

INPUT 61

**PREIS
1,90 DM**

Infos · News · Programme · Unterhaltung · Tips

DM 9,80

Unverbindliche

Logik-Simulation:

Digitest

Datei-Archivierung:

Diskettenverwalter

Platz-und-Maus-Spiel:

Labyrinth

C64-Tool:

Screen-Change

C128-Tools:

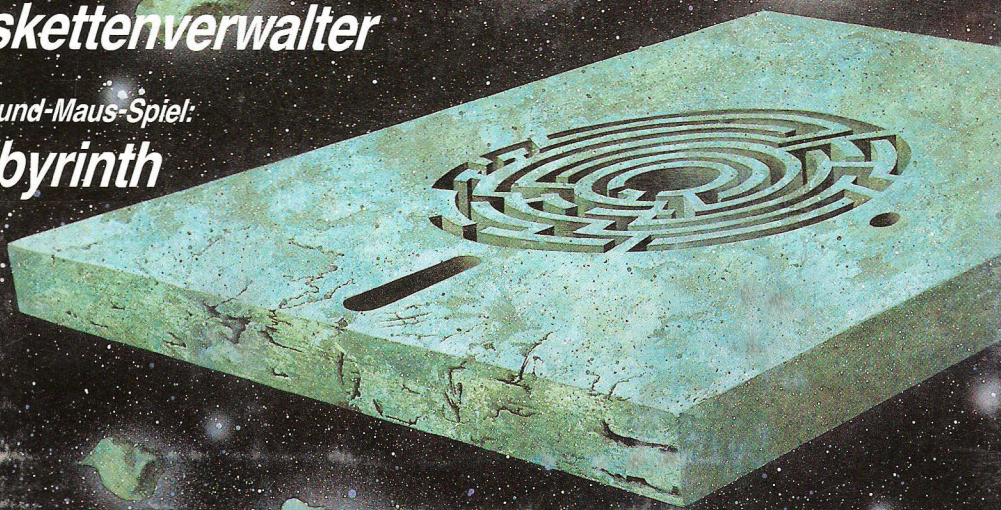
40/80-Zeichen-Hardcopy

REMOUT, RENEW!

Noch mehr Spiele:

Obsternte

Safari



**Über 140 KByte Software
Ohne Abtippen**

Hinweise zur Bedienung

INPUT 64 ist nicht nur einfach eine Programmsammlung auf Diskette, sondern ein Elektronisches Magazin. Es enthält ein eigenes Betriebssystem mit Schnelllader und komfortabler Programmauswahl. Die Bedienung ist kinderleicht:

Bitte entfernen Sie vor dem Laden eventuell vorhandene Steckmodule, und schalten Sie den Rechner einmal kurz aus und wieder ein. Geben Sie nun zum Laden der Diskette

LOAD "INPUT*", 8,1 und RETURN

ein. Alles Weitere geschieht von selbst.

Es wird nun zunächst ein Schnelllader initialisiert. Besitzen Sie ein exotisches Laufwerk oder ist Ihre Floppy bereits mit einem hardwaremäßigen Beschleuniger ausgerüstet, kann es zu Konflikten mit unserem SuperDisk kommen. In diesem Falle sollten Sie versuchen, die Diskette mit

LOAD "LADER*", 8,1 und RETURN

zu laden.

Nach der Titelgrafik springt das Programm in das Inhaltsverzeichnis des Magazins. Hier können Sie mit der Leertaste weiter und mit SHIFT und Leertaste zurückblättern. Mit RETURN wird das angezeigte Programm ausgewählt und geladen.

Das Betriebssystem von INPUT 64 stellt neben dem Inhaltsverzeichnis noch weitere Funktionen zur Verfügung. Diese werden mit der CTRL-Taste und einem Buchstaben aufgerufen. Sie brauchen sich eigentlich nur CTRL und H zu merken, denn mit dieser Tastenkombination erscheint eine Hilfsseite auf dem Bildschirm, die alle weiteren System-Befehle enthält. Nicht immer sind alle Optionen möglich. Befehle, die zur Zeit gesperrt sind, werden auf der Hilfsseite dunkel angezeigt. Hier nun die Befehle im einzelnen:

CTRL und Q

Diese Tastenkombination hat nur während der Titelgrafik eine Bedeutung. Mit ihr wird

das Titelbild abgekürzt, und Sie landen sofort im Inhaltsverzeichnis.

CTRL und H

Haben wir schon erwähnt – damit wird die Hilfsseite ein- und ausgeschaltet.

CTRL und I

Sie verlassen das gerade laufende Programm und kehren ins Inhaltsverzeichnis zurück.

CTRL und F

Ändert die Farbe des Bildschirmhintergrundes. Diese Option funktioniert immer, wenn ein Programm läuft oder Sie sich im Inhaltsverzeichnis befinden, aber nicht auf der Hilfsseite.

CTRL und R

Wie CTRL-F, wirkt auf die Rahmenfarbe

CTRL und B

Sie erhalten einen Ausdruck der Textseite eines laufenden Programmes auf einem angeschlossenen Drucker. Diese Hardcopy-Routine ist angepaßt für Commodore-Drucker und kompatible Geräte. Das Programm wählt automatisch die richtige Geräteadresse (4, 5 oder 6) aus. Sie können diese Routine mit der ←-Taste abbrechen.

CTRL und S

Programme, die auch außerhalb von INPUT 64 laufen, können Sie mit diesem Befehl auf eine eigene Diskette überspielen. Wenn Sie diesen Befehl aktivieren, bekommen Sie unten auf der Hilfsseite angezeigt, wie viele Blocks das File auf der Diskette belegen wird. Geben Sie nun den Namen ein, unter dem das Programm auf Ihre Diskette geschrieben werden soll. In der Regel handelt es sich um Programme, die Sie ganz normal laden und mit RUN starten können. Ausnahmen sind in den jeweiligen Programmbeschreibungen erläutert.

CTRL und D

Gibt das Directory der eingelegten Diskette

aus. Die Ausgabe kann mit der Leertaste angehalten und mit RETURN wieder fortgesetzt werden. Ein Abbruch ist mit der ←-Taste möglich. Wenn das Directory vollständig ausgegeben ist, gelangen Sie mit der RETURN-Taste zurück ins unterbrochene Programm beziehungsweise auf die Hilfsseite.

CTRL und @

Disk-Befehle senden, zum Beispiel Formatieren einer neuen Diskette oder Umbenennen eines Files. Für den zu sendenden Befehls-String gilt die übliche Syntax, natürlich ohne ein- und ausführende Hochkomata. CTRL-@ und RETURN gibt den Zustand des Fehlerkanals der Floppy auf dem Bildschirm aus. Weiter im Programm oder zurück auf die Hilfsseite führt ein beliebiger Tastendruck.

CTRL und A

Sucht auf der Diskette nach einem INPUT 64-Inhaltsverzeichnis. Mit diesem Befehl ist es möglich, ohne den Rechner auszuschalten, Programme von anderen INPUT 64-Disketten zu laden. Das funktioniert aber nur bei den Ausgaben ab 4/86.

Bei Ladeproblemen

Bei nicht normgerecht justiertem Schreib-/Lesekopf oder bei bestimmten Serien wenig verbreiteter Laufwerke (1570) kann es vorkommen, daß das ins INPUT-Betriebssystem eingebaute Schnelladeverfahren nicht funktioniert. Eine mögliche Fehlerursache ist ein zu geringer Abstand zwischen Floppy und Monitor/Fernseher. Das Magazin läßt sich auch im Normalverfahren laden, eventuell lohnt sich der Versuch.

LOAD "LADER", 8,1

Sollte auch dies nicht zum Erfolg führen, senden Sie bitte die Diskette mit einem kurzen Vermerk über die Art des Fehlers und die verwendete Gerätekonstellation an den Verlag (Adresse siehe Impressum).

Liebe(r) 64er-Besitzer(in)!

Kinder spielen gern. Spielen Eltern mit ihren Kindern, können sie sich wohlthuende Bemerkungen wie „Das ist aber ein toller Papi“ oder „Ihr habt aber eine klasse Mutti!“ anhören. Erwachsene spielen auch gern, nur wenn sie allein spielen, dann wird oft geringschätzig über sie gelächelt.

„Moment, werden Sie sagen, Erwachsene spielen ja auch Skat, Canasta und ähnliches. Da wird man nicht belächelt. Und was hat das alles überhaupt mit Computern zu tun?“ Seit es Computer gibt, gibt es auch Computerspiele. Spielt man – vor allem als Erwachsener – solche Spiele, muß man sich schon mal Sprüche anhören wie: „Du hast wohl nichts Besseres zu tun?“ Es sei denn, man befaßt sich mit 'ernsthafteren' Spielen. Das klassische Beispiel: Schach. Da wird niemand von 'nur' spielen reden, das gilt sozusagen als die 'hohe Schule' der Freizeitgestaltung. Wer aber spielt schon Schach am Computer? Fehlt da nicht ein menschlicher Spielpartner? Da fällt einem dann die ganze Palette gängiger Vorurteile über den Computer-Freak ein: sich in sein stilles Kämmerlein zurückziehen, keine Kontakte mehr zur Außenwelt, völlig vereinsamt.


Zugegeben, gelegentlich trifft dies zu, aber es gibt auch Spiele, die man zu zweit spielen kann. So'n richtiger gemütlicher Abend wie beim Skat wird es zwar nicht, aber 'völlige Vereinsamung' ist ja wohl auch verfehlt. „Ja, aber meistens geht es doch um 'Ballerspiele', die . . .“ O.K., O.K., erstens ist Joystick-Artistik immer noch besser, als den ganzen Abend im Fernsehsessel zu hocken, und zweitens gibt es ja auch noch Lern-, Denk- und Strategiespiele. Vor allem die Lernspiele sind hervorragend geeignet, den geistigen Horizont zu erweitern.

Adventures zum Beispiel. Das Denken und taktische Vorgehen wird geschult. Man muß genau überlegen, was zu tun ist, denn man will die Prinzessin ja unbedingt retten – koste es, was es wolle (Abenteuer, wie sie das tägliche Leben nicht mehr bietet). Jeder fanatische Computerspieler kennt das, auch wenn er es nicht gerne zugibt. Dabei kann dann schon einmal die eine oder andere Nacht draufgehen.

Wir erleben immer wieder, daß uns dienstags die Frauen unserer Kunden anrufen (ich finde es auch schade, daß die meisten Computerbesitzer Männer sind) und uns flehentlich um die Auflösung des letzten Adventures bitten. Der Mann sei schon seit vier Tagen keine Nacht vom Rechner weg-zukriegen – „Wenn Sie mir nicht die Lösung sagen, lass' ich mich scheiden.“

'Fanatisch' ist also nicht übertrieben. Durch diese Phase der Besessenheit muß wahrscheinlich jeder einmal durch – in den meisten Fällen also kein Grund zur Sorge, das gibt sich wieder.

Man könnte das Thema noch weiter vertiefen, aber ich will vorm Feierabend schnell noch 'ne Runde 'Safari' spielen.



K.-F. Probst

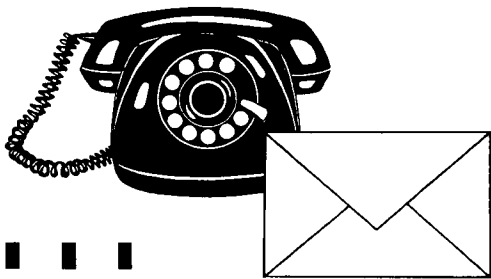
2/88



INHALT

Leser fragen	2
Rückblick: INPUT-BASIC	3
Diskettenverwalter Ordnung für die Programmsammlung	4
Obsterne Das Spiel mit dem Igel	7
Digitest Simulation logischer Schaltungen	8
Gatterlogik Teil 1: Grundlagen digitaler Schaltungen	12
C128-Tools Hardcopy, REMOUT, RENEW	16
Safari Unternehmensplanspiel	18
Labyrinth Irrgartenspiel	20
Nur nicht einmauern! Labyrinth-Algorithmen	21
64er Tips Die FOR-NEXT-Schleife	24
Shape-Editor für Vishypnotica	28
Scrangle Vier Bildschirme im Rechner	30
Vorschau	31
Impressum	32

Leser fragen . . .



MLM64plus auch mit Kassette

Ihr in Ausgabe 11/87 veröffentlichter Maschinsprache-Monitor funktioniert nicht im Kassettenbetrieb. Nach einem S-Befehl zum Abspeichern von Adreßbereichen kommt zwar die Aufforderung „press record & play on tape“, das Programm reagiert aber nicht auf die gedrückten Record-Tasten.

Einer unserer Autoren, Herr Astrath, hat dafür den passenden POKE-Befehl herausgefunden. In die Speicherstelle START plus 877 muß der Wert 47 (\$2F) geschrieben werden. Mit START ist natürlich die Anfangsadresse gemeint, beispielsweise in der Monitor-Fassung ab \$9000 die Adresse 37741. Dadurch initialisiert der Monitor das Datenrichtungs-Register des Prozessorports, die Adresse 0, wie vorgeschrieben mit \$2F statt mit \$FF wie in der veröffentlichten Version. (d. Red.)

Schneller Start für CAD

... Ich habe bei dem Programm „INPUT-CAD“ (10/86-2/87) versucht, den Floppy-Speeder (1/87) anzuhängen, der dann das CAD-Programm nachladen sollte. Doch als ich bei einer Zeichnung erst 10 Prozent des freien Speicherplatzes (ohne Einbindung von SuperTape) verbraucht hatte, stürzte das Programm ab. Deshalb meine Frage: könnten Sie mir eine Anpassung von „SuperDisk“ an „INPUT-CAD“ zuschicken?

M. Rauscher, St.Johann

Das Problem ist lösbar, wenn man verhindert, daß SuperDisk die nachgeladenen

Programme mit RUN startet. Dadurch ist auch gleich die vielgestellte Frage beantwortet, ob auch Maschinenprogramme automatisch nachgeladen werden könnten. Dabei stellt sich nämlich dasselbe Problem.

Die Befehls-Sequenz zum Starten eines BASIC-Programms aus einem Maschinsprache-Programm heraus lautet

```
JSR $A659  
JMP $A7AE
```

Wenn innerhalb von SuperDisk das erste Byte durch ein RTS ersetzt wird, ist der gewünschte Effekt erreicht. Wir veröffentlichten drei Versionen dieses Programms, in jeder Version beginnen die betreffenden Befehle an einer anderen Adresse:

```
Version 1 $08FE d 2302  
Version 2 $0900 d 2304  
Version 3 $0924 d 2340
```

Diese Adressen müssen mit '96', dem Code für den RTS-Befehl, beschrieben werden. Zur Sicherheit sollte nach dem Laden eines Programmes mit einem so veränderten SuperDisk ein NMI (durch RUN/STOP-RESTORE) ausgelöst werden. (d. Red.)

Selbstbestimmter RESET

... habe ich in Ihrem Magazin festgestellt, daß nach einem RESET das Inhaltsverzeichnis angesprungen wird. Wie macht man das, läßt sich das eventuell auch in BASIC-Programmen einsetzen? Leder, Hannover

Im Verlauf der RESET-Routine testet das Betriebssystem des C64, ob sich an den Speicherstellen \$8004 bis \$8008 die Byte-Folge \$C3, \$C2, \$CD, \$38, \$30 befindet, im

Klartext also „CBM80“. Ist das der Fall, wird der Programmzähler auf eine an Adresse \$8000 im Low/High-Byte-Format hinterlegte Adresse gesetzt. Ähnliches gilt für die NMI-Routine, mit dem Unterschied, daß diese sich das Sprungziel aus den Adressen \$8002/\$8003 holt. Diese Abfrage, die eigentlich dem Test auf ein vorhandenes ROM-Modul dient, läßt sich für eigene Zwecke mißbrauchen, sogar in BASIC. Das folgende kleine BASIC-Programm verdeutlicht das Prinzip:

```
10 FOR I = 32768 TO 32782  
20 READ A:POKE I,A  
30 NEXT  
40 DATA 9,128,9,128,195,194,205,56  
50 DATA 48,32,89,166,76,174,167
```

Sind diese Zeilen ausgeführt, holt sich der Prozessor bei einem RESET und bei Betätigung der RESTORE-Taste die weiteren Befehle bei Adresse \$8009 und findet dort ein RUN in Maschinsprache. Dies kann jeder leicht ausprobieren, wenn er obiges Programm nach getaner Arbeit mit NEW löscht, zur Probe die Zeile

```
10 PRINT"TEST":GOTO 10
```

eingibt und RESTORE drückt. (d. Red.)

INPUT muß abonniert werden!

Ich lese INPUT seit Ausgabe 10/86 ... Meine Begeisterung stieg mit jeder neuen Ausgabe. Nach einem halben Jahr war für mich klar: INPUT muß abonniert werden. Jeder neuen Ausgabe sehe ich mit Ungeduld entgegen. Ein besonderes Lob für Ihren Speedcompiler (Ausgabe 10/87). Habe gleich mein erstes BASIC V2-Programm hervorgeholt, einige wenige Änderungen vorgenommen, und es hat geklappt. Super!!!

Vor dem Spielen steht bei mir das Entwickeln und Programmieren. Da ich dies ausschließlich in BASIC V70 mache, würde ich mich freuen, wenn Sie mehr Tips und Hilfsprogramme für den C128 bringen würden.

U. Orth, Mayen

In dieser Ausgabe beginnt eine in loser Reihenfolge erscheinende Tool-Sammlung für den C128. (d. Red.)

Blick zurück nach vorn

INPUT-BASIC und der Rest der Welt

INPUT-BASIC ist eine BASIC-Erweiterung mit über 40 neuen Befehlen, die in Ausgabe 1/86 erschienen ist. Eine Reihe wichtiger Programme, die seitdem veröffentlicht wurden, sind auf ein „Zusammenspiel“ mit dieser Spracherweiterung abgestimmt, wie der Floppy-Beschleuniger SuperDisk (1/87) oder der in INPUT 64, 10/87 veröffentlichte BASIC-Compiler.

INPUT-BASIC + Ist BASIC = BASIC100

Außerdem brachten wir in der letzten Ausgabe eine weitere BASIC-Erweiterung, Ist BASIC, die INPUT-BASIC-verträglich ist. Hat man diese beiden Erweiterungen zusammen im Speicher, stehen fast 100 neue Befehle zur Verfügung.

