18 - 20 Uhr >

Programmbibliothek

Kontaktadresse

Klaus-Jürgen Mühlenbein
Am Mönchgarten 28
6940 Heinsheim
Tel.: 06201 /55452

Redaktion

Kontaktadresse

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

Adventure-Ecke

Kontaktadresse

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

Hardware

Kontaktadresse

Halter ZWICKEL
Lengfelder 123
5181 Bergheim (Austria)
Tel.: 0043662 /51130

Achtung:

Hier die PDRIVE-Werte unseres neuen
Club-Programm-Bibliothekars
für selbstbootenden Disketten:

40 TRACK -- SS -- DD
Ansonsten ist auch
40 TRACK -- DS -- DD
möglich.

Redakteure

dieser Ausgabe

Hanfred Held * Klaus-J. Mühlenbein
Jens Neueder * Hartmut Obermann
Jutta Obermann * Walter Piller
Bernd Reizlaff * Gerald Schröder
Peter Stevens * Günther Wagner
Walter Zwickel *
sowie Artikel aus: Dr. Dobbs's Journal, c't und
Computerheft

Bankverbindung des CLUB 88

Spartkassa Rosenheim, BLZ 711 500 00
auf Konto-Nr. 194 712
Postcheckkonto der Spartkassa
Nr. 0077-001

Das INFO erscheint zweimonatlich.

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

Schlupf

Hallo Club-88er,

auch in dem neuen Jahr findet Ihr den Schlupf in altbewährter
Weise - wie auch im letzten Jahr - wieder fast ganz am Ende des
CLUBINFO's.

Zuerst aber noch einen herzlichen Dank für die Weihnachts-
und Neujahrswünsche die ich von Euch erhalten habe. Ich habe mich
sehr darüber gefreut. Ich hoffe, daß das Neue Jahr auch Eure
Wünsche erfüllt und Ihr den Jahreswechsel gut überstanden habt.
Um dies mit den Worten von Ulrich zu beschließen:

"Habt immer eine arbeitende CPU"

Am Infoende findet Ihr diesmal einen einfachen Kalender, der
Euch an die Redaktionsschlupftermine in diesem Jahr und an das
Clubtreffen erinnern soll. Als Weiteres habe ich einmal eine
Zusammenfassung geschrieben, die Euch als INFO-Mitarbeiter den
"Umgang" mit der Redaktion erleichtern bzw. vereinfachen soll.
Diese beiden Blätter sind zum Abtrennen gedacht. Somit habt Ihr
ein paar Unterlagen mehr an Eurem "Computerarbeitsplatz" zur
Verfügung. Vielleicht sind sie Euch nützlich.

Ich persönlich hatte diesmal wenig Zeit etwas Eigenes zu
schreiben. Ich hoffe, daß ich bis zum nächsten Mal wieder einen
Beitrag für Euch habe.

So, dies war es mal wieder. Nun habt Ihr ein weiteres INFO in
Eurer Sammlung. Ich wünsch Euch viel Spaß beim Lesen und melde
mich pünktlich mit dem nächsten INFO wieder.

Euer

J. Neueder

CLUB 88 Mitgliederadressenliste

Name	Vorname	Straße	PlZ	Stadt	Telefon
Alber	Herbert	Alemannenstr. 20	7732	Niedereschach	07721 /7102
Albers	Herbert	Zum Duwelshopen 14	2117	Westedt	
Beckhausen	Wolfgang	Vuerfelsen-Kaule 30	5060	Bergisch-Gladbach 1	02204 /62781
Boecker	Dieter	Lehmweg 4	2930	Varel 1	04451 /7640
Boeckling	Ulrich	Am Sonnenhang 11	5414	Vallendar	0261 /69522
Buskowiak	Thomas	Eschersheimer Landstr. 257	6000	Frankfurt 1	069 /5601621
Dreyer	Gerald	Am Speiergarten 8	6200	Wiesbaden-Bierstadt	06121 /508218
Drowälder	Bernd	Buchentalsweg 8	4939	Steinheim	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
Hummel	Anton	Schubertstr. 2	7612	Haslach	07832 /8289
Jablotschkin	Rainer	Thiekamp 29	4780	Lippstadt 8	
Kasper	Dieter	Zeppelinstr. 9	8952	Marktoberdorf	08342 /1630
Koenig	Hans J.	Hebbelstr. 25	2080	Pinneberg	04101 /209444
Konrad	Josef	Anzengruberstraße 35	8038	Gröbenzell	08142 /8494
Kuhn	Eckehard	Im Dorf 14	7443	Frickenhäuser 1	07022 /45417
May	Holger	Marienstr. 9	5768	Sundern 2	02935 /1668
Mühlenbein	Klaus-Jürgen	Am Mönchgarten 28	6940	Weinheim -Lü.	06201 /55052
Neueder	Jens	Panoramastr. 21	7178	Michelbach/Bilz	0791 /42877 (dienstl.44-667)
Obermann	Hartmut	Schwalbacher Str. 6	6209	Heidenrod 1	06124 /3913
Perschbach	Patrick	Waldstr. 52	5000	Koeln 91	0221 /872118
Piller	Walter	Rohnenstraße 8	CH-8935	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	Annulf	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

