

INPUT 64  
auf der CeBIT '88  
16. 3. bis 23. 3.  
Halle 7, Stand C57

DAS ELEKTRONISCHE MAGAZIN 3/88

# INPUT 64

Infos · News · Programme · Unterhaltung · Tips **DM 19,80**

Unverbindliche Preisempfehlung

Klänge erzeugen

## SANDI

Sound-Sampler und -Sequencer

Bytes erforschen

## C128-Analyzer

Hintergrundmonitor

Höhlen erobern

## Lynx the ball

Video-Spiel

Tool: Super-Directory

Neuer Kurs:

Französische Grammatik

Serien: 64er Tips

Digitale Logik

Einer gegen Alle



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 Schnellader 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 Schnellader 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 belegt 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 Hochkomma. 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 64er-BesitzerInnen!

Es gibt einen jahrhundertealten philosophischen Streit um Löcher. Die endgültige Definition dieses Begriffs ist nach wie vor heftig umstritten. Das Problem von richtigen und falschen Löchern zur richtigen Zeit am richtigen Ort stellt sich ja auch bei Disketten. Nebenbei bemerkt: Für einige stellen Disketten schon fast so etwas wie ein Fossil dar, andere bestaunen diese Dinger immer noch als Wunder der Technik. Diese polarisierte Betrachtungsweise scheint ein Phänomen unserer schnellebigen Zeit zu sein – kaum erblickt ein Ding das Licht des Markts, welkt es auch schon wieder im Schatten der neuen Entwicklungen.

Diese Polarisierung spiegelt sich alltäglich im Umgang mit den Dingen wider. Ganz vorsichtige Naturen behandeln ihre Disketten fast wie ein rohes Ei. Und richtig dramatisch wird es erst, wollen sie eine Diskette per Post auf die Reise schicken. Sicher hat sich eine feste Pappe gegen Knicke und ein gepolsterter Umschlag als Schutz vor dem Poststempel gut bewährt, doch kann man es auch übertreiben. Die mehrfachen Lagen Aluminiumfolie beispielsweise benutze ich lieber fürs Butterbrot. Diskettenverpackungen in der Größe einer Waschmaschine (und sei's auch nur ein Untertisch-Modell) haben meist den berühmten „Nadel im Heuhaufen“-Effekt.

Respektlose Redaktionskollegen haben im Rahmen der Modewelle „Back to the roots“ zahlreiche Versuche unternommen, die Speicherscheiben beim Schreibtisch-zu-Schreibtisch-Transport im Sinne des Wortursprungs als Diskus beziehungsweise als Frisbee einzusetzen – auf Grund der mangelhaften aerodynamischen Form meist nur mit mäßigem Erfolg: landeten die Disketten doch allzu häufig im Papierkorb.

Bei der „Rundablage“ muß ich daran denken, wie ich vor längerer Zeit einen besonders ordentlichen Bekannten hatte. Dieser lochte seine Disketten, um sie danach fein säuberlich abzuheften. Tief beeindruckt davon entschloß ich mich, mein Chaos zu ordnen. Ein schönes Loch zwischen den Spuren eins bis fünf schmälerte allerdings die erfolgreiche Realisation meines guten Wil-

lens. Ich kehrte ohne schlechtes Gewissen sofort zur „kreativ-notwendigen“ Unordnung zurück, und seitdem ist meine Wand um ein Schmuckstück reicher.

Aber um diese Löcher sollte es eigentlich ja nicht gehen. Die ganz Respektlosen nämlich scheuen vor nichts zurück. Da wird einfach eine zweite Schreibschutzkerbe in die Scheiben „gebissen“, und schon ist die Speicherkapazität der Disketten verdoppelt – bei gleichzeitiger Halbierung ihrer Anzahl. Wahre Kunstwerke sind auf diese Art und Weise schon entstanden. Zugegebenermaßen ist es schwierig, mit einem normalen Locher eckige Löcher zu stanzen.

So oder so sollte man Abstand davon nehmen, Original-Disketten zu knipsen. Da die Laufwerke in der Kopieranstalt einfach alles beschreiben, was ihnen in den Schacht kommt, können wir Ihnen die INPUT-Disketten ohne Schreibschutzkerbe anbieten. Der Vorteil für Sie liegt darin, daß die Disketten gegen unbeabsichtigtes Löschen und Formatieren geschützt sind, was ja auch nicht unerheblich im Rahmen der „Sachmängelgewährleistungspflicht“ ist. Und Platz für eigene Programme werden Sie auf der Diskette auch kaum finden, denn meist bekommen wir die gesamte INPUT nur mit Mühen und Tricksereien auf die Scheibe.



Peter S. Berk

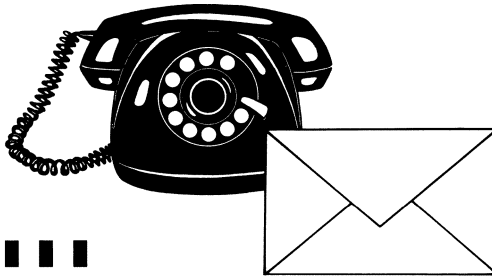
3/88



## INHALT

<b>Leser fragen</b>	2
<b>News</b> Neuer C64 als C64D	4
<b>SANDY</b> Ein Soundstudio im C64	5
<b>SuperDir</b> Directory für BASIC-Programmierer	12
<b>Französische Grammatik</b> Folge 1: Les Articles définis et indéfinis	13
<b>Lynx the ball</b> Helikopter-Spiel	14
<b>Einer gegen Alle</b> Kodier-Wettbewerb	16
<b>C128-Analyzer</b> Single-Stepper und Scanner	18
<b>G-Studio</b> Viernial Musik	22
<b>Gatterlogik</b> Teil 2: Flipflops	23
<b>Schloßgeist</b> Mini-Adventure	27
<b>64er Tips</b> Unterprogramm-Programmierung	28
<b>Vorschau</b>	31
<b>Impressum</b>	32

# Leser fragen...



## Damen „abgespeichert“

*Ich würde ein Programm benötigen, das es mit einem C 64 ermöglicht, z.B. für einen Herrn mit folgenden Daten:*

- Alter 25
- Zivilstand ledig
- Schulbildung Diplom
- Tätigkeit Angestellter
- Höhe 1,76
- Position Wohlhabend
- (etwa 10 Daten)

*alle, zwischen den 1000 eingespeicherten, passenden Damen herausucht.*

*W. Rode, 1 Fano*

Hier müssen wir passen. Wir kennen kein Programm, in dem die Daten von 1000 passenden Damen enthalten sind. Aber vielleicht kann ja ein(e) Leser(in) weiterhelfen? (d. Red)

## RS-232 „entwanzt“

Beim Experimentieren mit den RS-232-Routinen haben wir im C64-Kernal einen Fehler entdeckt. Ehrlich gesagt – nicht sofort; aber wenn kontinuierlich das erste empfangene Zeichen verstümmelt ist, wird auch der selbstkritischste Programmierer irgendwann die Ursache bei anderen suchen: in diesem Fall in den Kernal-Routinen. Die Analyse der beteiligten Routinen hat ergeben, daß das erste Startbit der Übertragung bereits als gültiges Datenbit interpretiert wird. Wollen Sie also auch das erste gesendete Byte richtig empfangen, sollten Sie vor

dem Öffnen des Datenkanals die Adresse \$a9 (Startbit-Flag) mit dem Wert \$92 versorgen. Für BASIC-Programmierer heißt das: POKE169,146. (d. Red.)

## V2-BASIC „gekürzt“

*Ich bin gerade dabei ein Adventure zu schreiben, und dabei stoße ich auf das Problem, daß ich die Programme IMC-BASIC (INPUT 9/87) und das Rasterzeilen-Tool (INPUT 6/85) beide zusammen betreiben möchte. Können Sie mir da helfen?*

*Und ich habe noch einen Tip für alle INPUT 64-Leser: Im Handbuch steht, daß es für den Befehl >PRINT#fn< keine Abkürzung gibt, aber es gibt sie doch! Mit >prl< wird dieser Befehl abgekürzt.*

*A. Czernay, Hamburg*

Ein 'Zusammenspiel' der beiden Programme ist leider nicht möglich. Ihren Hinweis geben wir aber gerne weiter. (d. Red)

## Ist BASIC „geladen“

*Ich habe auf einen Datenträger 'SuperDisk' unter dem Namen 'fast' gespeichert. Hier tauchten schon am Anfang Probleme auf. Welche SuperDisk-Version aus INPUT 64 1/87 war nun richtig? Weder das Studium des Begleitheftes noch das der Programmdemo auf der Diskette gaben mir die gewünschte Auskunft. Logischerweise nahm ich die INPUT-BASIC-kompatible Version mit dem Initialisierungsbereich 39183 – 41071 am Ende des BASIC-RAM. Nach dem*

*Anstarten des Laders "hängte sich die Floppy auf" und lief und lief...! Nach Drücken der STOP-Taste gab das Lade-Programm eine Meldung "Ladefehler!" aus. Bei mehreren verschiedenen Versuchen mit allen SuperDisk-Versionen passierte das gleiche.*

*Abgesehen von den kleinen Schwierigkeiten ist Ist BASIC in Verbindung mit INPUT-BASIC eine hervorragende Erweiterung der aus dem Hause Commodore stammenden armseligen Ausstattung des BASIC V2 des C 64-Interpreters. BASIC V70 bietet da einiges mehr. Allerdings ist der 64er-Modus auf meinem 128er unentbehrlich, da Ihre Programme bisher die sinnvollsten Ergänzungen zu dem "Volkswagen" der Home-Computer darstellen.*

*H. Kalipke, Recklinghausen*

Das mitgelieferte Ladeprogramm und die Informationen zum Zusammenspiel mit INPUT-BASIC wurden vom Autor der BASIC-Erweiterung kurz vor Redaktionsschluß als "Beigaben" nachgereicht und konnten deswegen leider nicht mehr ausreichend getestet werden.

Folgende Vorgehensweise ist richtig:

- Aus dem Modul Ist BASIC in Ausgabe 1/88 die Lader-Version ohne SuperTape abspeichern.
- Aus dem Modul SuperDisk in Ausgabe 1/87 Version 3 des Floppy-Beschleunigers abspeichern ("Kompatibel zu INPUT-BASIC"). Die Vorgaben können übernommen werden.
- Dieses Programm muß noch verändert werden: Laden Sie "SuperDisk" von Ihrer eigenen Diskette, geben im Direktmodus POKE 2267,96 ein und speichern es unter dem Namen "fast" auf der Diskette wieder ab, auf der sich auch der Lader, Ist BASIC und INPUT-BASIC befinden.

Außerdem ist in dem Kasten auf Seite 7 der Ausgabe 1/88 ein Schreibfehler enthalten. Richtig muß es heißen:

- x = 0 (keine Erweiterung im Speicher)
- x = 1 (INPUT-BASIC ohne SuperTape)
- x = 2 (INPUT-BASIC mit SuperTape)

(d. Red.)



## ID-Werkstatt „überlistet“

... übersende ich Ihnen in der Anlage eine Diskette mit dem CAD 4/1 und der aus der ID-Werkstatt kopierten Datei "ANALOG .1" mit der Bitte um Prüfung, da das CAD diese anscheinend nicht akzeptiert.

Liegt es u.U. daran, daß die CAD-Teile 1-4 von der Cassettenversion auf Diskette überspielt wurden?

P. Borchers, Langenhagen

Nein, an Ihrer CAD-Version liegt es nicht! Wir wollten es nicht wahrhaben, denn natürlich haben wir die Funktionen der ID-Werkstatt vor der Veröffentlichung geprüft. Nun gut, einige kleine Programmveränderungen unter der Benutzeroberfläche, aber – hier hilft kein aber; nur ein ehrliches 'tschuldigung. Diese kleine Änderung (Voraussetzung für die Abspeicherung von sequentiellen Files) wird dummerweise erst initialisiert, wenn die Cursor-Tasten mindestens einmal benutzt worden sind (wir finden das auch nicht plausibel). Wenn also die CAD-Datei als zweites File aus der ID-Werkstatt abgespeichert wird, ist alles in Ordnung (hierzu muß die besagte Taste ja bewegt worden sein). Somit können Sie sich bei dem Problem auch selbst behelfen. Die nächste ID-Werkstatt wird dann diese versteckte Gemeinheit nicht mehr enthalten. (d. Red.)

## Multicolor-BASIC „aufgebohrt“

Multicolor-BASIC ist genau, was ich schon lange suchte: eine vielseitige (fehlerfreie) Erweiterung, die gleich mit dem Anwenderprogramm geladen werden kann. Nun aber ein Problem: ich möchte den Zeichensatz teilweise ändern bzw. auch Umlaute einbauen. Ich finde ihn aber nicht und kenne das Format für den schmalen Satz nicht. Können Sie mir dazu Genaueres sagen?  
J. Willmann, CH Reisiswil

Die ausführliche Antwort gibt für uns der Autor von Multicolor-BASIC, Oliver Kraus:

IMC-BASIC greift auf zwei verschiedene Zeichensätze zu, auf den normalen ROM-Zeichensatz und einen speziellen Multicolor-Zeichensatz.

Der Zugriff auf den ROM-Zeichensatz kann einfach durch Änderung des High-Bytes der Startadresse geändert werden. Es wird ein neuer Zeichensatz an eine freie Stelle im RAM (siehe Speicherbelegung) geladen und dann das High-Byte in die Grafikroutinen geschrieben. Programm 1 lädt einen Zeichensatz mit dem Namen "fantasy" nach \$C000 und schreibt anschließend den Wert 192 (\$C0) in die Speicherzelle 7904 (Version 1 von IMC-BASIC).

Der spezielle Multicolor-Zeichensatz kann nicht so einfach geändert werden. Ich habe deshalb ein Programm (prog 2) geschrieben, das ein einzelnes Zeichen ändert. Die Daten des Zeichens stehen in den Zeilen 1020 bis 1100. Diese Daten besagen, daß ein 'A' anstelle des "Klammeraffen" (BS=0 steht für Bildschirmcode 0) ausgegeben werden soll. Zunächst wird die Position des Zeichens berechnet (1140 bis 1160). 'Z' enthält dann die Startadresse. Die Variable 'P' sagt aus, an welcher Bit-Position das Zeichen steht. Die Zeichen sind etwa so angeordnet:

Bit:	7	6	5	4	3	2	1	0
0000	-	*	-	-	*	*	-	-
0001	*	-	*	-	*	-	*	-
0002	*	-	*	-	*	-	*	-
0003	*	*	-	*	-	*	-	*
0004	*	-	*	-	*	-	*	-
0005	*	-	*	-	*	-	*	-
0006	*	-	*	-	*	-	*	-
0007	-	-	-	-	-	-	-	-

Alle 32 Punkte eines Zeichens werden zeilenweise gelöscht (Zeile 1200), berechnet (Zeile 1210) und schließlich wieder in Abhängigkeit von 'S' gesetzt (Zeile 1220). In Zeile 1240 wird dann das berechnete neue Byte wieder in den Speicher geschrieben und eine neue Zeile begonnen (Zeile 1250).

Das High-Byte des ROM-Zeigers liegt bei der Version 1 (\$A0) in der Adresse 7904 (\$1EE0) und bei der Version 2 (\$E) in der Adresse 7907 (\$1EE3). Der Start der Multicolor-Zeichen liegt bei der ersten Version bei 9574 (\$2566) und bei der zweiten Version bei 9587 (\$2573).

Es werden sowohl bei den ROM-Zeichen als auch bei den Multicolor-Zeichen nur die Bildschirmcodes von 0 bis 95 berücksichtigt. (Oliver Kraus, Forchheim)

## Programm 1: Zeichen anzeigen

```
10 mload 49152."fantasy".8
20 poke 7904,192 :rem version 1
30 rem version 2: poke7907,192
40 :
1000 rem alle zeichen anzeigen
1010 :
1020 multi
1030 text 0,0,1,1,1
1040 for i=32 to 128
1050 printchr$(i);
1060 nexti
1070 geta$:ifa$=""then1070
1080 mcoff
1090 end

ready.
```

## Programm 2: Zeichen ändern

```
10 ad=9574 :rem version 1
20 rem version 2 : ad=9587
30 gosub 1000
40 gosub2000
90 end
1000 rem zeichen aendern
1010 :
1020 a$(0)=" * "
1030 a$(1)="* * "
1040 a$(2)="* * * "
1050 a$(3)="*** "
1060 a$(4)="* * * "
1070 a$(5)="* * * "
1080 a$(6)="* * * "
1090 a$(7)=" "
1100 bs=0 :rem bildschirmcode
1120 : rem des zeichens
1130 p=0
1140 z=bs/2
1150 if z=int(bs/2) then p=4
1160 z=int(z)*8+ad
1170 for i = 0 to 7
1180 w=peek(z+i)
1190 for j = 0 to 3
1200 w=w and 255-2^(j+p)
1210 s=abs(mid$(a$(i),4-j,1))+"*")
1220 w=w or (s*2^(j+p))
1230 nextj
1240 poke z+i,w
1245 printw
1250 nexti
1260 return
1900 :
2000 rem alle zeichen anzeigen
2010 :
2020 multi
2030 text 0,0,2,1,1
2040 for i=32 to 128
2050 printchr$(i);
2060 nexti
2070 geta$:ifa$=""then2070
2080 mcoff
2090 return

ready.
```

## è „vergessen“

*Nach der Anpassung des Vokabeltrainers (Ausgabe 3/87) an französische Unmlaute habe ich das Problem, daß ein 'è' zwar eingegeben werden kann, aber nach dem Abspeichern und anschließendem Laden nicht mehr existiert. Woran liegt das?*

*Steinkampf, Azberg*

Dieser Fehler ist, wie uns der Autor des Vokabeltrainers mitteilt, durch eine Änderung dieses Programms beherrbar, und zwar durch

POKE 6190,10

(d. Red.)

## UniDat-Datensammlung „gefilmt“

*Ihr elektronisches Magazin INPUT 64 finde ich wirklich gut. Ich programmiere zwar nicht selber und bin nur einfacher Anwender, aber einige Ihrer Programme sind mir wirklich eine große Hilfe im täglichen Leben.*

*Der eigentliche Grund meines Schreibens ist Ihre Veröffentlichung des Programms UniDat im INPUT 64 vom Dezember 1987. Für dieses Programm kann ich eine Datensammlung für Filmfans anbieten. Im Moment habe ich ca. 2000 Titel auf Diskette gespeichert ... wird laufend fortgesetzt. Ein Datensatz besteht aus 1. Titel, 2. evtl. Originaltitel, 3. Herstellungsland und Datum, 4. Regie und 5. - 9. Besetzung.*

*Falls solche Daten Ihrer Meinung nach für andere INPUT-Leser von Interesse sein könnten, bin ich gerne bereit, sie zur Verfügung zu stellen.* S. Meuters, Wegberg

Für eine Veröffentlichung innerhalb von INPUT ist so eine große Datei kaum geeignet, es handelt sich um insgesamt drei Disketten. Wir haben uns deshalb mit Herrn Meuters in Verbindung gesetzt und seine Einwilligung zur Veröffentlichung seiner vollständigen Anschrift eingeholt. Wenn Sie also an der beschriebenen Datei interessiert sind, senden Sie bitte einen mit 1,90 DM frankierten, gefütterten DIN-C5-Rückumschlag und 10,- DM an Siegfried Meuters, Philosophenweg 47, 5144 Wegberg Dalheim. Herr Meuters wird seine Datei dann für Sie kopieren. (d. Red.)

