

INPUT 64

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

Unverbindliche Preisempfehlung

BASIC-Übersetzung

Speedcompiler

Systemumrechnung

HexDezBin-Tool

Kugelumleitung

Bouncing Ball

Serien:

64er Tips

Englische GRAMmatik

Rätsellösung

ID-Werkstatt

Spiel:

Chameleon



**Über 140 KByte Software.
Ohne Abtippen.**

Hinweise zur Bedienung

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

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

LOAD "INPUT*" 81 und RETURN

ein. Alles Weitere geschieht von selbst.

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

LOAD "LADER*" 81 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 Systembefehle 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 Hilfseite 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-Druker und kompatible Geräte. Das Programm wählt automatisch die richtige Geräteadresse (4, 5 oder 6) aus. Sie können diese Routine mit der ←-Taste abbrechen.

CTRL und S

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

CTRL und D

Gibt das Directory der eingelegten Diskette

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

CTRL und @

Disk-Befehle senden zum Beispiel Formatieren einer neuen Diskette oder Umbenennen eines Files. Für den zu sendenden Befehls-String gilt die übliche Syntax, natürlich ohne ein- und ausführende 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" 81

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

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

Kennen Sie das Gefühl der Hilflosigkeit, wenn Sie zum ersten Mal in einem unbekannten Automodell sitzen, beim Aussteigen verzweifelt den Türöffner suchen und, wenn Sie schließlich irgendwo einen Hebel gefunden haben, alle vier Himmelsrichtungen ausprobieren müssen, bis sich die Fahrzeuggür endlich öffnen läßt?

So richtig unangenehme Überraschungen können Sie als Fahrer in einem exotischen Auto erleben, wenn statt des gewohnten Blinkers plötzlich die Hupe betätigt wird oder das Licht ausgeht, wenn Sie nur einmal den Lüfter einschalten wollen. Amüsant für Außenstehende wirkt vielleicht auch Ihr Versuch, im dritten Gang anzufahren.

Gut, aber was hat das alles mit Computern im allgemeinen und mit dem Commodore 64 im besonderen zu tun? Was dem einen sein Turbolader, ist dem anderen sein Floppyspeeder, und an Stelle des Katalysators gibt es einen hochauflösenden flimmerfreien Monitor. Nun, sowohl Autos als auch Computer sind technisch hochentwickelte Gebrauchsgüter und benötigen eine differenzierte Einflußnahme durch den Benutzer. Gerade auch auf diesem Gebiet waren die Ingenieure nicht tatenlos und haben uns einerseits automatische Getriebe und Servounterstützung und andererseits Betriebssysteme spendiert, die – wie GEOS – eine grafische Benutzeroberfläche haben. Nur, jeweils vereinheitlicht wurde die Benutzerschnittstelle damit nicht.

Ein wirklich gelungenes Beispiel für eine anwenderfreundliche Eingabeschnittstelle ist die Tastatur des C128. Die bis zu fünf-fach(0) belegten Tasten lassen beim Benutzer sicherlich echte Freude aufkommen, besonders wenn er zwischen den beiden Betriebsmodi öfter hin- und herschalten muß.

Ein anderes weites Feld, Anwender restlos zu beglücken, besteht in der Chance, neue Programme zu schreiben, bei denen bekannte und liebgewonnene Befehlssequenzen eine überraschende Wirkung zeitigen.

Ich könnte mir vorstellen, daß die Funktionstaste 'f1' nicht wie bei der Textverarbeitung XYZ die aktuelle Zeile löscht, sondern (ohne Sicherheitsabfrage, versteht

sich) gleich den ganzen Text, und danach freundlicherweise zum Kaltstart verzweigt. Diese Vorstellung ist übrigens gar nicht so weltfremd, wie sie klingen mag. Wohlverstanden, gemeint sind hier nicht etwa Programm(ier)fehler, sondern von Programmentwicklern „wohlüberlegte“ Benutzerführungen.

Wo bleibt jetzt das Positive, das Konstruktive, werden Sie vielleicht fragen. Erstens muß man nicht in allem etwas Positives finden, und zweitens hat das mit dem Konstruktiven in diesem Fall so seine Schwierigkeiten. Wer kann und wer will schon allen Programmierern haarklein vorschreiben, wie sie ihre Ideen realisieren müssen. Irgendwo lebt die Computerei von kreativen Einfällen, und ein Gängeln (egal in welcher Form) würde sicherlich den einen oder anderen Farbtupfer im Keim ersticken lassen. In diesem Sinne müssen wir wohl alle mit diesem Chaos leben, und – vielleicht ist das ja auch gar nicht so schlimm.

In dieser hoffnungslosen Zeit kann es für Sie ein Trost sein, daß wir nicht beabsichtigen, die Bedienung dieses Magazins zu ändern. Nur Mut! Vertrauen Sie auf die SPACE-Taste und blättern Sie unbesorgt einmal das Inhaltsverzeichnis dieser Ausgabe durch.



Wolfgang Möhle

10/87



INHALT

Leser fragen	2
News	4
Test: Printerface 24-Nadel-Drucker am C64	5
Speedcompiler BASIC-Programme bis zu 20mal schneller	6
Datenfutter Fertige Dateien für INPUT-Calc und INPUT-CAD	14
Adventure gesucht! 3000 DM zu gewinnen im neuen Programmierwettbewerb	17
Englische GRAMmatik Teil 10 Leicht verwechselbare Verben	18
HexDezBin-Tool Zahlenumwandlung als Hintergrundprogramm	18
Bouncing Ball Spiel mit Kugeln und Schleusen	20
ID-Werkstatt Renummer, Kompaktor, Dekompaktor und INPUT-Calc-Patch	22
64er Tips Kreise in hochauflösender Grafik	23
Rätsellösung aus 7/87 Die schnellsten und die kürzesten Kreisalgorithmen	26
INPUT-Calc-Korrektur	28
Chameleon Spiel zur Mustererkennung für ein oder zwei Personen	30
Vorschau	31
Impressum	32

Leser fragen . . .

Laden mit Dolphindos

... kann ich aus INPUT64 abgespeicherte Programme mit meinem Floppy-Speeder nicht laden. Es werden auch ganz merkwürdige Startadressen ausgegeben!

(tel. Anfrage)

Bis zur Ausgabe 9/87 mußten alle Programme, die aus unserem Magazin abgespeichert wurden, mit Sekundär-Adresse Null geladen werden, beispielsweise LOAD"NAME",8,0 oder LOAD"NAME",8. Die zweite Form funktioniert aber nur beim Original-Betriebssystem, das bei fehlender Sekundäradresse automatisch eine Null einsetzt. Die gängigen Hardware-Speeder (Dolphindos, Turbo-Access, Speeddos) sind in diesem Punkt nicht kompatibel, eine fehlende Sekundäradresse wird hier durch eine Eins ersetzt. Ab Ausgabe 9/87 braucht man darauf nicht mehr zu achten, das INPUT-Betriebssystem speichert BASIC-Files jetzt immer mit der Ladeadresse 2049 (\$0801) ab. (d. Red.)

INPUT-BASIC mit Tricks

Drei Fragen:

1) Können Sie mir beschreiben, wie sich INPUT-BASIC (die BASIC-Erweiterung aus INPUT 1/86) mit dem Autostart-Generator (3/85) starten läßt?

2) Läßt sich ein in INPUT-BASIC geschriebenes Programm mit dem Kompaktor (6/85) bearbeiten?

3) Wie läßt sich in einem INPUT-BASIC-Programm die RUN/STOP(+RESTORE)-Taste sperren? Die „üblichen“ POKEs versagen – Programm hängt sich auf oder liefert Unsinn. W. Schneider, Eberhardzell 4

Drei Antworten:

1) Der Autostart-Generator muß mit 'Startadresse+24' statt mit 'Startadresse' angesprungen werden, beispielsweise SYS 49176 statt SYS 49152. Dies gilt nicht nur für INPUT-BASIC, sondern für alle Programme, die nach den sicht- und listbaren BASIC-Zeilen noch Maschinenprogramme, Grafikdaten oder ähnliches beinhalten.

2) Ja, allerdings sollte man die Finger vom LIST-Befehl lassen. Da das Commodore-BASIC die INPUT-BASIC-Befehle nicht kennt, ist sowohl vor als auch nach dem Kompaktieren wenig Sinnvolles zu sehen, oder es kommt sogar zu einem „syntax error“.

3) Die RUN/STOP-Taste läßt sich auch unter INPUT-BASIC durch 'POKE 788,52' sperren. Dadurch erfolgt der Einsprung in die Interrupt-Routine drei Adressen später, und die Abfrage der STOP-Taste wird übergangen. Allerdings zählt dann auch die Systemvariable TI nicht mehr hoch, und TI\$ bleibt stehen. Mal ausprobieren:

```
10 POKE 788,52
20 PRINT TI$:GOTO 20
```

(d. Red.)

INPUT mit Komma

Wie kann man mit dem INPUT-Befehl Strings einlesen, die ein Komma enthalten? Normalerweise ergibt dies nur die „extra ignored“-Meldung, und der Rest der Eingabe wird nicht eingelesen. (tel. Anfrage)

Zur Lösung dieses Problems gibt es zwei Möglichkeiten. Man kann den String in Hochkommata eingeben, also "NAME,VORNAME" statt nur NAME,VORNAME. Das ist für das Lesen von Strings aus Disketten-Dateien praktisch, für Tastatureingaben aber unschön. Hierfür empfehlen wir unser in Ausgabe 11/86 veröffentlichtes Tool INLINE, das auch noch einen PRINTAT-Befehl enthält und obendrein alle anderen Nachteile des INPUT-Befehls beseitigt. (d. Red.)

INPUT-ASS mit POKEs

... habe ich endlich rausgefunden, wie man das gelegentlich auftauchende Textvergessen beim INPUT-ASS-Editor beseitigt: POKE 6258,12. Dadurch wird ein Fehler bei der Behandlung einer Seitenüberschreitung in einer Kopieroutine behoben.

O. Kraus, Forchheim

Stimmt. Die so geänderten Assembler laufen seit drei Wochen in der Redaktion einwandfrei. Bleibt nur noch nachzutragen, daß INPUT-ASS der in Ausgabe 6/86 veröffentlichte Makro-Assembler ist. (d. Red.)

Kritik am System

Im großen und ganzen bin ich ja mit Ihrem Magazin einverstanden. Aber es sind drei Punkte, die mich an Ihrem Betriebssystem stören:


1) Es ist nicht sehr schön, ein Programm dadurch zu verlassen, indem man den Computer aus- und wieder einschaltet. Dadurch wird die Hardware doch nur unnötig belastet. Ihr Betriebssystem ist ja nicht einmal durch einen normalen Reset zu stoppen ...

2) Was ist das für eine seltsame Bildschirm-Hardcopyroutine, die da innerhalb von INPUT64 installiert ist? Kann man diese nicht abändern, da ich sonst Angst habe, daß mein schöner Drucker auseinanderfällt?!