Besser zwei als eins

Aus Platzgründen waren die Hinweise für den gleichzeitigen Umgang mit beiden Programmen in der Beschreibung zu Ist BASIC auf einige knappe Bemerkungen beschränkt, die sicherlich nicht alle auftauchenden Fragen beantworten. Darum jetzt etwas ausführlicher.

Wenn sie also mit beiden Erweiterungen arbeiten wollen, gehen Sie wie folgt vor:

1. Laden Sie INPUT-BASIC und starten es mit RUN. Wählen sie aus, ob Sie mit oder ohne SuperTape arbeiten möchten.
2. Laden sie Ist BASIC und starten auch dieses mit RUN.
3. Teilen Sie dem Programm durch 'POKE 39093,x' mit, welchen Modus Sie benutzen möchten. 'x' kann die Werte 0, 1 und 2 annehmen, die – in dieser Reihenfolge – die Bedeutung 'keine Erweiterung im Speicher', 'INPUT-BASIC ohne

Gibt es einen vernünftigen Grund, sich über ein vor zwei Jahren veröffentlichtes Programm noch einmal auszulassen? Es gibt sogar mehrere. Zum Beispiel, daß diese BASIC-Erweiterung im Rahmen einer Leserumfrage zum beliebtesten Programm gekürt wurde. Oder daß es immer noch zahlreiche Anfragen dazu gibt. Und daß der Speedcompiler darauf abgestimmt ist. Und und und . . .

SuperTape' und 'INPUT-BASIC mit SuperTape' haben. (Die Zuordnung der POKE-Werte war in der Beschreibung zu Ist BASIC falsch.)

Lösen Sie einen Reset aus, nachdem Ist BASIC initialisiert ist, können Sie zwischen Ihrem normalen BASIC oder Ist BASIC/INPUT-BASIC wählen. Haben Sie sich für Ist BASIC/INPUT-BASIC entschieden, werden beide Versionen, wenn sie vorhanden sind, neu initialisiert.

Wie man sich bettet, . . .

Die Erweiterung belegt den Speicherplatz von \$8000 bis \$98B5. Beim Start schützt sich das Programm selbsttätig vor dem Überschreiben durch BASIC-Daten. Wie man sieht, belegt die Erweiterung zwar einen Teil des BASIC-Speichers, der Ihnen dadurch nicht mehr zur Verfügung steht, aber aufgrund der neuen Befehle fallen die so erstellten Programme viel kürzer aus als bisher. Der verringerte Speicher fällt somit kaum ins Gewicht.

Nach einem Reset haben Sie die Wahl zwischen INPUT-BASIC (1), damit ist die ge-

samte INPUT-BASIC-Erweiterung gemeint (INPUT-BASIC und Ist BASIC) und dem normalen BASIC (1).

. . . so liegt man

Wählen Sie die erste Möglichkeit, wird INPUT-BASIC so geändert, daß Sie bei einer neuen Initialisierung mit SYS 49152 beide Erweiterungen neu starten. Entscheiden Sie sich für die zweite Möglichkeit, müssen Sie INPUT-BASIC mit SYS 49152 und Ist BASIC mit SYS 36474 einzeln neu starten.

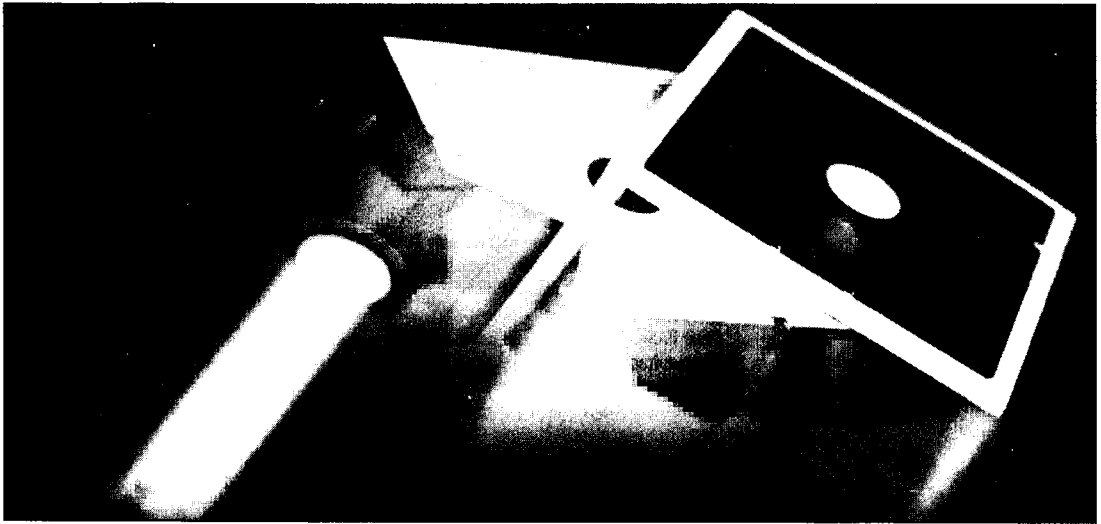
Außerdem soll nicht unerwähnt bleiben, daß die Erweiterung mit Zwei-Byte-Token arbeitet. Das erste Byte ist immer ein \$FE, das zweite je nach Befehl ein Zeichen von \$B0 bis \$E3. Diese Wahl der Token garantiert ein korrektes Umnúmerieren mittels 'RENUMBER' aus INPUT-BASIC.

INPUT-BASIC-Kennern ist bekannt, daß der Befehl AUTO des INPUT-BASIC bei Zeilennummern, die größer als 32768 sind, nicht einwandfrei funktioniert. Der Fehler kann beseitigt werden, wenn Sie beide BASIC-Erweiterungen im Speicher und initialisiert haben. Geben sie einfach ein DOKE 50462,34567 ein, und es sind wieder Zeilennummern bis 63999 möglich.

INPUT-BASIC + BASIC V2.0 = BASIC V7.0

Ebenfalls in der letzten Ausgabe haben wir den Wandler „BASICon“ veröffentlicht, der in BASIC V7.0 geschriebene Programme für den C16 für den C64 umsetzt. Eine genaue Zuordnung von INPUT-BASIC- und V7.0-BASIC-Befehlen finden Sie in der Tabelle. Bei INPUT-BASIC wird mit MODE der Zeichenmodus festgelegt. Mit 'MODE 1' werden die Punkte gelöscht, mit 'MODE 0' gezeichnet. Beim C16 bedeutet die erste Zahl hinter einer Malanweisung (DRAW . . .) die Farbzonennummer (##), das heißt, '0' ist Hintergrund (also löschen), '1' Zeichenfarbe (also setzen), '2' und '3' beziehen sich auf den Mehrfarbenmodus, welcher nicht direkt nachvollzogen werden kann. Ist '# #' ausgelassen, wird 1 angenommen.

Forts. S. 28



Einträge über Einträge

Diskettenverwaltung

Ein Dateiprogramm zur Diskettenverwaltung ist also ein Programm, das die Inhaltsverzeichnisse von verschiedenen Disketten lädt und verwaltet: sortiert, druckt, sucht und so weiter. Zwei Ziele hatte ich mir dabei gesetzt:

1. Das Programm sollte einfach zu bedienen sein. Die Bedienung erfolgt deswegen über die altbewährte Menütechnik – auf dem C64 immer noch das Beste und Schnellste. Der komplette Dialog zwischen User und Programm erfolgt fast immer auf dieselbe Art und Weise.
2. Das Programm sollte „hart im Nehmen“ sein. Dies ist eigentlich selbstverständlich. Wer aber selbst schon Programme geschrieben hat, wird wissen, wie viel Schweiß es kostet, den Computer zum Beispiel richtig auf Laufwerksfehler reagieren zu lassen und sich nicht einfach mit blinkender Floppy-LED zu verabschieden.

Ein Programm, das nichts anderes tut, als eine Diskettendatei zu verwalten, die nichts anderes enthält als die Inhaltsverzeichnisse von anderen Disketten? Daß so etwas sinnvoll sein könnte, ist sicherlich nicht nur dem Autor dieses Programms angesichts immer unübersichtlicher werdender Diskettenberge aufgefallen. Er hat seine Überlegungen in die Praxis umgesetzt, herausgekommen ist das vorliegende Programm, der „Diskettenverwalter“.

Automatischer Anfang

Innerhalb von INPUT 64 werden fast alle Funktionen der Diskettenverwaltung demonstriert. Um damit Ihre real existierenden Scheiben verwalten zu lassen, muß das Programm erst auf einen eigenen Datenträger abgespeichert werden. Wie das geht, steht auf der zweiten Umschlagseite,

wie auch alle anderen notwendigen Hinweise zum Umgang mit unserem Diskettenmagazin.

Ist das Programm geladen und mit 'RUN' gestartet, sucht es zunächst auf der Diskette nach einer festen Datei. Wenn sie vorhanden ist, wird sie automatisch geladen, und es entfällt das Anwählen des entsprechenden Menüpunkts („Datei“). Anschließend erscheint das Hauptmenü. Es bietet folgende Möglichkeiten:

1. Neue Diskette laden
2. Programmverwaltung
3. Diskettenverwaltung
4. Datei
5. Disketteninhalt
6. Diskettenbefehl
7. Programm beenden

Die folgende Beschreibung orientiert sich an diesen einzelnen Menüpunkten.

Datenaufnahme

1. Neue Diskette laden

In diesem Programmteil werden die notwendigen Informationen aus dem Directory der eingelegten Diskette in die Verwaltungsdatei übernommen. Das Programm fordert Sie auf, entweder mit RETURN neue Daten aufzunehmen oder die RUN/STOP-Taste zu drücken. Mit RUN/STOP kehrt man wieder ins Hauptmenü zurück; dies gilt übrigens auch für (fast) alle anderen Untermenüs und Eingabeschleifen. Tut's RUN/STOP einmal nicht, hilft statt dessen ein Druck auf die '←'-Taste.

Nach Drücken von RETURN läuft die Floppy an, und das Inhaltsverzeichnis wird, soweit möglich, geladen. Soweit möglich meint, daß Disketten mit manipulierten Directories (beispielsweise INPUT-Disketten!) nicht korrekt eingelesen werden. Danach befindet sich das Inhaltsverzeichnis bereits komplett im Arbeitsspeicher. Directory-Einträge, die nicht übernommen werden sollen, können nur durch nachträgliches Löschen entfernt werden. Deshalb springt das Programm nach dem Laden in den sogenannten Editor, in dem umfangreiche Korrekturmöglichkeiten zur Verfügung stehen.

Das Fenster zum Speicher

Die Idee war, einen einzigen Programmteil zu schreiben, der nur für den Dialog zwischen Benutzer und Speicher zuständig ist. Dieses Programm sollte wie ein „Fenster mit Filter“ wirken. „Fenster“ deshalb, weil der Bildschirm wie ein Fenster mit Ausblick auf den Speicher wirkt. Es wirkt auch wie ein „Filter“, da der Zugriff auf den Speicher immer bestimmten Kriterien unterworfen ist. Beispielsweise ist der Zugriff nach dem Laden eines neuen Inhaltsverzeichnisses nur auf eben dieses möglich.

Bedienung des Editors: In der Mitte des Bildschirms werden die geladenen Programmnamen dargestellt. Rechts daneben ist Raum für beliebige Bemerkungen. Links unten stehen zwei Zahlen. Die linke zeigt die aktuelle Position des Pfeils an, die rechte gibt an, auf wie viele Programmnamen insgesamt zugegriffen werden kann. Mit ↑ (Cursor hoch) und ↓ (Cursor runter) kann

der Pfeil nach oben und unten bewegt werden. Stößt der Pfeil oben oder unten an, wird das Fenster entsprechend verschoben.

F5 und F7 wirken wie die Cursor-Tasten, aber der Pfeil wird jeweils um drei Zeilen nach oben oder unten verschoben.

F6 und F8 ermöglichen ein kontinuierliches Verschieben nach oben oder unten.

F1 führt zum Anfang der Tabelle.

F3 zeigt das Ende an.

'N': Namen editieren. Mit dieser Taste ist es möglich, den Programmnamen, auf den der Pfeil zeigt, zu ändern. Dazu wird der Programmname in eine Eingabezeile in der vorletzten Bildschirmzeile geschrieben, in der er nach Belieben editiert werden kann.

Editieren kann man, wie üblich, mit ← und ⇒ sowie den Tasten INSERT und DELETE. Der geänderte Text wird mit RETURN übernommen. Mit der STOP-Taste kann die Eingabe abgebrochen werden.

'B': steht für „Bemerkung“. Damit kann man zu jedem Programmnamen eine beliebige Bemerkung schreiben. Ansonsten gilt das bei Taste 'N' Gesagte.

In der Bemerkung könnte zum Beispiel stehen, auf welcher Diskettenseite (A oder B) das Programm steht, um welche Version es sich handelt und ähnliches.

SHIFT + 'B': ändert alle Bemerkungen, auf die zugegriffen werden kann. Da die Funktion unter Umständen etwas länger dauert, kann sie mit der STOP-Taste abgebrochen werden.

RETURN: zeigt den Namen der Diskette an, auf der sich das Programm befindet.

SHIFT + RETURN: der Diskettenname kann geändert werden.

'L': löscht den Programmnamen unwiderruflich aus dem Speicher.

SHIFT + 'L': löscht kontinuierlich so lange, bis eine Taste gedrückt wird

'D': gibt den aktuellen Programmeintrag als Text im ASCII-Format auf dem seriellen Bus (Geräteadresse 4 bis 15) aus. Nach Drücken von 'D' werden Geräte- und Sekundäradresse in hexadezimaler Form erwartet, eventuell auch ein File-Name.

\$0801–\$2FEE	Programm
\$2FFF–\$CFFF	Arbeitsspeicher
\$D000–\$DFFF	Quicksort-Stack
\$E000–\$F3FF	und Zeigertabelle

Speicherbelegung der Diskettenverwaltung:
Fast die gesamten 64 KByte des C64 werden zur Ablage von Programmcode und Daten benutzt.

Beispiele:

47: gibt den Eintrag auf den Drucker mit Geräteadresse 4 und Sekundäradresse 7 aus.

82datnam: speichert den Eintrag auf der Floppy mit Device-Nummer 8 unter dem Namen „datnam“. Die so erzeugten Dateien können mit einer Textverarbeitung oder einem Editor, zum Beispiel dem von INPUT-ASS!, weiterverarbeitet werden.

SHIFT + 'D': gibt alle Programmnamen, numeriert mit Bemerkung und zugehörigem Diskettennamen, aus. Es werden wiederum nur die Einträge ausgegeben, auf die der Editor zugreifen kann. Für das Ausgabemedium gilt das unter 'D' gesagte.

'←': mit der ←-Taste schließlich wird der Editor verlassen.

Ordnen, suchen, orientieren

2. Programmverwaltung

In der Programmverwaltung hat man folgende fünf Möglichkeiten:

- nach Programmen suchen
- nach Bemerkungen suchen
- Programme sortieren
- Bemerkungen sortieren und
- die ursprüngliche Reihenfolge wiederherstellen

Wollen Sie nach Programmen oder Bemerkungen suchen, erwartet der Computer die Eingabe der Suchmaske. Es können, ähnlich wie bei der Arbeit mit der Floppy, die Joker '?' und '*' verwendet werden. Wird beispielsweise nur '*' eingegeben, findet die Dateiverwaltung alle Programmnamen im Speicher.

Mit den Punkten '3' und '4' können die Programmnamen und Bemerkungen sortiert werden. Zum Einsatz kam hier der sehr schnelle Quicksort-Algorithmus, der 2000 Elemente in der atemberaubenden Zeit von nur 10 Sekunden sortiert².

Mit der Taste '5' kann die ursprüngliche Reihenfolge, also die Reihenfolge des Einlesens der Disketten in den Speicher, wiederhergestellt werden. Diese Funktion organisiert und initialisiert den Speicher neu. Außerdem werden Diskettennamen, unter denen keine Programmnamen eingetragen sind („leere“ Disketten), entfernt. Jeder angezeigte Strich steht somit für 26 freie Bytes mehr. Eine „leere“ Diskette entsteht dann, wenn nach dem Laden einer neuen Diskette alle Programmnamen mit SHIFT + 'L' gelöscht werden. Der Diskettenname wird dabei nämlich **nicht** gelöscht.

Noch mehr Ordnung

3. Diskettenverwaltung

Dieser Menüpunkt ist eigentlich nur der Vollständigkeit halber eingebaut. Es werden in ähnlicher Weise wie beim Programm-Editor die Diskettennamen angezeigt. Mit den Cursor-Tasten kann man sich nach oben und unten bewegen. Mit RETURN werden die Programme, die sich auf dieser Diskette befinden, im Programm-Editor dargestellt. Mit der STOP-Taste verläßt man den Disketten-Editor wieder.

Die Datei der Dateien

4. Datei

Im Menüpunkt „Datei“ sind einige Funktionen zusammengefaßt. In der unteren Bildschirmhälfte werden ein paar statistische Angaben gemacht. Darüber befindet sich wieder ein kleines Menü:

- Datei speichern
- Datei laden
- Speicher löschen

Der Speicherinhalt (die „Datei“) kann unter einem beliebigen Namen abgespeichert werden. Der Vorschlag vom Computer ist der Name der festen Datei, die am Anfang immer geladen wird. Nach RETURN fragt der Computer, ob der Name übernommen werden soll. Die Taste 'J' bewirkt dann, daß auch beim Laden nach dem Starten und beim Speichern vor dem Verlassen des Diskettenverwalters dieser neue Name benutzt wird.

Sollte eine Datei bereits auf der Diskette vorhanden sein (Fehlermeldung: 63, file exists), will das Programm weiter „wissen“, ob es die Datei vorher löschen darf. Wird ihm das „erlaubt“, geht der Computer unter Beachtung aller Regeln der Datensicherheit nach folgendem Schema vor:

- a) Löschen der Datei „+back up datei“
- b) Umbenennen der alten Datei in „+back up datei“
- c) Löschen der alten Datei auf der Disk
- d) Abspeichern der neuen Datei

„Speicher löschen“ löscht den Speicher nach einer Sicherheitsabfrage. Verlassen können Sie diesen Menüpunkt mit der STOP-Taste.