Wegen Systemwechsel /Computerei aufgegeben ausgetreten

Hans Baldes (zum 1.1.86)

Matthias Hallup (zum 1.1.86)

Andreas Marx (zum 1.1.86)

Jean-Claude Wies (zum 1.1.86)

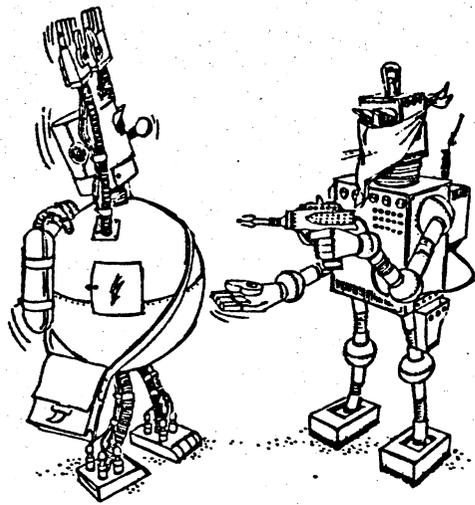
Neuzugänge :

Herbert Albers

Klaus-Jürgen Mühlenbein

Hans Raggan

Oliver Volz



CLUBBO TERMIN - KALENDER 1986

JANUAR

Mo	Di	Mi	Do	Fr	Sa	So
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

FEBRUAR

Mo	Di	Mi	Do	Fr	Sa	So
						1 2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
◆ 25	26	27	28			

MÄRZ

Mo	Di	Mi	Do	Fr	Sa	So
						1 2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

APRIL

Mo	Di	Mi	Do	Fr	Sa	So
		1	2	3	4	5 6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
◆ 29	30					

MAI

Mo	Di	Mi	Do	Fr	Sa	So
						1 2 3 4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

JUNI

Mo	Di	Mi	Do	Fr	Sa	So
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
◆						

JULI

Mo	Di	Mi	Do	Fr	Sa	So
		1	2	3	4	5 6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

AUGUST

Mo	Di	Mi	Do	Fr	Sa	So
						1 2 3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
◆ 26	27	28	29	30	31	

SEPTEMBER

Mo	Di	Mi	Do	Fr	Sa	So
						1 2 3 4 5 6 7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

OKTOBER

Mo	Di	Mi	Do	Fr	Sa	So
						1 2 3 4 5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
◆ 28	29	30	31			

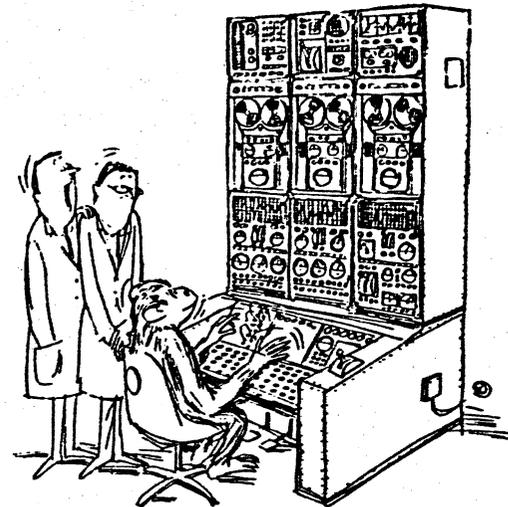
NOVEMBER

Mo	Di	Mi	Do	Fr	Sa	So
						1 2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