3) Als großer INPUT64-Fan stecke ich jeden Monatsanfang sofort nach Erscheinen in freudiger Erwartung die schöne neue Diskette in mein Laufwerk. Doch spätestens nach dem ersten Versuch, ein Programm auf eine eigene Disk abzuspeichern, meldet

Dienstag ist Lesertag

*Technische Anfragen:
nur Dienstag von 9 — 16.30 Uhr*

 *(05 11) 53 52-0*

sich die Floppy mit einem „disk full“. Ist es nicht möglich, die Directory innerhalb von INPUT64 einzulesen und vielleicht auch noch Befehle zur Floppy zu senden?

Aber ansonsten hätte ich nur noch zu sagen: Weiter so! A. Klak, Beelen

Punkt drei ist mittlerweile erledigt – seit Ausgabe 9/87 sind die Funktionen „Directory“ und „Disk-Befehle senden“ im INPUT-Betriebssystem verfügbar. Eine bessere Hardcopy-Routine setzt die Möglichkeit einer flexiblen Druckeranpassung voraus, die aber scheitert zur Zeit an Speicherplatzproblemen. Aber wahrscheinlich fällt uns dazu noch was ein. Eine „Belastung der Hardware“ durch Aus- und Einschalten des Computers ist unseres Wissens beim 64er nicht gegeben. Außerdem finden wir die neue Lösung (Reset führt ins Inhaltsverzeichnis) als „Notausgang“ sehr gelungen. (d. Red.)

Bingo mit Risiko

Im BINGO-Spiel aus der Ausgabe 8/87 ist dem Autor Andreas Dresbach ein Fehler unterlaufen. Beim Erreichen eines Gewinnstandes von über 32 767 Punkten stürzt das Programm mit der Fehlermeldung „illegal quantity error“ ab. Dies resultiert aus der Verwendung einer falschen Variablen. Das Programm benutzt in den Zeilen 710, 722 und 724 zur Übergabe der gewonnenen Punkte die Integer-Variablen p%. Da in hohen Spielen (zum Beispiel Faktor 9) bei entsprechend hohen Gewinnen der Wert von 32 767 überschritten wird, erfolgt der Abbruch in Zeile 724 bei der Befehlsfolge P%=P%+J%. H. Kalipke, Recklinghausen

Herr Kalipke hatte freundlicherweise auch gleich ein Listing der korrigierten Zeilen beigelegt. Die auf dieser Seite abgedruckten Zeilen stammen allerdings vom Autor des Spiels, den wir auf die Korrektur des Fehlers angesetzt hatten. (d. Red.)

SuperDisk mit Wunschfarben

Ich arbeite jetzt schon sehr lange mit SuperDisk und finde diesen Fastloader sehr gut. (SuperDisk ist der in Ausgabe 1/87 veröffentlichte Schnellloader; d. Red.) Nur: Was müsste am Programm verändert werden, damit man auch Maschinenprogramme mit ‚8,1‘ laden kann? Und: Nach dem Starten wird der Bildschirm schwarz und die Schrift gelb. Wie kann man diese Einstellungen ändern? D. Kutsch, Köln

SuperDisk lädt von Haus aus Maschinenprogramme in der gewohnten Form (mit Sekundäradresse 1), es muß also nichts geändert werden. Die Adressen, die für eine Änderung der Bildschirm- beziehungsweise Schreibfarbe durch ‚POKE‘ neu beschrieben werden müssen, sind je nach Version von SuperDisk unterschiedlich. Die an unterschiedliche Adreßlagen angepaßten Versionen sind in der BASIC-Zeile des Programms vermerkt, und zwar durch die Kürzel \$EO, \$DO, \$AO. Folgende Tabelle gibt Aufschluß über die Änderungsmöglichkeiten:

Version	\$EO	\$DO	\$AO
Schreibfarbe	2353	2389	2351
Schirmfarbe	2342	2378	2340

So führt die Befehlsfolge POKE2353,0:POKE2342,0 dazu, daß in der Version \$EO nichts mehr zu sehen ist, weil mit schwarzen Buchstaben auf schwarzen Hintergrund geschrieben wird. (d. Red.)

Drucken mit Pfiff

Nicht nur beim Punkt „Druckereinstellung“ innerhalb von INPUT-Calc 64/128 (der Tabellenkalkulation aus INPUT 8/87; d. Red.), sondern auch schon bei anderen Software-Produkten ist mir unklar geblieben, was es mit der „Byte-Folge“ auf sich hat, die man seinem Drucker senden soll. Helfen Sie mir weiter? (tel. Anfrage)

Es geht darum, daß einem Drucker nicht nur Zeichen übermittelt werden können, die er zu Papier bringen soll, sondern auch Befehle, die ihm sagen, wie er Zeichen zu Papier bringen soll. Diese Befehle bestehen meist aus nicht druckbaren Zeichen, die nur durch ihren ASCII-Wert beschrieben werden können. Zum Beispiel schaltet CHR\$(14) einen MPS803 auf Breitschrift, die angesprochene „Byte-Folge“ bestände in diesem Fall aus dem einzelnen Wert 14.

Kommandos, die aus mehreren Bytes bestehen, werden meist mit einem festgelegten Wert eingeleitet, in der Regel mit CHR\$(27), auch ESCAPE genannt. Der STAR NL10 etwa wird durch die Byte-Folge 27, 120, 49 in den Schöndruck-Modus gebracht. In BASIC entspräche dies den Befehlen

```
OPEN4,4:CMD4:PRINTCHR$(27);
CHR$(120);CHR$(49);PRINT #4:CLOSE4
```

Welche Kommandos was bewirken, ist dem Druckerhandbuch zu entnehmen. Vergewissern sollte man sich dabei, ob diese Angaben dezimal oder hexadezimal sind. Gegebenenfalls müssen letztere umgerechnet werden, etwa mit dem HexBinDez-Tool in dieser Ausgabe. (d. Red.)

Weiter mit Diskette

Nun ist es soweit, Ihr stellt die Kassettenproduktion ein – leider, wenn ich das hier mal sagen darf. Ich – Neufreak – habe vor eineinhalb Jahren mit der Computerei angefangen. Grundausstattung C64, Datensette, Monitor und Printer.

Bis jetzt war meine Welt in Ordnung, gab es doch INPUT64 auf Kasette für Datensette-Besitzer. Nun werde ich wohl gezwungen sein, meine Ausrüstung um eine Floppy zu erweitern. Denn INPUT64 ist gut, zu gut, um einfach darauf zu verzichten, nur weil es keine Kassetten mehr gibt.

D. Hirn, Seltkant

```
710 gw=(gs(0)+gs(3))*gs(6):gw=gw+gs(5)+gs(1)*gs(7)+(gs(4)+gs(2))*gs(8):gw=gw*mu
711 pi=int(p):p=p+gw-g:g=gw:fori=1to7:p(i)=peek(1064+i)-128:next:j=7:j%=1
722 pp=p-pi:ifj%<100thenifpp/(j*10)=int(pp/(j*10))thenj=j-1:j%=j*10:gosub234
724 i=j:pi=pi+j%:ifpi>pthen736
```

News

Datensicherung gegen Netzausfall

Programmabstürze durch Netzstörungen oder kurzzeitige Stromausfälle soll eine Notstromversorgung für den C64 verhindern, die das Ingenieurbüro Drust anbietet. Zum Lieferumfang gehören ein 220-V-Netzteil, ein Akku und die Anpassung für den C64 und die Floppy 1541. Der Akku ist für einen Netzausfall von maximal 40 Minuten ausgelegt; ein größerer Akku ist lieferbar. Ein quarzstabiler 50-Hz-Generator versorgt die interne Uhr mit den erforderlichen Impulsen.

Die Notstromversorgung wird als Bausatz mit weitgehend vorgefertigten Komponenten zum Preis von 188 DM geliefert. Der Umbausatz nur für eine 12-Volt-Versorgung kostet 50 DM.

Ing.-Büro I.Drust
Darmstädter Straße 77
6103 Griesheim

Computergesteuertes Audio-Meßsystem

Dort wo Frequenzverläufe, Amplituden oder Impedanzen von elektroakustischen Geräten gemessen werden müssen, werden in der Regel elektromechanische Linienschreiber eingesetzt. Dies ist oft eine große Hilfe bei der Fehlersuche, Beurteilung oder Anpassung von Geräten.

Das AMS 64 besteht aus einem Analog-Interface 1652 und der Software 1661. Das Interface wird an den Userport eines C64 angeschlossen. Nach der Aufnahme von Meßreihen werden die Werte in einem Bode-Diagramm auf dem Bildschirm dargestellt. Man kann bis zu drei Kurven für Vergleichszwecke gleichzeitig darstellen. Diese Darstellungen lassen sich nachträglich verändern.

Man kann die Meßreihen mit Namen und Datum kennzeichnen und auf Diskette abspeichern. Die dargestellten Kurven kann man auf einem Drucker entweder als Dia-

gramm oder als Zahlentabelle ausgeben. Das Programm unterstützt alle gängigen Drucker mit Commodore-Schnittstelle sowie den Plotter 1520.

Vom Hersteller wird das Mikrophon 1622 angeboten, das über mitgelieferte Korrekturdaten linearisiert wird. Diese Daten werden an einem hochwertigen Meßmikrophon kalibriert. Das Programm ermittelt über diese Daten die tatsächlichen Pegel. Eine sehr geschickte Methode, einen Meßaufnehmer zu erhalten, der erheblich preisgünstiger ist als ein entsprechendes Meßmikrophon.

Die Hersteller stellen noch einige Programmiererweiterungen in Aussicht, so zum Beispiel zur automatischen Einmessung von Bandmaschinen und Plattenspielern sowie zur Hallzeitbestimmung von Räumen. Erhältlich ist bereits ein Impedanzadapter, mit dem sich die wesentlichen Parameter eines Lautsprechers ermitteln lassen.

KEMTEC
Avenwedder Str. 490
4830 Gütersloh
Tel. 0 52 09/54 29

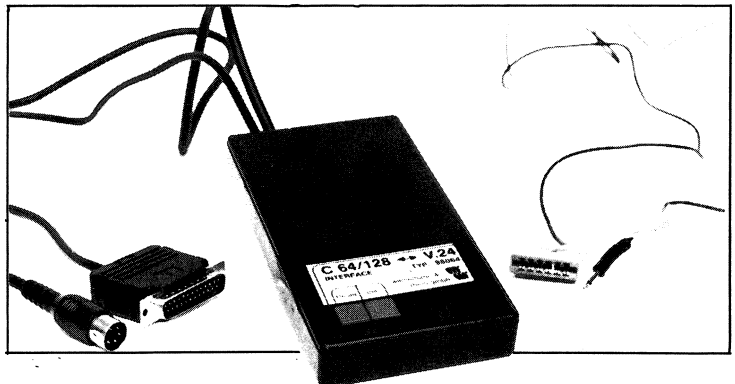
Seriell mit 57,6 Kbaud

Ein serielles Interface, das dem C64 und dem C128 die Kommunikation mit allen Geräten ermöglicht, die über eine RS-232-Schnittstelle verfügen, ist bei der Firma Wiesemann erhältlich. Die etwa taschenbuchgroße Hardware-Erweiterung im stabilen Plastikgehäuse wird über den seriellen Bus – etwa an der zweiten Buchse der Floppy – mit dem Rechner verbunden. Über die Geräteadresse 6 (wählbar) können dann Daten gesendet und empfangen werden.