Reale Zugriffe

5. Disketteninhalt

Zeigt das Inhaltsverzeichnis der eingelegten Diskette Seite für Seite an. Das Auflisten kann nach jeder Seite mit der STOP-Taste abgebrochen werden.

6. Diskettenbefehl

In einem weiteren Menü gibt der Computer Ihnen die Möglichkeit, die eingelegte Diskette zu manipulieren:

- File löschen
- File umbenennen
- Disk initialisieren
- Validate
- Diskettennamen ändern
- Neue Disk formatieren

Bis auf den vorletzten Punkt handelt es sich dabei um Standardkommandos, die über den Befehlskanal geschickt werden.

Sicheres Ende

7. Programm beenden

Das Programm fragt, ob die Systemdatei (die feste Datei, die auch am Anfang . . .) automatisch abgespeichert werden soll. Drücken Sie 'J' für „ja“, wird die Datei abgespeichert und das Programm verlassen. 'N' beendet das Programm, ohne die Datei abzuspeichern. Mit der STOP-Taste kann man aber auch wieder ins Hauptmenü zurückkehren.

Der Diskettenverwalter kann ohne Textverlust und ohne Laden der Systemdatei mit SYS 2140 gestartet werden.

Volle Speicher

Konfrontiert der Computer Sie mit der Meldung „Es ist kein Speicher mehr frei“, so kann entweder kein Platz mehr für neue Programmeinträge vorhanden oder die Obergrenze von 2041 Programmnamen erreicht sein. Das Programm gibt jedoch nicht immer diese Meldung aus. Im Editor etwa würde es bei bestimmten Operationen ('N', 'B' und SHIFT + 'B') schlicht deren Ausführung verweigern.

Schnelle Floppies

Das vollständig in Maschinensprache geschriebene Programm benutzt nur die „offiziellen“ Einsprungadressen in das 64er Betriebssystem. Es kann also jederzeit ein Floppy-Spieder eingebunden werden. Man muß jedoch auf zweierlei Dinge achten. Erstens sollte der Spieder nur den Bereich von \$C000-\$DFFF benutzen und zweitens sollte man im Programm darauf achten, daß immer etwa 4 KByte frei bleiben, damit der Bereich von \$C000-\$CFFF, der nämlich sonst auch benützt werden würde, nicht den Floppy-Spieder überschreibt. Die genaue Speicherbelegung finden Sie im Kasten.

¹ Der in Ausgabe 6/86 veröffentlichte Makro-Assembler mit integriertem Editor.

² Nach: 64'er Magazin 8/85 und 12/85, Haar bei München

Nicht gleich in die Luft gehen

**Geschicklichkeitsspiel:
Obsternte**



Hier wird das Obst einfach mit einem gezielten Stockwurf vom Baum geholt. Daß dabei eher Fallobst denn Obst der Handelsklasse 1 eingesammelt wird, braucht uns hier nicht weiter zu stören.

Von Igel, Vögeln . . .

Sicher haben Sie schon den nervösen Igel unter dem Baum gesehen. Was passiert, wenn Sie – in Form des Farmers – mit diesem Tierchen in Berührung kommen, können Sie in der Überschrift nachlesen.

Jedenfalls besteht Ihre Aufgabe darin, das Obst vom Baum zu werfen, persönlich einzusammeln und anschließend auf dem Wagen abzulegen. Dabei müssen Sie wohl oder übel ein paarmal über das Stacheltierchen springen.

Es gibt sicherlich unterschiedliche Methoden, an Baumobst zu gelangen. Mal abgesehen von der Möglichkeit, im Geschäft selbiges in Tüten zu erstehen, könnten Sie direkt auf die Bäume klettern und Stück für Stück herunterpflücken oder vom sicheren Boden aus mittels einer langen Stange . . . Eine ganz andere Technik können Sie in dem Spiel „Obsternte“ kennenlernen.

. . . und Insektiziden.

Was so ruhig und streßfrei beginnt, produziert in den höheren Levels doch schon einige Adrenalinstöße.

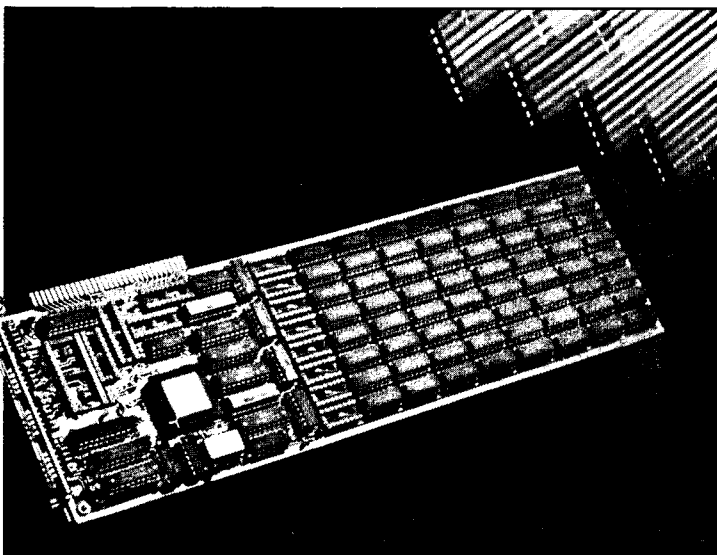
Sie haben nicht nur einen Neider, der das Obst viel lieber selbst fressen möchte, nein,

allen Umweltbelastungen zum Spott werden von einem Flugzeug Insektenvertilgungsmittel verstreut. Es dürfte sich inzwischen herumgesprochen haben, daß dieses Zeug auch den Farmern nicht sonderlich bekömmlisch ist.

Bedienung auf einen Blick

Zu Beginn können Sie den Schwierigkeitsgrad mit der Funktionstaste 'f1' beeinflussen. Mit Space gelangen Sie dann zum Spielfeld.

Das Spiel ist sowohl mit der Tastatur als auch mit einem Joystick zu bedienen. Da sich die Joystick-Bedienung von selbst ergibt, hier nur die Tastenbelegung: '/' und 'A' sind für die jeweilige Laufrichtung, 'A' für einen Sprung und die Space-Taste für einen Wurf zu betätigen. T. Ladenberger/WM



Erst studieren, dann probieren . . .

Digitest: Simulator für integrierte Digitalschaltungen

Genau betrachtet handelt es sich bei diesem von Gerrit Knoef geschriebenen Programm um einen Zwitter: einerseits ist es ein „ernsthafte“ Anwenderprogramm zur Erstellung von TTL-Schaltungen, andererseits ist es ein Lehr- und Übungssimulator, der den leichten Einstieg in die Digitaltechnik ermöglicht. Überhaupt bietet die Digitaltechnik einen guten Einstieg in die Elektronik, ist sie doch zwischenzeitlich auch gar nicht mehr wegzudenken aus dem täglichen Leben. Welches Elektrogerät ist heute noch ohne mikroprozessorgesteuertes Elektronengehirn denkbar? Doch ist es nicht jedermanns Sache, am heimischen Küchentisch den Lötkolben zu schwingen und in eine neue Welt aufzubrechen. Als Alternative bieten wir allen Interessierten

Moderne elektronische Rechner arbeiten mit nur zwei unterscheidbaren Zuständen - Null und Eins. Alle daraus resultierende Intelligenz der Rechner ist nur das Ergebnis der Verknüpfung dieser Elementarzustände. Wer tiefer in dieses Geheimnis einsteigen möchte, kann sich mit diesem Logiksimulator die Grundlagen der digitalen Schaltungslogik erarbeiten. Der in diesem Heft beginnende Grundlagenkurs ermöglicht es auch allen Anfängern, Einblicke in die Computerhardware zu bekommen.

dieses Programm mit der in diesem Heft startenden dreiteiligen Einführung in die Boolesche Algebra. Eine kinderleichte Bedienung des Programms ist durch die grafische Benutzeroberfläche gegeben, so daß auch der absolute Anfänger schnell die ersten Erfolge verbuchen kann. Der erfahrene Anwender hingegen kann auf die schnelle eine Schaltungsidee ausprobieren. Natürlich sind dabei ein paar Restriktionen zu beachten, doch dazu später.

Aus zwei mach eins

Vor der Erstbenutzung von Digitest sind einige Vorbereitungen nötig, an deren Ende eine fertige Arbeitsdiskette steht. Innerhalb von INPUT ist das Programm wegen seines Umfangs nicht lauffähig; im Magazin können und müssen zwei Dateien auf eine eigene Diskette abgespeichert werden. Beachten Sie dabei die Bedienungshinweise auf Seite 2 und die Anmerkungen, die innerhalb des Digitest-Moduls stehen. Haben Sie beide Installationsteile gesichert, können Sie das Magazin verlassen und die Digitest-Programmdiskette erzeugen. Hierfür muß das erste File (das längere der beiden) geladen und gestartet werden. Jetzt ist unbedingt ein Diskettenwechsel erforderlich, da das Programm nach Drücken von Shift-S die Diskette formatiert und mehrere Programmteile auf diese speichert. Ist dieser Vorgang beendet, laden Sie den zweiten Teil des Generierungsprogramms, indem Sie vorher den Rechner kurz aus- und wieder einschalten. Mit Shift-S werden die restlichen Programmteile auf die angefangene Programm-Disk übertragen.

Öffnen der Testbench

Das erste Programm auf der Arbeitsdiskette heißt "DIGITEST" und ist der Lader für den Simulator. Dieser muß absolut geladen werden, also startet man wie folgt: LOAD "DIGITEST",8,8. Für eine bequeme Handhabung sollte ein Joystick in Port 2 benutzt werden. Ersatzweise läßt sich das Programm auch mit der Tastatur bedienen, benutzen Sie die Tasten W, S, C, V und die Leertaste. Nach dem Laden erscheint eine Infoseite, ein Druck auf den (Feuer-)Knopf blättert um, und die Arbeitsfläche, auch

Testbench genannt, erscheint. In der linken oberen Ecke ist ein Pfeil zu sehen, den wir auch (Maus-)Zeiger oder Maus-Cursor nennen. Er wird mittels Joystick oder entsprechenden Tasten über den Bildschirm bewegt. Im oberen Teil des Bildes ist das Hauptmenü sichtbar, darunter die Anschlüsse der Testbench und die Versuchsfläche. Anschlußleiste und Arbeitsfläche sind erheblich größer als der dargestellte Ausschnitt. Daß dieser Platz notwendig ist, werden Sie bei der ersten Schaltung merken. Es gibt zwei Möglichkeiten, den Ausschnitt zu verschieben. Bewegen Sie den Zeiger einmal auf eines der vier Pfeilkästchen in der Menüleiste, bis das Feld sich umfährt. Auf Knopfdruck bewegt sich der Ausschnitt in die gewünschte Richtung. Die andere Möglichkeit, den Bildschirm zu scrollen, ist oftmals schneller. Dafür muß nur der Mauszeiger an den Rand des Ausschnitts bewegt und der Joystick losgelassen werden. Nach einer kurzen Verzögerung setzt sich die Fläche in Bewegung. Gestoppt wird das Scrollen, indem der Zeiger vom Rand weg bewegt wird. So schwer, wie es sich anhört, ist es aber nicht.

Die einzelnen Funktionen werden alle nach dem gleichen Schema aufgerufen. Der Zeiger wird auf den gewünschten Menüpunkt

plaziert, bis sich das Feld invertiert. Diese Auswahl muß mit Knopfdruck bestätigt werden. Ein kurzer, hoher Ton signalisiert, daß alles okay ist. Er klingt ein langes, tiefes Signal, dann ist die Funktion nicht korrekt eingeleitet oder durchgeführt worden.

Von oben nach unten

Bei grafisch-/menüorientierten Programmen läuft ein großer Teil der Bedienung intuitiv. Der beste Weg, solch ein Programm zu erkunden, besteht darin, daß man einfach ein bißchen damit „rumspielt“. Das ist ja das Schöne an einer Simulation gegenüber der „Wirklichkeit“: es geht nichts ernsthaft kaputt. Die ICs brennen sozusagen nur symbolisch durch, was sich auch auf der Kostenseite sehr angenehm auswirkt. Im folgenden Teil wollen wir uns durch alle Menüs „durchhangeln“, dabei ist die Vorgehensweise nicht nach Funktionen strukturiert, sondern orientiert sich an den Menüs.

GET CHIP

Über diesen Befehl erreicht man alles, was mit der Auswahl von verschiedenen ICs zu tun hat. In der Bibliothek sind sechzehn verschiedene TTL-ICs der 74xxx-Serie verzeichnet, diese dienen als Grundbaustein

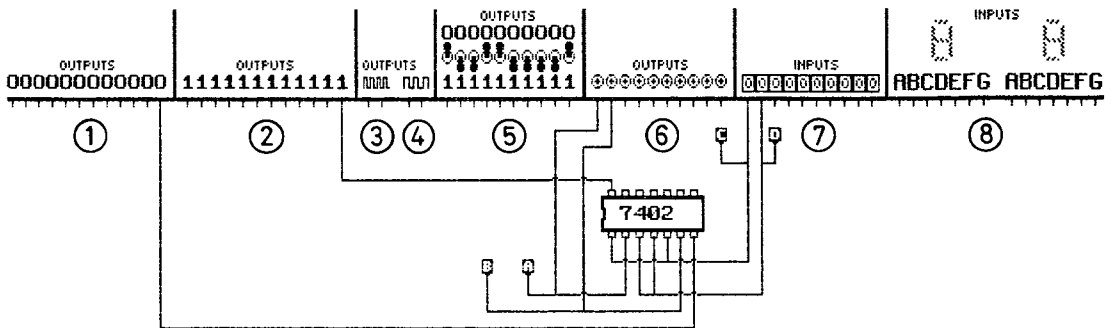
für die Schaltungen. Mit GET CHIP ins Untermenü gelangt, wählt man mit dem Zeiger ein IC aus; die Wahl wird in der obersten Zeile angezeigt. **INFO ON SCREEN** oder **INFO ON PRINTER** liefern eine Kurzbeschreibung des ICs. Bei fehlendem oder abgeschaltetem Drucker gibt das Programm eine entsprechende Fehlermeldung aus; über die OK-Box gelangt man eine Ebene höher. Ist der angewählte Baustein der richtige, kann er mit **GET CHIP** auf die Arbeitsfläche übernommen werden. Dort „klebt“ er am Mauspfeil. Sichtbar wird er erst, wenn er sich auf der Testbench befindet und an jener Stelle auch genügend Raum ist, um ihn zu plazieren. Ist eine entsprechende Stelle gefunden, kann das IC mit dem Knopf abgesetzt werden.

ERASE CHIP

Nach dem Anklicken dieser Funktion wird mit der Maus der versehentlich gesetzte oder nicht mehr benötigte Chip angefahren und dann mit dem Knopf endgültig entfernt.

CONNECT

Allein nebeneinander liegende elektronische Bausteine machen noch keine Schaltung, eine entsprechende Verdrahtung muß her. Ist der Connect-Modus aktiviert, kann ver-



- | | | | |
|--------------------|------------------|------------|----------------------|
| ① Low Level (0 V) | ③ Schneller Takt | ⑤ Schalter | ⑦ Indikator-Lämpchen |
| ② High Level (5 V) | ④ Langsamer Takt | ⑥ Taster | ⑧ Ziffernanzeigen |

Die umfangreiche Ausstattung der Standard-Testbench mit einem Versuchsaufbau.

drahtet werden. Nach dem Anfahren eines Anschlußbeinchens (IC oder auch Testbench) erscheint eine kurze Bezeichnung des Anschlusses im Menüstreifen. Anklicken färbt den Maus-Cursor gelb, das eigentliche Verbinden beginnt. Jetzt wird die Verbindung zum gewünschten zweiten Kontakt gezogen und abermals mit dem Knopf der Punkt angeklickt; dies beendet die Verbindung. Auch eine bestehende Leitung kann als Anknüpfungspunkt dienen, die Auswahl erfolgt genau wie bei jedem anderen Punkt auch. Ist der Verbindungsmodus (gelber Zeiger) aktiv, können Sie diesen durch die RUN/STOP-Taste oder durch **CUT CONNECTION** verlassen.

Ist der Zeiger blau, wird er bei der Anwahl von **CUT CONNECTION** in eine Schere verwandelt, um bestehende Verbindungen zu löschen.

SPECIAL

CLEAR PROJECT löscht nach Rückfrage das gerade im Speicher befindliche Projekt. Eine Bestätigung erfolgt über die YES-Box, ein Abbruch mit NO.

Der Menüpunkt **LOAD TESTBENCH** ist im Augenblick bedeutungslos, er ist für mögliche Erweiterungen vorgesehen.

Die vier möglichen Mini-Scope-Anschlüsse mit den Bezeichnungen A bis D setzt man mit **PUT PROBE** oder löscht sie entsprechend mit **ERASE PROBE**.

Da sich die Schaltungen alle ein bißchen ähnlich sehen (viele ICs und Verbindungen), ist jedem Projekt eine Namenszeile zugeordnet, die im Hauptmenü angezeigt wird. **ENTER INFO-LINE** ruft einen Line-Editor, mit dem das Kind seinen Namen bekommt. Bestätigen Sie die Zeile mit ENTER.

DISK

Projekte können auf Diskette gesichert und auch von dort wieder geladen werden. **SAVE PROJECT** ruft den Line-Editor zum Eingeben des File-Namens auf. Nach Eingabe des Namens führt PROCEED den Speicherbefehl aus.

Sind auf der eingelegten und mit **OTHER DISK** eingeloggten Diskette Projekte, werden diese auf dem Bildschirm angezeigt.

Ein angeklickter Projektname erscheint in der obersten Menüzeile, **LOAD PROJECT** lädt die gewählte Schaltung. Der aktuelle Speicherinhalt geht hierbei verloren, weshalb Sie ihre Absicht nochmals mit **PROCEED** bekräftigen müssen.

Ganz erwartungsgemäß formatiert der Befehl **FORMAT DISK** eine im Laufwerk befindliche Diskette. Die auf der Diskette vorhandenen Daten gehen dabei verloren.