## Dienstag ist Lesertag

Technische Anfragen:  
nur Dienstag von 9 — 16.30 Uhr  
☎ (05 11) 53 52-0

### News

# Läuft und läuft und . . .

## Neuer C64 mit integriertem Laufwerk

**Commodore beschert den 8-Bit-Rechnern ein „ewiges Leben“: noch in diesem Jahr soll ein C64D mit eingebaute 3,5- oder 5,25-Zoll-Floppy erscheinen.**

Unklar ist nach Auskunft von Commodore nur noch die genaue technische Ausstattung, klar ist aber, daß der C64D kommt. Das 'D' steht wie auch beim C128D für ein eingebautes Diskettenlaufwerk. Über das Format, ob also 3,5- oder 5,25-Zoll, sind



noch keine endgültigen Entscheidungen gefallen. Durch die Integration der Floppy sei aber in jedem Fall ein günstiger Preis möglich. Um circa 100 Mark werde man den jetzigen Preis für einen C64 mit Floppy 1541 unterbieten können, somit dürfte der C64D für etwa 500 DM in den Handel kommen.

Natürlich sei der neue 64er zu 100 Prozent software-kompatibel zum bisherigen Modell, das mit Erscheinen des D-Modells ab

**So könnte der designierte Nachfolger des C64, der C64D, aussehen: Rechner und Floppy in einem Gehäuse.**

gelöst werden soll. In der Gerüchteküche ist zu hören, daß auch die Integration eines Btx-Chips und des Netzteils geplant sei. Die Zeitplanung war zum Redaktionsschluß folgende: Genauere Vorstellung im März '88 auf der CeBIT, Erscheinen auf dem Markt zur Jahreswende 88/89. JS



# Digitales Soundstudio

## SANDI: Soundsampler-Sequencer-Editor

Der C64 liefert mit dem eingebauten Soundchip, dem SID, bereits den Anreiz, „klangvolle“ Programme zu schreiben. Doch um wirklich ansprechende Klänge aus dem SID herauszuholen, sind neben musikalischen Grundkenntnissen auch routinierte Programmiererfahrungen notwendig. Zudem sind den Möglichkeiten dieses Chips Grenzen gesetzt. So lassen sich bestimmte Instrumentalklänge nur mit großen Schwierigkeiten, teilweise überhaupt nicht synthetisieren. Diese klingen dann eben doch nicht wie ein echtes Klavier. Soundsamples hingegen sind Klänge, die über Mikrofon oder Tonabnehmer direkt vom Instrument abgenommen und in digitale Signale umge-

**In der Musik und für akustische Spezialeffekte beim Film sind Klangverfremdungen nicht mehr wegzudenken. Professionelle Geräte kosten aber meist viel Geld und sind für das Tonstudio daheim zu teuer. Für Ihre eigenen Experimente und zur Einbindung von Effekten in Ihre eigenen Programme wurde SANDI konzipiert. Das Thema? Soundsampling, Klang-Digitalisierung mit Ihrem Homecomputer.**

wandelt wurden. Der aufgenommene Naturklang ist, wird er mit einem geeigneten Gerät oder Programm abgespielt, wirklich wiedererkennbar. Dies gilt selbst für Nuancen der eigenen Stimme. Bei entsprechend großem Aufwand ist die Wiedergabe sogar fast vollkommen originalgetreu.

Experimentierfreudige Musiker haben darüber hinaus die Möglichkeit, solche Samples per Computerprogramm zu verändern, zu analysieren und mit Effekten anzureichern. Einzelne Samples lassen sich in der Tonlage verschieben und zu Sequenzen zusammensetzen, so daß aus einem einzigen kurzen Klang oder Geräusch ein komplettes Musikstück entsteht.

SANDI bietet hierfür eine komfortable Anwenderoberfläche, sowohl für Soundbegeisterte als auch für Programmierer. Wer sich mit diesem Spezialgebiet erst einmal vertraut machen möchte, sollte sich dem Kastentext „Vom Brummer zur Nummer“ widmen. Fachbegriffe werden im Kastentext „Ins Fach“ erklärt.

Mit SANDI sind Sie in der Lage, Sample-Dateien von Diskette zu bearbeiten. Wenn Sie die Multifunktionskarte zum C64 aus c't 9/86 besitzen, können Sie Sounds auch direkt mit SANDI digitalisieren.

Es gibt inzwischen eine Reihe von Programmen, die digitalisierte Klänge enthalten. Ein Beispiel ist der Funky-Drummer aus INPUT 64, Ausgabe 12/85. Solche Programme können mit SANDI komplett geladen werden. Durch Abhören und Verändern der Speichergrenzen können Sie mit etwas Geschick den Sound heraus schneiden und abspeichern. Sie haben damit ein Basis-Sample, das Sie anschließend nach Belieben verändern können. Ebenso kann man mit einem Programm wie zum Beispiel INPUT-Graph aus Ausgabe 9/87 mathematisch Samples synthetisieren.

Wer eine Hardware-Erweiterung besitzt, mit der sich Klänge digitalisieren lassen, kann natürlich auch damit erzeugte Sample-Files in SANDI übernehmen und weiterbearbeiten. Zum Digitalisierer auf der Multifunktionskarte zum C64 (c't, Ausgabe 9/86 und INPUT 64, Ausgabe 8/86 bis 11/86) hat SANDI ein besonders gutes Verhältnis. SANDI kann über diese Karte Sounds direkt aufnehmen und wiedergeben. Dafür steht wahlweise ein 4-Bit- oder ein 8-Bit-Modus zur Verfügung. Ersterer paßt sich den Ausgabemöglichkeiten des SID an. Damit sind 16 Abstufungen möglich. Im 8-Bit-Modus muß der Sound über einen Digital/Analog-Wandler, wie er auf der Multifunktionskarte zu finden ist, ausgegeben werden.

Die Wiedergaberoutine wurde gegenüber den Sample-Tools des SAM-Projektes in INPUT 64, Ausgabe 8/86 bis 11/86 erheblich verbessert. Die Sample-Rate wird bei SANDI von den Timern des Complex-Interface-Adapters (CIA) kontrolliert und nicht wie bei den älteren Tools durch Befehlszyklen gesteuert. Daher klingen Samples, die mit den

SAM-Tools erzeugt wurden, unter SANDI verzerrt. Entsprechendes gilt auch für Sample-Files anderer Digitalisierer. So sollten nach dem Editieren mit der ursprünglichen Wiedergaberoutine (zum Beispiel SAM II aus INPUT 64, Ausgabe 9/86) abgespielt werden.

Damit Sie sich ein Bild vom jeweiligen Sample machen können, haben Sie mit SANDI die Möglichkeit, sich Ihr gelassenes „Hallo“ grafisch darstellen zu lassen und es mittels Cursor-Tasten oder Joystick direkt in der Grafik zu verändern. Darüber hinaus können Sie mit Hilfe dieses grafischen Sound-Editors auch Klänge in den Speicher „hineinmalen“ – ein fast spielerisches Vergnügen.

## Ins Soundstudio

SANDI verwaltet 10 Sound-Puffer, deren Inhalte jeweils grafisch und akustisch ausgegeben und editiert werden können. Parameter wie Speichergrenzen, Rate, grafische Vergrößerung, 4/8-Bit-Format und Effekte lassen sich für jeden Puffer getrennt direkt am Bildschirm einstellen. Sie können die Klangmuster je nach Wunsch schneller oder langsamer abspielen oder auch mal „das Band“ rückwärts ablaufen lassen. SANDI gibt Ihnen umfangreiche Manipulationsmöglichkeiten in die Hand, mit denen Sie die Klangmuster verfremden oder mit Effekten mischen können.

Neben der Basis-Ebene, dem Speichereditor, gibt es vier weitere Bearbeitungsebenen:

Im **Verknüpfungsmodus** können Sie die Inhalte verschiedener Puffer miteinander mischen, logisch verknüpfen, vom 4-Bit-Modus in den 8-Bit-Modus und umgekehrt umwandeln, Puffer kopieren oder auch einen Puffer mit Konstanten und verschiedenen logischen Operationen manipulieren.

Ein **Keyboard-Modus** simuliert eine Klaviatur, so daß Sie den jeweils ausgewählten Puffer mittels Tastatur über fast zwei Oktaven abspielen können. Die Effektunterlegung kann dabei mit wiedergegeben werden.

Im **Grafikmodus** können Sie jedes einzelne Bit Ihres Samples grafisch untersuchen

## Befehlsübersicht:

0-9,A-F	Werte für Tabelle
K	Kontrollmodus mit Aussteuerung
G	Grafikanzeige
L	Laden
P	Wiedergabe
R	Aufnahme
S	Speichern
U	Umdrehen des Puffers
@	DOS-Befehle senden
CLR	Puffer löschen
RETURN	Wert erhöhen
SHIFT-RETURN	Wert erniedrigen
f1	Verknüpfen
f3	Keyboard
f5	Klangeditor
f7	Sequenz
f8	Exit

und mit Cursor-Tasten oder Joystick verändern. Hier können Sie auch selbst Samples in den Speicher zeichnen.

Ein eingebauter **Sequenz** erlaubt die Komposition von minutenlangen Musikstücken, die sich aus Klangelementen (Samples) zusammensetzen. Hierzu können Sie alle 10 Puffer über 21 Tonstufen variieren. Kenner älterer INPUT 64-Ausgaben dürften hier einige Grundfunktionen des Funky-drumpers aus INPUT 64, Ausgabe 12/85 wiedererkennen. SANDI kann mit digitalisierten Drum-Sounds auch als Drum-Computer eingesetzt werden, eine der vielfältigen Einsatzmöglichkeiten von SANDI.

## Soundkonzepte

Die Idee zu SANDI basiert auf einer Entwicklung, die vor zwei Jahren begann. In enger Zusammenarbeit mit unserem Schwesternmagazin c't entstand die bereits oben erwähnte Multifunktionskarte für den C64. Die Hardware präsentierte c't in Ausgabe 9/86, wir brachten in INPUT 64, Ausgabe 8/86 bis 11/86 unter INPUT-SAM die zugehörigen BASIC-Tools. Die Karte beherr-



## Vom Brummer zur Nummer

Was passiert eigentlich beim Digitalisieren, und wie funktioniert es? Zur tiefergehenden Information über die Hardware verweisen wir auf die entsprechende c't, Ausgabe 9/86, die die technische Seite des Projekts beleuchtet.

Töne, Klänge, Sprache und alles, was der Mensch mit dem Ohr wahrnehmen kann, basiert von der physikalischen Seite aus betrachtet auf demselben Prinzip, nämlich auf der Schwingung von Luftteilchen. Diese versetzen beim Hörvorgang das Trommelfell im Ohr in Schwingungen, und über eine recht komplizierte Umwandlung der Schwingungen in der Gehörschnecke in Nervenimpulse entsteht dann im Gehirn die eigentliche Tonwahrnehmung. Soweit die organische Seite.

Für das Digitalisieren ist vor allem der physikalische Aspekt interessant. Jede Art von Schwingung läßt sich als Überlagerungen von Sinuskurven beschreiben. Leider sind diese Schwingungen in der Natur so komplex, vor allem bei Geräuschen und bei Sprache, daß die augenblicklichen Luftdruckschwankungen nicht mehr dem Verlauf einer Sinuskurve entsprechen. Der Wellenverlauf ist meist sehr kompliziert. Von dem Aussehen einer Welle auf ihren Klang zu schließen, dürfte in den meisten Fällen jedenfalls sehr problematisch sein.

Wenn Sie ein Mikrofon benutzen oder Musik aus der Stereoanlage mit einem Recorder aufnehmen, hat bereits ein wichtiger Umwandlungsprozeß stattgefunden. Die Schwankungen der Luft werden jetzt durch elektrische Spannungsschwankungen dargestellt, wobei diese Schwankungen analoge Werte darstellen, mit denen der Computer nichts anfangen kann, weil dieser bekanntlich mit digitalen Werten arbeitet.

Hier vermittelt nun die Hardware eines Soundsamplers, wie er auf der Multifunktionskarte für den C64 zu finden ist: Der Analog/Digital-Wandler ist in der Lage, die Spannungsschwankungen in digitale Werte mit einer Bandbreite von 8 Bit umzusetzen. Das bedeutet, daß die abgelesenen Werte einer Skala mit 256 Abstufungen zugeordnet werden.

Diese Werte werden in einem Register abgelegt; dies ist ein kleiner Zwischenspeicher, der genau ein Byte, eben diese acht Bits, festhält, bis sie ausgelesen werden. Ein ent-

sprechendes Programm liest den Wert von dort aus und speichert ihn ab. Anschließend wird der Wandler wieder gestartet und ein neuer Wert ins Register geschrieben. Dabei entspricht ein Wert von \$80 (128) der Nulllage, er entsteht, wenn kein Signal anliegt. Je mehr ein Wert von \$80 abweicht, um so größer ist der momentane Abstand der Klangkurve von der Nulllage. Bei einer entsprechend hohen Lesefrequenz läßt sich ein recht getreues Abbild eines Naturklanges durch Zahlenwerte im Speicher eines Computers erreichen.

Mit dieser Technik können Sie einen sonst flüchtigen Klang wie das Zuschlagen einer Tür „dingfest“ machen. Über den Soundchip oder den Wandler kann dieser Klang dann wiedergegeben werden. Sie haben den Klang ja nun greifbar im RAM vor sich, was es erlaubt, die großen Zahlenmengen vor der Wiedergabe zu bearbeiten – eine Aufgabe, für die der Computer denkbar gut geeignet ist.

## Der gute Ton

Der Weg zu einem guten Klang ist weit. Viele Parameter beeinflussen die Qualität einer Aufnahme. Ein wesentlicher Punkt ist das Aufnahmeformat. Beste Voraussetzungen haben Sie mit dem 8-Bit-Format. Ein solches Sample kann allerdings nur mit dem Digital/Analog-Wandler über die Multifunktionskarte wiedergegeben werden. Wollen Sie den SID zur Wiedergabe benutzen und somit Samples auch unabhängig von einer Hardware-Erweiterung abspielen können, müssen Sie sich mit dem 4-Bit-Format begnügen. Der vorhandene Speicherplatz verdoppelt sich bei dieser Methode jedoch gegenüber der höheren Auflösung. In einem Byte sind dann zwei Klangwerte enthalten. Der zeitlich erste Wert befindet sich im unteren Nibble, in den unteren vier Bytes, der nachfolgende Wert im oberen Nibble, entsprechend Bit 4–7.

Ein weiteres Kriterium ist die Aufnahmegeschwindigkeit. Sie wird über die Spalte ZT geregelt. Erhöhen Sie den Wert und spielen den Puffer dann noch einmal ab, sind zwei Effekte die Folge:

- Der Sound klingt tiefer.
- Die Soundwiedergabe dauert länger.

Der Verlängerungseffekt gilt auch für die Aufnahme. Wollen Sie also längere Klangpassagen aufnehmen, ist dies durch Heraufsetzen

des ZT-Wertes möglich. Einen Haken hat dieser Trick allerdings. Je höher der Wert ist, um so geringer wird die Auflösung und damit die Klangqualität. Genauer läßt sich diese Regel leider nicht präzisieren, da die Qualität sehr stark von der Natur des Klanges abhängt. Ein sehr komplexer Klang erfordert meistens eine hohe Aufnahme Frequenz und damit einen niedrigen Timer-Wert. Hier hilft nur experimentieren. Der Effekt der unterschiedlichen Wiedergabelänge macht sich vor allem im Klaviermodus bemerkbar. Dieses Problem tritt allerdings auch bei professionellen Samplern auf, die zum Teil sehr viel mehr Geld kosten als die Kombination C64, Floppy und INPUT 64.

Bei Klängen, die einen gleichmäßigen Verlauf haben, wie dies zum Beispiel bei einem Sinusklang der Fall ist, kann das Problem umgangen werden, indem Sie die LOOP-Funktion anwenden. Sie müssen dann nur darauf achten, daß die Klangwerte am Pufferanfang und Pufferende sich annähernd entsprechen. Ansonsten hören Sie nach jedem Zyklus ein Knackgeräusch.

Über die Aussteuerungsanzeige können Sie die richtige Lautstärke für die Aufnahme kontrollieren. Unter Umständen kann eine leichte Übersteuerung das Ergebnis verbessern. Wenn Sie das Signal von einer Anlage aufnehmen, sollten Sie die Bässe herunter und die Höhen aufdrehen, da eine Baß-Anhebung meist sehr kräftige Signale erzeugt, was leicht zu einer Übersteuerung führt.

Enthält ein Sample Hintergrundrauschen, kann dies zum Teil mit dem Grafikeditor wieder beseitigt werden. Rauschen oder Knacken äußert sich in der Grafikanzeige meist als „Ausreißer“ aus der sonst geschlossenen Wellenlinie. In nicht so schwerwiegenden Fällen kann so ein Störgeräusch wieder zu rechtgemacht werden. Sie sollten sich allerdings bewußt machen, daß der jeweils sichtbare Teil des Samples nur einen Bruchteil der gesamten Sample-Information darstellt und sich eine kleine Änderung nur gering auf den Klang auswirkt.

Wie Sie sehen, läßt das Experimentieren mit den Samples schon bei der Aufnahme viele Möglichkeiten offen. In einer der nächsten Ausgaben von INPUT werden wir noch einige Tools veröffentlichen, mit denen Sie 4-Bit-Samples in eigene Programme einbinden können.

bergt neben Schalt-Interface, EPROM-Bank und MIDI-Schnittstellen auch einen 8-Bit-/D-Wandler (Analog-Digital-Wandler) und einen 8-Bit-/A-Wandler, der unter anderem zur Synthese von akustischen Informationen verwendet werden kann.

Da Klangsynthese ein relativ großes Speicherreservoir erfordert, mußte für SANDI ein Kompromiß zwischen Bedienerfreundlichkeit und Funktionsvielfalt gefunden werden. Um möglichst viele Funktionen unterzubringen, wurde auf umfangreiche kommentierende Darstellungen auf dem Bildschirm verzichtet. Trotzdem dürfte der größte Teil der Funktionen selbsterklärend sein, die Spezialitäten lassen sich ohnehin viel einfacher im Magazin nachschlagen. Von der aufwendigen Programmierertechnik bekommen Sie als Benutzer nicht viel mit, so liegen zum Beispiel viele Funktionen unterhalb des Kern-ROMs, um Ihnen möglichst viel freien Speicher im Rechner offen zu halten. SANDI kann im Bereich von \$2000 bis \$CF00 (8192 bis 52992) Sounds aufnehmen, insgesamt sind das 44 800 Bytes, also fast 44 kB (43.75 kB). Sie können sich also ganz auf Ihre eigene Kreativität konzentrieren.