Die notwendigen Parameter wie Baudrate, Anzahl der Daten- und Stoppbits, Parität und so weiter sind floppy-üblich über einen Befehlskanal, der mit Sekundäradresse 15 eröffnet wird, einstellbar. Bei der Übertragungsgeschwindigkeit steht dem Benutzer die beachtliche Bandbreite von 225 bis 57 600 Baud zur Verfügung. Zwischen Rechner und Interface selbst ist das Tempo des Datentransfers natürlich durch den seriellen Bus auf rund 5 000 Baud beschränkt. Um die schnelle Kommunikation mit der Außenwelt zu ermöglichen, hat das mit einem eigenen Mikroprozessor ausgestützte Interface je 32 KByte Ein- und Ausgabepuffer „on board“.

Software-Handshake über XON/XOFF-Protokoll ist programmierbar, für Hardware-Handshake sind die Leitungen RTS und DTR angeschlossen. Probleme dürfte es mit vorhandener Software geben; die in der Regel auf Geräteadresse 2 die RS-232-Schnittstelle erwartenden Terminal-Programme scheiden aus. Denn die Gerätenummer des Interface ist zwar zwischen 0 und 30 wählbar, nur schickt das 64er Betriebssystem kleinere Adressen als 4 gar nicht erst auf den seriellen Bus. Dafür ist man die Sorgen mit dem nicht normgerechten Commodore-Zeichensatz los: eine Code-Wandlung in Richtung German-ASCII oder IBM-PC-Code ist ebenso einstellbar wie ungewandelter Transfer von Binär-Daten. Im Preis von 298,-DM ist eine für DFÜ-Geübte ausreichende dreisprachige Anleitung enthalten.

C64/C128: V.24/RS-232-Interface
Wiesemann & Theis
Postfach 20 16 05
5600 Wuppertal 2



Printerface

Drucker-Interface für 24-Nadel-Drucker

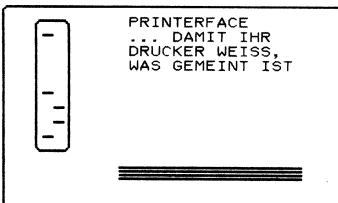
Das Printerface erhebt den Anspruch, universelle Verbindung zwischen Druckern mit Centronics-Schnittstelle und dem seriellen Ausgang der Commodore-Computer VC-20/C64/C128 zu sein. Mit folgenden Merkmalen stellt der Hersteller das Gerät vor:

- Selbsttesteinrichtung
- Auch für Typenradrucker
- Pufferspeicher
- ASCII- und DIN-Zeichensatz
- CP/M-kompatibel
- GEOS-kompatibel
- DIP-Schalter leicht von außen zugänglich
- Druckeranpassung durch mitgelieferte Software auf Diskette
- Einstellung erfolgt durch interaktives Vorgehen
- Es können andere Interfaces simuliert werden

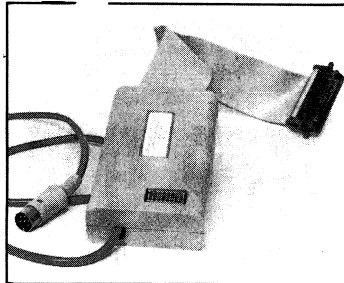
Beim Test in der Redaktion gefiel als erstes das lange Anschlußkabel für den seriellen Bus. Der Drucker kann also ruhig in gebührender Entfernung seinen Dienst versehen.

Die Bedienungsanleitung, im DIN-A5-Format, verfügt zwar über ein kurzes Stichwortverzeichnis, doch leider fehlt ein Inhaltsverzeichnis. In die Funktionskürzel und die Tabellen muß man sich zuerst ein wenig einlesen. Überzeugender wirkt die mitgelieferte Software auf Diskette. Über das Einstellprogramm können Sie die kor-

So sieht das Ergebnis des Einstellprogramms aus.



Die grafischen Möglichkeiten, die Computer inzwischen bieten, haben auch die Entwicklung der Druckertechnologie vorangetrieben. In letzter Zeit wurden von mehreren Herstellern 24-Nadel-Drucker zu erschwinglichen Preisen auf den Markt gebracht. Doch leider ist der Einsatz dieser neuen Geräte in Verbindung mit dem C64/C128 nicht ohne weiteres möglich. Die Hersteller des Printerface haben diesen Bedarf erkannt.



Durch das lange Anschlußkabel bereitet die Verbindung C64-24-Nadel-Drucker keine Probleme.

rechte Stellung der DIP-Schalter für den jeweiligen Drucker erfahren und testen. Als Abschluß erhalten Sie den abgebildeten Testausdruck, auf dem oben links die Stellung der DIP-Switches zu sehen ist.

Mit den Demo-Programmen erhalten Sie ein Printout der Zeichensätze und der Steuercodes und können die Funktionen in verschiedenen Modi erproben.

Eine umfangreiche Tabelle auf der Rückseite des Heftes nennt für etwa 53 verschiedene Druckermodelle die notwendige Einstellung der DIP-Schalter am Interface. Diese befinden sich an der Außenseite des Gehäuses und sind gut lesbar beschriftet.

Falls man das Printerface an verschiedenen Druckern betreiben möchte, reicht hier ein kurzer Blick aufs Heft.

Anhand der 8 DIP-Schalter lassen sich folgende Modi auswählen:

- Geräteadresse
- Druckertyp
- Sekundäradresse beim Einschalten
- ASCII-/DIN-Zeichensatz
- Auto-Line-Feed
- Interface-Modus

Unter dem letzten Punkt ist es möglich, das Interface so einzustellen, daß andere handelsübliche Interfaces simuliert werden können. Außerdem ist ein sogenannter VIZA-Modus bereits voreinstellbar.

Sämtliche Presets der DIP-Switches können auch per Software eingestellt werden. Außerdem können die Sekundäradressen beliebig den entsprechenden Funktionen zugeordnet werden.

Das Interface wird in zwei Versionen vertrieben, die sich durch die Größe des Puffers unterscheiden. Außerdem gibt es noch eine Puffererweiterung für die Grundversion:

GH 8 708 248,- DM mit 8K Puffer
GH 8 732 298,- DM mit 32K Puffer
GH 8 724 128,- DM Erweiterung auf 32K (nach Herstellerangaben)

Bei den Tests mit verschiedenen Programmen im C128- und im C64-Modus zeigten sich bei der Text- und Grafikausgabe keinerlei Probleme, die Darstellungen unter „Dateiservice“ in diesem Heft wurde mit einem NEC P6 und INPUT-Graph erstellt. Insgesamt ein problemlos einsetzbares und bedienerfreundliches Gerät. RH

positiv	negativ
leichte Bedienung problemloser Anschluß gute Anpassungsmöglichkeiten	unübersichtliche Anleitung

RTK
Postfach 71 03 44
8 000 München 31
(089) 79 51 10



BASIC auf Trab gebracht

Speedcompiler — Unterstützung für BASIC-Programmierer

Schon wieder ein Compiler für den C64? In der Tat gibt es eine unüberschaubare Vielfalt von BASIC-Compilern für diesen Computer. Dies ist aber der unseres Wissens leistungsfähigste Compiler, der bis jetzt in einer Zeitschrift vorgestellt wurde – und wahrscheinlich auch einer der leistungsfähigsten überhaupt. Darüber hinaus gehorcht er einer besonderen Konzeption, der bis jetzt kaum ein Compiler gerecht geworden ist.

Neues Konzept

Meist gibt es bei einem Compiler nur die Möglichkeit, ein vorhandenes Programm zu kompilieren, um es damit ein wenig schneller zu machen. Mit dem Speedcompiler kompilierte Programme sind zwar oft auch schneller als der von anderen Compilern erzeugte Code. Der Speedcompiler hat näm-

Bei vielen Programmier-Projekten steht man mit BASIC schnell vor unüberwindlichen Zeitschranken. Umsteigen auf Maschinensprache? Es geht auch anders. Mit diesem Compiler wird der Ablauf von BASIC-Programmen bis zu 20mal schneller. Selbstverständlich „kennt“ er auch unseren Publikumsliebbling Nummer eins, die Spracherweiterung INPUT-BASIC.

lich einige besonders schnelle Maschinenspracheroutinen zur Verfügung. Darüber hinaus stellt er Hilfen zur Programm-Optimierung und -Verbesserung bereit, die der BASIC-Interpreter nicht bietet. Zum Beispiel:

- Eigene Integer-Arithmetik
- Eigene String-Verarbeitung

- Keine Wartezeiten durch Garbage Collection
- Bis zu 50 KBytes für Programm und Daten verfügbar
- Overlays (Nachladeprogramme) einfach realisierbar
- Platz für Sprite- oder Graphikdaten reservierbar
- Abfangen von System-Fehlermeldungen

Mit all diesen Vorteilen ist dieser Compiler das ideale Werkzeug für engagierte Programmierer, die auch aufwendige Anwenderprogramme erstellen wollen. Der Programmlänge ist beim Speedcompiler nur durch den Speicherplatz des C64 eine Grenze gesetzt. Die Programme kann man mit Hilfe des Compilers so sicher machen, daß sie völlig absturzfrei funktionieren. Durch die Vielzahl von Möglichkeiten stellt er sozusagen ein „Entwicklungssystem für gute Programme“ dar.

Am eigenen Schopf

Der beste Beweis für die Leistungen des Speedcompilers ist der Compiler selbst. Er ist zu 100 % in BASIC V2 geschrieben und mit sich selbst kompiliert. Übrigens ist der Compiler im Quellcode 153 Blöcke lang,

kompiliert dagegen nur 98 Blöcke. Somit ist der Compiler unkompiliert gar nicht lauffähig, da der Speicherplatz nicht ausreicht.

Super-Normal . . .