TEST

Der Befehl **TEST** startet den Simulator, und in der Kopfzeile erscheint die Meldung „CLEANING UP CONNECTIONS“. Sind in der Schaltung logische Fehler vorhanden, die Kurzschlüsse verursachen, gibt es eine eindeutige optische und akustische Meldung. Anschließend zeigt der Maus-Cursor auf

die fehlerhafte Stelle. Die Prüfung der Schaltung kann je nach Umfang einen Moment dauern, danach beginnt die eigentliche Simulation. Sollte die Überprüfung eines Testaufbaus nicht enden wollen, kann das daran liegen, daß die Stromversorgung der ICs nicht oder nicht korrekt angeschlossen ist. In jedem Fall muß an den mit „+V“ bezeichneten IC-Anschlüssen 5 Volt liegen, dafür wird eine Verbindung mit „OUTPUTS 1“ hergestellt. Der mit „Ground“ bezeichnete IC-Anschluß ist die Masse und benötigt eine Verbindung zu „OUTPUTS 0“. Fehlt bei einem IC die Spannung, hängt das Programm in einer Endlosschleife, der Zeiger läßt sich nur noch ganz langsam und mit viel Geduld zu der CANCEL-Box bewegen.

Soll der Mauszeiger als Logiktester benutzt werden, muß er mit der Spitze auf einen Anschlußpunkt zeigen. Er zeigt dann auf

Hauptmenü	Untermenü	Bedeutung
GET CHIP	INFO ON SCREEN INFO ON PRINTER GET CHIP	Auswahl und Beschreibung der verwendbaren TTL-ICs.
ERASE CHIP		Löschen von ICs.
CONNECT	CUT CONNECTION	Herstellen und Entfernen von „elektrischen“ Verbindungen.
SPECIAL	CLEAR PROJECT LOAD TESTBENCH PUT PROBE ERASE PROBE ENTER INFO-LINE	Einrichten eines neuen Projektes. Anschlußpunkte für das MINI-SCOPE setzen.
DISK	LOAD PROJECT SAVE PROJECT FORMAT DISK ERASE PROJECT OTHER DISK	Fertige Projekte laden und speichern. Diskettenoperationen durchführen.
TEST	MINI-SCOPE	Schaltung simulieren und testen.
PRINT	PRINT	Ausdruck der Schaltung auf einen Drucker.

Ein schneller Überblick: Was geht wo?

Typ	Schaltung	Typ	Schaltung
7400	4 NAND Gatter	7430	1 NAND Gatter 8 Eingänge
7401	4 NAND Gatter	7445	BCD Decoder
7402	4 NOR Gatter	7449	Displaydecoder
7404	6 Inverter	7473	2 JK Flipflops
7408	4 AND Gatter	7474	2 D Flipflops
7410	3 NAND Gatter 3 Eingänge	7476	2 JK Flipflops
7411	3 AND Gatter 3 Eingänge	7486	4 EXOR Gatter
7420	2 NAND Gatter 4 Eingänge	74174	6 D Flipflops

Mit 16 verschiedenen TTL-ICs läßt sich eine Menge konstruieren

blauem Grund die anliegenden logischen Zustände 0 oder 1, eine 0 auf rotem Grund kennzeichnet einen offenen Eingang.

Sind „Probes“ in einer Schaltung angeschlossen, schaltet **MINI-SCOPE** das „Oszilloskop“ ein, für jeden Anschluß erscheint am oberen Bildschirm ein eigener Kanal. Am besten probieren Sie das einfach einmal aus, schließen Sie je einen Mini-Scope-Anschluß an den schnellen und den langsamen Takt (auch Clock genannt) und schauen, was passiert. Nun sollte die Funktionsweise eines einfachen Signalschreibers klarer sein: auf der horizontalen Achse ist die Zeit aufgetragen, auf der vertikalen die beiden logischen Zustände 0 und 1. Als Meßgerät läßt sich das Mini-Scope nicht einsetzen, doch erlaubt es einen schnellen Überblick über den Signalverlauf an den Leitungen.

Die Bedienungselemente der Testbench lassen sich mit dem Mauszeiger bedienen. Die Ausgänge der Taster führen im Ruhezustand Low Level; stellt man die Pfeilspitze auf den jeweiligen Taster und drückt dann den Feuerknopf, wechselt der Ausgang auf High Level. Der Ausgang eines Schalters führt den Level, auf den der Schalter zeigt. Um die Schalterstellung zu ändern, wird der Schalter einfach angeklickt und mit gehaltenem Joystick-Knopf in die andere Stellung gezogen.

PRINT

Um Schaltungen auch im Ganzen betrachten zu können und nicht immer nur in kleinen Bildschirmportionen, erzeugt **PRINT** ei-

nen Ausdruck auf Papier. Die Druckroutine unterstützt den Commodore MPS 801 und dazu kompatible Matrixdrucker. Für eine ausreichend große Auflösung erfolgt der Druck bei großen Schaltungen in zwei Teilen, die sich leicht zur vollständigen Schaltung zusammenfügen lassen.

Wie soll man's sagen?

Auch wenn das Programm nur in einer englischen Fassung vorliegt, dürfte das eigentlich keine allzu großen Schwierigkeiten bereiten. Damit niemand nach einem Lexikon suchen muß, sind hier die wichtigsten Begriffe kurz übersetzt. Als erstes die Dialogboxen:

cancel ungültig; Abbruch von angefangenen Anweisungen
enter eingeben; Abschluß einer Eingabe
load laden; Dateien von Diskette in den Speicher holen
no nein
ok okay; zur Bestätigung
proceed ausführen; eingeleitete Prozesse ausführen
yes ja; zur Bestätigung

Der Vollständigkeit halber auch noch die wichtigen Meldungen:

Please insert the Digitest program-disk Bitte die Digitest-Programmdiskette ins Laufwerk legen

There is no probe used Es ist kein Mini-Scope-Anschluß angeschlossen

Unable to access printer, printing aborted Der Drucker spricht nicht an, Drucken abgebrochen

Cleaning up connections Die Verbindungsliste wird erstellt

Loading this project will destroy the current project Das Laden eines neuen Projektes zerstört das im Speicher befindliche

Are you shure you want to erase this project? Sind Sie sicher, daß Sie das Projekt löschen wollen?

All probes are used Alle Mini-Scope-Anschlüsse sind in Gebrauch

Switch on printer to print project Schalten Sie den Drucker ein

Grenzgänger

Anfänglich wird es schwerfallen, die Kapazitätsgrenzen des Programms zu erreichen. Damit Sie sich aber nicht wundern, wenn Digitest keine weiteren ICs mehr setzen mag oder der Verbindungsmodus „streikt“, sind hier die Grenzwerte beschrieben: Ein Projekt darf aus höchstens 16 Chips und 500 einzelnen Verbindungen bestehen. Damit keine unvorhersehbaren Effekte auftreten, sollten alle Inputs an einem definierten Potential liegen; offene Eingänge sind zu vermeiden!

Zum Autor

Der 20jährige Autor Gerrit Knoef ist vielen sicherlich bekannt, denn seit über einem Jahr veröffentlicht er in INPUT viele seiner Programme, Darunter Spiele wie „Happy Hacker“, „Bouncing Ball“ und kleine Anwendungen wie „Electrobrief“. Wir freuen uns besonders, daß wir Ihnen sein neuestes Werk, dieses Simulationsprogramm, vorstellen dürfen. Eigentlich hätten wir gerne diesem Programm den Wettbewerbspreis von DM 3000 zuerkannt, doch würde da nicht jeder an Schiebung denken?

Der Autor ist dennoch zufrieden, schließlich hat er mit Digitest den ersten Platz des holländischen Wettbewerbs „Micro-Master-Competition 1987“ in Utrecht belegt. Es versteht sich fast von selbst, daß wir ihn natürlich wie einen Wettbewerbsieger für seine Arbeit entlohnen.

Vom Zählen mit zwei Fingern

Teil 1: Gatter, Wahrheitstabellen und Verknüpfungen

Der Ausdruck TTL ist eine Abkürzung und bedeutet Transistor-Transistor-Logik. In TTL-Chips stecken also in der Hauptsache Transistoren, die dazu verwendet werden, logische Verknüpfungen zu realisieren. In diesem Artikel soll es uns aber nicht darum gehen, wie diese ICs technisch realisiert sind, sondern was man mit ihnen machen kann. Dazu ist ein wenig Theorie unumgänglich, aber wir werden auch Beispielschaltungen und praktische Anwendungen vorstellen.

Null ist nicht Eins

Bei dem Logik-System, das mit TTL-Schaltungen realisiert werden kann, handelt es sich um eine sogenannte binäre oder auch zweiwertige Logik. Das heißt, es existieren genau zwei verschiedene Zustände, die mit 'wahr' und 'falsch', 'ja' und 'nein' oder '0' und '1' bezeichnet werden können. Zwischenwerte wie 'vielleicht' oder 'jein' existieren nicht. In digitalen Schaltungen wird der 1-Zustand dadurch gekennzeichnet, daß eine Spannung anliegt. 0 heißt dann logischerweise 'keine Spannung'. Bei TTL-ICs wird eine 1 durch eine Spannung von +5 V dargestellt, die 0 entspricht 0 V beziehungsweise Masse. Zwischenwerte sind unzulässig.

Die einfachsten Elemente digitaler Schaltungen sind sogenannte Gatter. Dabei handelt es sich um Einheiten, die in den entsprechenden Chips enthalten sind und eine logische Funktion realisieren. Da Gatter grundlegend für digitale Schaltungen sind, hat man für sie eigene Symbole eingeführt.

Das einfachste Gatter ist der sogenannte Inverter oder Umkehrer. Sein Schaltbild und

Das Programm Digitest, das wir Ihnen in dieser Ausgabe vorstellen, dient dazu, Schaltungen mit TTL-Chips zu entwerfen und am Rechner zu simulieren. Es enthält eine Reihe der gängigsten ICs dieser Familie. Um jedoch digitale Schaltungen zu entwickeln, muß man wissen, was in diesen Chips steckt und wie man sie benutzt.

seine Wahrheitstabelle sind in Bild 1 dargestellt. Ein Inverter besitzt einen Eingang und einen Ausgang. Dabei führt der Ausgang immer den entgegengesetzten Pegel des Eingangs.

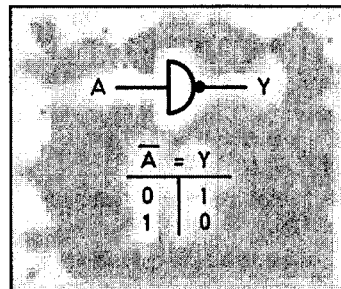


Bild 1: Ein Inverter negiert den Eingangszustand.

Zwei in Eins

Von großer Bedeutung in der Digitaltechnik sind Gatter, bei denen der Zustand des Ausgangs von zwei oder mehr Eingängen abhängt. Bild 2 zeigt ein UND-Gatter. Sein Ausgang führt genau dann den 1-Pegel, wenn Eingang A UND Eingang B auf 1 liegen. In der Familie der TTL-ICs enthält beispielsweise der 7408 mehrere AND-Gatter mit je zwei Eingängen. Sie können ja mal die in Bild 3 gezeigte Schaltung mit Digitest aufbauen und überprüfen, ob die angegebene Wahrheitstabelle für ein UND-Gatter stimmt.

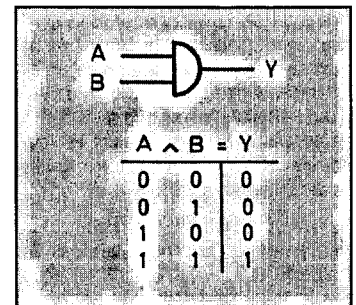


Bild 2: So beschreiben Digitalisten das Wörtchen UND.

Eine weitere wichtige Verknüpfung mehrerer digitaler Größen wird durch ein ODER-Gatter realisiert (Bild 4). Sein Ausgang geht auf 1, wenn an Eingang A ODER an Eingang B (oder an beiden) eine 1 anliegt.

Im praktischen Gebrauch in Schaltungen verwendet man sehr gerne AND- und ODER-Gatter, bei denen vor dem Ausgang

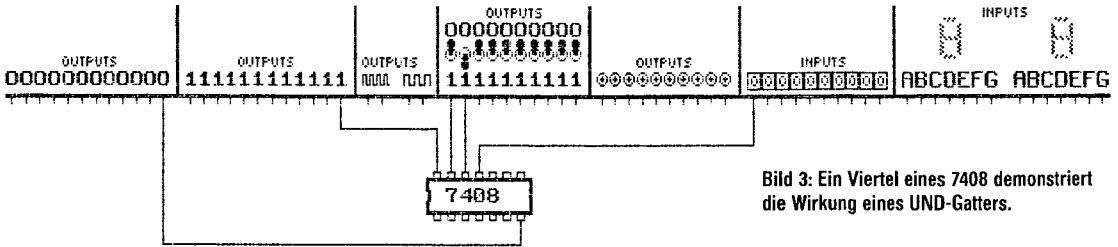


Bild 3: Ein Viertel eines 7408 demonstriert die Wirkung eines UND-Gatters.

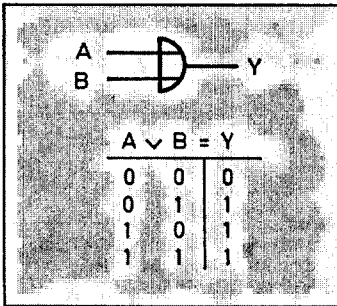


Bild 4: A ODER B — das ist keine Frage

eine Invertierung an dieser Stelle. Warum diese Gatter eine so große praktische Bedeutung haben, wird klar, wenn man sich ein wenig mit der Schaltalgebra beschäftigt.

Rechnen mit Zuständen

Hinter dem Begriff Schaltalgebra oder auch Boolesche Algebra verbirgt sich die mathematische Theorie, die in digitalen Schaltungen mit Gattern steckt. Man kann mit den beiden Zuständen, die eine Leitung in einer solchen Schaltung annehmen kann, fast wie mit Zahlen rechnen. Dazu benutzt man wie in der normalen Mathematik Buchstaben für Variablen, die Zahlen 0 und 1 für die entsprechenden Zustände sowie einige besondere Rechenzeichen.

noch ein Inverter sitzt. Solch ein invertierendes AND-Gatter heißt NAND, ein invertierendes ODER-Gatter NOR. Die Schaltsymbole und Wahrheitstabellen dieser beiden Gatter sind in Bild 5 dargestellt. Allgemein bedeutet bei logischen Schaltsymbolen ein Punkt an einem Ein- oder Ausgang

Ein invertierter Zustand wird durch einen Strich über dem entsprechenden Ausdruck gekennzeichnet, zum Beispiel \bar{A} (gelesen 'nicht A' oder 'A quer'). Es gilt: $\bar{0} = 1$ sowie $\bar{1} = 0$. Wird ein Wert zweimal invertiert, so entsteht wieder der Ausgangszustand (nicht nicht A ist gleich A).

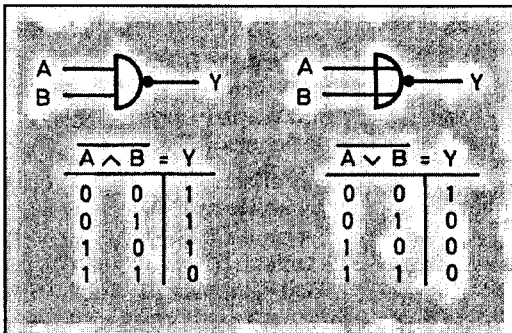


Bild 5: Ein Pünktchen am Ausgang dreht diese Gatter komplett um.

Die UND-Verknüpfung wird durch das Zeichen \wedge dargestellt, für ODER schreibt man ein \vee . Bei dem UND-Gatter aus Bild 2 gilt zum Beispiel: $Y = A \wedge B$

Gesetze der Logik

In der Tabelle auf dieser Seite finden Sie einige Verknüpfungen, die in Gleichungssystemen häufig vorkommen und die die Vereinfachung logischer Ausdrücke erleichtern.

$A \vee 0 = A$	$A \wedge 0 = 0$
$A \vee 1 = 1$	$A \wedge 1 = A$
$A \vee \bar{A} = 1$	$A \wedge \bar{A} = 0$

Einige einfache Rechenregeln der Digitaltechnik.

Booleschen Algebra sind die sogenannten de-Morganschen Gesetze. Sie lauten:

$$\overline{A \vee B} = \bar{A} \wedge \bar{B} \text{ und} \\ \overline{A \wedge B} = \bar{A} \vee \bar{B}$$

Mit diesen Rechenregeln erkennt man die Bedeutung von NAND- und NOR-Gattern: Alle denkbaren logischen Verknüpfungen zweier Variablen lassen sich nämlich so umformen, daß nur noch NANDs oder nur noch NORs erforderlich sind. Dadurch lassen sich die Kosten einer Schaltung reduzieren, denn viele gleiche Gatter sind billiger herzustellen als lauter verschiedene. Außerdem gibt es noch einen technischen Grund

für den Einsatz von Gattern mit invertierendem Ausgang. Intern enthalten die meisten Gatter am Ausgang noch einen Verstärker zur Erhöhung der Störsicherheit. Und dieser Verstärker bewirkt automatisch eine Invertierung.

Einer für alle

Daß man mit NAND- beziehungsweise NOR-Gattern tatsächlich alle anderen Verknüpfungen simulieren kann, soll folgendes Beispiel zeigen. Ausschließlich mit NAND-Gattern werden die Funktionen Invertierung, UND und ODER realisiert:

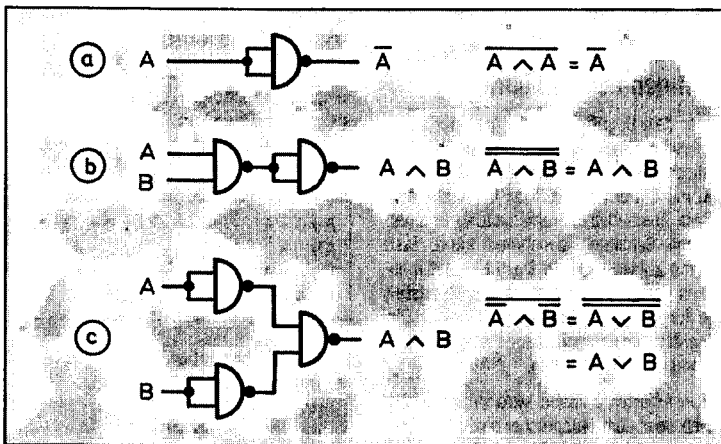


Bild 6: Es sind nur NANDs zu sehen, trotzdem sind alle anderen Verknüpfungen vorhanden.