## Direkt geregelt

Starten Sie SANDI, befinden Sie sich auf der obersten Benutzerebene. Aus allen anderen Ebenen gelangen Sie stets mit der Taste RUN/STOP hierhin zurück.

Die merkwürdigen Hexadezimalzahlen mögen am Anfang verwirren; letztlich sind sie es aber, durch die die vielen Funktionen von SANDI möglich sind. SANDI verwendet diese Zahlendarstellung, da sich diese mit der Struktur von SANDI sehr gut verträgt und eine einheitliche Darstellung ermöglicht. Sollte Ihnen die Zahl 'AA00' befremdlich vorkommen oder Sie glauben lassen, der Druck sei durcheinandergeraten, empfiehlt sich die Lektüre des Kastentextes „Dezimal Hexereien“.

Jede der 10 Zeilen auf dem ersten Bildschirm präsentiert einen Sound-Puffer, dessen Anfangs- und Endadressen durch die ersten beiden 4stelligen Hexadezimalzahlen

bestimmt wird. Die Endadresse auf dem Bildschirm ist die um eins erhöhte Endadresse des jeweiligen Puffers.

Der Cursor läßt sich wie gewohnt mit den Cursor-Tasten bewegen, ein Verändern der Werte ist durch einfaches Überschreiben möglich. Mit RETURN oder SHIFT-RETURN kann der Wert, an dem der Cursor steht, erhöht oder erniedrigt werden. Wer es bequemer findet, kann den Cursor auch mit dem Joystick in Port 2 bewegen. Bei gedrücktem Feuerknopf ist mit Joystick-Bewegungen nach vorne und hinten ein Erhöhen und Erniedrigen der jeweiligen Werte möglich. Der aktuelle Puffer ist stets derjenige, in dessen Zeile sich der Cursor befindet. Stellen Sie doch einmal zur Einübung für den Puffer 0 den Bereich von \$3000 bis \$4FFF ein (den Endwert müssen Sie dann auf \$5000 setzen). Die weiteren Zahlen lassen Sie erst einmal, wie sie sind.

## Funktionstests

Einige Tasten der normalen Tastatur haben spezielle Funktionen:

Drücken Sie die **Taste 'L'**, gelangen Sie ins Lade-Menü. Mit der **Taste 'R'** können Sie einen kompletten Speicherinhalt ins RAM laden, mit **'K'** holen Sie sich ein Sample-File in den aktuellen Puffer. 'T' erlaubt es, Tabelleninhalte für die Tabelle, die Sie gerade vor Augen haben, zu laden. Die **Taste 'S'** holt Tabellen für den Sequenzer wieder ins Programm. Spaßeshalber können Sie mit 'K' auch Ihr letztes Programm laden und sich das grausige Getöse mit 'P' anhören. Alle anderen Dateien müssen von SANDI selbst stammen und werden deshalb beim Abspeichern über die **Taste 'S'** mit einer Kennung versehen. Beim Laden können Sie das gewünschte File direkt aus dem angezeigten Directory mit dem Cursor auswählen und mit RETURN den Ladevorgang einleiten. Ist das File zu lang oder hat es die falsche Kennung, wird es bei einem Ladeversuch nicht akzeptiert. Benötigt das Directory mehr als einen Bildschirm, können Sie mit der Leertaste weiterblättern.

Die **Taste '@'** hilft Ihnen bei Problemen mit der Diskette. Es erscheint die Aufforderung „DOS:“, wonach Sie die üblichen DOS-Befehle

## Ins Fach

Die Thematik, die SANDI dem Benutzer eröffnet, legt es nahe, auch gleich einige Begriffe aus der Physik zu erläutern, die im Zusammenhang mit Schwingungen und Wellen häufig auftauchen.

**Amplitude:** Die maximal erreichte Distanz einer Schwingung von der Nulllage (größtmögliche Elongation).

**Elongation:** Die momentane Entfernung eines Schwingungspunktes von der Nulllage.

**Frequenz:** Die Anzahl der Schwingungen in einer festgelegten Zeiteinheit. Sie wird gemessen in Hertz (Hz). Eine Frequenz von einem Hertz entspricht einer Schwingung pro Sekunde. Je höher die Frequenz ist, um so höher wird der entsprechende Ton, und umgekehrt. So entspricht zum Beispiel eine Sinusschwingung mit der Frequenz von 440 Hertz dem Kammerton A.

**Periode:** Der sich stetig wiederholende Teil einer periodischen Schwingung.

fehle absenden können, wie zum Beispiel „I“ zum Initialisieren einer Diskette. Die Befehlssyntax finden Sie in Ihrem Floppy-Handbuch. Sie entspricht dem String, der sonst nach der Befehlsfolge OPEN 15,8,15,“ gesendet werden würde, natürlich ohne Hochkomma.

Die **Taste 'P'** stellt die Verbindung zu Ihrem Gehör her. Haben Sie einen Monitor oder Fernseher mit Lautsprecher angeschlossen, wird der mit dem Cursor markierte Puffer abgespielt. Dieser **Play-Modus** ist so lange aktiv, wie die Taste 'P' niedergedrückt bleibt. So sind durch stotterndes Betätigen der Taste 'P' Rap-Effekte möglich, wie sie in der Musik derzeit gern verwendet werden. Auch vom Joystick können Sie den aktuellen Puffer mit abspielen, indem Sie die Kombination **Feuer-Links** benutzen. Abgebrochen wird der Spielvorgang dann mit **Feuer**. Durch Variation des Wertes in der **Spalte ZT** kann die **Geschwindigkeit** und damit auch die **Tonhöhe** der Wiedergabe verändert werden – natürlich ändert

sich hierbei auch die Länge des Samples. Doch auch dazu später mehr.

Im mittleren Kasten finden Sie drei Spalten mit den Angaben **WT**, **A**, **H**, von denen die ersten beiden zusammengehören. Mit der 'P'-Taste sind Stottereffekte nur am Pufferanfang möglich.

## Trickkiste

Geben Sie nun für **H** (= **Handmodus**) den Wert '1' ein und spielen den dazugehörigen Puffer ab ('P' gedrückt halten), können Sie während der Wiedergabe durch Betätigen der **RESTORE**-Taste mitten in der Wiedergabe Stottereffekte erreichen. Je größer der Wert von 'H' ist, um so mehr wird beim Stottern vom Puffer wiederholt. Bei einigen Rechnern ist die Bedienung der **RESTORE**-Taste übrigens nur mit einem kräftigen Schlag möglich. Eine Einheit entspricht einer Menge von 256 zu wiederholenden Bytes.

Die **Spalte A** (= **Auto-Modus**) entspricht der Spalte H, nur daß hier das Stottern nicht von der Hand, sondern vom Programm aus gesteuert wird, was im Vergleich zum Handmodus sehr viel schonender für die Tastatur ist.

Der Wert unter **WT** bestimmt, wie groß die **Abspielzeit** zwischen zwei Wiederholungen sein soll. Je kleiner er ist, desto häufiger wird das Stottern ausgelöst. Der Klang wird quasi im „Pilgerschritt“ (300 Bytes vorwärts, 256 zurück, 300 Bytes vorwärts . . .) wiedergegeben. Mit der richtigen Einstellung sind dann Echo und Halleffekte möglich. Eines ist hier jedoch sehr wichtig: Wird der Wert **WT** im Verhältnis zu **A** zu klein gewählt, werden die Rücksprünge im Puffer sehr häufig ausgelöst, und es kann passieren, daß **SANDI** regelrecht rückwärts durch den Speicher liest (300 Bytes vorwärts, 512 zurück, 300 Bytes vorwärts . . .). Unter Umständen kann dies aber auch zu interessanten Klängen bei der Wiedergabe führen.

Wollen Sie einen Puffer wirklich **rückwärts** anhören, hilft Ihnen die **Taste 'U'**. Diese dreht den Pufferinhalt um. Wenn Sie nun 'P' betätigen, hören Sie den Klang natürlich rückwärts. Richtig eingesetzt eröffnen die

se Funktionen völlig neue Klangdimensionen.

Wollen Sie den Inhalt eines Puffers **löschen**, können Sie dies mit **SHIFT-CLR/HOME** erreichen.

## Vom Ohrenschaus zur Augenweide

Durch Betätigen der **Taste 'G'** erreichen Sie die **grafische Ausgabe** des aktuellen Pufferinhalts – allerdings sehen Sie nur die ersten 320 Bytes des Puffers bedingt durch die Auflösungsgrenze des Videochips im C64. Wenn Sie auf solche Detailbetrachtungen verzichten können und lieber eine Übersicht über den ganzen Puffer haben wollen, hilft Ihnen die **Spalte GR**, deren Inhalt normalerweise „01“ ist. Durch Heraufsetzen dieses Wertes wird die Sprungweite bei der Grafikausgabe erhöht. Wählen Sie hier zum Beispiel „05“, so wird nur das 0., 5., 10., 15., . . . Byte angezeigt. Allerdings erhalten Sie so einen Überblick über den fünffachen Speicherbereich. Ein nochmaliger Druck auf die Leertaste schaltet die Grafik wieder ab. Mit dieser Funktion steht Ihnen nebenbei ein einfaches Speicheroszilloskop zur Verfügung. Statt eines akustischen Signals können Sie ja auch eine andere Meßgröße an den Wandler anschließen und deren zeitlichen Verlauf grafisch betrachten.

Bei der grafischen Darstellung können Sie sehr gut die Qualität einer Aufnahme ablesen. Wenn Sie dort zum Beispiel Schwingungen sehen, denen die Spitzen fehlen, war die Aufnahme höchstwahrscheinlich übersteuert.

Die **Tasten 'I' und 'O'** für „in“ und „out“ beziehen sich auf einen Speicherbereich, der unterhalb des I/O-Bereichs von \$D000 bis \$E000 im RAM des C64 liegt. Sie können dort mit 'I' einen der Puffer für die spätere Wiederverwendung ablegen. Mit 'O' holen Sie sich den Inhalt in den angezeigten Puffer zurück.

In Verbindung mit der oben erwähnten Multifunktionskarte gewinnt die **Taste 'K'** an Bedeutung. Mit ihr wird der **Kontrollmodus** angeschaltet. Vorausgesetzt, ein Signal liegt

am Wandlereingang der Multifunktionskarte an, können Sie jetzt durch Regulieren der Lautstärke (Stereoanlage, Walkman, Tonbandgerät oder ähnliches) und des Trimmers auf der Karte die auf dem Bildschirm sichtbare Aussteuerungsanzeige so einstellen, daß die Farbe Rot (ganz rechts bei monochromen Monitoren) für kurze Augenblicke sichtbar ist, auf keinen Fall jedoch die ganze Zeit. Damit vermeiden Sie Übersteuerungen bei der Aufnahme, die sich bei der Wiedergabe als meist unangenehme Verzerrungen bemerkbar machen.

Nun können Sie das anliegende Signal durch Betätigen der **Taste 'R'** aufnehmen. Um Zeitprobleme zu vermeiden, schaltet **SANDI** während der **Aufnahme** alle Sprites ab. Da der Videochip zu deren Darstellung verständlicherweise zusätzlich Zeit benötigt, ergäbe sich eine Verzögerung der Prozessortätigkeit, was sich bei solch zeitkritischen Verfahren wie Soundsampling hörbar bemerkbar macht. Erscheint der Cursor wieder, ist die Aufnahme beendet.

Der Wert in der **Spalte M** bestimmt, ob es sich im dazugehörigen Puffer um ein 4-Bit-Sample (0) oder um ein 8-Bit-Sample handelt (<>0). Dieser Wert wird stets vom Programm aus gesetzt, unter Umständen ist es aber sinnvoll, diesen Modus nachträglich zu ändern.

Tips zum Erreichen einer guten Klangqualität finden Sie am Ende des Artikels.

Mit **f8** können Sie **SANDI** vom Hauptmenü aus verlassen und in den **BASIC-Editor** gelangen.

## Beziehungskiste

Ein sehr leistungsfähiges Element von **SANDI** sind die **Verknüpfungsmöglichkeiten**. Mit **f1** gelangen Sie in das entsprechende Menü. Die voreingestellten Werte können Sie mit **RETURN/SHIFT-RETURN** ändern. Die Position des Cursors wird mit **CURSOR-UP/DOWN** verändert.

Im ersten Kasten wählen Sie den zu bearbeitenden Puffer (**QU1**: – Quelle 1) und die **Lautstärke (VOL**: – Volume). Bei kleiner werdenden Lautstärkewerten werden die Bits vor der Verknüpfung nach rechts ver-

schoben, wodurch die Intensität bei der späteren Wiedergabe abnimmt.

Im zweiten Kasten wählen Sie die Art der **Verknüpfung**. Sie können hier unter anderem die logischen Verknüpfungen **AND**, **OR** und **EOR** benutzen. Zusätzlich stehen Ihnen die arithmetischen Operationen **ADD** und **SUB** für Addition und Subtraktion zur Verfügung. Die Addition zweier Puffer dürfte

wohl die am meisten benötigte Funktion bei den Verknüpfungen sein, weil sich mit ihr der physikalische Vorgang der Überlagerung von Wellen nachbilden läßt.

Wollen Sie also wissen, wie zwei Puffer gleichzeitig abgespielt klingen, können Sie dies erreichen, indem Sie in einem dritten Puffer die Summe der beiden Ausgangspuffer schreiben. Übrigens wird das Ergebnis

der Addition noch halbiert, um das Problem der sonst entstehenden Übersteuerung zu umgehen.

Eine weitere Funktion ist der Befehl **CPY**, bei dem der unter **QU1** angegebene Puffer in den Zielpuffer kopiert wird. Die dritte Spalte hat bei **CPY** keine Funktion.

Dies gilt auch für die Funktionen **8>4** und **4>8**, die den Puffer **QU1** von einem For-

## Dezimale Hexereien

Die Darstellung einer Zahl als \$AEF8 löst bei Nichteingeweihten gelinde Befremdung aus und liest sich eher wie ein Schreibfehler. Dabei ist das Geheimnis gar nicht so hintergründig, wie es auf den ersten Blick den Anschein haben mag.

Bei der ersten Begegnung fragt man sich natürlich, wieso solch kuriose Schreibweisen überhaupt erst erfunden werden, schließlich sind Generationen von Menschen mit dem bekannten Zahlensystem ausgekommen und haben sich auch dabei oft genug plagen müssen. Der Umgang mit den Ziffern von Null bis Neun ist uns vertraut und erlaubt es einem in kritischen Fällen, sich auch mal selbst auf die Finger zu schauen, immerhin hat man mit seinen beiden Händen einen kostenlosen stromlosen Rechenapparat. Ganz selbstverständlich ist dies aber nicht. Die Inkas hielten es für notwendig, beide Füße mit hinzuzunehmen. Deren Zahlensystem basierte auf der Zahl Zwanzig.

Wie Sie vielleicht bereits wissen, rechnet der Prozessor 6502 im C64 mit 8 Bit und kann so mit Zahlen von Null bis 255 rechnen. Jede Zahl in diesem Bereich wird in einer achtstelligen Zahl mit den Ziffern Null und Eins dargestellt. Leider sind diese Zahlen für unsere Anschauung schwer verständlich, und man benötigt schon ein längeres Training, um 00010001 als 17 zu lesen. Vergleichen Sie einmal die Aufschlüsselung der Zahlensysteme in der Tabelle.

Ein wenig einfacher ist es, wenn diese Zahlen in zwei Vierergruppen zerlegt werden, in sogenannte Nibbles. Mit einer vierstelligen Binärzahl, gebildet aus Null und Eins, läßt sich von Null bis 15 zählen.

0000 = 0  
0001 = 1  
0010 = 2  
0011 = 3  
...  
1000 = 8  
1001 = 9  
...  
1111 = 15

Um diese Zahlen mit nur einer Ziffer darzustellen, fehlen in unserer Zahlenschrift die Zeichen für die Zahlen 10 bis 15. Dieser Mangel wird einfach behoben, indem dafür die ersten sechs Buchstaben des Alphabets verwendet. 'C' stellt in diesem Hexadezimalsystem (16er System) die Dezimalzahl 12 dar. Hier brauchen Sie schon drei Hände, um mitzählen zu können.

In dem Fall, in dem der Prozessor mit Adressen hantiert, muß er auf einen Be-

reich von Null bis 65 535 zugreifen können, was eben genau 64 kB entspricht. Für die Adressierung benutzt der C64 daher zwei Byte, also zwei mal acht Bits, woraus sich rechnen läßt:  $256 * 256 = 65 536$ ; was genau stimmt, da die Adresse Null ebenfalls mitgezählt wird.

Die berühmte Adresse 49152 liest sich binär dargestellt: %1100 0000 0000 0000, hexadezimal: \$C000.

Die Zeichen '\$' und '%' werden oft von Monitor-Programmen oder Assemblern zur Unterscheidung von Hexadezimal- und Binärzahlen verwendet und gehören selbstverständlich nicht zum Zahlenwert.

In der Tabelle von SANDI können Sie mit RETURN und der Kombination SHIFT-RETURN die hexadezimalen Werte an jeder Stelle rauf und runter zählen.

## Systematisches Zählen

Binärsystem		Hexadezimalsystem		Dezimalsystem	
%10100010 00001110		\$A20E		41 486	
Einer	2 <sup>0</sup>	Einer	16 <sup>0</sup>	Einer	10 <sup>0</sup>
Zweier	2 <sup>1</sup>	16er	16 <sup>1</sup>	Zehner	10 <sup>1</sup>
Vierer	2 <sup>2</sup>			100er	10 <sup>2</sup>
Achter	2 <sup>3</sup>			1000er	10 <sup>3</sup>
				10 000er	10 <sup>4</sup>
16er	2 <sup>4</sup>	256er	16 <sup>2</sup>		
32er	2 <sup>5</sup>	4096er	16 <sup>3</sup>		
64er	2 <sup>6</sup>				
128er	2 <sup>7</sup>				
...					
32 768er	2 <sup>15</sup>				

Hochgerechnet wird's durchschaubar

mat in das jeweils andere konvertieren. Geschrieben wird auch hier in den Zielpuffer.

Bei allen anderen Funktionen wird im dritten Kasten entschieden, ob der Ausgangspuffer mit einem weiteren Puffer oder einer Konstante verknüpft werden soll. Wählen Sie einen Puffer, so muß als Wert für die Konstante der Schrägstrich gewählt werden: **KON:/**. Enthält die Konstante einen Zahlenwert, besitzt das Feld **QU2** (Quelle Nummer 2) keine Funktion. Arbeiten Sie mit 4-Bit-Samples, sollten Sie den Wert für die Konstante doppelt (AA zum Beispiel) angeben, da sonst jedes zweite Nibble (4-Bit-Wert) mit einem anderen Wert verknüpft würde. Ein Pfeifton im Hintergrund wäre die Folge.