Der Speedcompiler ist relativ einfach zu bedienen. Er muß wie ein BASIC-Programm

geladen und mit RUN gestartet werden. Normalerweise gibt man dann den Namen des zu kompilierenden Programms ein. Befindet sich das Programm auf der Diskette im Laufwerk, beginnt der Compiler den Kompilations-Vorgang mit dem ersten Durchlauf, also Pass 1. Treten dabei keine Fehler auf, wird Pass 2 gestartet. Ist auch dieser fehlerfrei beendet, so meldet der Compiler alle wichtigen Code-Daten und kehrt in den Direktmodus zurück. Die kompilierte Fassung des Programms ist unter dem Namen „C/“ + alter Programmname auf Diskette abgelegt. Existiert bereits ein solches File, wird es gelöscht. Der Kompilations-Vorgang läßt sich jederzeit durch Betätigen von RUN/STOP und gleichzeitig RE-STORE abbrechen. Kompilierte Programme werden wie BASIC-Programme geladen und mit RUN gestartet.

Kleine Compiler-Kunde

Fachbegriffe schnell erklärt

Ohne die Programmierung durch sein menschliches Gegenüber versteht der Computer überhaupt nichts. Prinzipiell gibt es zwei Möglichkeiten, einem Computer seine Wünsche mitzuteilen: durch eine **Hochsprache** oder durch **Maschinensprache**. Maschinensprache ist das, was der Prozessor direkt versteht. Hochsprachen sind die, die der Mensch gut versteht. Sie müssen in Maschinensprache **übersetzt** werden, weil der Prozessor sonst nichts versteht.

Zur Übersetzung gibt es wiederum zwei Möglichkeiten: **Interpretieren** oder **Kompilieren**. Ersteres ist das, was der BASIC-Interpreter normalerweise mit den eingetippten oder geladenen BASIC-Programmen macht: den Text (das heißt das Listing) nacheinander auswerten und **unmittelbar ausführen**, indem die passenden Maschinensprache-Routinen angesprungen werden. Kompilieren bedeutet strenggenommen, daß das vorhandene BASIC-Programm, der **Quelltext**, Stück für Stück in Maschinensprache umgesetzt wird, das Endergebnis nennt sich **Kompilat**.

Aus programmtechnischen Gründen (flapsig ausgedrückt: damit der Compiler in Zeile 1 schon weiß, wo sich das dort mit GOTO oder GOSUB angesprungene Unterprogramm in Zeile 60000 befindet) muß der Compiler den **Source-Text** (was dasselbe ist wie Quelltext, aber entschieden fachkundiger klingt) mehrere Male von Anfang bis Ende durchgehen. So ein Durchgang heißt auf Englisch **pass**, damit

dürften die Bezeichnungen **1-, 2- oder sogar 3-Pass-Compiler** klar sein.

Das wäre eigentlich schon alles, wenn sich die Dinge in der Praxis nicht meist komplizierter darstellen als in der reinen Theorie. Tatsächlich erzeugen nämlich die meisten Compiler einen **Zwischen-Code**, den sogenannten **P-Code**. Dieser P-Code besteht aus Maschinenbefehlen einer **virtuellen CPU** (sprich: eines nicht real existierenden Prozessors). Dieses Konzept wurde bei der Entwicklung von Pascal erdacht. Der P-Code ist nicht nur kürzer als eine direkte Umsetzung in Maschinenbefehle, sondern auch unabhängig vom eingesetzten Prozessor. Diesen P-Code versteht der Prozessor nämlich nicht (siehe oben), deswegen wird zur Laufzeit des Programms ein **Übersetzer** eingeschaltet.

Das den P-Code interpretierende Programm ist das **Runtime-Modul**, auch bekannt als **Runtime-Interpreter** oder **P-Code-Interpreter**. Bei einer Anpassung an andere Maschinen muß (theoretisch) nur dieser P-Code-Interpreter geändert werden, denn die Compiler selbst sind in einer Hochsprache geschrieben und mit sich selbst kompiliert.

Das ist dann wirklich alles, abgesehen von Feinheiten wie den Problemen bei der Behandlung von **Zahlenformaten**, **Overlays** (nachgeladene Programme) und **Speicherverwaltung**. Solche Fragen, die der **normale Anwender** sich nicht unbedingt stellen muß, werden in der demnächst erscheinenden Serie zum Thema behandelt. JS

. . . oder mit Extras

Bestimmte Optionen (Overlays, Platz für Grafik oder Sprites freihalten) sind einstellbar, wenn dem Programmnamen ein Befehl vorangestellt wird. Befehl und Name sind durch einen Doppelpunkt zu trennen, beispielsweise „EV:NAME“ statt nur „NAME“. Möglich sind die drei folgenden Voranstellungen:

EV Bei dieser Angabe werden nach dem Kompilieren die gefundenen Variablen und eventuelle Adressen auf dem Bildschirm ausgegeben.

EP bewirkt eine Ausgabe aller wichtiger Compiler-Meldungen auf dem Drucker. Fehlermeldungen erscheinen zusätzlich auf dem Bildschirm. Wie bei 'EV' werden auch alle Variablen-Namen, diesmal aber auf dem Drucker, ausgegeben. Der Drucker muß die Geräturnummer 4 haben.

ER ist die stärkste Eingriffsmöglichkeit, denn nach dieser Angabe wird der Programmierer nach den wichtigsten Grundeinstellungen gefragt, die er dann selbst modifizieren kann.

Drückt man bei der Frage nach dem File-Namen nur RETURN, wird das Directory der Diskette auf dem Bildschirm ausgegeben. Dies kann durch einen Tastendruck angehalten werden. Die Directory-Funktion ist auch über die Eingabe des Dollar-Zeichens (\$) erreichbar.

Disk-Befehle senden kann man durch '@'+ Befehls-String, etwa „@S:TESTPROGRAMM“, um das Programm „TESTPROGRAMM“ zu löschen. Die Directory-Funktion und das Senden von Disk-Befehlen wirken auf das Laufwerk mit der Geräte- nummer 8.

Selbstbestimmung nach Maß

Die ER-Option bietet eine Vielzahl von Ein- stellungsmöglichkeiten:

Was nicht geht: Inkompatibilitäten zum BASIC V2

GET, GET#: Nicht mit mehreren Variablen möglich (also nicht GETA\$,B\$ sondern GETA\$,GETB\$).

CLOSE: Wie GET nur mit einer Angabe er- laubt.

SYS: Es sind keine Multiparameter und keine Einsprünge ins BASIC-ROM mög- lich.

ON: Die Konstruktionen ON .. GOTO ... und ON .. GOSUB ... müssen alleine in der Zeile stehen.

ST: Nimmt den Wert 128 statt -128 an, darum immer: „ST AND 255“. Bei abge- schalteten I/O-Fehlern (zum Beispiel REM I04) ergibt ST/256 zusätzlich die Fehler- nummer (0-9) des Betriebssystems.

QA: In keiner Form als normale Variable benutzen! Das ist nämlich die Systemsta- tus-Variable des Runtime-Moduls.

VERIFY, SAVE, NEW, LIST: Diese vier Be- fehle sind nicht mehr im Sprachschatz enthalten. SAVE läßt sich notfalls mittels OPEN und PRINT# emulieren.

INPUT: Als Trennzeichen wird bei norma- ler Einstellung nur „;“ verwendet. „:“ ist nor- malerweise kein Trennzeichen. Weiterhin erfolgt bei Runtime-Fehlern keine Fehler- meldung „Extra Ignored“ oder „Redo from

Beginn für Programmcode: Hier kann der Start für den Programmcode festgelegt werden, der normalerweise auf 8192 liegt und auch nicht niedriger liegen darf. Ist das Programm kein Nachlader (siehe unten), so wird der Bereich zwischen dem Ende des Runtime-Moduls und dem Programmstart mit Null-Bytes aufgefüllt, dieser Bereich wird also mitgespeichert.

Druckerausgabe: Ist quasi identisch mit dem Befehl 'EP', außer daß die gefundenen Variablen nur dann ausgegeben werden, wenn auch der nächste Punkt mit 'j' beant- wortet wird.

Start“, da diese Fehlermeldungen bei vielen Programmen unerwünscht sind und im Programm selbst abgefragt werden können. Beim INPUT von numerischen Variablen wird bei Eingabe von nur Carriage Return nicht die alten Werte, sondern 0 übernom- men.

ARRAYS: Sind nur bis maximal 3dimensio- nal erlaubt. Mehrdimensionale Variablen werden im Gegensatz zu 1dimensionalen nicht automatisch auf 11 Elemente dimen- sioniert, sondern müssen unbedingt vom Programmierer dimensioniert werden. Ar- rays verschiedener Dimensionen müssen auch verschiedene Namen erhalten.

DIM: Muß alleine in der Zeile stehen. REMs dahinter sind nur erlaubt, wenn sie der Di- mensionierung der String-Länge dienen. DIM muß ganz am Anfang des Programm- textes stehen. Dimensionierungen in Unter- programmen oder ähnliches sind verboten!

NEXT: Zu jedem Befehl 'FOR' darf immer nur genau ein 'NEXT' angegeben werden. Dieses muß im Programmtext nach dem 'FOR'-Befehl kommen. Alle anderen Kon- struktionen zeugen von einem miserablen Programmierstil und sind für einen Compil- er, der bemüht ist, schnellen Code zu er- zeugen, undurchschaubar. Ein überzähliger NEXT-Befehl für eine FOR-Schleife läßt sich dadurch eliminieren, daß man ihn durch einen Sprung auf den allerletzten NEXT-Be- fehl der Schleife ersetzt. NEXT mit meh- reren Angaben für Schleifen-Variablen ist natürlich erlaubt.

Variablen-Ausgabe: Ist identisch mit 'EV', bei 'j' werden die Variablen mit Adressen ausgegeben.

Nachlader: Wird diese Frage mit 'j' beant- wortet, so wird das Programm ohne Run- time-Modul auf die Diskette geschrieben. Es kann dann von einem anderen kompilierten Programm nachgeladen werden (Overlay). Dies geht auch, wenn der Beginn des Pro- grammcodes höher liegt als 8192. Aller- dings erscheint dann eine Warnung, denn aus technischen Gründen muß ein Nachla- der an eine Adresse geladen werden, die

String-Variable: Wie schon oben be- schrieben, werden String-Variablen an- ders abgelegt als beim normalen BASIC- Interpreter. Das kann dazu führen, daß ein Programm, das String-Arrays benutzt, nicht mehr mit den Variablen in den Spei- cher paßt („Speicherplatz zu gering“-Feh- lermeldung). Dann müssen die String- Längen der String-Arrays entsprechend gekürzt werden. Beachten Sie bitte auch die Bemerkungen zur String-Verknüpfung im Hauptteil der Beschreibung.

Bei der Benutzung von Integer-Werten kann es in wenigen Fällen bei Berech- nungen zu Abweichungen im Vergleich zu den Ergebnissen des BASIC-V2-Interpre- ters kommen. Dieses kann durch unter- schiedliche Rundung oder auch in sehr wenigen Fällen durch Probleme beim Konvertieren von Integer nach Gleitkom- ma herrühren.