Aus einem NAND-Gatter einen Inverter zu machen ist noch relativ einfach: man schaltet einfach die beiden Eingänge zusammen (siehe Bild 6a).

Bild 6b zeigt, wie aus zwei NAND-Gattern eine AND-Verknüpfung entsteht: Der (invertierende) Ausgang des ersten NANDs wird einfach noch einmal invertiert.

Etwas komplizierter ist schon die Erzeugung einer ODER-Funktion aus NAND-Gattern. Man benötigt dafür drei Stück, wovon

die ersten beiden als Inverter funktionieren. Die Schaltung und der Rechenweg sind in Bild 6c dargestellt.

Gatter als Speicher

Von den in Digitest enthaltenen ICs sollten Sie mit den bisherigen Erklärungen folgende Typen einsetzen können: 7400, 7401, 7402, 7404, 7408, 7410, 7411, 7420 und 7430. In der nächsten Ausgabe von INPUT 64 werden wir ausführlich auf die Funktionsweise der restlichen ICs eingehen. Sie enthalten Decoder und Flipflops. Decoder sind Netze von UND- und ODER-Gattern mit mehreren Ein- und Ausgängen und rea-

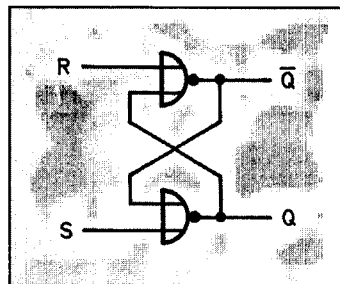


Bild 7: So miteinander verbunden, können sich die beiden Gatter etwas merken.

Literatur
Texas Instruments (Hrsg.): Das TTL-Kochbuch, Texas Instruments Deutschland GmbH, Freising 1980

INPUT 64 BASIC-Erweiterung

Die BASIC-Erweiterung aus INPUT 64 (Ausgabe 1/86), gebrannt auf zwei 2764er EPROMs für die C-64-EPROM-Bank.

Keine Ladezeit mehr — über 40 neue Befehle und Super-Tape integriert.

Preis: 49,— DM
zuzüglich 3,— DM für Porto und Verpackung (V-Scheck)

Bestelladresse:
Verlag Heinz Heise GmbH & Co KG
Postfach 61 04 07
3000 Hannover 61

lisieren sehr komplexe Funktionen, die den möglichen Eingangs- bestimmte Ausgangskombinationen zuordnen.

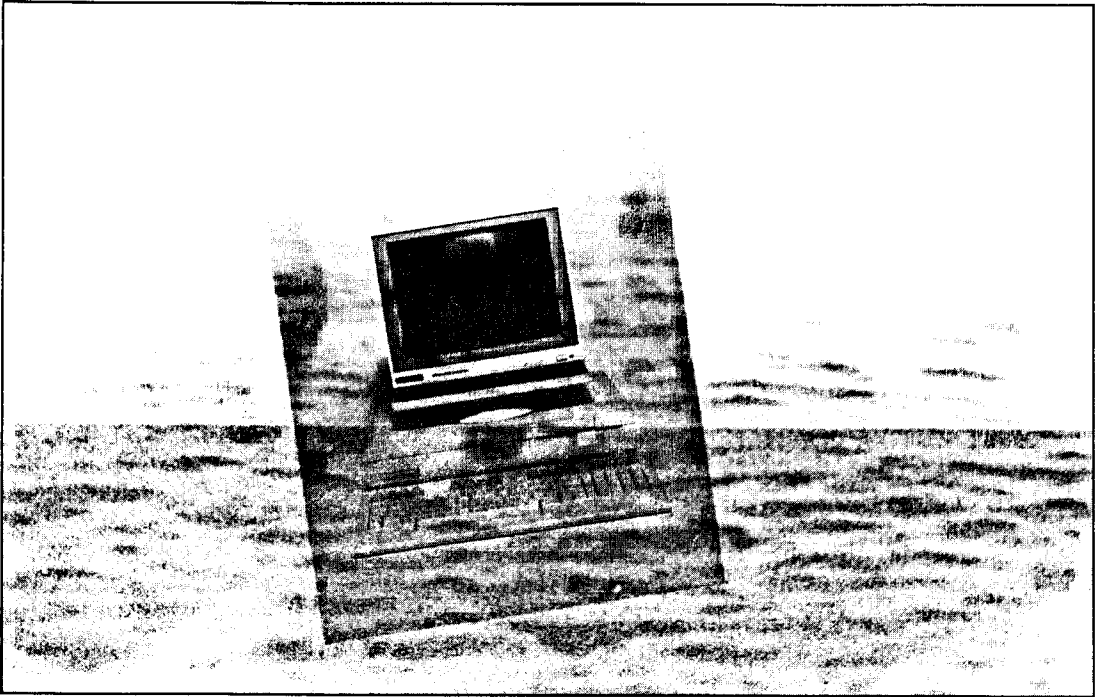
Flipflops sind digitale Speicher. Einen sehr einfachen Vertreter dieser Gattung zeigt Bild 7. Sie können ihn mit Digitest aus einem 7402 leicht selber 'bauen'. Schließen Sie an die Eingänge R und S Taster und an die Ausgänge Q und Q-bar Indicator-Lämpchen an, und lassen Sie sich überraschen. Die genaue Funktionsweise dieser Schaltung finden Sie in der nächsten Ausgabe von INPUT 64.
Hajo Schulz

Fast wieder Tango!

extra 6, HiFi-Boxen selbstgemacht. Mit fünfzehn gelungenen Selbstbau-Konzepten namhafter Entwickler. Mit wichtigen Grundlagen: mit News, News, News. . . Und natürlich in Farbe. Für 16 Mark 80 beim Verlag erhältlich.



Verlag Heinz Heise GmbH, Helstorfer Str. 7, 3000 Hannover 61



Mit Nadeln und Haken

128er-Tools: Programmier-Werkzeuge auf dem C128

In dem Modul 128er-Tools finden Sie diesmal drei Werkzeuge:

RENEW
REMOUT
Hardcopy

Sollten Sie INPUT 64-Ausgaben aus den Jahren 1985 und 1986 besitzen, kennen Sie diese Tools bereits für den C64. In INPUT 64, Ausgabe 12/87, erschien PLH (Programmer's Little Helper), ein komfortables Programmier-Werkzeug für den C64, das unter anderem diese Tools enthält.

In dieser Ausgabe können Sie sich innerhalb von INPUT 64 eine kurze Demonstra-

Obwohl das BASIC V7.0 des C128 über erheblich mehr Programmierhilfen verfügt als das etwas magere BASIC V2.0 des C64, fehlen einem dennoch bei der Erstellung von Programmen einige Möglichkeiten. INPUT 64 eröffnet hiermit eine Serie, die diese Lücken schließen wird.

tion der einzelnen Tools anschauen. Auf der letzten Bildschirmseite haben Sie dann die Möglichkeit, diese Tools mit CTRL-S auf eigenen Datenträger abzuspeichern.

Aus der Versenkung

Jedem Programmierer ist es wahrscheinlich schon einmal passiert, daß ein Programm mitten in der Entwicklung versehentlich durch einen Reset oder ein NEW von der BASIC-Oberfläche verschwand. In einer solchen Situation vermißt man dann das Gegenstück zum NEW-Befehl, denn der Rechner hat nur einige Werte im Speicher gelöscht, das Programm selbst verbirgt sich immer noch in den Tiefen der RAMs. Kenner vermögen natürlich mit der Sequenz

```
POKE 256*PEEK(46)+PEEK(45),1  
DELETE 1 (oder RENUMBER)
```

ihr BASIC-Programm wieder hervorzubekommen, doch der kleinste Tippfehler hierbei kann unangenehme Folgen haben.

Um ein BASIC-Programm wieder zum Leben zu erwecken, laden Sie das RENEW-Tool einfach mit

BLOAD "RENEW"

Es startet automatisch und verschwindet danach wieder aus dem Speicher des Rechners.

Nur in einem Stück

Programme, die aus einem Maschinensprache- und einem BASIC-Teil zusammengesetzt sind, können nicht mit diesem „Bergungs“-Programm gerettet werden. Sie müßten Programmumfang und -Ende je nach Programmstruktur mit Hilfe eines Monitors von Hand restaurieren, da sonst der Maschinenteil beim Abspeichern verlorengehe.

Ohne Bemerkungen

Wer übersichtliche Programmierung bevorzugt, wird sein Programm mit erläuternden Kommentaren in REM-Zeilen beschreiben. Für die endgültige Version wünscht man sich dann die Möglichkeit, diese Zeilen mit einem Befehl entfernen zu können, um ein kürzeres RUN-File auf der Diskette zu erzeugen, was einem längere Wartezeiten beim Laden erspart.

Hierzu dient das REMOUT-Tool, das Sie vom eigenen Datenträger laden können:

BLOAD "REMOUT"

Die Startadresse entspricht der Ladeadresse und liegt bei \$1500 (5376). Laden Sie anschließend ein BASIC-Programm, entfernt
SYS DEC("1500")

alle REM-Zeilen aus diesem Programm.

Erscheint Ihnen eine andere Adreßlage sinnvoll, können Sie REMOUT auch anderswo unterbringen:

BLOAD "REMOUT" P adr

Beseitigen können Sie dann alle REM-Zeilen mit

SYS adr

Vor der Anwendung sollten Sie jedoch folgende Hinweise beachten:

- Untersuchen Sie Ihr Programm vorher auf GOTO-, IF- oder GOSUB-Sprünge, die in REM-Zeilen führen! Verändern Sie diese Sprünge vor dem REMOUT entsprechend, da sonst Ihr Programm nicht mehr lauffähig ist. Grundsätzlich gehören REM-Zeilen nicht zum Programmablauf. Verzweigungen dorthin zeugen von schlechtem Programmierstil.
- Die letzte Zeile im Programm sollte eher END oder RETURN enthalten, als ausge-rechnet ein REM. REMOUT akzeptiert diese Unsitte nicht.

Selbsterkennend schwarz auf weiß

So mächtig das BASIC V7.0 des C128 ist – der Befehl „hardcopy“, um den Bildschirminhalt ohne Programmieraufwand zu Papier zu bringen, wird häufig schmerzlich vermißt. Die von uns veröffentlichte Routine besticht durch ihre Universalität (gelegentlich darf man sich ja auch mal selbst auf die Schulter klopfen . . .), hat deswegen aber auch die typischen Mängel aller „Allroundgenies“. Nämlich, in den Feinheiten zu versagen.

Im Einzelnen: Vorausgesetzt werden Commodore-Drucker der MPS-Serie und kompatible, bei Fremdfabrikaten hängt der Erfolg des Zusammenspiels vom Interface ab. Die Hardcopy-Routine läuft in jedem Speicherbereich, assembliert ist sie für die Startadresse \$1300. Nach dem Abspeichern auf eine eigene Diskette wäre die typische Vorgehensweise:

BLOAD"HARDCOPY"
SYS DEC("1300")

Der SYS-Befehl dient der Einbindung ins Betriebssystem, führt also nicht zum Ausdruck. Die Druckanforderung teilt der Benutzer durch die Tastenkombination CTRL-B mit

ist der Bereich ab \$1300 nicht verfügbar, kann durch

BLOAD "HARDCOPY", P adr
SYS adr

an eine beliebige Adresse im RAM geladen werden. „adr“ steht für eine Zahl zwischen (theoretisch) 0 und 65535, das Programm paßt sich selbst an die neue Lage im Speicher an. Ebenfalls selbsttätig wird ermittelt, ob der 40- oder der 80-Zeichen-Modus aktiviert ist. Das Programm berücksichtigt in beiden Modi reverse Zeichen und die Groß/Klein-beziehungsweise Groß/Grafik-Darstellung.

Was nicht geht

Einschränkungen gibt es beim Ausdruck des DIN-Zeichensatzes. Es ist zwar bei vielen Druckern möglich, durch Senden der entsprechenden Escape-Sequenzen zwischen verschiedenen nationalen Zeichensätzen zu wählen. Allerdings ist die Kodierung der Umlaute je nach Modell unterschiedlich, deswegen werden – um der Universalität willen – die deutschen Umlaute im DIN-Zeichensatz von dieser Hardcopy-Routine fast immer falsch ausgedruckt. „Das läßt sich doch vom Benutzer anpassen!“ könnte man dagegen einwenden. Schon, dann wäre aber die „Relokaltibilität“ (gleich: Lauffähigkeit an einer beliebigen Adresse) nur unter unvertretbar hohem Programmieraufwand zu gewährleisten.

Immer schön angepaßt

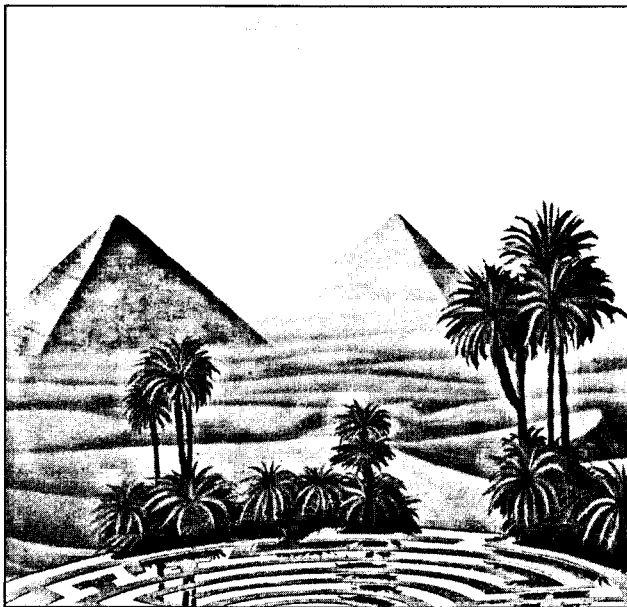
Dafür sind die Geräteadresse, die Sekundäradresse und sogar die Kanalnummer für den Ausdruck veränderbar. Voreingestellt ist alles entsprechend der BASIC-Sequenz 'OPEN 4,4:'. Wer anpassen will oder muß, kann dies durch POKE-Befehle und anschließendes Abspeichern tun, es gilt folgende Zuordnung:

KANAL Startadresse+4
GERÄT Startadresse+6
SEKA. Startadresse+8

Das Programm ist 325 Bytes lang. Benutzt werden die Zero-Page-Adressen von \$FA bis \$FF, „verbogen“ wird – für die CTRL-B-Abfrage – der Vektor auf die Tastaturabfrage (\$033a).
Astrath/rh/j5

Als Tierfang- Unternehmer unterwegs

Spiel: Safari



Um solch ein Unternehmensplanspiel, die Betonung liegt dabei auf "spiel", handelt es sich auch bei 'Safari'. Der Name läßt das Wirtschaftsobjekt erkennen, mit dem sich der von uns, dem Spieler, geleitete Betrieb befaßt: Exotik, Expeditionen und wilde Tiere. Konkreter gesprochen: Der Spieler soll zunächst seine Ausrüstung für Jagd und Reise und seine erforderliche Mannschaft geschickt zusammenstellen, im weiteren die Expeditionsschiffe buchen, Versicherungen abschließen oder es lassen, die Safari gut überstehen, mit reichem Fang gesund nach Hause kommen, um dann den Zoos seine Tiere zum Kauf anzubieten. Mit dem so gewonnenen Erlös wiederum kann man dann auf neue, jetzt möglichst größere Tour gehen, um immer reicher zu werden, wenigstens aber nicht pleite zu gehen.

Krokofant und Eledil . . .

Welche Entscheidungen, Kern jedes Betriebes, Kern auch jedes Unternehmensplanspieles, kann bei unserem 'Safari'-Betriebsmodell jeder der teilnehmenden Unternehmer-Spieler treffen? Zunächst wird das

Bekanntlich gibt es außer den drei Spielarten, die mit A beginnen (Arcade, Adventure, Animation), auch die reizvolle Gruppe der Unternehmensplanspiele. Diese Spiele ähneln den Adventures, aber mit dem Unterschied, daß letztere meist eine konkrete Situation zum Dreh- und Angelpunkt ihrer Spielidee haben, Unternehmensspiele dagegen nur (?) Kennziffern wie Umsatzsteigerung, Gewinnmaximierung und so weiter. Typisches Adventure: „Du befindest Dich im Schloß des Grafen Dracula, vor Dir eine Tapetentür, die in den Keller führt . . .“; typische Entscheidung: „nehme Schlüssel“, „laufe weg“, „hebe Kreuz“. Beim Unternehmensspiel dagegen hat man andere Kompromisse einzugehen, zum Beispiel: „Ich nehme einen Kredit von 10 000,— DM auf und stelle fünf neue Arbeiter ein“.

Spiel geladen, dann die Anzahl und Namen der miteinander konkurrierenden Unternehmer (gleich Mitspieler) eingetippt. Danach erscheint das Hauptmenü, dessen oberste Zeile vom Balken-Cursor hell unterlegt ist.

Drückt man die RETURN-Taste, erscheint Menü Nr. 2 und fragt ab, für welche Truppe die Ausrüstung sein soll. In Frage kommen dabei, wiederum mit Menü und Balken-Cursor zur Auswahl gestellt: Elefanten, Löwen, Eisbären, Tiger, Papageien und Wellensittiche. Auch hier wird mit der RETURN-Taste das Entsprechende ausgewählt. Rechts daneben erscheint jetzt eine Materialliste mit Stückzahl. Darunter steht in einer besonderen Zeile der Gesamtpreis der Ausrüstung. Da das Anfangskapital nur 70 000 DM beträgt, tut man gut daran, die erste Safari mit kleinen, leichten Tieren zu beginnen, ansonsten ist man schnell in Liquiditätsnöten, und einen Kredit gibt's nicht vor der zweiten Runde.

Mit dem Balken-Cursor wählt man, wie gesagt, die Tierart aus und stellt anschließend, bis "Ausrüstung komplett" erscheint, das benötigte Material zusammen.