Im letzten Kasten wählen Sie den **ZIEL**-Puffer, in den das Endprodukt der Verknüpfung geschrieben wird.

Die Berechnung wird mit der Leertaste eingeleitet. Ein grundsätzliches Problem bei Verknüpfungen ist die meist unterschiedliche Länge der einzelnen Puffer. Zu diesem Zweck dienen die Einstellungen **MOD: O/C** (O = One-Shot-Mode / C = Continuos-Mode). Diese Modi kommen dann zur Geltung, wenn ein Puffer kürzer ist als der andere. Im One-Shot-Mode wird am kürzeren Puffer ein simulierter Bereich mit Nullbytes, im Continuos-Mode der Puffer noch einmal an sich selbst angehängt und die Verknüpfung so lange fortgeführt, bis der längere Puffer durchgearbeitet ist. Sollte der Zielpuffer kürzer sein als die Ausgangspuffer, wird der Vorgang dann abgebrochen, wenn der Zielpuffer voll ist. Im One-Shot-Modus wird der Zielpuffer mit Nullen aufgefüllt, wenn der Quellenpuffer kürzer ist.

## Heiße Scheiben

Die **Taste f3** schaltet den **Keyboard-Modus** an. Mit den Tasten, die oben auf dem Bildschirm zu sehen sind, können Sie mit dem Klang des aktuellen Puffers musizieren. Die Frequenzverhältnisse der Tasten entsprechen denen der Tonleiter. Der Wert von ZT entspricht dabei der Note A, die Sie über die Taste H erreichen. Alle anderen Tasten richten sich bei einer Änderung von ZT entsprechend aus. Auf diese Weise ist es mög-

lich, ein Sample zu stimmen! Mit der **Taste '←'** können Sie den Loop-Modus an und abschalten. Bei eingeschaltetem Loop-Modus wird der aktuelle Puffer nicht nur einmal durchgespielt, sondern so lange immer wieder, bis die Taste losgelassen wird. Wenn Sie gleichzeitig **SHIFT-LOCK** einrasten, wird der Puffer kontinuierlich gespielt. Der Puffer kann durch Betätigen der entsprechenden Zifferntaste gewechselt werden.

Mit der **Taste f5** können Sie einen **Editor** aufrufen, der den Inhalt der Klangpuffer grafisch ausgibt und mit einem Cursor die Manipulation der Wellenwerte ermöglicht. Der Cursor wird mit den Cursor-Tasten bewegt, ein neuer Wert kann an der Cursor-Position mit RETURN gesetzt werden. Wer es bequem haben möchte, kann aber auch mit einem Joystick in Port 2 editieren. Der Feuerknopf entspricht dann der RETURN-Taste. Am rechten oder linken Bildschirmrand angekommen, scrollt die Anzeige entsprechend weiter. Die Geschwindigkeit des Cursors kann mit den Zifferntasten in 10 Abstufungen erfolgen. Je größer der Wert ist, um so langsamer reagiert der Cursor auf Ihre Eingabe.

Bei einem 8-Bit-Sample reichen natürlich 16 Zeilen nicht für die Darstellung aus. Es wären 256 nötig. Mit der Leertaste schalten Sie den Zoom-Modus an, in dem auch ein wertegenaues Editieren möglich ist. Diese Funktion ist natürlich nur bei einem 8-Bit-Sample möglich. Im Zoom-Modus ist es dann sogar möglich, nach oben und nach unten zu scrollen.

Die **Taste f7** führt in den **Sequencer**. Es stehen Ihnen hier drei Funktionen zur Verfügung, die Sie mit **CURSOR UP/DOWN** und **RETURN** auswählen können. Wählen Sie zuerst einmal **EDIT**. Sie befinden sich jetzt in einem Editor und können eine Tastenfolge eintippen, wie Sie es auch im Keyboard-Modus machen, um eine Klangfolge zu erhalten. Um den gewünschten Puffer anzusprechen, geben Sie vor jeder Tastenfolge die entsprechende Ziffer ein.

Beispiel:  
0ASDF-41GFSS6KLHG←

spielt die Klänge ASDF aus Puffer Null, gefolgt von einer vier Zyklen dauernden Pau-

se. Dann erklingen die Folgen GFSS aus Puffer Eins, KLHG aus Puffer Sechs und damit Schluß.

Das Zeichen '←' dient als Enderkennung. Das Pfundzeichen als Enderkennung bewirkt, daß die Sequenz zyklisch wiederholt wird, wodurch Endlosmelodien möglich sind.

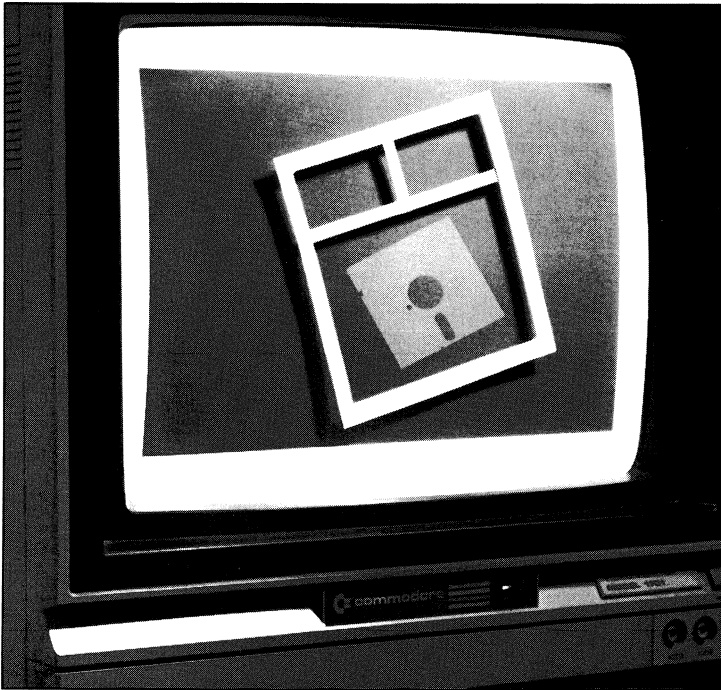
Mit **RETURN** gelangen Sie wieder in die Auswahl **TEST/PLAY/EDIT** und können mit **TEST** die ersten 32 Spielanweisungen sowie mit **PLAY** die gesamte Komposition anhören.

Manfred Sonnenberg/RH

**WIE**  
bekommt man 12mal  
**INPUT 64**  
zum Preis von 10?  
Plus kostenloser Zustellung  
in Haus?

**SO**  
Durch das  
**INPUT 64-Abonnement.**  
198,— 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!**





# Fenster zur Diskette

## Hilfsprogramm: SuperDir

Die Ausgabe des Directory kann in einem vorbestimmten Fenster geschehen. Da der alte Bildschirminhalt zwischengespeichert wird, geht die Bildinformation nicht verloren. SuperDir erlaubt die komfortable Auswahl von File-Namen aus einem Directory. Der ausgewählte Name wird in einer String-Variablen übergeben. Zusätzlich erhalten Sie noch Informationen über den File-Typ und die Blockzahl des Files.

### Oberflächlich betrachtet

Für den Anwender ist die Bedienung von SuperDir kinderleicht. An der entsprechen-

**Daß von BASIC aus das Disketten-Inhaltsverzeichnis nicht so einfach ausgelesen werden kann, hat sich inzwischen sicherlich rumgesprochen. Es gibt bereits eine große Anzahl von Hilfsprogrammen, die diese Schwierigkeiten umschiffen. Wir stellen Ihnen nun das dreieundfünfzigste (oder ist es erst das zweiundfünfzigste?) vor. Aber nicht ohne Grund, wie Sie im folgenden sehen werden.**

den Programmstelle (zum Beispiel beim Menüpunkt „Inhaltsverzeichnis“) öffnet sich ein Fenster, in dem das Disketten-Inhaltsverzeichnis erscheint.

Mit den Cursor-Tasten ↑ und ↓ kann ein File ausgewählt werden, beziehungsweise mit der SPACE-Taste kann gegebenenfalls das Directory innerhalb des Fensters geblättert werden. Die Auswahl des gewünschten Files geschieht mit der RETURN-Taste. Wenn das Directory den gesuchten Eintrag nicht enthält oder wenn aus einem anderen Grund die Funktion nicht mehr benötigt wird, kann diese auch mit der Taste ← abgebrochen werden.

In jedem Fall wird nach Beenden des Menüpunktes das Fenster wieder geschlossen und der Bildschirminhalt wieder hergestellt, der vor dem Aufruf der Funktion vorhanden war.

## Die inneren Werte

Wenn Sie jetzt befürchten, daß so viel Komfort programmtechnisch extrem kompliziert zu steuern und zu verwalten sei, können wir Sie beruhigen. Ihr Programmieraufwand reduziert sich auf die Formulierung eines SYS-Befehls und anschließende Auswertung einiger Variablen.

Hier nun die Syntax für den SYS-Befehl:

**SYS AD,BU,OB,UN,BL,TP,ER,NS**

Diese ersten Parameter bestimmen Sie:

**AD** = Startadresse  
**BU** = Adresse des Bildschirm-Buffers  
**OB** = obere Fenstergrenze  
**UN** = untere Fenstergrenze

Die folgenden Variablen werden vom Programm mit Werten gesetzt:

**BL** = Blockzahl des gewählten Files  
**TP** = File-Typ des gewählten Files  
**ER** = Fehler-Flag  
**NS** = String für File-Name

Es versteht sich von selbst, daß Sie auch andere Variablenamen verwenden können. Die ersten vier Variablen können sogar durch Konstanten ersetzt werden. Vorgeschieden ist nur der Datentyp; so erwartet SuperDir als achte Variable zwingend eine String-Variablen.

## Die Steuerung

Nun zu den Parametern im einzelnen. Nach der Adresse (auf sie gehen wir noch gesondert ein) wird mit dem ersten Wert (BU) die Lage des Buffers für die Bildschirm- und Farbinformationen bestimmt. Er ist 2000 Byte groß und kann überall im RAM-Bereich liegen, also auch unter dem ROM und unter dem I/O-Bereich. Da dieser von den meisten Programmen am wenigsten benötigt wird, bietet er sich als Buffer-Speicher besonders an. Als Buffer-Adresse wählen Sie dann den Wert 53248. Übergeben Sie direkt oder indirekt den Wert 0, wird der Bildschirm nicht gepuffert!

Die Größe des Fensters, das von SuperDir benutzt wird, kann durch die Angabe der obersten (OB) und der untersten (UN) Zeile bestimmt werden.

Die folgenden Variablen(!) dienen der Rückgabe von Informationen. Die Variable (BL) enthält die Anzahl der für das File benötigten Blöcke. Wurde kein File ausgewählt, wird der Wert 0 zurückgegeben.

Der Variablen TP übergibt SuperDir Informationen über den File-Typ. Dieser numerischen Variablen wird der ASCII-Code des ersten Buchstabens des File-Typs übergeben, also

TP = 80 = ASC("P") = PRG  
TP = 83 = ASC("S") = SEQ  
TP = 85 = ASC("U") = USR  
TP = 82 = ASC("R") = REL

Ist ein File nicht ordnungsgemäß geschlossen worden, wird dies im Directory durch ein dem File-Typ vorangestelltes \* angezeigt. Zu dem Wert TP wird dann noch die Zahl 128 addiert.

Die Variable ER enthält hoffentlich stets den Wert 0. Dies zeigt nämlich an, daß die Routine ordnungsgemäß abgeschlossen worden ist. Ansonsten enthält sie die Fehlernummer der Floppy-Fehlermeldung. Ein Wert von 128 zeigt an, daß kein File-Name ausgewählt wurde.

Die letzte Variable (N\$) muß eine String-Variable sein und beinhaltet den File-Namen. Bei einem vorzeitigen Abbruch wird ein Leer-String übergeben.

## Frei und ungebunden

Das Programm SuperDir besitzt eine Besonderheit: Es ist völlig relokatable, das heißt, es ist frei im Speicher verschiebbar.

Das Tool kann somit von BASIC-Programmierer mit anderen Tools wie PRINTAT und INKEY zusammen benutzt werden. Dazu laden Sie zuerst das alte Tool und starten es mit RUN. Nun wird SuperDir mit ,8,0 (!) an den neuen BASIC-Anfang geladen und ebenfalls gestartet. Die Aufrufadresse der anderen Tools hat sich natürlich nicht verändert, die von SuperDir muß noch berechnet werden. Mit LIST erhalten Sie nun eine REM-Zeile, die die Berechnungsvorschrift für die Adresse enthält.

Wenn Sie aus dieser Zeile nur das REM entfernen (die Wertzuweisung an die Variable 'S' also stehenlassen), können Sie im folgenden Programm diese Variable als Startadresse benutzen. Wollen Sie die Vorgabe nicht akzeptieren (zum Beispiel, weil Sie die Variable 'S' für andere Zwecke reserviert haben), können Sie in die Berechnungsvorschrift auch einen anderen Variablennamen schreiben. Sie können die Berechnung auch im Direktmodus durchführen und das Ergebnis dann als Konstante in den SYS-Befehl übernehmen.

Danach können Sie wie gehabt programmieren, und wie immer gilt auch hier: vor dem Abspeichern POKE 44,8:POKE 43,1 eingeben. Frank Börncke/WM



# Lernen im Dialog

## Frage: Französische Grammatik

**Gerade mühsam erworbenes Wissen über die grammatischen Feinheiten einer Fremdsprache geht leicht verloren, wenn nicht regelmäßig in der Sprache geübt wird. Hier haben Sie einen gedulden Trainer vor sich, der jederzeit zu einem Übungsstündchen bereit ist und es Ihnen auch nicht übelnimmt, wenn Sie ihn mittendrin unterbrechen.**

In der ersten Runde können Sie mit diesem französischen Grammatik-Trainer den richtigen Gebrauch von bestimmten und unbestimmten Artikeln üben. Dabei geht es ein wenig um die Welt. Sie sollten mit den Formen

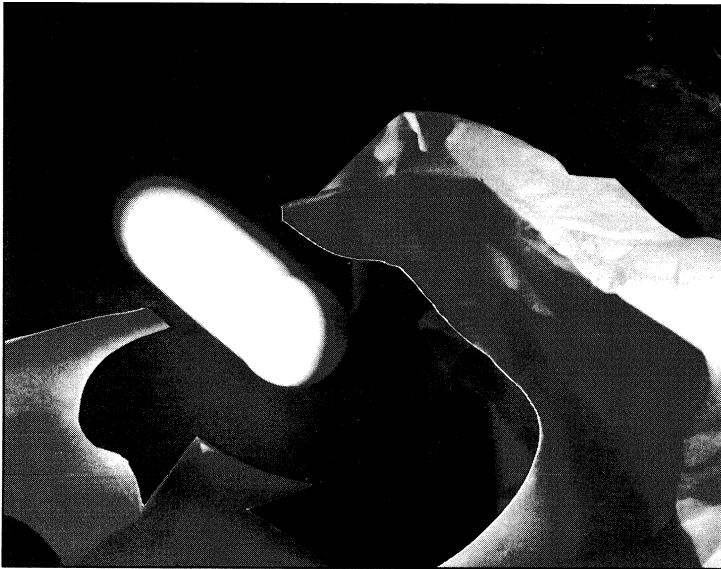
Article défini:  
— le, la, les, l'

Article indéfini:  
— un, une

und natürlich mit einem französischen Grundwortschatz vertraut sein. Wenn mal ein Wort nicht sofort klar ist, wird es wahrscheinlich aus dem Zusammenhang verständlich.

Die Bedienung ist denkbar einfach: Der Trainer wartet auf Ihre Eingaben. Haben Sie sich vertippt, können Sie mit den Cursor-Tasten für rechts und links und den Editor-Tasten INST, DEL und CLR korrigieren. CLR löscht dabei das ganze Eingabefeld. Sind Sie der Meinung, Ihr eingegebener Text sei richtig, schließen Sie mit RETURN ab. Sind alle Lücken gefüllt, erfahren Sie sofort, ob Sie einen Fehler gemacht haben.

Wohlgemerkt, Frage ist ein Trainingsprogramm. Auch wenn Sie dabei sicherlich etwas lernen, sollten Sie schon Grundkenntnisse mitbringen, selbst wenn diese nach längerer Zeit ein wenig „angestaubt“ sind. Im Training lassen sie sich wieder auffrischen. RH



# Höhlenflieger

## Spiel: Lynx the ball

---

Das Spiel wird normal geladen und mit RUN gestartet. Zum Spielen drücken Sie einfach den Feuerknopf Ihres zuvor im Port 2 angeschlossenen Joysticks, und schon hebt er ab, Ihr freundlicher Helikopter namens Lynx.

Lynx the ball heißt er, weil es sich bei ihm um einen besonderen Hubschrauber handelt: Er ist rund, kugelig mit herzlich lachendem Gesicht, eben nett und erfreulich anzuschauen.

Ganz im Gegensatz zu unserem lachenden Kugelhelikopterchen steht seine furchterregende Labyrinthumgebung. Er startet (Feuerknopf!) und bewegt sich in einem futuristischen Höhlengebirge, dessen Gänge an qualvoller Enge nichts zu wünschen übriglassen und von lauernden Gefahren nur so strotzen.

**Sind Sie schon mal mit einem Hubschrauber geflogen, mit einem echten, versteht sich? Es ist schon ein irres Gefühl. Manchmal weiß man gar nicht wo einem der Kopf steht. Es geht aufwärts, abwärts, links oder rechts, immer sind Sie in eine, manchmal auch zwei Richtungen unterwegs. Es ist lange nicht so wie bei einem Flugzeug, das doch relativ gerade fliegt. Doch bei diesem Spiel fliegen Sie nicht etwa in einem Hubschrauber, nein, denn Sie selbst sind ein solcher.**

## Gefährliche Höhlen

Wenn Sie also zu Lynx' Start auf den Knopf drücken, seien Sie ab sofort keinen Sekundenbruchteil mehr unaufmerksam bis zur glücklichen Heimkehr; denn Sie müssen Lynx sehr sicher, ruhig und genau steuern, damit er mit keinem der Hindernisse zusammenstößt und mit keiner Höhlenwandung, weder oben, unten, links noch rechts kollidiert. Meiden Sie alle Berührungen mit Mauern, Tropfsteinen, Flugkörpern und mit sonstigen Sachen. Bedenken Sie: Sie haben pro Spiel nur ein Leben, also einen einzigen Lynx.

Einmal so richtig angeeckt, und schon ist es vorbei mit dem Fliegen im laufenden Spiel. Statt einmal richtig anzustoßen, genügt für den Exitus Ihres kugeligen Flugmännchens auch ein drei- bis viermaliges leichtes oder ein zwei- bis dreimaliges mittelschweres Anrempeln. Seien Sie also gewarnt und auf der Hut.

Achten Sie zudem auch noch auf die unberechenbar schließenden Schiebetüren, beweglichen Wände und seien Sie besonders vorsichtig, wenn Sie die zahlreichen Lichtschranken passieren, damit Sie kein Laserstrahl trifft.