Bei der Ausgabe von Zahlen gibt der Runtime-Interpreter des Speedcompilers anschließend ein Leerzeichen aus. Der BASIC V2-Interpreter unterscheidet hier zwischen „normaler“ Peripherie und Bild- schirm, auf letzterem wird statt des Leer- zeichens ein Cursor-right ausgegeben, weil die Doppelzeilenorganisation des 64er Bildschirms dies erfordert. In be- stimmten Fällen kann es deswegen bei kompilierten Programmen zu uner- wünschten Scroll-Effekten kommen. Ab- hilfe: den Wert 29 in die Adresse 2912 im Runtime-Modul schreiben. Vor Drucker- ausgaben zurücksetzen (auf 32) nicht vergessen!

256 Bytes niedriger liegt als der Programmstart.

Beginn für Variablen: Diese zwei Punkte (für allgemeine Variablen und Integer-Variablen) sind speziell für Nachlader gedacht. Sie eröffnen dem Programmierer die Möglichkeit, mehrere Programme gleichzeitig im Computer zu haben und sie zusammenarbeiten zu lassen. Genaueres wird in einer demnächst folgenden Artikelserie beschrieben werden, in der diese und andere Compiler-Möglichkeiten im einzelnen erklärt werden.

Bereichstest ein: Mit dieser Option erzeugt der Compiler zusätzlichen Programmcode für Bereichstests von Arrays (Bad Subscript Error) sowie zum Test der korrekten String-Längen (String too long), außerdem wird beim RETURN-Befehl abgefragt, ob überhaupt eine Rücksprungsadresse auf dem Stack liegt (Return without Gosub). Die Ausführung der Programme wird dadurch etwas langsamer und der Code wird länger. Diese Option ist nur zum Ausstesten von kritischen Programmen gedacht. Bei fertigen Programmen wird man in aller Regel diese Option ungesetzt lassen.

INPUT-BASIC und der Speedcompiler

True Leser unseres Magazins wissen, warum es geht: INPUT-BASIC ist eine in der Januarausgabe 1986 veröffentlichte BASIC-Erweiterung, die bei der Leserumfrage im Herbst vorigen Jahres zum beliebtesten Programm gekürt wurde.

Mit INPUT-BASIC erstellte Programme können ganz normal kompiliert werden. Um dieses Programm laufen lassen zu können, muß INPUT-BASIC initialisiert sein, anschließend kann man das Programm laden und starten. Nach einem Abbruch des Programms kann es sein, daß INPUT-BASIC „abgehängt“ wird. Für einen Neustart des Programms muß INPUT-BASIC nicht neu initialisiert (also auch nicht neu geladen) werden; es darf natürlich nicht zerstört worden sein. Mit INPUT-BASIC hat man nicht sämtlichen

Zeilenadressen speichern: Die Zeilennummern mit ihren Referenzadressen im Objektcode werden nach dem Kompilieren als Programm mit dem Namen „Z/“ + Programmname abgespeichert.

Floppy-Nummern: Durch diese Abfrage nach den Geräteadressen der Quell- und Ziel-Floppy-Nummer wird das Kompilieren mit zwei Diskettenstationen unterstützt oder die Angabe einer beliebigen Geräteadresse. Beim Kompilieren mit einer Floppy ist die Ziel-Floppy-Nummer gleich der Quell-Floppy-Nummer. Bei zwei Floppys muß das Ursprungsprogramm im Quelllaufwerk sein. Übrigens lassen sich auch andere Laufwerke als die 1541 zum Kompilieren benutzen, etwa die 1570 oder die 1571².

Flexible Responses

Der Compiler „kennt“ über 50 Fehlermeldungen in deutsch. Taucht ein Fehler in Pass 1 auf, so wird er ausgegeben, gefolgt von der Zeilennummer, in der der Fehler aufgetaucht ist. War der Fehler nicht zu schwerwiegend, versucht der Compiler sei-

ne Arbeit fortzusetzen, um mögliche weitere Fehler zu finden.

Pass 2 wird nur gestartet, wenn im Pass 1 keine Fehler aufgetreten sind. Sonst bricht der Compiler nach Pass 1 ab. Um dem Programmierer die Möglichkeit zu geben, alle Fehler mitzubekommen, wird jeweils nach 10 Fehlern angehalten und gefragt, ob weiterkompiliert werden soll. Wird dies mit Nein beantwortet, erfolgt ein sofortiger Abbruch. Treten erst im Pass 2 Fehler auf, so wird bis zum Programmende kompiliert und eine Warnung ausgegeben, denn das erzeugte File ist nicht lauffähig.

Damit der Programmierer weiß, wo der Compiler gerade arbeitet, wird die Zeilennummer, die er gerade liest, links oben auf dem Bildschirm eingeblendet.

Am Ende der Kompilation gibt der Compiler alle wichtigen Meldungen aus: zunächst die Anzahl der aufgetretenen Fehler, dann die wichtigsten Adressen, die das Programm betreffen. Der 'Beginn des Programmcodes' zeigt die Startadresse des P-Codes an. Der 'Beginn der Datas' ist die Adresse nach dem Ende des P-Codes. Der 'Beginn der Variablen' liegt hinter dem Data-Bereich und zeigt die Anfangsadresse der Laufzeit-Variablen (außer Integer-Variablen) an. 'Ende der Variablen' ist die erste Adresse hinter den Variablen. Alle Speicherstellen hinter dieser Adresse sind frei für andere Dinge (es sei denn, INPUT-BASIC wird benutzt), etwa für Maschinenprogramme.

Nach diesen Angaben werden, falls 'Variablen-Ausgabe' eingeschaltet ist, die verwendeten Variablennamen ausgegeben. Als erstes in loser Folge alle Integer-Variablen-Namen, jeweils mit zwei Spaces getrennt; zwei Doppelpunkte in dieser Liste bedeuten, daß die vorige Variable im Programm als Schleifen-Variable benutzt wird. Nach den Integer-Variablen werden alle anderen Variablen-Typen untereinander aufgelistet: in der Reihenfolge Name, Kennziffer und Adresse, auf die die Variable gelegt wurde. Die Kennziffern haben folgende Bedeutungen:

- 0 Keine Bedeutung
- 1 String-Variable
- 2 Integer-Array
- 3 String-Array
- 4 Gleitkomma-Variable
- 6 Gleitkomma-Array

Speicherplatz zur Verfügung, sondern nur die gewohnten 38 KByte.

Einige INPUT-BASIC-Befehle sind bereits im Runtime-Modul des Speedcompilers implementiert, sie können genutzt werden, ohne auf INPUT-BASIC zurückgreifen zu müssen. Und zwar

PRINTAT, RESTORE n, DOKE, VOLUME

Allerdings kann man DOKE und VOLUME nur unter INPUT-BASIC eingeben, wegen der „Tokenisierung“. Der PRINTAT-Befehl wird in der Syntax PRINT@ zeile,spalte,„text“ erwartet.

Einige INPUT-BASIC-Befehle aus der Kategorie „Programmierhilfen“ sind nicht unterstützt, weil ihr Einsatz zur Laufzeit wenig sinnvoll ist, als da sind:

@G, ALLCLOSE, AUTO, CASE ERROR, DELETE, FIND, MERGE, NO ERROR, OLD, RENUMBER, RESUME, USR mit Spezialfunktionen

Zuallerletzt wird die Anzahl der vorgefundenen Zeilen ausgegeben.

Echte, ehrliche Integer . . .

Der Speedcompiler ist ein Spezialist für Integer-Rechnungen. Das heißt, er erzeugt Code für echte Integer-Operationen bei der

Verarbeitung von Integer-Werten. (Um diese auf den ersten Blick selbstverständlich klingende Fähigkeit richtig würdigen zu können, muß man wissen, daß das BASIC V2 alle Berechnungen im Fließkomma-Format durchführt. Integer-Werte werden also einmal hin- und wieder hergewandelt.) Das bedeutet eine wesentlich schnellere Ver-

arbeitung durch Integer-Variablen. Denn Rechnungen in Gleitkomma-Arithmetik sind wesentlich langsamer als solche in Integer-Arithmetik. Für die meisten Zwecke ist übrigens Integer-Arithmetik ausreichend.

Für möglichst schnelle Verarbeitung bietet der Speedcompiler deshalb die Möglichkeit,

Gezielte Eingriffe: Compiler-Optionen

Compiler-Optionen sind Angaben im Programmtext, die der Compiler auswertet. Damit diese nicht beim Interpretieren des Programms stören, werden sie hinter REM-Befehle geschrieben. Das Kennzeichen für Compiler-Optionen ist '\$', somit lautet die Syntax für Compiler-Optionen REM\$Option. Folgende Optionen sind möglich:

STOP-Kontrolle

S0: Normaleinstellung, erlaubt Abbruch mit RESTORE und RUN/STOP-RESTORE

S1: Programmabbruch nur noch mittels RUN/STOP-RESTORE möglich

S2: kein Abbruch mehr möglich (außer durch Hardware-Reset)

I/O-Optionen

I00: Normaleinstellung, alle I/O-Fehler führen zum Programmabbruch

I01: „File not found“ führt nicht zum Abbruch

I02: „Device not present“ führt nicht zum Abbruch

I03: Zusammenfassung von I01 und I02

I04: keine Fehlermeldung führt zum Abbruch

INPUT-Befehl

?: Nach dem '?' kann ein String in Anführungszeichen angegeben werden, der statt des '?' beim INPUT-Befehl ausgegeben wird. So kann mit "" die Ausgabe des '?' verhindert werden.

Bereichstest

Mit der Compiler-Option 'C' läßt sich im Programmtext dasselbe erreichen wie mit der Voreinstellung „Bereichstest“. Mit dieser Option wird also auf jeden Fall immer Testcode erzeugt.

Integer-Bereich

+: Stellt die Integer-Wertigkeit vom Bereich - 32 768 bis 32 767 in den Bereich 0 bis 65 535 um. Es werden also alle Integers nur noch als positive Zahlen angenommen.

-: Stellt wieder auf Normal-Modus zurück.

Stack-Kontrolle

R: Nach dieser Option kann über die Variable QA der aktuelle Stand des System-Stackpointers ermittelt werden.

B: Mit dieser Option kann der Systemstackpointer berichtigt werden. Nach 'B' muß eine Integer-Variable (I) angegeben werden, die die Anzahl der Bytes enthält, um die der Stack berichtigt werden soll (nur positiv).

Variablen-Einstellung

V0: Ab dieser Option werden alle Variablen ohne Angabe (wie %, \$ oder &) als Integers behandelt. Um dann noch mit Gleitkomma-Variablen arbeiten zu können, müssen diese den Zusatz '&' erhalten (beispielsweise 'D&' statt nur 'D'). Diese Einstellung dient der leichteren Erstellung von Programmen, die hauptsächlich Integer-Variablen benutzen.

V1: Schaltet in den Normalmodus zurück.