... gibt's nicht nur am Nil

Mit der F1-Taste geht es zurück zum Grundmenü. Jetzt muß die Mannschaft zusammengestellt werden. Auch hier wählt man wieder die Tierart aus, für die man schon die Ausrüstung gekauft hat. Variationen sind im Verlauf des Spiels möglich.

Folgende Qualifikationsgrade oder Tarifgruppen stehen zur Qual der Wahl: Spezialist, Fachmann, Angelernter und Ungelernter. Die Zahlen daneben bedeuten: "DM pro Expedition und Mann an Heuer". Die unternehmerische Entscheidungskunst, wie im realen Betriebsleben auch, besteht nun darin, die rechte Kombination zwischen teureren, aber guten hochqualifizierten und den preiswerteren, aber leistungsschwächeren Vorarbeitern und Helfern zu finden.

Viel Spaß beim Herausfinden des Optimums. Ein Betrieb ist so gut wie seine Mitarbeiter. Beim Herausfinden hilft der im Menü erscheinende Indikator "Erfolg: xx %", "Mannschaft: yy %".

Hat man auch diese Unternehmerkunst gemeistert, geht's mit der F1-Taste übers bekannte Grundmenü (Nr. 1) zum Entscheidungsbild Nr. 4: "Schiffe buchen".

Reden ist Silber . . .

Beim Menü "Schiffe buchen" gibt's dank Monopolmacht der Reeder nicht viel für den Safariausrüster zu entscheiden. Die Schiffspreise stehen marktdiktatorisch fest, und da der Spieler Ausrüstung samt Mannschaft bereits für teures Geld erworben hat, verbleibt ihm jetzt kein Spielraum zum Verhandeln. Er muß die Schiffspassage zum vorgegebenen Preis abschließen. Hoffentlich sind hierfür in der Kasse noch Reserven vorhanden. Ist das geglückt, kommt die unscheinbar wirkende Frage "Ausrüstung versichern? - J/N".

Ein Preis für diese Versicherung erscheint hier nicht, und das Konto wird erst später belastet. Gar nicht so abwegig, das Verfahren, wenn man weiß, daß es in der Praxis oft auch so läuft: Versicherungen werden oft abgeschlossen, ohne daß groß Preisvergleiche mit dem Wettbewerb eingeholt werden. Oft wird nicht einmal nach dem Preis der Ware 'Sicherheit' gefragt.

Die Entscheidung für Versicherung "J/N" fällt hier nicht leicht; denn das "J" kann wegen möglicher hoher Versicherungsprämien unnötig teuer werden. Noch teurer freilich, bis zum Ruin, kann der Druck auf die "N"-Taste zu stehen kommen, weil nämlich, ähnlich den "Ereigniskarten" im MONOPOLY, auf der Safari Leute verletzt, Gegenstände beschädigt werden oder ganz verlorengehen. Hier würde sich natürlich eine vorher abgeschlossene Versicherung auszahlen.

... Reeder sein bringt Gold

Hat man schließlich entscheidungsschwer alles beisammen, was bei max. 8 Konkurrenten (Mitspieler) eine Weile dauern kann, geht es los: einmal die F7-Taste gedrückt, und schon befindet man sich auf Safari. Ergebnismeldungen der einzelnen Tierfanggruppen erscheinen: Katastrophen wie "Eisbärgruppe - Schiff gesunken, Ausrüstung weg, alle Mann ertrunken" - über schöne Erfolge wie "Papageiengruppe - guter Erfolg, 20 Papageien als Bonus extra erhalten", bis hin zu märchenhaftem Glück: "Wellensittichgruppe - Diamanten gefunden, Dein Anteil: 20 000,- DM".

Dann schließlich erscheint wieder das Grundmenü (Nr. 1), jetzt mit den Safari-Ergebnissen.

Profit mit Tieren

Das Grundmenü zeigt jetzt den neuen Bargeldbestand, die zu zahlende Versicherungsprämie, die erhaltenen Versicherungsleistungen und vor allem den Bestand an mitgebrachten Tieren an. Außerdem wird angezeigt, daß man pro Runde 5000,- DM erhält (einfach so). Das kann man auch gebrauchen, vor allem wenn man kein Bargeld mehr hat oder keinen Kredit erhält. Jetzt kommt die unternehmerisch schöne Spielphase des Verkaufs der Tiere an die konkurrierenden Zoos. So hat man die Ehre, im Grundmenü mit dem Cursor auf "Verkauf" zu schalten, Entscheidungsbild Nr. 5 erscheint und zählt die Namen deutscher Zoos auf. Stellt man den Cursor auf einen dieser Zoos und drückt RETURN, erhält man die variablen Preise, zu denen dieser

betreffende Zoo Tiere kauft, oder aber man erhält eine Meldung, wonach dieser Zoo derzeit keinen Bedarf an neuen Tieren habe.

Die einzelnen Zoos zahlen recht unterschiedliche Preise, so daß man zum übersichtlichen Preisvergleich alle Zoos nacheinander durchgehen und sich alle Daten per Block und Blei oder mit Hilfe eines zweiten Computers säuberlich notieren sollte, um anschließend seine Tierbestände jeweils den Zoos anzubieten, welche für die entsprechende Tierart auch am meisten zahlen.

Hat man alle seine Tiere verkauft, kann man sich bei gutem Erlös (Zoos zahlen prompt) neue Expeditionen zusammenstellen, wie gehabt, jetzt aber (vielleicht) dank gewonnener Liquidität aus Barvermögen plus neuem Kreditrahmen sich sogar mehrere Gruppen gleichzeitig leisten und sie versichert auf neue Safari schicken. So winkt die Gewinnmaximierung, und es bleibt das Restrisiko totalen Verlustes.

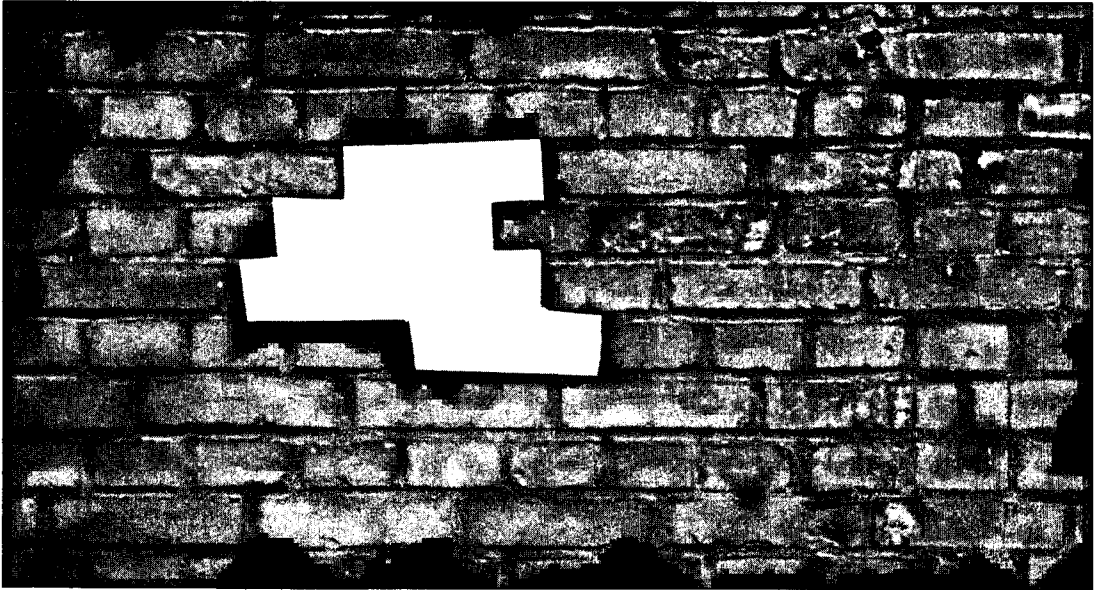
Kreditwürdig oder unwürdig

Besonders spannend ist im Grundmenü der Punkt 5 (Kredit). Hier gilt es, ähnliche Spielaktiken zu entwickeln, um so der Bank die benötigten Kredite zu entleihen. Wie unternehmensüblich, ist oft zu wenig Geld in der Kasse, um größere Expeditionen zusammenzustellen. Da kann geschickt getüftelte Kreditpolitik den Finanzengpässen zunächst, besser dann dauerhaft, abhelfen. Vorteile gegenüber den Mitspielern beziehungsweise gegenüber dem Vortagsergebnis sind hier gut möglich.

Gewonnen hat schließlich, wer nach vorher festgesetzter Rundenzahl das meiste Vermögen (Geld oder Schulden) besitzt. Verloren aber hat derjenige, der am Ende am wenigsten Geld auf seinem Konto hat.

Es kommt auch vor, daß andere Spieler gar fünf, maximal sechs Gruppen gleichzeitig auf Safari schicken, riesige Umsätze mit Mengen an verkauften Tieren machen, darauf von der Bank hohe Kredite erhalten und dabei immer mehr in die roten Zahlen rutschen, wie im Geschäftsleben oft auch: Gewinnsteigerung vor Umsatzmaximierung.

W. Schmidt-Pabst/kfp



Einen Weg gibt's immer

Spiel: Labyrinth

Ob Sie das Spiel allein oder zu zweit spielen, die Aufgabe bleibt die gleiche: Sie steuern mit dem Joystick oder mit der Tastatur eine oder zwei symbolische Mäuse vom Startpunkt zum Ziel.

Unterwegs könnten Sie mit einer der beiden Katzen zusammentreffen. Sollte Ihnen dieses Mißgeschick widerfahren, muß Ihre Maus wieder in die Startlöcher zurück.

Sichtbar . . .

Der besondere Reiz dieses Spiels liegt in der Beeinflussung der Labyrinth-Darstellung. Im Einstellmenü (wird vom Labyrinth aus mit der Taste 'F3' erreicht) können Sie festlegen, ob das Labyrinth von Haus aus sichtbar oder unsichtbar ist.

Wohl jeder von Ihnen wird in seiner Programmsammlung ein Spiel dieses Genres haben. Nur dürfte sich bei jenen wahrscheinlich die Anzeige nicht selbständig auflösen oder aber erst nach und nach aufbauen. Also wieder nur ein Labyrinthspiel? Ja, aber eins mit Pfiff.

Da die Katzen ihren gelaufenen Weg auch wahlweise farblich kenntlich machen können, gibt es interessante Kombinationsmöglichkeiten.

. . . oder unsichtbar

Versuchen Sie doch einmal: Modus „unsichtbar“ und Weg der Katzen „weiß“ oder

Modus „sichtbar“ und Weg der Katzen „grün“.

In dem Hauptmenü können Sie auch die Geschwindigkeit, die Anzahl der Spieler festlegen und die Neuberechnung eines Labyrinthes starten. Unter der Überschrift „Nur nicht einmauern!“ finden Sie in dieser Ausgabe eine Beschreibung der Algorithmen für die Labyrinthberechnung.

Gleich nach dem Titelbild müssen Sie festlegen, ob Sie mit der Tastatur oder mit Joystick(s) spielen wollen. Wenn Sie zu zweit spielen, gilt diese Festlegung für beide Parteien.

Die Tastatur-Belegung (für den ersten und zweiten Spieler) ist: 'Q' und '*' für hoch, 'A' und ';' für runter, 'Z' und '.' für links sowie 'X' und '/' für rechts. C. Wilhelm/WM

Nur nicht einmauern!

Ein Labyrinth-Algorithmus

Haben Sie sich auch schon mal den Kopf zerbrochen, wie Sie ein Labyrinth auf eine Fläche zaubern können — ein Labyrinth, das keine überflüssigen Freiräume und garantiert einen Ein- und einen Ausgang hat?

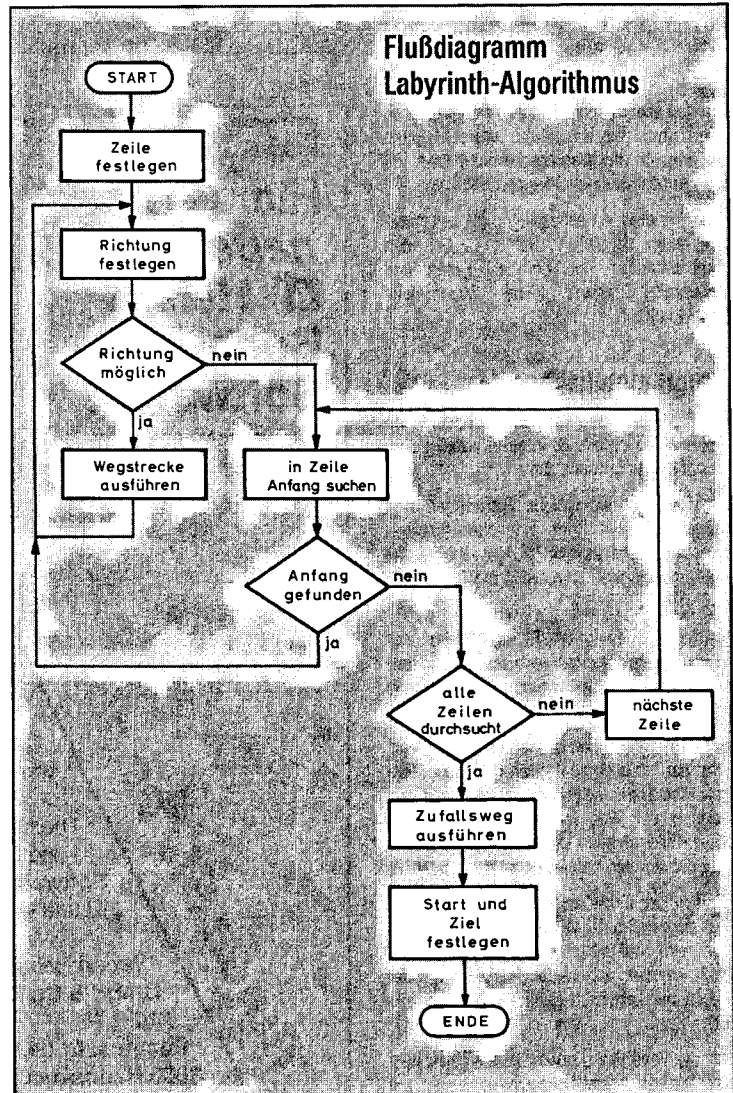
Im folgenden wollen wir den Algorithmus beschreiben, der bei dem Labyrinth-Spiel in dieser Ausgabe angewendet worden ist.

Es sind zwei grundsätzlich unterschiedliche Vorgehensweisen denkbar. Zum einen kann der mögliche (Ir)weg Stück für Stück und zum anderen können die hinderlichen Mauern berechnet werden. Der vorliegende Algorithmus verfolgt die erste Strategie.

Wenn von Zeilen und Spalten gesprochen wird, sind jeweils die durchnummerierten Bildschirmzeilen und -spalten gemeint. Die Numerierung beginnt jeweils bei der ungeraden Zahl Eins.

Axiomatisches

In welche Richtung sich die Wegstrecke weiterentwickeln kann, hängt in erster Linie von der Position im Zeilen- und Spaltensystem ab. Die theoretischen Möglichkeiten können Sie einer Tabelle in diesem Artikel entnehmen.



Erlaubte Richtungen für die Wegstrecke

Spalten \ Zeilen	ungerade	gerade
ungerade	↕ ↔	↔
gerade	↕	%

weils ein Zeichen lang) durch Zufall bestimmt und auf das Labyrinth gesetzt. Hierdurch können geschlossene Kreise entstehen.

Abschließend wird in den Spalten 0 und 38 an zufälligen ungeraden Zeilen je ein Zeichen als Eingang beziehungsweise als Ausgang gesetzt. Das Labyrinth ist damit fertig.

C. Wilhelm/WM

Eine dieser theoretischen Möglichkeiten kann aber nur ausgeführt werden, wenn die angestrebte Wegstrecke nicht an einen bereits bestehenden Weg anstößt.

Aus den obigen Festlegungen ist unschwer abzuleiten, daß sich auf den Schnittpunkten von geraden Zeilen und geraden Spalten keine Wegstrecke befinden kann.

Programmatisches

Die Erzeugung des Labyrinthes beginnt am linken Rand (Spalte 0) in einer zufällig ausgewählten ungeraden Zeile.

Es wird geprüft, welche Richtungen nun möglich sind. Aus den möglichen Richtungen wird eine zufällig ausgewählt. Sollte nur eine möglich sein, wird diese verwendet.

In der ausgewählten Richtung wird der Weg um ein Zeichen fortgesetzt, und anschließend werden wieder die möglichen Richtungen geprüft.

Sollte keine Richtung mehr möglich sein, wird ein neuer Anschluß an dem bestehenden Weg in der Zeile gesucht und, falls vorhanden, der Weg an dieser Stelle fortgeführt. Dies ist auch die einzige Möglichkeit, Weggabelungen entstehen zu lassen.

Sind in der aktuellen Zeile die Möglichkeiten ausgeschöpft, wird die folgende Zeile überprüft. Auf diese Weise kann sichergestellt werden, daß nach Überprüfung aller Zeilen die Labyrinth-Fläche optimal gefüllt ist.

Um die Verwirrung etwas zu steigern, werden nun noch drei kurze Wegstrecken (je-

Hier ist das *neue* Heise- Buch- und Software- Programm 1987/88



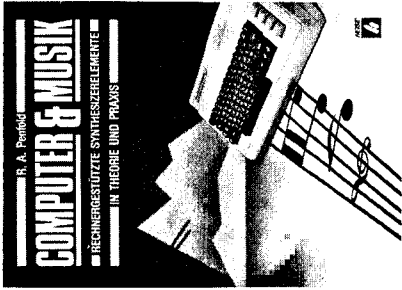
Rufen Sie es ab!

Elektronik — Schaltungen, Boxenselbstbau, computergestützte Klangsynthese.
Computertechnik — Anwendung, Programmierung und Weiterbildung.
 Themenschwerpunkte:
 KI und OCCAM.
Software — die Speed-, LIB und -FAKTURA.