Für Lynx gibt es aber auch Erfreuliches im Greulichen zu erleben: Der größte Trost für seinen genervten Lenker besteht darin, hoffentlich rechtzeitig die Space Bar, also die Leertaste zu drücken, um somit das Spiel erst mal einzufrieren. Ein paar Momente Atem schöpfen – dann erneut sammeln, den Feuerknopf gedrückt, und weiter geht das nervenaufreibende Flugspiel.

## Hamburger sind überall

Lynx selber hat auf seinem zunächst nach unten, später wieder nach oben führenden Flug durch die Korridore wenigstens eine Waffe zur Verfügung – oder ist's ein Werkzeug? –, er kann (Feuerknopf schnell mehrmals hintereinander drücken!) nach unten und zugleich in Richtung seiner aktuellen Flugbahn schießen beziehungsweise Tropfen spritzen. Dies kann er, nicht um damit jemandem wehzutun, sondern um hinderliche Türen, Wände und Mauern zu perforieren und sie so für ihn durchgängig zu machen.

Aber bitte: den Feuerknopf oft genug tüchtig drücken, da nur leichtes, zaghaftes Tröpfeln hier auch nicht weiterhilft.

Wenn zu Anfang erwähnt wurde, der Spieler solle alle Hindernisse meiden und nirgendwo gegenkommen, dann ist das im Prinzip auch schon richtig, aber mit Ausnahmen: Da gibt es überall im Höhlengang, hier und da immer am Boden verstreut, merkwürdige Gebilde, die man, solange einem nichts Besseres einfällt, am ehesten mit Hamburgern vergleichen kann. Gemeint sind damit aber keine Bewohner einer bekannten Hansestadt, sondern die beliebten Erzeugnisse amerikanischer Eßkultur, sieht halt oben und unten so aus wie ein Brötchen, in der Mitte reines Rindfleisch, würzige Gurken und so weiter, jedenfalls klappen diese Brötchenhälften schnell und rhythmisch auseinander und schließen sich auch so. Jedenfalls erkennt man sie nach dieser Beschreibung ganz gut.

Unser Lynx soll diese Hamburger berühren, aber auch dabei nirgendwo hart aufsetzen. Jede Berührung ergibt mindestens 10 Punkte auf dem mitlaufenden Punktezähler, gut für Leute mit Ambitionen, das High-Score-Board nach oben zu verschieben. In der Scrolling-Erklärung des Spieles werden diese Lynx-Targets übrigens als „Virtuals“ bezeichnet, was soviel bedeuten könnte wie „Lynx soll diese Hamburger-Wesen retten“.

Weitere Gadgets des Lynx-Fluges sind zum Beispiel Gebilde, ähnlich dem Eisernen Kreuz, hier im Spiel vorgestellt als „Power“, was nichts anderes bedeutet, als daß Lynx diese Zeichen anfliegen und berühren soll, damit er bei Kräften bleibt und keinen Kraftstoffmangel erleidet.

## Rudern im dunklen Licht

Die anderen noch zu erwähnenden Zielpunkte sind zweischneidig, und als „zwei, drei Groschen zwischen zwei Bierdeckeln“ zu beschreiben. Berührt er sie, so wird die Schwerkraft im System vorübergehend aufgehoben. Damit muß man sich dann erst einmal zurechtfinden.

Im übrigen ist diese Schwerkraft, solange sie wirkt, der Faktor dafür, weshalb unser Lynx so flatterhaft vorankommt, und auch

der Grund, daß der geplagte, geforderte Spieler ständig wild und doch ruhig mit dem Joystick rudern muß. Ein ständiges Anrudern gegen die Schwerkraft, das zusätzlich streift. Doch auch die Berührung mit dem Anti-Gravity-Gadget macht zu schaffen, am besten einfach ausprobieren!

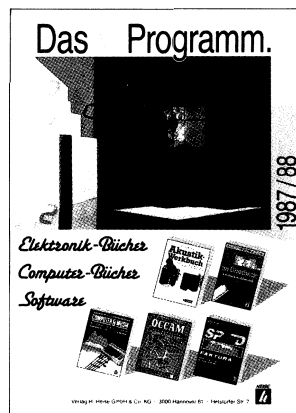
Meiden Sie unter allen Umständen die Zielpunkte, die sich als Dunkellicht-Gadgets bezeichnen lassen. Diese sehen aus wie ein kleiner Stapel aus drei oder vier gleich großen Scheiben. Stößt unser Lynx daran,

ist es um den Durchblick im System geschehen: Das Licht im Höhlengebilde erlischt, nur noch das freundlich lachende Gesicht unseres Flugmaschinchens ist zu erkennen, sonst gar nichts. Da hilft, leichter gesagt als getan, nur eines – den Joystick ganz ruhig halten, bis nach einem Weilchen das Licht wieder von allein angeht und man die vielen Hindernisse klar erkennt. Meiden Sie also diese Lichtschalter, wenn Sie der Haben-Seite auf Ihrem Punktekonto den Vorzug geben.

W. Schmidt-Pabst/kfp

## Hier ist das *neue* Heise- Buch- und Software- Programm 1988

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

Erbitte ein Exemplar. Meine Anschrift:

# Ist der einzelne chancenlos?

## Chiffrierungsaufgabe schon wieder gelöst

Aus den vielen Zuschriften entnehmen wir, daß, trotz des so einseitigen Spielverlaufes, die Knobeleyen bei Ihnen auf ein großes Interesse stoßen. Neben dem abstrakten Thema bleibt doch zuerst die Herausforderung für jeden einzelnen, denn wenn sich alle hinter allen verstecken (die anderen werden das schon machen), hat das Spiel seinen Sinn verloren.

### Die Lösung für alle

Die Dechiffrierung der Aufgabenstellung aus der Ausgabe 12/87 können nun alle innerhalb von INPUT 64 verfolgen. Außerdem können Sie insgesamt vier kleine Programme zum Chiffrieren und Dechiffrieren von A. Hüwe aus Ratingen auf Ihren Datenträger überspielen und anschließend analysieren.

Mit diesen dokumentierten Programmen (innerhalb von INPUT 64 mit "code1" bis "code4" bezeichnet) haben Sie auch gleichzeitig ein kleines Beispiel für eine datei-orientierte Verarbeitung von Daten auf der Diskettenstation.

Aus den vielen Einsendungen wurden mittels Losentscheid die folgenden fünf Gewinner ermittelt: Reinhold Böhm, 8500 Nürnberg; Frank Scholz, 4800 Bielefeld; Bettina Klemm, 6000 Frankfurt; R.-F. Hufner, 7541 Straubenhardt; H.-D. Rocke, 4650 Gelsenkirchen. Den Gewinnern unseren herzlichen Glückwunsch!

### Auf ein neues

Die neue Aufgabe wurde von G. Philippe aus Petange in Luxemburg entwickelt. Die üblichen vier Code-Folgen finden Sie auch

**So hatten wir uns diese Serie eigentlich nicht vorgestellt. Wir müssen Ihnen nun schon zum vierten Mal das gleiche Spielergebnis mitteilen: „Der Code ist geknackt!“**

diesmal wieder in einem kleinen Programm, das Sie aus INPUT 64 abspeichern können. Dieses Programm kann auch als Rahmenprogramm für Ihre Versuche dienen, den Kodierungs-Schlüssel zu knacken.

Den „Klartext“ der ersten beiden Code-Folgen können Sie durchaus als ernstgemeinten Hinweis auffassen. Wir sind sicher, daß Ihnen diese Chiffrierung trotz dieser Hilfestellung einiges an Kopfzerbrechen bereiten wird.

Abschließend noch die obligatorische Aufforderung an Sie, uns Ihre selbstentwickelten Chiffrierungs-Programme einzusenden. Sie sollten sich auch den Hauch einer Chance nicht entgehen lassen. Vielleicht haben Sie ja das nötige Quäntchen Glück!

WM

### Die Spielregeln

Mit dieser Ausgabe stellen wir Ihnen die fünfte Dechiffrierungs-Aufgabe. Für alle neu hinzugekommenen INPUT-Leser hier noch einmal die Spielregeln: Wir geben jedesmal eine chiffrierte Code-Folge vor und erwarten von Ihnen nicht mehr und nicht weniger als den Klartext, der sich hinter dieser Code-Folge versteckt. Um Ihnen die Aufgabe zu erleichtern, erhalten Sie mehrere Code-Folgen (natürlich mit dem gleichen Algorithmus kodiert), wovon wir Ihnen für zwei auch gleich den Lösungstext mitliefern.

Natürlich können Sie auch etwas gewinnen. Wenn Ihr Chiffrierungs-Programm (Sie können also auch ein Programm einschicken) innerhalb von vier Wochen nicht geknackt wird, bekommen Sie ein INPUT 64-Jahresabonnement. Sie spielen

also alleine gegen 100 000 INPUT-Leser. Sollten – was bisher jedesmal der Fall war – in dieser Frist richtige Lösungen bei uns eintreffen, verlosen wir unter den Einsendern fünf Bücher. Der Programmentwickler geht dann leer aus.

Noch eine Bemerkung zu den Chiffrierungs-Programmen. Es sollte theoretisch auch die Möglichkeit bestehen, den Code zu knacken. Eine Verknüpfung beispielsweise mit dem Geburtstag des Autors (oder mit anderen externen Zahlen oder Texten) ist zwar ein sehr wirkungsvoller Kodier-Schlüssel, bietet aber kaum Ansatzpunkte für eine Dechiffrierung. Also: alle Informationen, die für die Dechiffrierung benötigt werden, müssen als Code irgendwie und irgendwo in der gesamten Code-Folge auch enthalten sein.



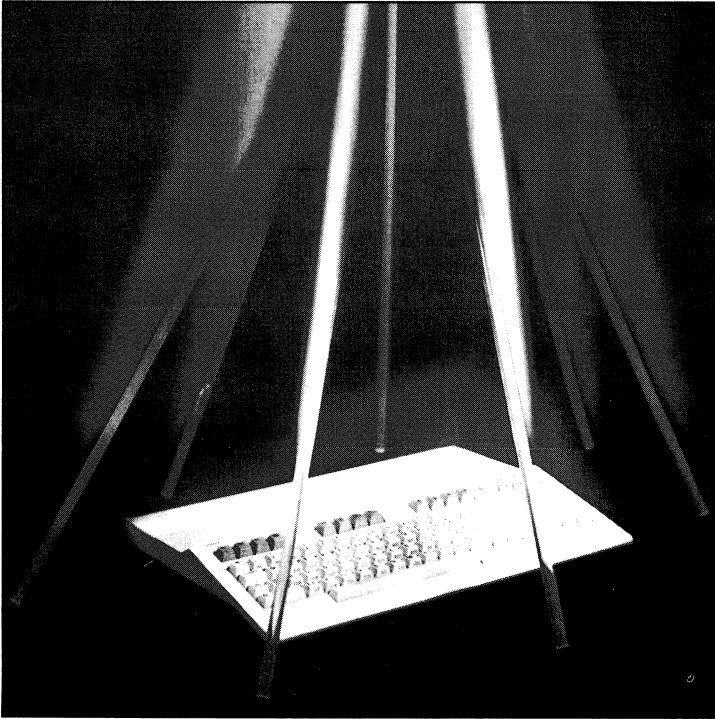
# INPUT 64 SPECIAL 1

## Englische GRAMmatik

- Ein interaktives Lernprogramm für den Commodore 64
- 800 Übungssätze in 10 Trainingseinheiten.  
Aus dem Inhalt: „If-clauses“, „Reported Speech“, „Irregular plural forms“, „Personal and reflexive pronouns“, „Will-future and going-to-future“, „Prepositions of place, movement and time“, „Question tags and short answers“ u.v.a.m.
- Erstellen eigener Übungseinheiten möglich
- Im stabilen, vierfarbigen Umschlag
- Für 19,80 DM ab 11. März 1988 im Bahnhofsbuchhandel und direkt beim Verlag.

Diese INPUT 64-Sonderausgabe ist eine Zusammenfassung aller in INPUT 64 erschienenen Folgen dieses Lernprogramms. Erweitert um Druck-Optionen und die Möglichkeit, eigene Übungen zu erstellen.

**Bestelladresse:**  
**Verlag Heinz Heise GmbH & Co KG**  
**Helstorfer Straße 7**  
**3000 Hannover 61**  
 (Nur gegen Verrechnungsscheck)



# Zugriff im Hintergrund

## Der C128-Analyzer

Der C128-Analyzer ist in zwei Teile gegliedert, den Scanner und den Single-Stepper. Beiden Programmen ist gemeinsam, daß, während im „Vordergrund“ ein BASIC- oder ein Maschinenprogramm abläuft, Sie die Möglichkeit haben – sozusagen im „Hintergrund“ –, sich einen beliebigen Speicherbereich innerhalb der momentanen Speicherbank anzusehen. Der Single-Step-Modus bietet darüber hinaus noch die Möglichkeit, jeden Prozessor-Befehl einzeln abzuarbeiten. Dabei werden, wie bei einem Monitor, alle Register und Flags angezeigt und der aktuelle Befehl disassembliert. Zusammen

**Geplagt von scheinbar unerklärlichen „Hängern“ in Maschinenprogrammen? Genervt von völlig unbeabsichtigten Nebenwirkungen von BASIC-PEEKs und -POKEs? Dann ist der C128-Analyzer Ihr Programm. Denn er ermöglicht es, dem Rechner während des Programmablaufs „auf die Finger“ zu sehen.**

mit dem von Haus aus im 128er eingebauten Monitor hat der Programmierer so ein mächtiges Entwicklungswerkzeug in der Hand. Dieses ursprünglich von Peter Seeliger für den C64 geschriebene und bei uns im Mai 1985 veröffentlichte Analyseprogramm hat Dirk Astrath auf den C128 angepaßt.

## Achtung, Demo!

Innerhalb von INPUT 64 läuft eine Demonstration der beiden Programmteile, die den Umgang mit dem INPUT-Betriebssystem geringfügig einschränkt:

1. Während der Analyser in Betrieb ist, sind die Betriebssystemaufrufe von INPUT 64 (CTRL-H, CTRL-I und so weiter) nicht möglich. Erst dann, wenn wieder eine normale Textseite erscheint, kann man die INPUT 64-Funktionen wieder nutzen.
2. Im Gegensatz zum „echten“ C128-Analyzer ist beim schnellen Scrollen nicht die CTRL-Taste, sondern die **C**-Taste zu benutzen. Das Verlassen des Analyzers ist in der Demonstration einfach mit F1 beziehungsweise F3 möglich, es ist also nicht ein gleichzeitiger Druck auf die CTRL-Taste notwendig.

Wenn Sie aus dem Hauptmenü mit '1' den Scanner angewählt haben, sehen Sie ein kurzes BASIC-Programm auf dem Bildschirm, welches eine einfache Aufgabe zu erfüllen hat. Mit F3 starten Sie dies Programm und gelangen in den Analyser, damit Sie sich mit dem Scanner den Speicher ansehen können. Die Ausgaben sind hexadezimal!; am linken Bildschirmrand erscheinen in 8er-Schritten die Adressen, deren Inhalt rechts daneben als Zahlwert und noch einmal im ASCII-Format ausgegeben wird.

## Tiefe Einblicke

Interessant sind die erweiterte Zero-Page und der Stack (\$0000 bis \$033C), der Bildschirm (\$0400 bis \$0800) und der Ein-/Ausgabebereich (\$D000 bis \$DBFF). Ändert sich der Wert einer Adresse, wird er auch sofort auf dem Bildschirm aktualisiert. Ausprobieren: mit **C**-F7 und **C**-F5 die Adresse \$DC00 einstellen, dann die RUN/STOP-Taste drücken. Der Inhalt der Speicherzelle

\$DC01 ändert sich dabei von \$FF nach \$7F, über diese Adresse (unter anderem) kommuniziert der Rechner nämlich mit der Tastatur. Der offensichtlich ständig wechselnde Inhalt der Register \$DC04 und \$DC05 stellt die periodisch herunterzählenden Timer dar.

Oder stellen Sie einmal die Adresse \$0720 ein, das entspricht dem unteren Drittel des Bildschirms, auf den das BASIC-Programm ja nach wie vor „C128-Analyzer“ schreibt. Bei aufmerksamem Hinsehen ist sehr schön zu sehen, wie der Bildschirm scrollt, indem Speicherinhalte verschoben werden. Zurück zur Menüseite geht es mit F3.

Der zweite Teil ist mehr für Maschinensprachler. Wenn mit der Taste '2' in den Single-Step gesprungen wird, ist auf dem Bildschirm ein kurzes Maschinenspracheprogramm sichtbar, das die Rahmenfarbe mit Zufallswerten belegt. Der Single-Step selbst wird mit F1 aktiviert. Sie sehen nun bis auf die letzten Zeilen das gleiche Bild wie beim Scanner. In den unteren Zeilen stehen die Registerinhalte des Prozessors, der momentan bearbeitete Befehl und ein paar Steuerzeichen.

Unter 'PC' finden Sie einen Wert aus dem BASIC-ROM (\$Axxx), dem Kernal-ROM (\$Exxx) oder der Charget-Routine (dieses Programm steht im C64 in der Zero-Page ab Adresse \$0073 und liest die Zeichen aus dem BASIC-Text). Mit der £-Taste wird vom Single-Step- in den Continuous-Mode geschaltet; die PC-Adresse rauscht jetzt munter durch BASIC, Kernal und Zero-Page. Mit der †-Taste können Sie eigene Werte in die Prozessorregister übernehmen, nämlich das oben erwähnte Maschinenprogramm. Der Bildschirmrahmen ändert nun periodisch seine Farbe.

Ihnen stehen jetzt Befehle zur Verfügung, mit denen Sie sich den Speicher ansehen können (F5/F7) und Befehle zur Steuerung des Single-Steppers. 'c' schaltet den Einzelschritt-Modus ein und aus. Mit '\*' wird im Einzelschritt-Modus der nächste Befehl geholt. Der Rückweg ins Menü steht mit F1 offen. Aber: Wenn Sie jetzt ein Farbflimmern sehen, haben Sie den Prozessor im Testprogramm gelassen. Es ist also wichtig, vorher die Registerinhalte mit † wieder zu restaurieren, also auf die BASIC-Routinen zu setzen.

Um die Tools auf dem 128er einsetzen zu können, müssen diese zunächst auf eine

## Die Befehle des C128-Analyzers

TASTE	Funktion	TASTE	Funktion
F1	Einschalten des Single-Step-Modus	CTRL F5	Scrollen nach oben (32 Zeilen)
F3	Einschalten des Scanner-Modus	CTRL F7	Scrollen nach unten (32 Zeilen)
F5	Scrollen nach oben (eine Zeile)	£	Ein-/Ausschalten des Einzelschritts
F7	Scrollen nach unten (eine Zeile)	*	Einen Befehl weiter
CTRL F1	Ausschalten des Single-Step-Modus	=	Setzen des IRQ-Flags unterbinden
CTRL F3	Ausschalten des Scanner-Modus	†	Eigene Werte in die SINGLE-STEP-Register

Adresse	\$9F9	\$9FA	\$9FB	\$9FC	\$9FD	\$9FE	\$9FF
Register	CR	YR	XR	Akku	Status	PC/L	PC/H

eigene Diskette gespeichert werden. Sie können vor dem Abspeichern zwischen der BOOT-Version und der RUN-Version wählen. Letztere hat vor dem Maschinenteil noch einen BASIC-Starter.