INPUT-Trennzeichen

Z: Nach 'Z' kann ein beliebiges Zeichen folgen, das die Rolle des Trennzeichens „,“ in INPUT-Eingabe-Strings übernimmt. „,“ ist dann kein Trennzeichen mehr, so setzt

„REM: „,“ als Trennzeichen. Space ist aus technischen Gründen mit dieser Option nicht als Trennzeichen einsetzbar. Aber mittels 'POKE \$1EFC,ASC(Zeichen)' läßt sich jeder ASCII-Wert als Trennzeichen definieren. So läßt sich mittels 'POKE \$1EFC,13' das zweite Trennzeichen ganz abschalten, denn Carriage Return ist bereits das erste Trennzeichen bei INPUT, das den Eingabe-String beendet.

Gleitkommafehler

E2: Schaltet die Gleitkommafehlermeldungen, wie zum Beispiel „Division by Zero“, völlig ab. Diese Option ist allerdings nicht so einfach zu handhaben, da bei manchen Fehlern der System-Stack verändert wird. Darum müssen diese Fehler, wenn sie auftreten können, nachträglich abgefragt werden. Dies ist eine Option für absolute Spezialisten.

E0: Schaltet wieder auf Normalmodus

String-Länge einstellen

\$: Nach dem '\$' steht ein String-Name, der noch nicht benutzt worden ist. Dann kommt 'E' mit der String-Länge dazwischen. Damit kann man bei Einzel-Strings (keine Arrays) eine bestimmte String-Länge (kleiner 256) festsetzen.

Steht 'REM' mit einer angefügten Zahl direkt hinter der Dimensionierung eines String-Arrays, so werden alle Einzel-Strings dieses Arrays auf diese Länge begrenzt.

Alle Compiler-Optionen gelten nur von da an, wo sie im Programmtext stehen. Sie können teilweise auch später im Programm wieder aufgehoben werden. Sie gelten dann also genau im Bereich zwischen diesen beiden Angaben.

FOR-NEXT-Schleifen zu programmieren, deren Lauf-Variable vom Typ Integer ist, wie die Konstruktion „FOR I% = 0 TO 100“. Auch mit einer Schrittweite von -1 läßt sich eine Integer-Schleife bilden. Um dem Programmierer noch weitere Arbeit abzunehmen, kennt der Compiler zusätzlich noch die Möglichkeit, Hexadezimal-Zahlen im Bereich von \$0000 bis \$FFFF im Programmtext zu erkennen und auszuwerten. Dazu muß nur jeweils das Zeichen „\$“ vor der Zahl stehen.

Im kompilierten Programm werden für eine beschleunigte String-Verarbeitung und vor allem, um die bei Programmierern berücksichtigte „Garbage Collection“ zu vermeiden, die String-Variablen anders verwaltet. Diese Verwaltung ist dabei so ähnlich wie in Pascal- oder C-Programmen. Für jeden String wird immer nur die Anzahl von Zeichen im Speicher reserviert, die maximal benötigt werden, um den String aufzunehmen. Diese Anzahl kann der Programmierer angeben; gibt er keinen Wert an, so geht der Compiler von einer Länge von 255 für normale Strings und 254 für String-Arrays aus.

... und gute alte statische Strings

Dadurch kann es vor allem bei String-Arrays zu Speicherplatzproblemen kommen, denn jeder Einzel-String nimmt dann einen Speicherplatz von 255 Bytes ein. Gibt der Programmierer eine kleinere Zahl an, sollte er genau wissen, wie viele Zeichen der String maximal während des Programmlaufs einnehmen kann. Dies kann notfalls durch Testen herausgefunden werden, indem man einen geschätzten Wert angibt und das Programm mit der Option 'Bereichstest ein' kompiliert. Wenn beim Austeilen aller Funktionen die String-Länge überschritten wird, kommt die Fehlermeldung „String too long“: die Angabe war zu klein.

Bei Unsicherheiten sollte man immer soviel Platz für Strings reservieren, wie der verfügbare Speicher zuläßt. Die 255 Zeichen für Einzel-Strings sind normalerweise nicht problematisch, aber trotzdem einstellbar. Die Einstellung für String-Arrays erfolgt direkt nach deren Dimensionierung in der gleichen Zeile des DIM-Befehls, mit

Normierter Tempo-Test

Benchmark-Tests, das sei vorweg gesagt, sind natürlich nicht die Meßlatte für die Qualität von Hard- oder Software. Verglichen wird lediglich die Ausführungszeit für mehrere kleine Programme. Wir haben die erstmals 1977 von Tom Rugg und Phil Feldman in der US-amerikanischen Zeitschrift Kilobaud veröffentlichten 'Benchmarks' zugrunde gelegt, da sich diese zu einem Quasi-Standard entwickelt haben. Um auch die Verarbeitungsge-

schwindigkeit der String-Verarbeitung vergleichbar zu machen, haben wir noch ein neuntes Programm hinzugefügt.

Vergleicht man die notwendigen Zeiten zur String-Verarbeitung, muß man noch folgendes berücksichtigen: wäre beim Interpretieren durch das 'normale' Commodore-BASIC eine Garbage Collection aufgetreten, so hätte sich die Zeit um ein Vielfaches verlängert; beim kompilierten Programm kann keine Garbage Collection auftreten.

Bench	BASICV2 (Interp.)	Austro-Compil.	BASIC64 Compil.	Speedcompiler
BM1	1.4	1.16	1.05	0.74
BM2	11.23	2.08	1.85	1.47
BM3	20.6	7.56	7.35	7.25
BM4	22.9	8.1	7.05	7.56
BM5	25.1	8.3	7.37	7.7
BM6	36.54	15.43	13.94	11.7
BM7	56.54	22.12	19.67	17.54
BM8	12.02	10.38	10.26	10.22
BM9	21.34	12.75	13.00	11.92

Deutlich die Nase vorn hat der Speedcompiler beim Geschwindigkeitstest — gegenüber dem BASIC-Interpreter sowieso, aber auch gegenüber zwei anderen Compilern. (Zeiten in Sekunden)

„REM&String-Länge“. Diese Angabe gilt nur für das letzte Array im DIM-Befehl und für alle Einzel-Strings im Array.

Die Tempo-Optimierung des Speedcompilers geht leider zu Lasten seiner Fähigkeit, komplexe Konstruktionen von String-Formeln auszuwerten. Darum kommt es manchmal zu der Fehlermeldung „Stringformel zu komplex“ in solchen Ausdrücken, in denen mehrere String-Verarbeitungsbeefehle ineinandergeschachtelt vorkommen. Der Grund dafür liegt eigentlich nicht im Compiler selbst, sondern im Runtime-Modul, in dem die Verarbeitung zur Laufzeit stark vereinfacht ist (um Zeit zu sparen), so daß manche Konstruktionen zu schwierig sind. Dieses läßt sich vermeiden, indem

man diesen Ausdruck auf mehrere kleinere Formeln aufteilt, also zunächst die innersten String-Verarbeitungsbefehle abarbeitet, das Ergebnis in einem Hilfs-String speichert und dann die äußeren Funktionen auf diesen anwendet. Eine genauere Beschreibung dieses Sachverhaltes wird ebenfalls in der oben erwähnten Serie folgen.

Variable Dimensionierungen sind nicht möglich. Wird keine feste Dimensionierung bei einem DIM-Befehl angegeben, so wird die Anzahl der Elemente direkt beim Kompilieren abgefragt.

Der Speedcompiler erzeugt einen speziellen Hochgeschwindigkeits-P-Code. Das ist eine Art Zwischensprache zwischen BASIC und Maschinensprache, die von einem speziellem P-Code-Interpreter abgearbeitet wird. Dieser P-Code-Interpreter ist im Compiler enthalten und wird zu Beginn von Pass 2 auf Diskette geschrieben (Ausnahme: Nachlader). Dieser Interpreter darf beliebig kopiert und weitergegeben werden. Darum sind kompilierte Programme sogar kommerziell nutzbar! Manche mögen sich jetzt

vielleicht fragen, warum der Compiler nicht gleich Maschinencode erzeugt. Der Grund ist, daß Maschinencode nur noch wenig Geschwindigkeitssteigerung bringt und vor allem, daß Maschinencode wesentlich längere Programme ergeben würde. Der P-Code-Interpreter hat eine Länge von 24 Blöcken, auch ein einzeliges Programm wird

durch Kompilation also mindestens 25 Blöcke lang.

Eigener Status

„QA“ ist die interne Status-Variablen des Runtime-Modules, die von Programmen wie

eine normale Variablen abgefragt werden kann. Sie hat verschiedene Bedeutungen:

(1) Bei Division von Integer-Werten erhält man nach der Division den Divisionsrest.

(2) Bei den Integer-Operationen $*$, $+$, $-$ kann man einen Überlauf des Integer-Bereiches feststellen, wenn diese Variablen ungleich Null ist.

(3) Nach einem String-Array-Zugriff wird diese Variablen 1, wenn ein schwerwiegender Adressierungsfehler aufgetreten ist.

(4) Nach der Compiler-Option „ER“ enthält QA den Stand des Runtime-Stackpointers.

Der Befehl LOAD darf nur mit Sekundäradresse 1 angewandt werden, da durch die interne Technik des Runtime-Moduls alle Nachlader (oder nachzuladende Maschinenprogramme) absolut geladen werden müssen. (Das ist natürlich keine Einschränkung.) Ein kompiliertes Programm, das ohne Runtime-Module (Nachlader) kompiliert wurde, wird mit Sekundäradresse 2 nachgeladen. Es wird dann automatisch nach dem Laden gestartet. Daß bei diesem Nachladen eventuell das erste Programm überschrieben wird, ist nicht wichtig, wenn man es nicht mehr braucht.

Mit „GOTO# Adresse“ kann man direkt eine bestimmte Adresse im P-Code eines kompilierten Programmes anspringen. Das ist dann nützlich, wenn man ein zweites Programm nachgeladen hat, das das erste nicht überschrieben hat. Dann kann man nämlich zwischen den Programmen hin- und herspringen. Um ein Programm zu starten, muß man die Adresse angeben, die beim Kompilieren des Programms unter „Beginn Programmcode“ ausgegeben wurde. Weitere Erläuterungen folgen in der bereits erwähnten Serie.