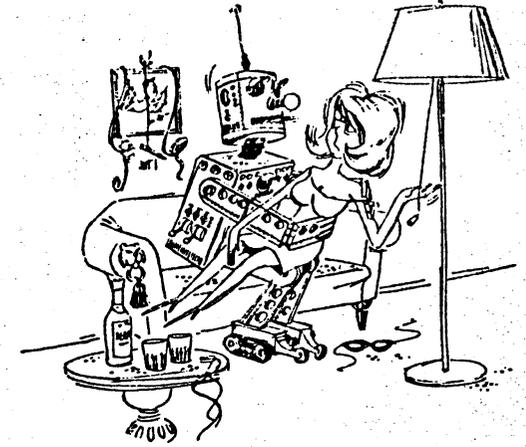
DEZEMBER

Mo	Di	Mi	Do	Fr	Sa	So
						1 2 3 4 5 6 7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
◆ 30	31					

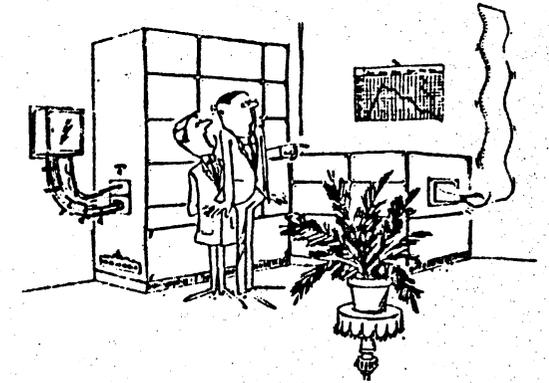
»GELD HER ODER BATTERIE!«



»Menschliches Versagen ist bei uns jetzt völlig ausgeschlossen.«



»ICH HABE IHNEN DOCH GLEICH GESAGT, DASS ICH ECHT UND NICHT MASKIERT BIN!«



»Wer hatte eigentlich die famose Idee, den Computer aus Indien zu importieren?«

◆ → Redaktionsschluß des INFO's

⚡ → 2. Clubtreffen in Holzhausen (siehe INFO 10)

Bitte denkt an die Termine und nehmt sie rechtzeitig wahr !!!

CLUBBO TERMIN - KALENDER 1986

JANUAR

Mo	Di	Mi	Do	Fr	Sa	So
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

FEBRUAR

Mo	Di	Mi	Do	Fr	Sa	So
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
◆	25	26	27	28		

MÄRZ

Mo	Di	Mi	Do	Fr	Sa	So
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

APRIL

Mo	Di	Mi	Do	Fr	Sa	So
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

MAI

Mo	Di	Mi	Do	Fr	Sa	So
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

JUNI

Mo	Di	Mi	Do	Fr	Sa	So
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

JULI

Mo	Di	Mi	Do	Fr	Sa	So
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

AUGUST

Mo	Di	Mi	Do	Fr	Sa	So
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

SEPTEMBER

Mo	Di	Mi	Do	Fr	Sa	So
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

OKTOBER

Mo	Di	Mi	Do	Fr	Sa	So
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

NOVEMBER

Mo	Di	Mi	Do	Fr	Sa	So
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

DEZEMBER

Mo	Di	Mi	Do	Fr	Sa	So
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

◆ --> Redaktionsschluß des INFO's

≠ --> 2. Clubtreffen in Holzhausen (siehe INFO 10)

Bitte denkt an die Termine und nehmt sie rechtzeitig wahr !!!

I N F O R M

An alle Redakteure --- oder an die, die es noch werden wollen.

Aus gegebenem Anlaß möchte ich Euch einige Tips geben, die eine bessere Übersicht und Form der INFO ermöglicht. Gleichzeitig hilft es mir, die Redaktionsarbeit zu vereinfachen. Ich bitte Euch daher, sich an diese **Richtlinie** für Redakteure zu halten.