## Der Ernstfall

Die RUN-Version setzt voraus, daß vor dem Laden durch DLOAD und Start durch RUN die 40-Zeichen-Grafik ausgeschaltet ist, der BASIC-Anfang also bei \$1C00 liegt. Die BOOT-Version wird mit BOOT oder BLOAD geladen, die Startadresse ist \$2400 (dezimal 9216). Andere BASIC-Programme oder Maschinenspracheprogramme können jetzt nachgeladen werden. Bei BASIC-Programmen ist wichtig zu wissen, daß der BASIC-Start von beiden Versionen des C128-Analyzers auf \$4000 hochgesetzt wird. Maschinenspracheprogramme, die einen BASIC-Start mit 'SYS' besitzen, laufen dann nicht mehr. Ein Löschen der Grafik sollte man unbedingt vermeiden, da dies auch den Analyzer löscht.

Der Analyzer selbst belegt den Bereich ab \$2400 bis \$2B20, außerdem wird ein zweiter Bildschirm angelegt, der bei \$2000 anfängt und bis \$23FF geht und die Zero-Page-Adressen \$F8 bis \$FF benutzt (bei BASIC-Programmen dürften keine Probleme

entstehen). Der Analyzer vergrößert den Common-Bereich auf 16 KByte (bis \$3FFF) und bindet sich in die Interrupt-Service-Routine ein. Der Single-Stepper ist kein Emulator, sondern es handelt sich um einen echten „harten“ Einzelschritt. Dies hat den Vorteil, daß – abgesehen vom höheren Zeitbedarf – keine Abweichungen vom „normalen“ Prozessor-Betrieb vorhanden sind. Der Stepper nutzt die Tatsache aus, daß der Prozessor, wenn er eine Interrupt-Anforderung erhält, den begonnenen Befehl ausführt und dann erst in die IRQ-Routine verzweigt.<sup>2</sup> Im Gegensatz zum Scanner wird im Single-Step die Commodore-IRQ-Routine nicht durchlaufen. Das heißt: kein Cursor-Blinken. Die Tastaturabfrage wird aber eingebunden, um während des Einzelschritts Eingaben zuzulassen.

## Kollisions-Analyse

Der C128-Analyzer wurde so konzipiert, daß er mit BASIC-Programmen konfliktfrei zusammenarbeiten kann. In Maschinenspracheprogrammen muß darauf geachtet werden, daß es zu keiner Adreß-Kollision mit Zero-Page-Adressen oder dem Analyzer selbst kommt. Stackpointer-Manipulation sind mit Vorsicht zu genießen. Der IRQ-Vektor darf nach dem Start des Analyzers nicht mehr verändert werden.

Eine Übersicht der Befehle finden Sie im Kasten. Nach der Initialisierung des Analyzers springt dieser in den eingebauten Monitor. Wenn Sie jetzt mit F1 den Einzelschrittbetrieb aufrufen und mit '\*' Befehl für Befehl abarbeiten lassen, wird die Eingabewarteschleife des Betriebssystems durchlaufen (Ab Adresse \$C25E). Dasselbe geschieht, wenn zuvor der Monitor mit 'x' verlassen wurde und sich der Rechner im BASIC-Direktmodus befindet. Starten Sie ein BASIC-Programm und rufen dann den Stepper auf, können Sie die Verarbeitung dieses Programms im BASIC-Interpreter verfolgen. Für BASIC wichtiger ist aber der Scanner, mit dem während eines laufenden Programms der Speicher untersucht werden kann.

Maschinensprache-Routinen können auf zwei Arten untersucht werden.

Erste Möglichkeit: Den Analyzer starten, mit dem Monitor ein Programm eingeben (oder laden) und starten. Geben Sie beispielsweise mit dem Monitor an Adresse \$F1300 ein:

```
DEC $D020  
BNE $1300  
BEQ $1300
```

und starten dieses kleine Programm in Bank 15 mit dem GO-Befehl (G F1300). Die Farbe des Bildschirmrahmens wird schnell inkrementiert, dem Programm fehlt eine Abbruchbedingung, es läuft endlos weiter. Wenn jetzt mit F1 der Stepper aufgerufen wird, geht das Programm in den Einzelschritt-Modus und kann Befehl für Befehl verfolgt werden. Auf diese Art und Weise ist es möglich, auch sonst schwer aufzufindender Fehler Herr zu werden.

Zweite Möglichkeit: Die Parameter-Register des Analyzers mit Werten versorgen. Ihre Adressen und die zugehörigen Bedeutungen finden Sie im Kasten, diese Werte werden bei laufendem Stepper mit der ↑-Taste in die Register übernommen. Wenn Sie in diesen Adressen PC/L und PC/H auf \$00 beziehungsweise \$13 setzen, können Sie aus dem Monitor heraus den Einzelschrittbetrieb mit F3 aufrufen; mit '↑' das Programm ab \$1300 untersuchen, mit nochmaligem '↑' den Programmzähler wieder auf den Monitor richten und dann mit CTRL-F1 den Einzelschritt-Modus verlas-

sen. Das Kürzel CR in der Tabelle steht für das Konfigurations-Register, dieser Wert wird mit der ↑-Taste nach \$FF00 übernommen (siehe Handbuch Anhang B).

Wichtig ist dabei, daß Sie die Speicherzellen \$9FE und \$9FF (Startadresse Low- und High-Byte) definiert haben, da Sie sonst einen Systemabsturz riskieren. Normalerweise wird der Single-Step für die Dauer des gesetzten IRQ-Flags außer Kraft gesetzt. Wenn Sie auch die Routine zwischen SEI und CLI untersuchen möchten, müssen Sie die --Taste drücken. Dadurch wird das Setzen des IRQ-Flags unterbunden. Dieser Modus wird durch ein nochmaliges Drücken der --Taste wieder aufgehoben. Sie haben so die Möglichkeit, den Step-Modus von Ihrem eigenen Maschinenprogramm aus durch ein SEI abzuschalten und durch ein CLI wieder einzuschalten.

## Ende ohne Schrecken

Es gibt mehrere Möglichkeiten, das eigene Programm zu beenden. Die sicherste Möglichkeit ist es, ein BRK zu setzen. Sie können aber auch mit der ↑-Taste ins Betriebssystem zurückspringen.

Außer der £-Taste zum Umschalten zwischen dem Step-by-step-Modus und dem kontinuierlichen Einzelschrittbetrieb gibt es

noch einen Trick. Er besteht darin, die STOP-Adresse (\$2545/2546) des Analyzers mit der Adresse zu belegen, bei der der Stepper auf Einzelschritt umschaltet. Wenn Sie eine ROM-Routine verfolgen (zum Beispiel: Fehlerausgabe) und wissen wollen, ob ein bestimmtes Programmstück im ROM (etwa die Druckerausgabe) angesprungen wird, setzen Sie einfach die Startadresse dieses Programmstücks ein. Trifft der Prozessor im Single-Step-Betrieb nun auf diese Adresse, wird der Step-by-step-Modus eingeschaltet, und Sie können in Ruhe die Routine verfolgen.

Um nach dem Start des Scanners noch ein eigenes Programm im Interrupt laufen zu lassen, kann der Sprung im Analyzer zur IRQ-Routine (\$FA65) an Adresse \$25D3/25D4 auf die eigene Routine geleitet werden. Dies ist im Zusammenhang mit dem Stepper nicht möglich D. Astrath/JS

<sup>1</sup> Wenn Ihnen „hexadezimal“ nichts sagt: In dem Artikel zum Programm SANDI in dieser Ausgabe wird dieses Zahlensystem erklärt.

<sup>2</sup> Deshalb funktioniert dieser Analyzer auch mit der in Ausgabe 6/87 unserer Schwesterzeitschrift c't vorgestellten 16-Bit-CPU 65816, allerdings werden dabei anstatt der neuen Befehle nur Fragezeichen ausgegeben.

# **3000 Mark.**

---

**Beim INPUT-64-Programmierwettbewerb.**

---

**Monat für Monat.**

---

**Für Ihre Anwendungen, Werkzeuge, Spiele,  
Animationen und Lernprogramme.**

---

**Interessiert?**

---

**Hinweise für Autoren anfordern!**

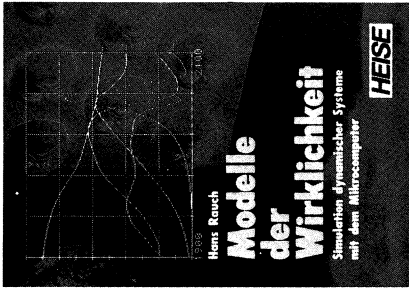
---

**Auch für**

---

# **128er Programme.**

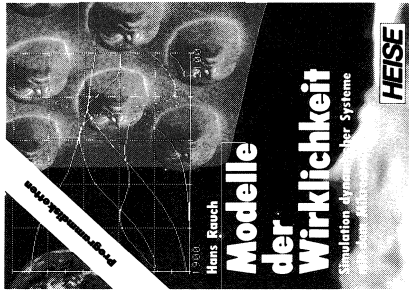
# Wie wirklich ist eine Computer- Simulation?



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



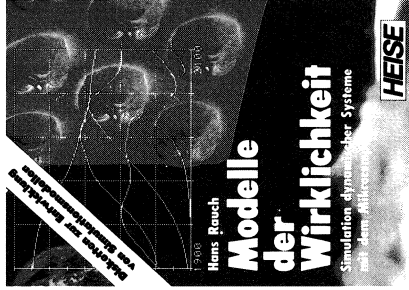
## DI Programmdisketten

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 Programmierung 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: 524/12

BUCH & SOFTWARE



# Musik, Musik, Musik, Musik



## ☛ Studio: Diesmal geht's um Musik

Vier Musikstücke, von 1 bis 4 nummeriert, stehen Ihnen zur Auswahl. Sie können das Stück auch während des Laufes wechseln. Jedes einzelne Stück können Sie auf Ihre eigene Diskette abspeichern. Das funktioniert wie immer mit CTRL und S. Dabei wird das Stück abgespeichert, welches gerade Töne von sich gibt.

Um das Programm von Ihrer eigenen Diskette zu laden und zu starten, geben Sie ein: LOAD "Name",8,1. Das Musikstück wird nun absolut geladen, und zwar an die Speicherstelle 33322 dezimal (\$822A). Mit SYS 33322 wird es dann gestartet. Jedes dieser Musikstücke ist nur ab der angegebenen Adresse lauffähig. Sie können sie also nicht wie ein BASIC-Programm laden und mit RUN starten.

### Besser als Wasser, so gut wie CD

Alle Musikstücke sind neustartfähig, Sie können sie in eigene Programme einbinden,

**Es gibt nicht nur schöne Dinge, die man mit den Augen wahrnimmt, es gibt auch Schönes für die Ohren. Also, dieselbigen aufgesperrt und aufgepaßt, diesmal werden Töne so aneinandergereiht und aufbereitet, daß mancher Hörer staunen wird, was aus dem C64 alles rauszuholen ist.**

abbrechen und mit dem schon erwähnten SYS-Befehl wieder aufrufen. Schreiben Sie einfach in Ihr BASIC-Programm:

```
1 IF A = 0 THEN A = 1: LOAD"Name",8,1
```

Diese BASIC-Zeile lädt ein Programm 'Name' (Musikstück) nach. Wenn Sie die Musik jetzt starten wollen, schreiben Sie in der nächsten Zeile:

```
2 POKE 55, 255: POKE 56, 129
```

```
3 CLR
```

```
4 SYS 33322
```

und schon spielt die Musik.

Um die Musik zu unterbrechen beziehungsweise abzuschalten, erstellen Sie sich am besten ein kleines Unterprogramm:

```
7000 C = 56333
```

```
7010 POKE C,127: REM Interrupt ausschalten
```

```
7020 REM Interrupt-Vektor auf normale Werte setzen
```

```
7030 POKE 788,3*16+1
```

```
7040 POKE 789,14*16+10
```

```
7050 POKE C,128+1: REM Interrupt einschalten
```

```
7060 RETURN
```

Dieses Unterprogramm schaltet den Interrupt ab, setzt die normalen Werte, die auch beim Einschalten des Rechners in den angegebenen Adressen stehen, und schaltet den Interrupt wieder ein.

Alle vier Musikstücke hat uns Michael Winterberg eingeschickt. Können Sie selber komponieren oder liegen schon fertige Titel in Ihrer Schublade, schicken Sie uns doch Ihre Kompositionen einfach zu. In welchem Speicherbereich sie lauffähig sind, ist dabei erst mal nicht von Bedeutung. Das Wichtigste dabei ist, daß Sie die Musikstücke selbst komponiert, und nicht schon vorhandene Titel nur umgesetzt haben. kfp

# Vom Schalten und Halten

## Teil 2: Flipflops und was man damit machen kann

In der Tat sind nur einige wenige Bauelemente nötig, um Schaltungen aufzubauen, die die grundlegenden Funktionsweisen von Speichern und Schnittstellen simulieren. Alle diese Elemente sind in dem Programm Digitest aus der letzten Ausgabe von INPUT 64 enthalten. So werden Sie in diesem Artikel nicht nur einige der in der letzten Ausgabe noch nicht beschriebenen Chips von Digitest kennenlernen, sondern gleichzeitig einen kleinen Einblick in die Funktionsweise Ihres C64 erhalten.

### Blitzmerker

Die einfachste Grundschaltung eines speichernden Elementes haben wir schon in der letzten Ausgabe kurz angesprochen: das Flipflop. Dabei handelt es sich um eine Kombination aus zwei NOR-Gattern, bei denen die Ausgänge jeweils auf einen Eingang des anderen Gatters zurückgekoppelt sind. Ein solches Flipflop heißt RS-Flipflop, benannt nach seinen Eingängen R (Reset) und S (Set). In Bild 1 sind das logische Schaltbild und das Symbol dieses Flipflops dargestellt. Die Festlegung, welchen der Eingänge man mit R und welchen mit S bezeichnet, ist übrigens wegen der Symmetrie der Schaltung völlig willkürlich.

Die Ausgänge von Flipflops bezeichnet man üblicherweise mit Q und  $\bar{Q}$ . Das kommt daher, daß  $\bar{Q}$  normalerweise (auf Ausnahmen kommen wir noch) genau den entgegengesetzten Zustand von Q annimmt. Bei einem RS-Flipflop wird mit dem S-Eingang der Ausgang Q gesetzt und mit dem R-Eingang zurückgesetzt beziehungsweise gelöscht.

Das bedeutet, daß beim Anlegen eines High-Pegels am S-Eingang auch der Q-Ausgang diesen Pegel zeigt. Der  $\bar{Q}$ -Ausgang

**Daß ein digitaler Rechner wie der C64 hauptsächlich aus digitalen Bauelementen besteht, ist wohl eine Binsenweisheit. Diese sind letztendlich aus so einfachen Komponenten wie den im letzten Heft beschriebenen Gattern zusammengesetzt. Doch wie kann man solche Elemente dazu benutzen, beispielsweise ein Portregister aufzubauen?**

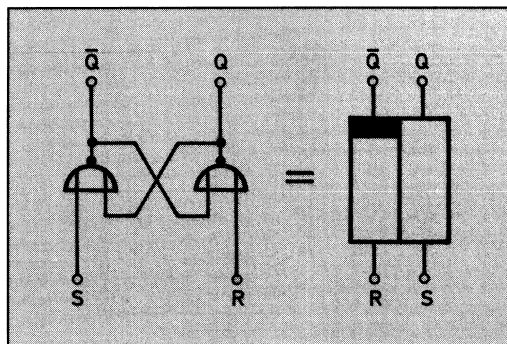
fällt dabei auf 0. Dieser Zustand wird gespeichert: Man kann die Spannung am S-Anschluß so oft wechseln wie man will, die Ausgänge verändern ihre Zustände nicht. Erst wenn man auf den Reset-Ein-

gang eine Spannung legt, schaltet das Flipflop wieder um und hält dann diesen Zustand.

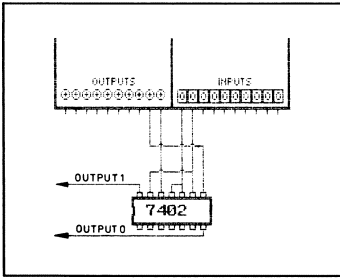
### Ausgeflippt

Wenn oben gesagt wurde, daß die beiden Ausgänge eines RS-Flipflops normalerweise immer entgegengesetzte Zustände haben, so gilt das bis auf eine Ausnahme: Legt man nämlich an beide Eingänge gleichzeitig eine Spannung an, so gehen beide Ausgänge auf Low. Dieser Zustand wird nicht gespeichert. Für ein RS-Flipflop ist diese Eingangskombination sozusagen unzulässig.

Wenn an beiden Eingängen eine Spannung anliegt und man anschließend gleichzeitig beide Leitungen wieder auf Masse legt, dann schwingt ein ideales Flipflop. Diesen Effekt haben Sie bestimmt schon beobachtet, wenn Sie die Schaltung aus Bild 2 (so war der letzte Absatz in dem Artikel in der Februar-Ausgabe gemeint) einmal mit Digitest aufgebaut haben. Die meisten Leser haben sicher bemerkt, daß die Schaltung im letzten Heft auch auf der Seite 9 in der Digitest-Version abgedruckt war. Das Programm legt bei der Initialisierung erst mal



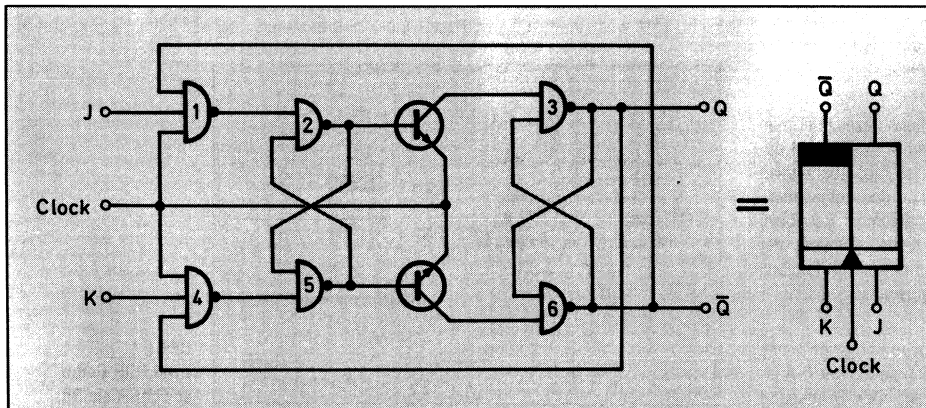
**Bild 1:**  
Zwei NOR-Gatter  
speichern ein Bit.