Mit acht Zeichen wird es gehen

Variablen-Namen dürfen beliebig lang sein. Davon werden die ersten acht Zeichen unterschieden, ein entscheidender Vorteil gegenüber den nichtssagenden Variablen-Namen des BASIC V2. Ansonsten besteht nur noch eine Beschränkung in der Anzahl der Variablen, die aber kaum erreicht wird. Zulässig sind maximal 120 Integer-Variablen

```

10 rem kleines programm zum verwalten
20 rem von telefonnummern
30 rem zum weiterexperimentieren
40 rem und kennenlernen der moeglich-
50 rem keiten des compilers
60 rem 010987 js
100 rem alle globalen compileroperationen
110 rem gehoeren an den programmanfang
120 rem fsl :fio3
130 rem abbruch nur run/stop-restore
140 rem kein abbruch durch
150 rem 'file not found' und
160 rem 'device not present'
170 rem f?:"
180 rem input befehl mit ':'
185 rem fv0
187 rem nur integer
190 rem
200 rem menue
205 max=100:dim d$(max,1):rem f20
206 rem stringlaenge 20,per hand dimensionieren
210 eingabe=0:ausgabe=3
220 rem 8 zeichen signifkaint
230 print"2kleines telefonverzeichnis"
240 print
250 print"bitte waehlen:"
255 print"programm beenden" 0"
260 print"ein-/ausgabe aendern" 1"
270 print"daten eingeben" 2"
280 print"daten ausgeben" 3"
290 print"daten suchen" 4"
292 print"daten ordnen" 5"
295 print">"
300 gosub 1000:if w<48 or w>53 then 300
305 print
310 on (w-48) gosub 2000,3000,4000,5000,6000
320 if w$("<0") then 240
330 close2:close3:close15
340 end
1000 rem -warten auf taste-
1005 poke 198,0:wait 198,1
1010 get w$:print <CRSR_Left>:w$;
1020 w=asc(w$)
1030 return
2000 rem -ein-/ausgabe aendern
2010 print"eingabegeraet 0";
2015 gosub1000:if w=13 then w=48
2017 eingabe=w-48:print
2020 print"ausgabegeraet 3";
2025 gosub1000:if w=13 then w=51
2027 ausgabe=w-48:print
2030 return
3000 rem -daten eingeben
3005 fi$=""
3010 if eingabe<8 then 3030
3020 gosub 9000
3030 open2,eingabe,2,fi$

```

```

3032 if st<0 then 3100
3035 if eingabe<0 then 3040
3037 print"name,telefonnummer ";
3040 input#2,n$,t$
3050 d$(anz,0)=n$:d$(anz,1)=t$
3060 anz=anz+1:if eingabe=0 then print
3065 if eingabe>7 then if st<64 then 3040
3066 if eingabe>7 then 3100
3070 print"weiter? (j/n) "
3080 gosub1000:if w$="n" then 3100
3090 if anz<max+1 then 3037
3095 print"datei voll!"
3100 close2
3110 return
4000 rem -daten ausgeben
4005 fi$=""
4010 if ausgabe<8 then 4030
4020 gosub 9000:fi=fi$+"s,w"
4030 open3,ausgabe,3,fi$
4035 if st<0 then 4075
4040 for i=0 to anz-1
4050 print#3,d$(i,0)
4060 print#3,d$(i,1)
4070 next
4075 close3
4080 return
5000 rem -daten suchen
5010 print"namen suchen" 1"
5020 print"oder"
5030 print"telefon-num." 2"
5040 gosub1000
5050 if w$("1" or w$)="2" then 5040
5060 such=w-49
5070 input"suchwort":such$
5080 for i=0 to anz
5090 if such$<>d$(i,such) then 5100
5095 print d$(i,0)
5097 print d$(i,1)
5100 next
5110 return
6000 rem -daten ordnen
6010 for i=0 to anz-2
6020 for j=i+1 to anz-1
6030 if d$(i,0)>d$(j,0) then gosub 6900
6040 next j,i
6060 return
6900 rem strings tauschen
6910 namtemp$d$(i,0):numtemp$d$(i,1)
6920 d$(i,0)=d$(j,0):d$(i,1)=d$(j,1)
6930 d$(j,0)=namtemp:d$(j,1)=numtemp
6990 return
9000 rem -dateinamen holen
9005 input"dateiname":fi$
9010 return

```

Den Einsatz der Compiler-Optionen demonstriert dieses kleine BASIC-Programm

<pre> 1000 rem bm1 1010 print"s"; 1020 for k=1to1000 1030 next k 1040 print"e", 2000 rem bm2 2010 print"s"; 2020 k=0 2030 k=k+1 2040 if k<1000then2030 2050 print"e", 3000 rem bm3 3010 print"s"; 3020 k=0 3030 k=k+1 3040 a=k/k*k+k-k 3050 if k<1000then3030 3060 print"e", 4000 rem bm4 4010 print"s"; 4020 k=0 4030 k=k+1 4040 a=k/2*3+4-5 4050 if k<1000then4030 4060 print"e", 5000 rem bm5 5010 print"s"; </pre>	<pre> 5020 k=0 5030 k=k+1 5040 a=k/2*3+4-5 5050 gosub5080 5060 if k<1000then5030 5070 print"e", 5080 return 6000 rem bm6 6010 print"s"; 6020 k=0 6030 dim m(5) 6040 k=k+1 6050 a=k/2*3+4-5 6060 gosub 6110 6070 for l =1to5 6080 next l 6090 if k<1000then6040 6100 print"e", 6110 return 7000 rem bm7 7010 print"s"; 7020 k=0 7030 dim m(5) 7040 k=k+1 7050 a=k/2*3+4-5 7060 gosub 7120 7070 forl=1to5 </pre>	<pre> 7080 m(l)=a 7090 next l 7100 if k<1000then7040 7110 print"e", 7120 return 8000 rem bm8 8010 print"s"; 8020 k=0 8030 k=k+1 8040 a=k^2 8050 b=log(k) 8060 c=sin(k) 8070 if k<1000then8030 8080 print"e", 10000 rem bm9 stringtest 10001 print"s"; 10002 mx=666 10003 dima\$(mx):rem50 10010 fori=1to mx 10020 a\$(i)=str\$(i)+"/"+ti\$ 10030 next 10070 fori=1to mx/2 10080 a\$(0)=a\$(i) 10090 a\$(i)=a\$(mx+1-i) 10100 a\$(mx+1-i)=a\$(0) 10110 next 10120 print"e", </pre>
---	--	---

Acht sind klassisch, das neunte ist speziell auf String-Operationen bezogen. Benchmark-Testprogramme.

dann nur noch kontrollieren, in welchem Bereich die ausgedruckte Nummer ist, und erhält die gesuchte Zeilennummer.

Alle Programme können nach einem Reset einfach durch „SYS 2061“ neu gestartet werden.

Vorsicht statt Nachsehen

Für die Kompilation von Programmen gelten folgende Richtlinien, die eingehalten werden sollten.

- Auf der Zieldiskette muß genügend Platz frei sein. Möglichst leere Disketten benutzen.

- Auf der Zieldiskette sollte kein File mit Namen 'D'+Programmname vorhanden sein. Ab einer bestimmten Menge von DATA-Statements werden diese nämlich in einem File mit diesem Präfix zwischengelagert.

- Außerdem sollte kein File 'C'+Programmname oder 'Z'+Programmname auf der Diskette sein, es sei denn, es wird nicht mehr gebraucht. (Etwa, wenn es von der vorigen Kompilation stammt.)

- Alle Programme auf der Quelldiskette sollten noch einmal auf einer anderen Diskette vorhanden sein; es sei denn es wird mit zwei Laufwerken gearbeitet, dann ist das nicht ganz so wichtig.

- Die BAM der Zieldiskette sollte man vorsichtshalber bei häufigem Kompilieren gelegentlich mit dem VALIDATE-Befehl aktualisieren, das vermeidet DOS-Fehler.

Die mit „sollte“ angegebenen Richtlinien sind nicht unbedingt nötig, dienen aber der Sicherheit der Programme und Disketten und gewährleisten damit ein rasches und zuverlässiges Arbeiten.

Um falschen Erwartungen vorzubeugen, sei zum Schluß noch folgendes bemerkt: Dieser Compiler ist nicht zu dem Zweck geschrieben worden, „mal eben schnell“ irgendein BASIC-Programm zu behandeln, sondern eher zur Programmentwicklung im Hinblick auf diesen Compiler. Nur dann kann er seine Fähigkeiten voll ausspielen. Wie wirklich das letzte aus dem Speedcompiler herausgeholt werden kann, wird in der anschließenden Serie aufgezeigt.

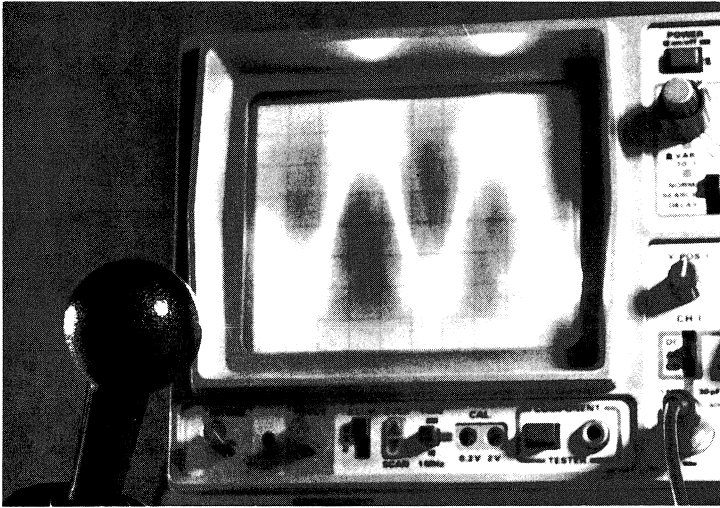
Jürgen Lindemeyer/JS

Runtime - Modul	\$ 0 8 0 1
Black - Board	\$ 1 e e 0
Variablen - Adressen	\$ 1 f 0 0
Programmcode dahinter: Variablen	\$ 2 0 0 0
Programmcode, Variablen oder INPUT - BASIC (oder leer)	"
evtl. Rest von INPUT - BASIC	\$ d 0 0 0
	\$ e 0 0 0

So nutzen kompilierte Programme den Speicher des C64. Der P-Code-Start ist variabel, die Lage des Runtime-Moduls ist unveränderlich

(außer Integer-Arrays). Von den anderen Variablen-Typen dürfen insgesamt 128 benutzt werden. Sollte das wirklich einmal nicht ausreichen, kann man wenig benutzte Variablen in einem Array zusammenfassen.

Alle mit dem Speedcompiler kompilierten Programme können auf zwei Arten beim Laufen abgebrochen werden, es sei denn, diese Möglichkeiten sind ausgeschaltet. Die erste ist ein Druck auf die RESTORE-Taste. Dies entspricht dem normalen Break mit der STOP-Taste. Diese Möglichkeit dient aber eigentlich nur zum Testen, da hier weder der Bildschirm gelöscht noch der Stand des Programmzählers ausgegeben wird. Die zweite Möglichkeit ist der Abbruch mittels RUN/STOP-RESTORE. Bei jedem Abbruch, der mit der Ausgabe von „Break in ...“ endet, außer dem mit dem STOP-Befehl, wird der Stand des Programmzählers und nicht die Zeilennummer des Quellprogramms ausgegeben. Um die entsprechende Zeilennummer zu erhalten, kann man sich die Zeilenadressen vom Compiler generieren lassen und das File dann einfach wie ein BASIC-Programm ausdrucken. Man muß



Dateiservice

Im Dienst der Graphen

Beispieldateien zu INPUT-Graph und INPUT-CAD

In vielen Bereichen der Meßtechnik und der Verarbeitung statistischer Daten fällt Zahlenmaterial an, aus dem sich selbst bei einer grafischen Darstellung keine genaueren Aussagen ableiten lassen. Mit solchen Problemen haben sich Mathematiker beschäftigt und Algorithmen, also Rechenverfahren, entwickelt, mit deren Hilfe besondere Eigenschaften von Zahlenmaterial ermittelt werden können. Zu den bekannten Verfahren gehört die Berechnung des Mittelwerts. INPUT-Graph aus INPUT 64, Ausgabe 9/87, bietet nicht nur die Möglichkeit Zahlenmaterial grafisch darzustellen, sondern darüber hinaus die mathematische Bearbeitung von bereits vorliegenden Daten. Grundsätzlich ist auch der umgekehrten Weg möglich, mit Hilfe von Funktionen Daten zu erzeugen.