**Verlag Heinz Heise
GmbH & Co KG
Abt. 178
Postfach 61 04 07
3000 Hannover 61**

Erlaubte ein Exemplar. Meine Anschrift:

Computer für Mucker



Der Homecomputer als Hilfsmittel zur elektronischen Klangsynthese
 — Stichworte Sequenzer, MIDI —
 Schnittstellen, Soundgeneratoren, Digitalumsetzer, Kompander, Mehrkanal-Generatoren.
 Sämtliche Themen werden leicht nachvollziehbar behandelt. Vorausgesetzt wird etwas Erfahrung in der Programmierung von Computern und im Aufbau einfacher Schaltungen.



Verlag
 H. Heise GmbH
 Postfach 61 04 07
 3000 Hannover 61

Broschur, 108 Seiten
DM 18,80
ISBN 3-922705-37-5

COMPUTER & ELEKTRONIK

Im Buch-, Fachhandel oder beim Verlag erhältlich. 537/2.4

Information schafft Wissens- vorsprung.



Datenbanksysteme speichern und verarbeiten Informationen. Das Buch vermittelt Einblick in die Thematik und unterstützt Entscheidungen in allen Wirtschaftsbereichen. Am Beispiel dBASE III, dem Marktführer bei PC-Datenbanksystemen, werden die dargestellten theoretischen Aspekte verdeutlicht. Ein eigener Teil ist dem hochaktuellen Thema „Online-Datenbanken“ gewidmet.



Verlag
 H. Heise GmbH
 Postfach 61 04 07
 3000 Hannover 61

Broschur, 173 Seiten
DM 36,80
ISBN 3-88 229-133-8

COMPUTER- BUCH

Im Buch-, Fachhandel oder beim Verlag erhältlich. 139/1.4

Zählmeister

Tips zur Schleifenkonstruktion in BASIC

Immer wiederkehrende gleichartige Aufgaben empfindet der Mensch als monoton und langweilig. Wer in der Schule hundertmal „Ich darf nicht dazwischenreden!“ als Strafarbeit anfertigen mußte, kann ein Lied davon singen. Der C64 ist dagegen recht leicht zu solcher Arbeit zu überreden und schreibt den geforderten Satz auch beim hundertsten Mal noch mit gleicher Geduld.

Zählwerk

Natürlich wird der C64 in der Regel nicht mit unsinnigen Strafarbeiten beschäftigt. Für ihn gibt es andere monotone Aufgaben, wie etwa ein Sprite von links bis zur Mitte des Bildschirms zu bewegen, wobei die einzelnen x-Koordinaten nacheinander in das entsprechende Video-Chip-Register geschrieben werden müssen. Wer geduldig ist und viel Speicherplatz frei hat, kann ja 160 POKE-Befehle eintippen. Alle anderen tippen nur einen POKE-Befehl und lassen ihn 160mal durchlaufen.

Alle BASIC-Versionen, die auf dem C64 existieren, und auch alle blockorientierten Sprachen wie Pascal oder C, bieten dazu eine Sprachkonstruktion an. Sie wird mit den Schlüsselwörtern FOR, TO, NEXT und eventuell STEP gebildet und wird kurz FOR-NEXT-Schleife genannt. Im Magazin sehen Sie dazu ausführliche Beispiele. Obwohl der Befehl oft benutzt wird, unterlaufen auch erfahrenen Programmierern häufig Fehler bei der Anwendung, weil aus der Fülle der Möglichkeiten nicht alle erlaubt sind. Ein kurzer Ausflug in die internen Vorgänge des C64 bei der Bearbeitung einer Schleife schafft Klarheit.

Eine typische Schleife hat folgenden Aufbau:

```
10 FOR Z = A TO E
20 REM Anweisungen
30 NEXT Z
```

Schleifen verbinden die meisten Menschen mit „nett, adrett und nützlich“. Schleifen sind oft bunt und manchmal zum Ärger des Schuhträgers nicht vorhanden. Programmierer sehen das anders: Programmschleifen leisten oft die grobe Schwerarbeit bei Routineaufgaben. Die von BASIC angebotene FOR-NEXT-Schleife verdient daher eine eingehende Betrachtung.

Der Zähler (Z) wird am Beginn der Schleife auf den Anfangswert (A) gesetzt. Dann werden die Anweisungen ausgeführt. Taucht der Befehl NEXT auf, wird der Zähler, soweit nichts anderes angegeben ist, immer um eins erhöht und alle Anweisungen noch mal ausgeführt. Erst wenn der Zähler gleich dem Endwert (E) ist, wird die Schleife verlassen. Wer eine andere Schrittweite (S) als 1 benötigt, muß sie mit dem Befehl STEP hinter dem Endwert angeben.

```
10 FOR Z = A TO E STEP S
20 REM Anweisungen
30 NEXT Z
```

Nun erhöht der C64 bei Erreichen des NEXT-Befehls den Zähler nicht um eins, sondern um die angegebene Schrittweite. Ist sie größer als eins, ist der Endwert mit weniger Durchläufen erreicht, ist sie kleiner, werden die Anweisungen öfter ausgeführt. Was passiert aber zum Beispiel, wenn der Anfangswert schon größer ist als der Endwert? Wird die Schleife dann überhaupt durchlaufen? Um diesen und anderen Fragen auf den Grund zu gehen, ist ein kleiner Ausflug in den BASIC-Interpreter nötig.

Hochstapelei

Der C64 verwaltet interne Informationen mit einem 255 Byte großen Stapelspeicher, der Maschinenprogrammierern als Stack bekannt ist (siehe Kasten). Für eine Schleife werden hier 18 Bytes an Informationen abgelegt, die beim Verlassen wieder entfernt werden. Das gleiche gilt für Unterprogrammaufrufe und einige andere BASIC-Befehle. Wenn der Stapelspeicher voll ist, verweigert der Computer jede weitere Mitarbeit mit der Meldung OUT OF MEMORY ERROR, auch wenn im Hauptspeicher noch genug Platz frei ist.

Der BASIC-Interpreter legt auf diesem Stack den Namen der Zähler-Variablen, den Endwert und die Schrittweite ab. Der Anfangswert wird dem Zähler zugewiesen und hier nicht mehr beachtet. Als Zähler kommt jede normale Real-Variable (zum Beispiel Z) in Frage. Nicht erlaubt sind Integer- und String-Variablen (Z% oder Z\$). Wollen Sie das gerade bearbeitete BASIC-Programm kompilieren und benutzen dafür den in der Ausgabe 10/87 veröffentlichten BASIC-Compiler, sind wiederum Integer-Variablen (Z%) von Vorteil, da solche Schleifen dann sehr schnell abgearbeitet werden. Der Anfangswert, der Endwert und die Schrittweite können übrigens beliebige Ausdrücke sein. Sie werden aber nur einmal, zu Beginn der Schleife, ausgewertet und auf den Stack gelegt. Die Schrittweite kann innerhalb der Schleife genausowenig verändert werden wie der Endwert, auch wenn beide aus einer Variablen bestehen. Trifft der C64 nun auf den NEXT-Befehl, sucht er den Datenblock auf dem Stapel. Findet er keinen, weil keine Schleife eröffnet wurde, meldet er ein NEXT WITHOUT FOR ERROR und bricht das Programm ab.

Ist der gesuchte Datenblock vorhanden, wird zuerst die Schrittweite zum Zähler addiert. Haben Sie eine negative Schrittweite angegeben (-3), wird der Zähler dabei natürlich kleiner, denn $24 + (-3) = 21$, es wird dann also rückwärts gezählt. Im nächsten Schritt wird der Zähler mit dem gespeicherten Endwert verglichen. Ist er erreicht oder überschritten, ist die Schleife beendet. Sonst werden die Anweisungen in der Schleife noch mal abgearbeitet. Beim Verlassen der Schleife werden die Informatio-

Aufstapeln

Wenn man in BASIC programmiert, kommt man selten in die Verlegenheit, sich mit den Innereien eines Rechners auseinandersetzen zu müssen. Programmiert man aber in Maschinensprache, ob mit einem assemblerfähigen Monitor (siehe INPUT64, Ausgabe 11/87) oder mit einem Assembler (siehe INPUT64, Ausgabe 6/86), wird man nicht umhinkommen, sich mit der Funktionsweise eines Rechners zu beschäftigen. Ein dritter Aspekt ist das hier veröffentlichte ist BASIC. Es stellt nämlich zwei Stack-Manipulationsbefehle zur Verfügung (PULLSUB, PULLLOOP).

Wenn man mit diesen Befehlen effektiv programmieren will, muß man zumindest wissen, was der sogenannte Stack, zu deutsch Stapel, ist. Das soll hier in diesem kleinen Beitrag erläutert werden.

Stapel werden im täglichen Leben oft benutzt. Es werden Bücher zu Stapeln aufeinandergelegt, Zeitungen werden gestapelt oder, was jeder bestimmt schon mal gesehen hat, Groschen in einem Groschenspeicher gestapelt. Damit sind die Groschen für die Parkuhr gemeint. Heute muß man ja schon 50Pfennig-Stücke dafür bereithalten. Na gut, wir wollen nicht abschweifen.



Bild 1: Ein solcher Zettelpieker verkörpert sehr gut einen Stack. Einzelne Zettel kann man nicht zwischenraus-oder unterwegnehmen.

Dreigroschenstapel

Bleiben wir noch etwas bei diesem Groschenspeicher. Man steckt also einen Groschen in diesen Speicher. Steckt man einen zweiten Groschen dazu, wird der erste nach unten gedrückt und der zweite liegt oben drauf. Beim Einschieben eines weiteren werden alle schon enthaltenen Groschen immer weiter nach unten geschoben. Entnimmt man jetzt aber einen Groschen, bekommt man den in die Hand, den man als letzten eingeschoben hat. Solch ein System nennt man dann auch LIFO-Speicher (Last in, first out). Das heißt nichts anderes als: als letztes rein, als erstes raus.

Ähnlich ist es auch beim Stack (Stapel) in unserem Rechner. Nur werden hier nicht die einzelnen Speicherinhalte immer um eins verschoben, sondern der sogenannte Stackpointer (Stapelzeiger). Der gesamte Stack nimmt eine ganze Speicherseite im Rechner ein, von \$01FF (511) bis \$0100 (256). Wenn der Stapelzeiger beim Einschalten neu initialisiert wird, wird der Stapelzeiger vom Betriebssystem auf die Adresse \$01FF gesetzt. Anschließend werden noch einige Routinen durchgeführt, die den Stack benutzen. Dabei wird der Stackpointer natürlich verschoben. Wenn wir also mit dem Stack arbeiten wollen, kann es sein, daß der Zeiger auf eine andere Adresse gesetzt ist, zum Beispiel auf \$01F3.

Steht der Zeiger durch den Gebrauch des Stacks auf \$0100, ist der Stack voll. Nun gut, das sollte bei unserer Programmierung auch nicht passieren. Falls dieses dennoch vorkommt, wird es vom BASIC abgefangen. Auf dem Bildschirm erscheint die Fehlermeldung: OUT OF MEMORY ERROR.

Auf den Stack damit

Wie anfangs schon erwähnt, stellt ist BASIC zwei Befehle zur Verfügung, mit denen man den Stack beziehungsweise den Stackpointer beeinflussen kann. Dazu muß man wissen, daß auch bei der BASIC-Programmierung der Stack benutzt wird, zum Beispiel bei jeden GOSUB. Dabei verzweigt das Programm zu einer Unterroutine, die dann mit einem RETURN abgeschlossen wird. Wie weiß das Programm aber, zu wel-

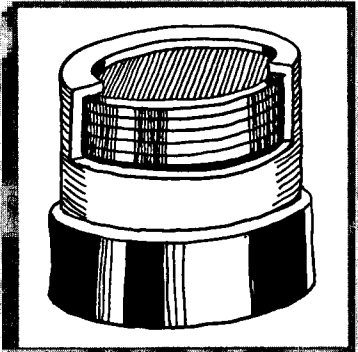


Bild 2: Der Letzte wird der Erste sein. Der zuletzt eingeschobene Groschen kommt als erster wieder raus.

chem Programmpunkt es zurückkehren muß?

Das ist im Prinzip ganz einfach. Bei einem Programmstart wird der Program-Counter (Programmzähler) auf die Adresse gesetzt, an der das Programm beginnt. Dieser Programmzähler zählt nun jeden Programmschritt mit, mal vorwärts, mal rückwärts. Bei Sprüngen wie GOTO oder GOSUB wird er auf die anzuspringende Adresse gesetzt und zählt dann an dieser weiter.

Bei einem GOSUB aber wird außerdem die Adresse des Befehls, der dem GOSUB-Befehl folgt, auf dem Stack zwischengespeichert. Stößt das Programm jetzt beim Abarbeiten des Unterprogrammes auf ein RETURN, wird die zwischengespeicherte Adresse wieder vom Stack geholt und in den Programmzähler geschrieben. Das Programm arbeitet dann genau den Befehl ab, der an dieser Adresse steht.

Wie man sieht, wird bei einem RETURN der Stack immer wieder bereinigt. Verläßt man das Unterprogramm aber mit einem GOTO, wird die Adresse nicht vom Stack entfernt. Wird dann solch ein Unterprogramm mehrmals aufgerufen, kann dadurch der Stack überlaufen.

nen vom Stapel entfernt, womit der Platz wieder frei ist.

Jetzt läßt sich auch die oben gestellte Frage beantworten: Wenn der Endwert kleiner als der Anfangswert ist und eine positive Schrittweite gewählt wurde, werden die Anweisungen in der Schleife trotzdem einmal ausgeführt, da die Überprüfung erst am Ende durch den NEXT-Befehl erfolgt.

Verbotene Schachteln

Aus dem oben Beschriebenen läßt sich ableiten, was bei Schleifen erlaubt und was verboten ist. Eine Schleife ist ein geschlossener Block, der mit einer FOR-Anweisung begonnen und einer NEXT-Anweisung beendet wird. Das bedeutet, daß nicht mit einem GOTO-Befehl hinein- oder hinausgesprungen werden darf. Während ersteres zu einem NEXT WITHOUT FOR ERROR führt, ist letzteres ein Fehler, der sehr schwer zu finden ist. Das Programm arbeitet scheinbar richtig weiter, auf dem Stapel bleiben aber die 18 Bytes, die zuvor als Informationen eingetragen wurden, liegen. Passiert das mehrmals, läuft der Stapel über und es kommt zum OUT OF MEMORY ERROR. Der ahnungslose Programmierer versucht daraufhin verzweifelt, sein Programm zu kürzen, hat aber keinen Erfolg. Der Sprung mit GOTO aus der Schleife heraus wird oft verwendet, wenn sie vorzeitig abgebrochen werden soll. Das läßt sich jedoch ohne Probleme auch durch Manipulation des Zählers erreichen. Setzt man den Zähler auf den Endwert und springt dann zum NEXT-Befehl, wird die Schleife ordnungsgemäß verlassen, und die Gefahr ist gebannt. Innerhalb einer Schleife darf natürlich beliebig gesprungen werden. Ebenso können Unterprogramme mit GOSUB aufgerufen werden, denn mit dem RETURN-Befehl kehren sie immer in die Schleife zurück.

Es lassen sich auch mehrere Schleifen verschachteln, das bedeutet, daß in einer Schleife eine weitere geöffnet werden darf, in dieser eine weitere, noch eine und noch eine, bis der Stapel voll ist. Informationen von maximal zehn Schleifen können aufgenommen werden. Es kann also durchaus passieren, daß Ihr Programm nur zwanzig Zeilen umfaßt, aber zehn ineinandergeschachtelte Schleifen enthält. Sie programmieren noch etwas, und

schon erscheint nach dem Starten des Programmes ein OUT OF MEMORY ERROR. Dabei ist nicht etwa der Programm-Speicher voll, wie irrtümlich angenommen, sondern nur der Stack. Schleifen, die nicht ineinander verschachtelt sind, können Sie natürlich unbegrenzt verwenden. Werden Schleifen verschachtelt, muß für jede eine andere Zählvariable verwendet werden, die jede für sich 18 Bytes des Stacks belegt.

Der Begriff „schachteln“ legt schon nahe, worauf dabei zu achten ist. So, wie Sie nur eine kleinere Schachtel in eine größere legen können, so müssen auch die Schleifen ineinander liegen. Im Programmablauf muß also zuerst das NEXT der inneren und danach das NEXT der äußeren Schleife kommen. Halten Sie sich nicht daran, ahndet der C64 das mit einem NEXT WITHOUT FOR ERROR, obwohl Sie doch den FOR-Befehl ordnungsgemäß im Programm stehen haben.

Weil eine Schleife ein geschlossener Block ist, erlaubt der C64, den NEXT-Befehl ohne Angabe der Zählvariablen zu gebrauchen. Ein NEXT gehört immer zum letzten FOR-

Befehl. Schreibfaule Programmierer machen davon regen Gebrauch, zumal die Bearbeitung dann etwas schneller erfolgt. Wer aber ein gut dokumentiertes, verständliches Programm schreiben möchte, gibt die Variable immer an, weil beim Lesen des Programms dann sofort klar ist, welche Schleife geschlossen wird.

Endlos

Zum Schluß noch ein Trick: Programmieren Sie doch einmal eine Schleife mit der Schrittweite 0 (also STEP 0). Da hier zum Zähler ständig null addiert wird, erreicht er nie seinen Endwert, es handelt sich also um eine Endlosschleife. Diese können Sie nur verlassen, wenn Sie den Zähler selbst auf den Endwert setzen, wie oben beschrieben. Das ist eine elegante Methode, auch in BASIC eine Repeat-Until-Struktur zu programmieren.

```
10 FOR Z = 1 TO 2 STEP 0
20 GET X$
30 IF X$ = "W" THEN Z = 2
40 NEXT Z
```

F. Dreismann/kfp

Assembler-Know-how für alle!

Ab sofort direkt beim Verlag erhältlich: Ein Leckerbissen für jeden Assembler-Programmierer und alle, die es werden wollen.