**Bild 2: So könnte ein selbstgebautes RS-Flipflop aussehen.**

sämtliche Leitungen auf Null. In der Praxis sind übrigens zwei Gatter nie genau gleich schnell, so daß sich bei einem realen Flipflop beim Einschalten immer einer der definierten Zustände einstellen wird, allerdings ist dieser Zustand dann zufällig ausgewählt.

Bild 3 zeigt eine alternative Schaltung eines RS-Flipflops. Statt der NOR- werden hier NAND-Gatter verwendet. Dieses Flipflop verhält sich genau umgekehrt wie das bisher bekannte – darauf sollen die invertiert bezeichneten Eingänge hindeuten. Bei diesem Flipflop ist die Eingangskombination 'beide auf Null' unzulässig. Dabei führen beide Ausgänge High-Pegel.



**Bild 4: Master und Slave bilden ein leistungsfähiges Team.**

## Taktgefühl

RS-Flipflops sind zwar sehr einfach zu realisieren, haben aber ihre Nachteile: Zum einen sollte es in real existierenden Schaltungen nicht die erwähnten 'unerlaubten Zustände' geben. Zum anderen kann es in einer Computerschaltung nachteilig sein, daß ein Register sofort auf eine Änderung reagiert. Man hat es lieber, wenn dieses Umschalten nur zu ganz bestimmten, festgelegten Zeitpunkten stattfinden kann.

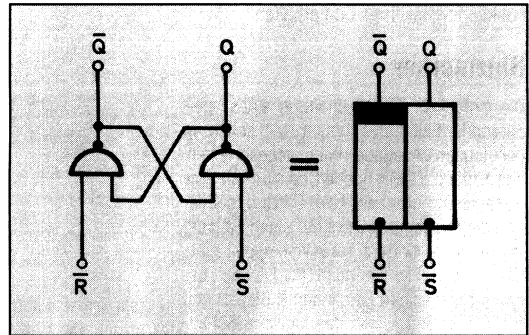
Beide Forderungen erfüllt das sogenannte JK-Master-Slave-Flipflop. Sein Schaltplan und das dazugehörige Symbol sind in Bild 4 gezeigt. Man erkennt, daß diese Schaltung zwei RS-Flipflops enthält: Das eine (das Master-Flipflop) wird aus den Gattern 2 und 5 gebildet, die Gatter 3 und 6 realisieren das Slave-Flipflop. Die Eingänge J und K entsprechen den Anschlüssen S beziehungsweise R eines RS-Flipflops. Der

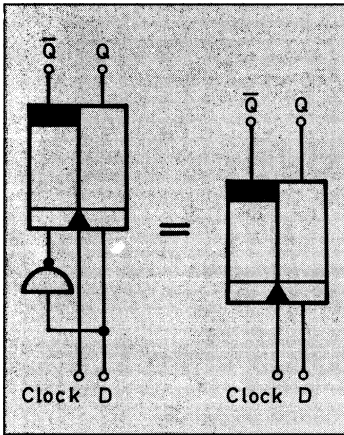
Clock-Eingang dient dazu, das Flipflop mit Taktimpulsen zu versorgen, die für den Umschaltzeitpunkt zuständig sind.

Ein JK-MS-Flipflop arbeitet folgendermaßen: Solange am Clock-Eingang eine Spannung anliegt, gelangen die an den Eingängen J und K liegenden Spannungen nur an das Master-Flipflop. In dem Augenblick, wo der Clock-Eingang auf 0 geht, werden die Gatter 1 und 4 gesperrt, so daß sich der Zustand des Master-Flipflops nicht mehr ändern kann. Gleichzeitig werden die beiden Transistoren leitend, und das Slave-Flipflop übernimmt den Zustand des Masters. Die Ausgänge können also nur genau zu diesem Zeitpunkt ihren Zustand ändern.

Die beiden Leitungen von  $\bar{Q}$  zum Gatter 1 beziehungsweise von Q zum vierten NAND sorgen dafür, daß auch die Eingangskombination  $J=K=1$  sinnvoll ist. In diesem Falle schalten nämlich die beiden Ausgänge mit

**Bild 3: Auch ein RS-Flipflop – aber andersrum.**





**Bild 5:** Auch in Computern geht es manchmal nicht ohne Verzögerung.

jeder negativen Flanke des Clock-Signals um. Aber es ist ausgeschlossen, daß beide denselben Pegel führen oder gar schwingen. Durch diese Eigenschaft kann man aus einem JK-MS-Flipflop sehr einfach einen Frequenzteiler bauen: Man legt die Eingänge J und K auf High und erhält am Q-Ausgang eine Schwingung mit der halben Frequenz des angelegten Clock-Signals.

### Setzen und Löschen

Im Programm Digitest sind zwei Chips enthalten, die aus je zwei JK-MS-Flipflops bestehen. Im 7473 haben die Flipflops noch je einen Clear-Eingang. Dabei handelt es sich um eine Leitung, die intern an je einen zusätzlichen Eingang der Gatter 1, 5 und 6 geführt ist. Legt man diesen Anschluß auf Masse, wird das Flipflop gelöscht, der Ausgang Q führt dann unabhängig vom Zustand der Leitungen J und K Low-Pegel,  $\bar{Q}$  ist auf High. Damit die Flipflops eines 7473 wie oben beschrieben funktionieren, muß an die  $\bar{Clear}$ -Eingänge eine Spannung gelegt werden.

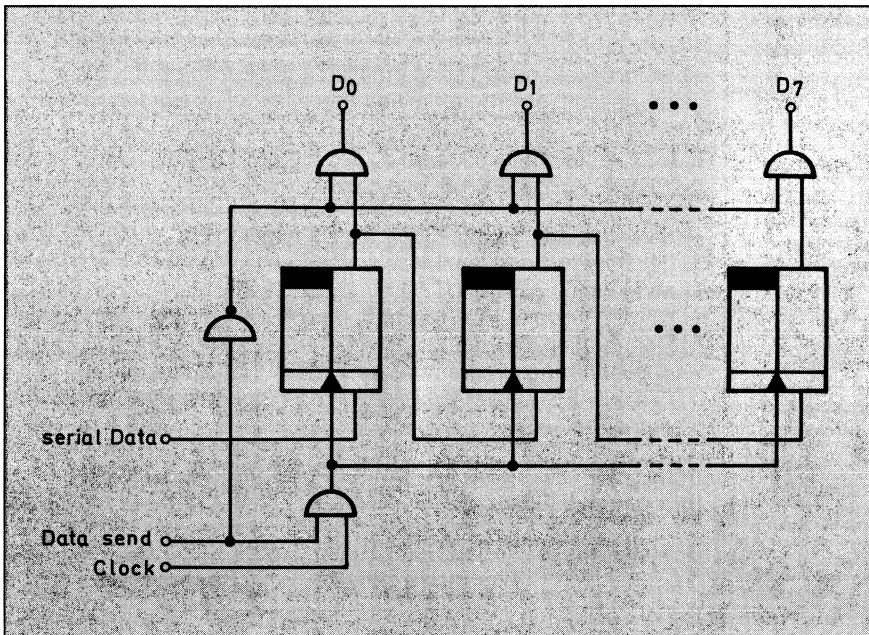
Die JK-MS-Flipflops im 7476 haben noch einen Eingang mehr, der mit Preset be-

zeichnet wird. Er dient dazu, das Flipflop in den entgegengesetzten Zustand zu bringen: Liegt er auf Masse, führt Q High und  $\bar{Q}$  Low. Auch er muß an eine Spannung angeschlossen werden, damit das Flipflop auf die Eingänge J und K reagieren kann.

### Verzögerungen

Durch geeignete Beschaltung der Eingänge (siehe Bild 5) entsteht aus einem JK-Flipflop ein D-Flipflop (Delay-Flipflop = Verzögerungs-Flipflop). Dieses verzögert die am Eingang D anliegende Information um eine Taktzeit. Das heißt, daß der an D anliegende Pegel erst mit der nächsten negativen Flanke am Clock-Eingang zum Ausgang Q weitergegeben wird.  $\bar{Q}$  weist auch bei diesem Flipflop immer den entgegengesetzten Pegel von Q auf.

Man kann durch Hintereinanderschaltung mehrerer D-Flipflops, die alle mit dem gleichen Taktsignal versorgt werden, eine Information um eine genau festgelegte Anzahl von Taktzyklen verzögern. Die Hintereinanderschaltung von D-Flipflops dient in der Computertechnik außerdem dazu, Schieberegister zu realisieren. Diese werden dazu



**Bild 6:** Flipflops sorgen für Schiebung.

benötigt, einen seriellen Datenstrom (beispielsweise an der RS-232- Schnittstelle) in parallele Daten, etwa für die Ansteuerung einer Centronics-Schnittstelle, umzuwandeln.

In Bild 6 ist solch ein Schieberegister als Empfänger für synchrone serielle Datenübertragung vereinfacht dargestellt: Wenn der Sender ein Byte übertragen will, legt er an den Data-Send-Eingang High-Pegel. Die Bits kommen seriell über den Serial-Data-Anschluß, und zu jedem Bit gibt es einen Impuls auf der Clock-Leitung. Wenn alle acht Bits übertragen sind, wird der Data-Send-Eingang auf Masse gelegt. Dadurch können sich die Zustände der Flipflops nicht mehr ändern, und das empfangene Byte wird zu den Ausgängen  $D_0$  bis  $D_7$  durchgelassen.

### (Dig)test it yourself

Für eigene Experimente mit D-Flipflops im Programm Digitest können Sie die Chips 7474 und 74174 benutzen. Der 7474 enthält zwei getrennte D-Flipflops mit zusätzlichen Preset- und Clear-Eingängen, die genau die gleiche Wirkung haben wie bei den JK-Flipflops im 7476. Der 74174 eignet sich sehr gut zum Erstellen eines Schieberegisters oder für lange Verzögerungen. Er besteht aus sechs D-Flipflops mit gemeinsamen Clock- und Clear-Eingängen.

Trotz der prallen Ladung Theorie in diesem Artikel sollten Sie sich nicht davon abhalten lassen, intensiv mit den vorgestellten Chips herumzuxperimentieren. Dank Digitest kann dabei ja nichts kaputtgehen. In der nächsten Ausgabe von INPUT 64 folgt der dritte und letzte Artikel dieser Reihe. Dann werden wir uns die bisher noch nicht erwähnten ICs zur Brust nehmen. Anhand von Beispielen werden wieder praktische Anwendungen gezeigt, die demonstrieren, mit welchem geringem Aufwand man dank der Digitaltechnik überraschend leistungsfähige Schaltungen erstellen kann. Hajo Schulz

#### 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**

## 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 Maschinensprache-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**





# Die Abenteuer des kleinen Geistes

**Spiel: Schloßgeist**

---

Bei diesem Spiel geht es darum, den kleinen Geist durch den großen Schloßkeller zurück in die Freiheit zu führen. Niemand weiß genau, aus wie vielen Räumen das alte Gemäuer besteht, in dem er eingesperrt ist. Ein gemütlicher Spaziergang wird es leider nicht werden, denn überall lauern Gefahren, die es zu bestehen gilt. Die unerwartet auftauchenden Sperrern und Barrieren gehören zu den einfachen Problemen. Die Fledermäuse sind da schon tückischer und lassen überhaupt nicht mit sich spaßen. Deshalb braucht der kleine Geist, der sonst gar keine Furcht kennt, jemanden, der mit ihm zusammen den richtigen Weg sucht und ihn führt. Mit einem Joystick in Port 2 kann man dem kleinen Geist den Weg weisen, oder es geht auch, wenn gerade kein Joystick zur Hand ist, mit der in INPUT üblichen Tastaturbelegung. Mit den Tasten A und Z geht es rauf und runter, mit Punkt und Schrägstrich kommt man nach rechts und links. Mit der Leertaste teilt man dem Programm mit, daß man anfangen möchte.

**Eigentlich ist es ungewöhnlich, daß Geister sich in Not befinden — versetzen diese Schwerenöter doch sonst ihre Mitmenschen regelmäßig in Angst und Schrecken. Ganz anders ist es allerdings mit dem kleinen Geist; nur ganz selten erlaubt er sich einen winzigen Scherz mit einem, und jetzt ist er im Keller seines Schlosses eingesperrt und fürchtet sich ein klein wenig. Wer hilft dem kleinen Geist und findet den Weg aus dem Keller zurück?**

## **Auch Geister lassen Leben**

Der kleine Geist verliert bei jeder Berührung mit den Wänden oder einem Hindernis ein „Leben“, also sollte man Obacht geben, denn auch Geister haben nicht unendlich viele Leben. Oft muß sich der kleine Geist blitzschnell in Ecken und Nischen zwängen, um sich vor den Fledermäusen in Sicherheit zu bringen. In einigen Räumen sind es gleich mehrere, die ihn am Weiterkommen hindern wollen. Der Weg in den nächsten Raum ist durch ein großes Eisgitter versperrt, dieses läßt sich nur mit einem Schlüssel öffnen. Es bleibt also nichts anderes übrig, als zuerst den Schlüssel zu holen. Befindet sich auf dem Weg gerade ein Goldtopf, ist die Freude groß, denn jeder eingesammelte Topf bringt 100 Extrapunkte. Als Gewinn erhalten die ganz Schnellen die noch verbliebenen Bonuspunkte, wenn sich das Gitter öffnet. Aber auch der kleine Geist geht nicht leer aus: für jeden erfolgreich durchquerten Raum bekommt er ein weiteres „Leben“.

pan

# Das Programm im Programm

## Zum Umgang mit BASIC-Unterprogrammen

Daß man das Rad nicht jeden Tag neu erfinden muß, ist eine Binsenweisheit. Daß man ein einmal gelöstes Programmierproblem nicht immer wieder aufs neue programmiert, sollte eigentlich ebenso klar sein. Dies scheint sich aber noch nicht ganz so herumgesprochen zu haben wie die erstgenannte Erkenntnis. Die Zahl der Programme, die nach der Methode „Immerhintereinanderwegprogrammieren“ erstellt werden, ist Legion. Diese Methode wird beschönigend auch die „direkte Methode“ genannt.

Nehmen wir als Beispiel die Aufgabe, den Cursor an eine bestimmte Position auf dem Bildschirm zu setzen. Bekanntlich wird dies mit drei POKE- und einem SYS-Befehl getan (siehe Kasten 1). Die „direkte Methode“

```
POKE 783,0
POKE 781,5
POKE 782,12
SYS 65520
```

**Diese vier Befehle steuern den Cursor**

besteht nun darin, diese vier BASIC-Befehle immer, wenn der Cursor gesetzt werden soll, einzutippen. Das Ergebnis ist in Listing 1 zu sehen. So kann man aus kleinen Problemen große Programme machen.

```
10 REM OHNE UNTERPROGRAMM
20 PRINT CHR$(147)
30 REM FOLGENDE 4 ZEILEN SETZEN DEN
40 REM CURSOR AUF ZEILE 5 UND SPALTE 12:
50 POKE 783,0
60 POKE 781,5
70 POKE 782,12
80 SYS 65520
90 PRINT "BEISPIELMENU"
100 POKE 783,0
110 POKE 781,7
120 POKE 782,8
```

**Eine durchdachte Struktur ist eines der wesentlichen Kennzeichen eines guten Programms. Dazu gehört die Aufteilung des Problems in sinnvolle Teilschritte, die in einzelnen Unterprogrammen gelöst werden können. Was dabei auf dem C64 in BASIC zu beachten ist, zeigen wir Ihnen in diesen Tips.**

Ein Programmierer (eine Programmiererin natürlich auch) macht daraus ein Unterprogramm, das sich mit einer Zeile aufrufen läßt. Listing 2 demonstriert den Unterschied. Die Platzersparnis wird hier noch nicht so deutlich, weil das Unterprogramm recht klein ist. Aber stellen Sie sich einmal vor, die Routine hätte 40 statt 4 Zeilen!

### Hochstapelei

Was passiert bei einem Unterprogramm-Aufruf eigentlich genau? Der entsprechende BASIC-Befehl (GOSUB Zeilennummer)

steht für GOTO SUBroutine und bedeutet "Gehe zum Unterprogramm". Hier besteht eine gewisse Ähnlichkeit zum GOTO-Befehl, denn das Programm wird ab der angegebenen Zeile fortgesetzt.

Damit hören die Gemeinsamkeiten auch schon auf. Nach einem durch GOTO veranlaßten Sprung „vergißt“ das BASIC, woher der Sprung kam. Bei einem GOSUB dagegen legt der BASIC-Interpreter die Adresse des nächsten Befehls, die Rückkehradresse, auf den Prozessor-Stack. Das ermöglicht, am Ende des Unterprogramms die Stelle wiederzufinden, von der aus das Unterprogramm aufgerufen wurde. Der RETURN-Befehl holt die Adresse vom Stack, die GOSUB hinterlegt hat, und setzt das Programm an dieser Adresse fort. Im Magazin finden Sie hierzu eine Demonstration.

Durch diese Technik wird es möglich, ein Unterprogramm im Unterprogramm aufzurufen oder Unterprogramme zu „verschachteln“, wie es in der Fachsprache heißt. Auf die erste Rückkehradresse legt das zweite GOSUB einfach die nächste. Ein folgendes RETURN nimmt zunächst die oberste Adresse vom Stack und kehrt in das erste, unterbrochene Unterprogramm zurück.

```
130 SYS 65520
140 PRINT "PUNKT NUMMER 1"
150 POKE 783,0
160 POKE 781,8
170 POKE 782,8
180 SYS 65520
190 PRINT "PUNKT NUMMER 2"
200 POKE 783,0
210 POKE 781,9
220 POKE 782,8
230 SYS 65520
240 PRINT "PUNKT NUMMER 3"
```

**Listing 1: Lang, unnützlich, umständlich – die sogenannte direkte Methode**

```

REM MIT UNTERPROGRAMM
10 PRINT CHR$(147)
20 GOTO 100
30 REM UNTERPROGRAMM SETZT CURSOR
40 AUF ZEILE Z UND SPALTE S
50 POKE 783,0
60 POKE 781,S
70 POKE 782,S
80 SYS 65520
90 RETURN

```

```

100 REM HAUPTPROGRAMM
110 Z=5:S=12:GOSUB 50
120 PRINT "BEISPIELMENUE"
130 Z=5:S=8:GOSUB 50
140 PRINT "PUNKT NUMMER 1"
150 Z=5:GOSUB 50
160 PRINT "PUNKT NUMMER 2"
170 Z=9:GOSUB 50
180 PRINT "PUNKT NUMMER 3"

```

**Listing 2: Unterprogramme am Anfang des Listings — schnell und strukturiert**

## Schnelle Klarheit

Aufmerksame Leser werden sich fragen, warum das Unterprogramm in Listing 2 so weit vorn steht und deswegen mit einem GOTO-Befehl übersprungen werden muß. Dies ist nicht in der Tölpelhaftigkeit des Beispiel-Erfinders begründet, sondern wohlüberlegt: Der BASIC-Interpreter muß die angegebene Routine nach dem GOSUB-Befehl nämlich erst einmal suchen, und damit beginnt er am Anfang des Programms. Wenn Sie mehrere Unterprogramme haben, die häufig aufgerufen werden, sollten Sie diese deshalb wie in Listing 2 am Anfang des Programms plazieren. Unter Umständen erreichen Sie so eine deutliche Laufzeitverkürzung.