-- DANKE --

1. ALLGEMEINES :

Durch das Verkleinern der Artikel bekommen wir viel Information auf kleinsten Raum. Dieses System wollen wir beibehalten, es ermöglicht uns nämlich das INFO mit geringem **Kosten- und Materialaufwand** zu erstellen. Um daher eine Verkleinerung Eures Beitrages in bester Qualität zu erhalten, ist folgendes nötig :
Textbeiträge oder Listings sollten ausgedruckt oder mit Schreibmaschine geschrieben sein. Als **Untergrund** für die Buchstaben und sonstige Zeichen empfiehlt sich weißes linienfreies Papier. Bitte sendet keine handschriftlichen Beiträge ein, da meist die persönliche **Randschrift** eines jeden nicht mehr an die in der Schule gelernte **Schönschrift** heranreicht und daher zu Entzifferungsproblemen führt. Wer keinen Drucker/Schreibmaschine hat, kann mir ja sein Textfile auf Diskette zusenden. (Bitte Textverarbeitungssystem und Titel des File angeben)

Verwendet auch nicht gerade die kleinste Schriftart Eures Druckers oder ein altes Farbband, denn auch dies führt nach dem Verkleinern und Drucken zur Unleserlichkeit.

Bei Einsendung eines mehrseitigen Artikels (mehrerer Artikel) denkt bitte daran, die Reihenfolge der Blätter numerisch festzulegen (auf der Rückseite der Beiträge genügt). Ich hoffe, dadurch Verwechslungen innerhalb eines Beitrages oder ein Vertauschen der chronologischen Reihenfolge der Beiträge entgegenzutreten.

2. FORM :

Die Ausdruckform sollte eine Breite von 15 - 17cm haben, und die maximale Zeilenzahl pro Seite darf 62 Zeilen nicht überschreiten. Ich benötige die dadurch erstandenen freien Ränder für die Seitennumerierung /-markierung sowie als Kleberand zum Binden und Abheftlochrand für Euch.

3. ARTIKEL AUS ZEITSCHRIFTEN :

Beachtet bei Artikeln aus Zeitschriften, daß Text, der in farbigen oder gerasterten Flächen steht, durch das Kopieren meist schon schwer leserlich wird. Versucht beim Kopieren die Kopie mit einer helleren Stufe als normal zu erstellen. Ist die Textstelle trotzdem noch nicht **schwarz auf weiß**, solltet Ihr vielleicht eine Abschrift der entsprechenden Textstelle machen.

Ihr habt aber auch die Möglichkeit, mir das Original zukommen zu lassen. Ich habe die Gelegenheit, **heller** zu kopieren mit gleichzeitiger Verkleinerung. Dies ergibt meist eine brauchbare Veröffentlichung für das INFO. Das Original erhaltet Ihr auf jeden Fall mit der INFO zurück.

4. VERSAND :

Nachdem Ihr nun Euer **Schriftwerk** fertig habt, und Ihr es der Redaktion zukommen lassen wollt, solltet Ihr Eure Arbeit nicht zusammenfallen und in ein kleines Kuvert stecken. Diese Knicke geben dann beim Kopieren/Verkleinern graue Streifen, die die Buchstaben an dieser Stelle unkenntlich machen. Darum **ungeknickt** in ein der Größe entsprechendes Kuvert (max DIN A4) und ab an die Post. Portomäßig sind die Kosten für Euch ja gleich.

5. BEITRÄGE ZUR BORSE :

Beiträge zu diesem Thema schreibt bitte in folgender Art und Weise auf:

Sparte (Wer hat was, wer will was oder Fragekasten)
Euren Text (Angebot ,Wunsch oder Frage) --
Euer Autogramm

Ich schneide die Teile aus Eurem Brief und klebe sie dann in die entsprechende Sparte.

6. DISKETTEN :

Bitte gebt grundsätzlich Eure **DISKETTES** an. Lesbar sind für mich nur Singel-Density-Disketten folgender Typen:
40 Track SS oder DS und 80 Track SS oder DS
Zurücksendung Eurer angekommenen Disketten ist Ehrensache. Ihr denkt bei der Diskversendung natürlich auch an die richtige Versendungsform Eurer **Scheiben**. Am besten sind Kuverts mit **Luftpolsterung** --- daß wißt Ihr ja selbst.

Wenn ihr Euch an diese kleine **Redakteuranleitung** haltet, dürfte das die Qualität unseres INFO's noch steigern. Ich hoffe, Ihr helft dabei mit.

Im voraus nochmals meinen DANK

Redaktion CLUB 88