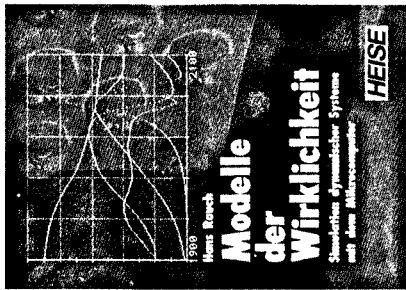
Eine Diskette mit dem Macro-Assembler INPUT-ASS aus INPUT 64, Ausgabe 6/86, und dazu

- der komplette Source-Code dieses Assemblers
- der Source-Code des Maschinsprache-Monitors MLM64plus aus INPUT 64, Ausgabe 11/87
- Library-Module: I/O-Routinen, Hex/ASCII/Dezimal-Wandlung, Multiplikation, Division
- Konvertierungsprogramme zur Format-Wandlung von PROFI-ASS- und MAE-Texten in das Source-Code-Format des INPUT-ASS

Preis: 49,— zuzüglich 3,— DM für Porto und Verpackung (nur gegen V-Scheck)

**Bestelladresse: Verlag Heinz Heise GmbH & Co KG
Postfach 61 04 07 · 3000 Hannover 61**

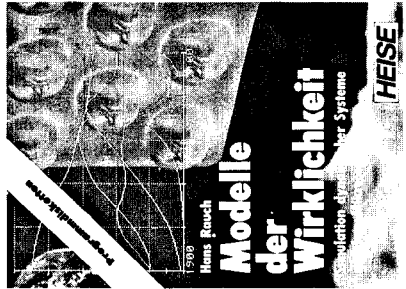
Wie wirklich ist eine Computer- Simulation?



Entwickelt werden in diesem Buch neun Simulationsmodelle zu unterschiedlichen Bereichen: radioaktive Strahlung, Räuber-Beute-Modell, Pflanzengift DDT in der Umwelt, Bevölkerungspyramide, das komplexe Weltmodell von J. FORRESTER u. a.

Durch die kritische Erörterung der Ergebnisse werden die Vorteile und Grenzen von Computer-Simulationsmodellen herausgearbeitet.

Broschur, 210 Seiten
DM 29,80
ISBN 3-922705-24-3



DI Programm disketten

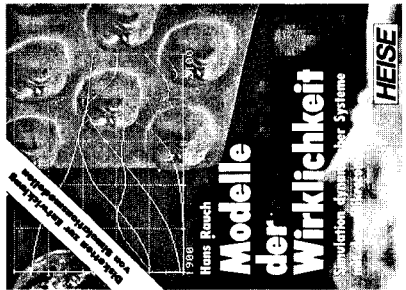
Auf den DI-Disketten befinden sich die im Buch behandelten lauffähigen Programme. Die eingestellten Standardwerte der Simulationen können sehr einfach und komfortabel verändert werden.

Die Sourcetexte sind nicht enthalten.

Die Disketten enthalten **nicht** das notwendige Betriebssystem.

Alle Versionen werden mit der jeweils notwendigen Diskettenanzahl geliefert. Unverbindliche Preisempfehlung.

Apple	Nr.: DI 0601-0	DM 58,—	DII 0611-8	DM 58,—
IBM PC	Nr.: DI 0602-9	DM 58,—	DII 0612-6	DM 58,—
Schneider	Nr.: DI 0603-7	DM 58,—	DII 0613-4	DM 58,—
Atari ST	Nr.: DI 0604-5	DM 68,—	DII 0614-2	DM 68,—



DII Disketten zur Entwicklung von Simulationsmodellen

Auf den Disketten befinden sich die vollständigen Sourcetexte der Programmierumgebung und der im Buch behandelten Programme. Als Programmiersprache wird Turbo- bzw. ST-Pascal benötigt.



Verlag
Heinz Heise
GmbH & Co KG
Postfach 61 04 07
3000 Hannover 61

Im Buch-, Fachhandel oder beim Verlag erhältlich. 524/1,2

BUCH & SOFTWARE

Kreative Gestalten

Eigener Shape-Editor für VisHypnotica

Das in der Dezemberausgabe 1986 vorgestellte Programm „VisHypnotica“, dessen einziger Sinn perfekt animierende Optik ist (aber das wenigstens richtig!), hat jetzt von seinen „Erfindern“, Christoph Gerlach und Wolfgang Schulze, einen Editor zum Erstellen eigener Shapes bekommen.

Zur Erinnerung: Mit Shapes sind die mit dem Joystick bewegbaren Figuren gemeint, die durch ständigen Farbwechsel eine unterschiedliche Verweildauer vortäuschen. Wem das zu abstrakt ist: angucken! Mit diesem Editor wird auch VisHypnotica noch einmal mitgeliefert.

Nach dem Start landet man zunächst in VisHypnotica, von dort wird mit F7 der Editor aufgerufen. Von den neun möglichen Shapes sind acht festgelegt, das neunte gilt als „User-Shape“, nur dieses ist editierbar. Um auch die fest installierten Shapes verändern zu können, wird beim Aufruf des Editors das angezeigte Shape übernommen, gilt aber ab sofort als Shape 9 und überschreibt dieses.

Wie vor zwei Monaten versprochen, schieben wir nun das Programm zum Selbsterstellen „visuell hypnotisierender Gestalten“ nach.

Der Editor selbst ist menügesteuert. Mit dem Joystick oder dem Tastenkreuz 'Q', 'W', 'E', 'A', 'D', 'Z', 'X' und 'C' wird der „Cursor“ bewegt, Aktionen löst man mit dem Feuerknopf beziehungsweise der Shift-Taste aus (Umschalten: F1). Am oberen Bildschirmrand ist eine Menüleiste aktivierbar (Feuer/Shift), die die Funktionen „Zurück zu VisHypnotica“ (der Punkt), „Disketten-Menü“ und „Editieren“ enthält.

Im Disketten-Menü ist nur das Laden und Speichern von Shapes („Load shape“, „Save shape“) erklärungsbedürftig. Die Shapes werden nämlich nicht mit frei wählbaren Namen auf Diskette abgelegt, sondern sind von '01' bis '99' durchnummeriert. Diese Nummern werden beim Laden und Speichern als Eingabe erwartet.

Im Editier-Menü ist der Shape-Ursprung, der Punkt, von dem aus sich der Shape entwickelt, bestimmbar und der Farbablauf. Ein Shape kann aus maximal 120 Punkten bestehen, die bis zu 240mal ihre Farbe ändern können. Im Farbablauf-Editor ist die vorgegebene Farbe Schwarz, jeder Druck auf den Feuerknopf inkrementiert den Farbwert, also von Schwarz über Weiß nach Rot und so weiter. Die eingestellten Werte werden so lange übernommen, bis nur noch Schwarz erscheint.

Alles weitere ist im Programm selbst durch ausführliche Hilfstexte erklärt. VisHypnotica läuft innerhalb und außerhalb von INPUT64. JS

Forts.: Blick zurück . . .

Die Tabelle

BOXf# ,lox,loy,rox,ruy	FRAME lox,loy,rox,ruy
BOXf# ,lox,loy,rox,ruy,1	BOX lox,loy,rox,ruy
CHARf# ,x,y,"text"	TEXT x,y,8,"text"
CIRCLEf# ,xm,ym,xr,yr	CIRCLE xm,ym,xr,yr
COLOR 0,hf,L,COLOR 1,zf,L	COLOR zf,hf
DRAWf# ,xa,ya to xe,ye	LINE xa,ya,xe,ye
DRAWf# to x,y	MOVE x,y
DRAWf# ,x,y	SET x,y
GRAPHIC 1	GRAPH
GRAPHIC 0	NORM
GRAPHIC 1,1	HIRES zf,hf
LOCATE x,y	SET x,y

PAINTf# ,x,y	FILL x,y
SCNCLR	CLS

SCALE existiert in INPUT-BASIC nicht, kann aber durch einen Multiplikationsfaktor leicht ersetzt werden. Nach SCALE 1 liegt die x-Achse nicht mehr im Bereich von 0 bis 319 und die y-Achse von 0 bis 199, sondern beide von 0 bis 1023.

Da der C16 keinen so tollen Sound-Chip wie der C64 hat, kann die Musik nicht so leicht übertragen werden. Vor allem die genauen Werte für ENVELOPE und WAVE sind nicht bekannt. Bei der Umsetzung des VOL-

Befehls muß man beachten, daß die Werte beim C16 nur von 0-7 gehen (bei INPUT-BASIC 0-15). Eine sehr gute Näherung erreicht man mit:

```
VOL vo =  
ENVELOPE1,2,2,5,5:  
WAVE1,16:VOLUME2*vo
```

```
SOUND v,freq,la -  
PLAY v,freq,la
```

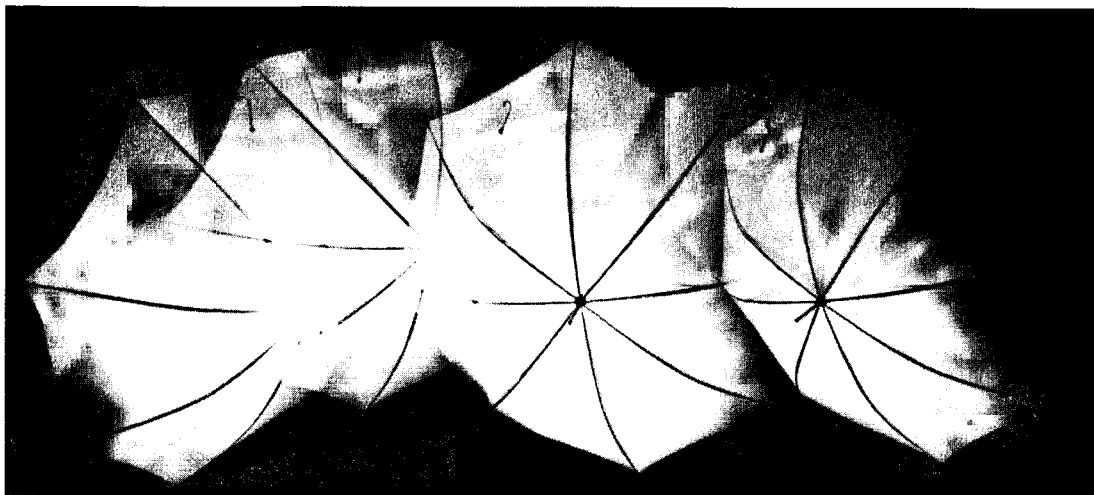
INPUT-BASIC enthält auch die wichtigen Programmierhilfen des C16 wie AUTO, RENUMBER und DELETE. kfp/rh/JS

WIE

bekommt man 12mal INPUT 64 zum Preis von 9,5. Plus kostenloser Zustellung ins Haus?

SO

Durch das INPUT 64-Abonnement. 198,— DM pro Jahr. Plus der Möglichkeit zur übernächsten Ausgabe kündigen zu können und auch noch überzahlte Gebühren zurückzubekommen. Die Abo-Bestellkarte ist auf der letzten Seite. ALSO!



Vier Schirme im Rechner

Scrange: Bildwechsel-Tool

Mit diesem von Marc Sinemus geschriebenen Tool stehen dem Programmierer vier verschiedene Bildschirme zur Verfügung. So ist ein Wechsel zwischen unterschiedlichen Ausgabemasken oder das Anlegen von „Notizzetteln“ schnell und ohne detaillierte Kenntnisse des Video-Chips möglich.

Da das Programm sowohl den Direkt- als auch den Programmmodus unterstützen soll, gibt es zwei Methoden, zwischen den Screens umzuschalten. Im Direktmodus entspricht jede Funktionstaste einem Bildschirm, es gilt folgende, nahegelegende, Zuordnung:

- F1 Screen 1
- F3 Screen 2
- F5 Screen 3
- F7 Screen 4

Schneller Schalter

Diese Funktionstasten-Steuerung sollte während eines Programmlaufs abgeschal-

Bei genauerem Hinsehen geben die Einzelbestandteile des Kunstnamens „Scrange“ einen Hinweis auf Sinn und Zweck dieses kleinen Hilfsprogramms: die englischen Begriffe „Screen“ und „Change“ standen Pate bei der Namensgebung.

tet werden, da dann ein direktes Umschalten per SYS-Befehl praktikabler ist. Die anzuspringende Adresse ist 49155 (\$C003), vorher muß in Adresse 49158 (\$C006) hinterlegt werden, was zu tun ist. Es gibt folgende Möglichkeiten:

- 0 F-Tasten-Steuerung ausschalten
- 1 Screen 1
- 2 Screen 2
- 3 Screen 3

- 4 Screen 4
- 5 F-Tasten-Steuerung einschalten

Beispiel: POKE 49158,2:SYS 49155 schaltet Bildschirm 2 ein. Das Ganze funktioniert natürlich auch in Maschinensprache.

Achtmal abgelegt

Das Programm wird, nach dem Abspeichern auf eine eigene Diskette und Laden von dieser, mit RUN gestartet. Es belegt dann den Bereich von 49152 bis 49563 (\$C000 bis \$C19B), die vier Bildschirm- und Farbspeicher befinden sich „unter“ dem BASIC-ROM von 40960 bis 48960 (\$A000 bis \$BF40).

Die Abfrage, ob eine Funktionstaste betätigt wurde, ist in die Interrupt-Routine eingebunden. Durch einen Reset oder RUN/STOP-RESTORE wird das Tool „abgehängt“, kann aber durch SYS 49152 neu initialisiert werden. JS

Am 26. Februar an Ihrem Kiosk:
INPUT 64, Ausgabe 3/88



Wir bringen unter anderem:

Sandy

Digitalisierte Geräusche synthetisieren, anhören, manipulieren, verfremden, verknüpfen und mit Effekten überlagern, grafisch analysieren und editieren: das alles und noch viel mehr, würde ich tun, wenn ich der Sandy von INPUT wär! Außerdem könnte ich Daten von INPUT-Graph und INPUT-SAM verkraften und selbstverständlich auch die – in der nächsten Ausgabe – mitgelieferten Beispiel-Samples verdauen. Wie gesagt: Das alles und noch viel mehr . . .

FraGram

Für alte „INPUT-Hasen“ sieht FraGram eher wie ein Schreibfehler aus. Es ist aber keiner! Wir beginnen mit der nächsten Ausgabe ein neues Fremdsprachen-Repetitorium – diesmal in Französisch.

Einer gegen Alle

Es ist wieder soweit. Die Lösungsstrategie für die Chiffrierungsaufgabe aus INPUT 12/87 und die Veröffentlichung der Gewinner – sofern die Aufgabe bis dahin gelöst wird – erfahren Sie in der nächsten Ausgabe.

SuperDir

Es gibt schon viele Hilfsprogramme, die den BASIC-Programmierern den Umgang mit dem Directory erleichtern. Mit SuperDir dürfte mithin das Nonplusultra sowohl an Bedienungskomfort als auch an programmtechnischer Auswertung erreicht sein.

c't – Magazin für Computertechnik

Ausgabe 3/88 – ab 19. Februar 1988 am Kiosk

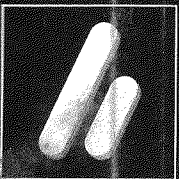
Projekt: A/D-Wandler für Atari ST, PC-Harddisk-Controller am Amiga * Grundlagen: Simulation chaotischer Systeme * Software-Know-how: TOS/GEM für 68020, Eigenschaften netzwerkfähiger Programme, Occam 3 – Sprache für Parallelverarbeitung, Amiga-Devices * u.v.a.m.

elrad – Magazin für Elektronik

Ausgabe 3/88 – ab 19. Februar 1988 am Kiosk

Bauanleitung μ P-Technik: Einplatinen-Mikrocomputer mit MIDI-Anschluß E.M.M.A. emanzipiert sich – zum Entwicklungssystem * Der Weg zum eigenen Meßlabor (4): Intelligente Meßgeräte, Zusätze für Heimcomputer, Logiktester bis zum Mini-Logikanalyzer * Bauanleitung Labor: Netzgerät 0 . . .16 V/20 A * Report Satellitentechnik * Bauanleitung Audio: High-End-Stereo-Vorverstärker * u.v.a.m.

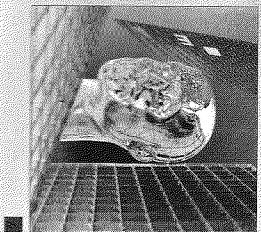
HEISE



Künstliche Intelligenz Die aktuelle Computer- anwendung

COMPUTER- BUCH

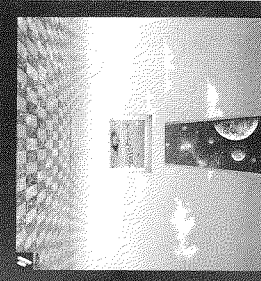
Urich Eisenbroder Künstliche Intelligenz und Musteranalyse



Ein wesentliches, wenn nicht sogar entscheidendes Problem in der Forschung zur künstlichen Intelligenz ist das selbständige Auffinden gänzlich neuer und das Wiedererkennen bekannter Muster in Texten, Bildern, Musikstücken usw. Der Autor stellt ein neues Verfahren zur Musteranalyse von Zeichenketten vor.

DM 39,80
Broschur, 189 Seiten
ISBN 3-88229-125-7

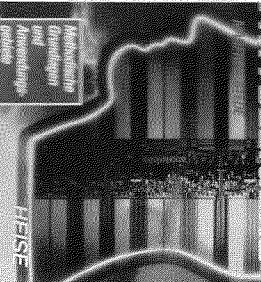
Manfred Stede PASCAL-PROGRAMME zur künstlichen Intelligenz



Theoretische Informationen über künstliche Intelligenz werden in konkrete Programme umgesetzt, die der Leser ausprobieren, verstehen und erweitern kann. Zum Experimentieren dienen dem fortgeschrittenen Hobby-Programmierer vor allem die Bereiche Suchverfahren und Spielstrategie.

DM 44,80
Broschur, 219 Seiten
ISBN 3-88229-126-5

EINFÜHRUNG IN DIE KÜNSTLICHE INTELLIGENZ



Der umfassende Einblick in diesen hochaktuellen Bereich der Computerprogrammierung ermöglicht es dem Leser, sich sein eigenes Urteil über Chancen und Grenzen der künstlichen Intelligenz zu bilden. Die methodischen Grundlagen der KI und ihre wichtigsten Anwendungsfelder werden vorgestellt.

DM 49,90
Broschur, 267 Seiten
ISBN 3-88229-018-8



Im Buch-, Fachhandel oder beim Verlag erhältlich. **KV 22**

Verlag
Heinz Heise
GmbH & Co KG
Postfach 61 04 07
3000 Hannover 61