Noch ein Tip: Eine geschickte Verteilung der Zeilennummern erhöht die Übersichtlichkeit. So kann Routine 1 ab Zeile 100, Routine 2 ab Zeile 200 und so weiter stehen. Diese Zahlen können Sie sich besser merken als „krumme“ Zeilennummern wie etwa 127, 3513 und ähnliche.

## Fälscher am Werk

Nun ist es beim Einsatz von Unterprogrammen mit sich gelegentlich abwechselnden GOSUBs und RETURNs nicht getan. Es gibt eine Reihe von Unterprogrammen, die diesen Namen nicht verdienen. Ein Beispiel finden Sie in Listing 3. Dieses Subroutinen verstößt gegen eine der zentralen Regeln modularer Programmierung: Unterprogramme haben genau einen Eingang und einen Ausgang. „Hintertüren“, Zweitausgänge wie in Zeile 1000, sind verboten.

Übertriebener Dogmatismus? Keineswegs. Das Problem ist in der Abteilung „Fehlersuche“ begründet. Die „saubere Lösung“, in

Listing 4 vorgestellt, ist nämlich vernünftig testbar. Ein STOP-Befehl statt des RETURN bricht das Programm ab, und man kann sich Variablenwerte ausgeben lassen und dergleichen. In der in Listing 3 gezeigten Lösung hätte ein zweiter STOP-Befehl in Zeile 1000 gesetzt werden müssen. Das ist umständlich — was verschmerzbar wäre —, vor allem aber fehlerträchtig. Wenn die Routine etwas umfangreicher ist und vor längerer Zeit programmiert wurde, ist der zweite Ausgang in der ersten Zeile schnell vergessen. Das Rätselraten, warum der STOP-Befehl am Unterprogrammende nicht erreicht wird, kann dann leicht zur abendfüllenden Beschäftigung ausarten.

Ähnliches gilt für Zweiteingänge. Wird unser Unterprogramm mal mit GOSUB 1000, mal mit GOSUB 1010 aufgerufen, ist es praktisch unveränderbar. Stellen Sie sich vor, sie kämen eines Tages auf die Idee, daß man diese Routine eleganter und kürzer halten könnte. Etwa so:

```

1000 IF A<>0 THEN A=A+1
1010 RETURN

```

Korrekt aufgerufen, wäre von außen betrachtet kein Unterschied erkennbar. Die Routine addiert '1' zu A, wenn A ungleich Null ist. Der Frevler, der durch den Aufruf GOSUB 1010 den Effekt haben wollte, in jedem Falle A zu inkrementieren, fällt auf die Nase. Darum: Unterprogramm und Hauptprogramm müssen sich aufeinander verlassen können.

## Hand in Hand

Leider hat das C64-BASIC bei der Unterprogrammmanwendung einige Hürden eingebaut. Ein Problem stellt die Parameterübergabe dar. Parameter sind Daten, die der

Routine mitteilen, was Sie genau tun soll. Für die Cursor-Routine in Listing 2 dienen die Variablen Z für die Zeilennummer und S für die Spaltennummer als Speicher für die Übergabeparameter des Hauptprogramms. Zeile 110 zeigt, wie der Aufruf vorbereitet wird. Z bekommt den Wert 5 und S den Wert 12 zugewiesen, worauf nach dem Aufruf der nächste PRINT-Befehl in Zeile 5 und Spalte 12 schreibt.

```

1000 IF A=0 THEN RETURN
1010 A=A+1
1020 RETURN

```

**Listing 3: Ein RETURN zuviel — dadurch schwer zu testen und nur als abschreckendes Beispiel tauglich**

Schön wäre es, diesen Variablen sinnvolle Namen geben zu können, etwa „Zeile“ statt Z oder „Spalte“ statt S. Dies ist aber so einfach nicht möglich, da nur zwei Zeichen signifikant sind. Wer im Laufe eines größeren Programms viele Unterprogramme zu bedienen hat, kann so leicht mit den Variablennamen durcheinanderkommen. Das Unterprogramm sollte natürlich keine Varia-

```

1000 IF A=0 THEN 1020
1010 A=A+1
1020 RETURN

```

**Listing 4: Der Sprung zum RETURN macht's — so muß ein Unterprogramm aussehen.**

Das Zusammenfügen von BASIC-Programmteilen geht auch „zu Fuß“:

1. PRINT PEEK(43)

Ergebnis merken (normal: 1)

PRINT PEEK(44)

Ergebnis merken (normal: 8)

2. LOAD"Unterprogramm",8,0 PRINT PEEK(45)

3. Ist das Ergebnis 0 oder 1:

POKE 43,256+PEEK(45)-2

POKE 44,PEEK(46)-1

NEW

sonst:

POKE 43,PEEK(45)-2

POKE 44,PEEK(46)

NEW

4. LOAD"Hauptprogramm",8,0

POKE(43),gemerkter Wert (siehe oben)

POKE(44),gemerkter Wert (siehe oben)

Die Zeilennummern von „Hauptprogramm“ müssen größer als die von „Unterprogramm“ sein.

FOR-NEXT-Schleifen (siehe 64er Tips 2/88) oder Maschinenprogramme. Wenn nicht mehr genug Platz auf dem Stack für die Rückkehradresse ist, bricht der GOSUB-Befehl mit der Meldung „OUT OF MEMORY ERROR“ ab. Dieser Fehler tritt oft dann auf, wenn ein Unterprogramm sich selbst aufruft. Diese rekursive Technik ist in anderen Programmiersprachen sehr beliebt, aber im C64-BASIC kaum sinnvoll. Bei sonst unbelastetem Stack können Sie immerhin 24 Unterprogramme verschachteln.

Das dürfte auch für komplexe Anwendungen ausreichen.

Wenn Sie jetzt darangehen, sich eine Unterprogrammbibliothek mit den gebräuchlichsten Routinen auf einer Diskette einzurichten, können Sie künftig beim Programmieren viel Arbeit sparen, weil Sie für Teilprobleme fertige Lösungen parat haben. Eine Schwierigkeit wäre noch zu lösen: Wie lassen sich solche Programmteile zu einem bestehenden Programm hinzufügen? Das BASIC V2 des C64 läßt Sie hierbei im Stich. Fast alle BASIC-Erweiterungen<sup>1</sup> bieten aber den Befehl MERGE, der eine Datei an das im Speicher stehende BASIC-Programm bindet.

Sind Sie nicht Besitzer eines entsprechenden Tools, können Sie behelfsmäßig nach dem Kasten vorgehen. Dabei laden Sie zuerst das Unterprogramm in den Speicher. Nach Verändern der BASIC-Zeiger laden Sie das Hauptprogramm, dessen Zeilennummern höher sein müssen als die des Unterprogramms. Zum Schluß fügen Sie in der ersten Zeile noch einen Sprung auf das Hauptprogramm ein.

## Blick über den Zaun

Moderne Compiler-Sprachen wie C oder Pascal haben die Möglichkeiten der Unterprogrammtechnik als wesentlichen Bestandteil integriert. Wie oben schon angedeutet, kommt es der menschlichen Denkweise sehr entgegen, ein größeres Problem zur Lösung in leichter überschaubare Teile zu zerlegen. Umgekehrt gibt es auch die Möglichkeit, aus bekannten Teillösungen ein komplexes Gebilde aufzubauen. In der Software-Technik heißt die erstgenannte Methode TOP-DOWN-, die andere BOTTOM-

UP-Strategie. Für jede Aufgabe wird ein Unterprogramm geschrieben, das dann entsprechend vom Hauptprogramm unter Angabe der Parameter aufgerufen wird. Das bringt große Vorteile in der Testphase, da die einzelnen Unterprogramme völlig selbstständig laufen und auch getrennt getestet werden können. In modular aufgebauter Software stecken dann auch in der Regel weniger „völlig unerklärliche“ Fehler.

In extremen Fällen führt eine durchgehaltene Unterprogrammtechnik beim C64-BASIC an die Grenzen erträglicher Verarbeitungsgeschwindigkeit. Eventuell kann es sich der Programmierer nicht leisten, aus jeder kleinen Aufgabe ein Unterprogramm zu machen, weil der Gesamt Ablauf sich dadurch zu sehr verlangsamt. In solchen – seltenen – Fällen muß ein Kompromiß zwischen Modularisierung und Geschwindigkeits-Optimierung gefunden werden. Eine weitere Schwierigkeit ist, daß dem Commodore-BASIC gegenüber strukturierten Sprachen die lokalen Variablen fehlen, Variablen also, die nur dem Unterprogramm bekannt sind.

## Fensterprogramm

Um Sie zum Anlegen einer eigenen Programmibibliothek zu ermuntern, haben wir für Sie diesmal wieder ein kleines Tool zum Abspeichern. Wenn Ihnen die Fensterrahmen mit Schatten aus den 64er Tips schon immer gefallen haben, können Sie dieses zum Experimentieren als BASIC-Unterprogramm verwenden, das die Fenster in Blockgrafik in wählbarer Größe auf den Bildschirm zaubert. Es benötigt vier Übergabe-Parameter, die die Lage und die Größe des Fensters bestimmen. Dazu dienen folgende Variablen:

Z,S:Zeilen- und Spaltennummer der linken oberen Ecke des Fensters

ZZ: Anzahl der Zeilen im Fenster (Höhe)

SZ: Anzahl der Spalten des Fensters (Breite)

Sollten Sie die Routine mit einem PRINT\*AT-Tool einsetzen, können Sie dieses einbauen und so für mehr Tempo sorgen.

Franz Dreismann/JS

blen des Hauptprogramms verwenden und umgekehrt. Es kommt bei solchen Fehlern oft zu Fehlfunktionen, die schwer erklärbar scheinen.

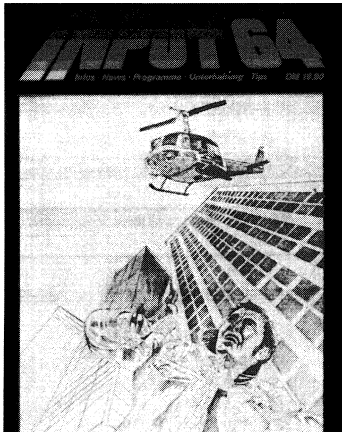
Unsere Cursor-Routine ist da recht harmlos, weil sie keine eigenen Variablen benötigt, aber bei der Routine, die Sie dieses Mal aus dem Magazin abspeichern können, ist das schon anders: dort dient die Variable II als Zähler einer Schleife. Wenn Sie das Programm in ein eigenes übernehmen, in dem II bereits Verwendung findet, werden Sie sich über lauter falsche Ergebnisse wundern. Hier hilft nur Aufpassen und schon beim Programmwurf genau Buch darüber führen, welche Variable wo verwendet wird.

## Nicht zu tief stapeln!

Beim Schachteln von Unterprogrammen stoßen Sie unweigerlich an eine Grenze, die von der Stack-Belastung abhängt. Wie oben erklärt, belegt jeder Aufruf Platz auf dem Stack. Andere Stack-Beleger sind

<sup>1</sup>Bei uns erschienen: INPUT-BASIC (1/86), IMC-BASIC (9/87), PLH (12/87).

**Am 31. März an Ihrem Kiosk:  
INPUT 64, Ausgabe 4/88**



**Wir bringen unter anderem:**

## IFC

Man kann mehrere BASIC-Programme kopieren, indem man das erste lädt, auf eine andere Diskette abspeichert, das nächste lädt, abspeichert und so weiter. Spätestens bei überlangen BASIC-Programmen, Maschinenprogrammen oder sequentiellen Files versagt diese ohnehin umständliche Methode. Mit INPUT-File-Copy ist das alles kein Problem mehr. Kopiert wird bis auf relative Dateien einfach alles, mit frei wählbarer Reihenfolge, allen denkbaren Disk-Kommandos und und und . . .

## Price of Peril

Jetzt ist es soweit: Der 3000-DM-Gewinner des Adventure-Wettbewerbs ist ermittelt. Ein spannendes Abenteuer steht Ihnen bevor, wenn Sie sich auf dieses Spiel einlassen, bei dem es an Ihre Nerven geht. Sie sind der Gejagte und müssen versuchen, eine bestimmte Zeit zu überleben. Die ganze Jagd wird im Fernsehen übertragen, und die Zuschauer geben Ihnen dabei Ratschläge, die Sie sehr genau befolgen sollten . . . Der Einsatz ist hoch: Ihr eigenes Leben.

## Abiturnoten

Mühsames Rechnen und Überlegen ist out. Für alle, die 1988 Abitur machen, gilt: Lassen Sie den Rechner für sich arbeiten, um Ihnen die Entscheidung schneller und vor allem leichter zu machen, ob Sie noch in das „Mündliche“ gehen. Dieses Programm errechnet die Notendurchschnitte, die sich aus den Punkten der einzelnen Kurse ergeben.

## **c't — Magazin für Computertechnik**

**Ausgabe 4/88 — ab 18. März am Kiosk**

Test: XT zum AT aufrüsten — vier Erfahrungsberichte \*  
SUN-4 — Was eine Super-Workstation zu leisten vermag \*  
Programm: Druckertreiber für Atari ST an Brother CE \*  
Karte: Schrittmotoren — wie man sie ansteuert \* u.v.a.m

## **elrad — Magazin für Elektronik**

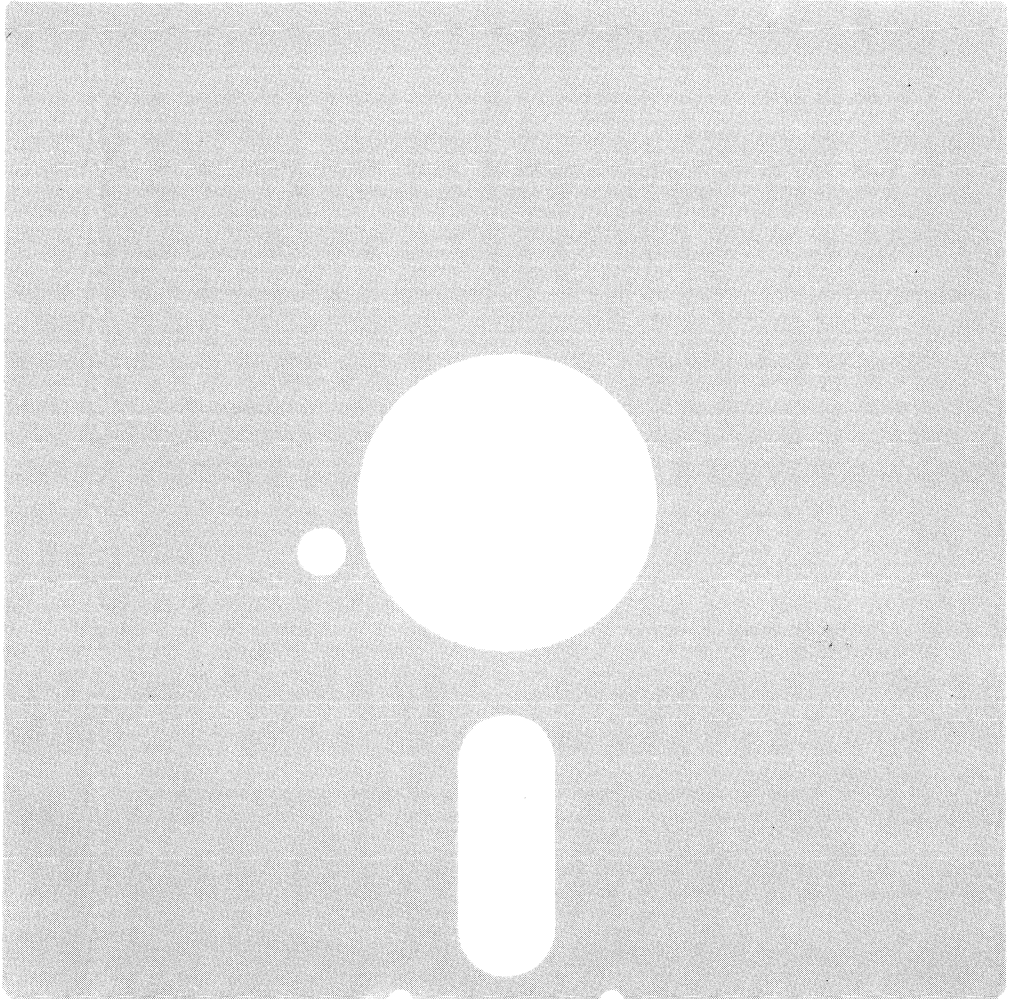
**Ausgabe 4/88 — ab 25. März am Kiosk**

Bauanleitung  $\mu$ P-Technik: DCF-77-Uhr mit acht programmierbaren Schaltausgängen — mit dem Einplatinen-Mikro E.M.M.A \*  
Grundlagen: der Transformator aus anderer Sicht \*  
Bauanleitung Audio: Anpaßverstärker gegen Pegeldifferenzen in Studioqualität \*  
Bauanleitung Bühne: Frequenzshifter und Effektgerät \*  
Die elrad-Laborblätter: IR-Fernsteuerungstechnik \* u.v.a.m





**Bitte zum Entnehmen der Diskette die Perforation  
an den markierten Stellen aufreißen.**



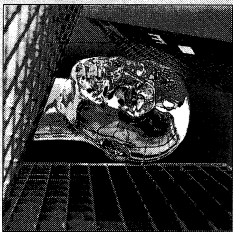
# HEISE



## Künstliche Intelligenz Die aktuelle Computer-anwendung

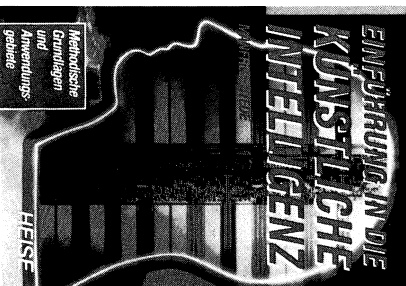
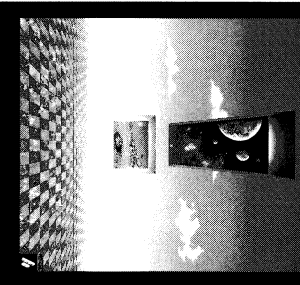
### Künstliche Intelligenz und Musteranalyse

Ulrich Eisenacker



### PASCAL-PROGRAMME zur künstlichen Intelligenz

Manfred Stede



## COMPUTER- BUCH

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

Theoretische Informationen über künstliche Intelligenz werden in konkrete Programme umgemünzt, 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

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 49,80  
Broschur, 267 Seiten



Verlag

Heinz Heise  
GmbH & Co KG  
Postfach 61 04 07

Buch-, Fachhandel oder beim Verlag erhältlich. Kr/2.2