Die Dateien aus dem Modul „Dateiservice“ sollten Sie auf einen eigenen Datenträger übernehmen. „Alphabet“ enthält einen feinen Schriftsatz aus der Gruppe „old english“ für INPUT-CAD. Die übrigen Dateien liefern Beispiele für INPUT-Graph. Dieses Werkzeug verfügt neben den grafischen Darstellungsmöglichkeiten auch über eine Schar von mathematischen Verfahren, mit denen Sie Ihr Datenmaterial überarbeiten können. Im folgenden können Sie sich den Methoden des Herrn Fourier und der Approximation annähern.

Es lassen sich aus einem Wust von Daten bestimmte Merkmale herausfiltern und aufgrund dieser Merkmale Aussagen über die Datenstrukturen machen. Welche inhaltli-

che Bedeutung diese Aussagen dann für die Situation haben, aus der die Daten gewonnen wurden, hängt in erster Linie von der kritischen Reflexion der Meßbedingungen und von dem Verständnis für Zahlenmaterial ab.

Ins Bild setzen

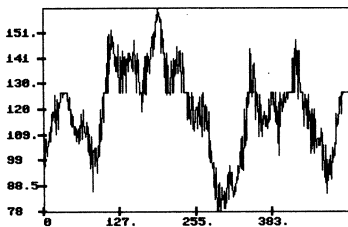
Es kann sehr interessant sein, Daten zu untersuchen, die von einem Sampler stammen, einem Gerät, das analoge Signale in digitale Signale umwandelt und diese in Form von Bits abspeichert (INPUT 64, Ausgabe 10/86–1/87). Diese Geräte lassen sich zur Sound- oder Sprachanalyse einsetzen. Schaut man sich einen Signalverlauf in der grafischen Darstellung an, sieht man meist nur ein Gemisch von unzähligen Impulsen, in dem man Andeutungen von Schwingungsverläufen höchstens errahnen kann.

Verfügen Sie über INPUT-Graph, können Sie sich eine solche Grafik anschauen: Sie speichern zuerst aus dem Modul „Dateiservice“ die Dateien „samplea“, „sampleb“ und „samples“ ab. Danach laden Sie diese mit INPUT-Graph als Datenfeld (siehe Anleitung zu INPUT-Graph in INPUT 64, Ausgabe 9/87) und übertragen Sie über den Menüpunkt „Byte“ im Lademenü ins Y-Feld. Wählen Sie den Menüpunkt „Grafik“, wird der Verlauf dieser Samples in Ausschnitten dargestellt. Verfahren Sie wie im Kastentext „Klangbild“ erläutert. Das Ereignis, das „samplea“ erzeugte, entspricht dem bekannten „A“, das der Herr Doktor einem im Anflug einer Grippe gelegentlich abverlangt. Die Dateien „sampleb“ und „samples“ entsprechen den Lauten „B“ und „S“.

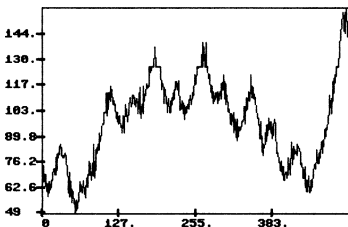
INPUT-Graph bietet nun die Verfahren, mit denen sich diese Signalverläufe analysieren lassen. Die Analyse liefert Aussagen über das Frequenzspektrum und eine angenäherte Funktion, mit der solche Signale wiederum synthetisiert werden können. Geeignete Verfahren hierzu wurden von Herrn Fourier entwickelt. Der mathematische Hintergrund wurde ausführlich und nachvollziehbar in c't, Ausgabe 8/86 erläutert.

Imaginäres

Die Fourier-Analyse/-Synthese verarbeitet ausschließlich komplexe Werte. Dies sind



Ausschnitt im berichtigten „A“



„B“-Landschaften

Werte, die einen Real- und einen Imaginäranteil enthalten.

Die Umkehrung des Quadrierens, das Wurzelziehen, liefert beispielsweise imaginäre Werte. Solange man bei diesem Rechenvorgang im positiven Zahlenbereich bleibt, erhält man jeweils eine positive und eine negative Zahl als Ergebnis. Wurzel aus '9' ergibt '+3' und '-3'. Jeder Versuch, die gleiche Prozedur im negativen Zahlenbereich durchzuführen, führt zwangsläufig in einen

Klangbilder

Wählen Sie den Menüpunkt „Laden“. Wenn Sie die Diskette mit den Dateien aus dem Dateiservice eingelegt haben, können Sie über „Directory“ das Inhaltsverzeichnis der Diskette kontrollieren. Mit dem „Zeigefinger“ setzen Sie die Position, an der nach RETURN oder nach Betätigen des Feuer-Knopfs das Menü wieder eingeblendet wird. Geben Sie den gewünschten Namen ein und wählen anschließend „Speicherbereich“. Als Startwert empfiehlt sich der Wert '49 152'.

Nach dem Ladevorgang können Sie über „Bytes“ im gleichen Menü die Daten ins Y-Feld kopieren. Verschieben Sie diesen Bereich in einer Schrittweite

von 1024, erreichen Sie so weitere Teile des geladenen Samples. Vermeiden Sie es, Samples in diesen Bereich zu laden, die mehr als 16 Blöcke haben. Sie schreiben sonst unweigerlich in geschützte Bereiche hinein.

Sie sehen, daß es bei einigen Samples erst ab einer bestimmten Speicherzelle zu interessanten Bildern kommt. Anhand der Bedienungsanleitung in Ausgabe 9/87 können Sie nach Belieben mit diesen Daten herumexperimentieren, die Zwischenergebnisse jederzeit speichern und ausdrucken. Sogar die Grafik läßt sich abspeichern, entweder als CAD-Datei oder als „Speicherbereich“ ab 40 960 und steht damit für andere Grafik-Programme zur Verfügung.

Zahlenraum, der sich jeglicher Vorstellbarkeit entzieht: der Raum der imaginären Zahlen. Wurzel aus '-9' liefert als Ergebnis '-3i' und '+3i'.

Imaginäre und reelle Zahlen bilden zusammen einen Zahlenraum, die komplexen Zahlen. Komplexe Zahlen bestehen also immer aus zwei Anteilen, dem reellen und dem imaginären Anteil (es sei daran erinnert, daß reelle Werte einen Spezialfall komple-

xer Daten darstellen: ihr Imaginärteil ist Null).

Fourier behauptete und bewies, daß jeder periodische Signalverlauf durch eine Summe von Sinus- und Kosinusschwingungen angenähert nachgebildet werden kann. Der Einfluß der einzelnen Sinus- und Kosinusfunktionen wird durch Frequenz und die Amplitude bestimmt.

INPUT-Graph muß bei diesen Verfahren mit komplexen Zahlen hantieren, wozu der C64 von Haus aus nicht in der Lage ist. Der Versuch mit $SQR(-1)$ zu rechnen, endet unweigerlich mit einem „ILLEGAL QUANTITY ERROR“. Unser Programm behilft sich, indem es für diese Operationen in der ersten Hälfte des Zahlenfeldes (y-Feld) den Realteil, in der zweiten Hälfte den Imaginärteil ablegt. Der Realteil enthält die Koeffizienten der Kosinusfunktion, der Imaginärteil die der Sinusfunktion.

Praktischer Beweis

Wählen Sie „Rechnen“ auf dem Bildschirm an und löschen dort erst einmal sämtliche 1024 Werte des gesamten Feldes, und zwar von Null an. Setzen Sie anschließend die Feldgrenzen für Y-Feld auf '512' und X-Feld auf „zählen“.

Unterhalb des Menüpunktes '9' geben Sie die folgende Funktion ein:

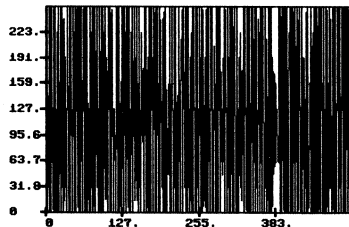
$$\sin(2 * \pi * x / 512)$$

Verlassen Sie das Menü über „QUIT“ und schalten Sie das Grafikmenü ein. Bestätigen Sie „Bildschirm, Linien“ mit RETURN oder dem Feuer-Knopf des Joysticks in Port 2, übernehmen die Grenzwerte unter „Weltfenster“ mit der f3-Taste, RETURN und wieder der f3-Taste und starten die

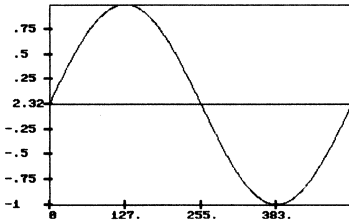
grafische Darstellung mit der f1-Taste. Sie erhalten eine Darstellung, die Bild 1 entspricht.

Um der Fourier-Analyse auf den Zahn zu fühlen, wählen Sie wieder das Menü über „QUIT“ und „Rechnen“ an. Dort schalten Sie unter „Feldgrenzen setzen“ um auf „X-Daten aus Feld“ und setzen „X-Anfang“ auf 512. Löschen Sie vorsichtshalber 512 Werte im Feld ab 512.

Über „Transformation / Approximation“ geht es zur Fourier-Analyse. Es dauert einen kleinen Moment, bis INPUT-Graph diese Berechnungen abgeschlossen hat. Erscheint der „Zeigefinger“ wieder, können Sie sich das Ergebnis als Grafik anschauen oder weiter arbeiten, indem Sie den „Betrag des Spektrums“ ausrechnen lassen.



Kurzes gezichtigtes „S“



Glatter Sinus ...

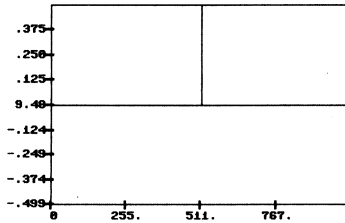
Damit Fourier-Analyse und -Synthese nicht unerträglich lange dauern, wird in INPUT-Graph der sogenannte FFT-Algorithmus (Fast-Fourier-Transformation) verwendet. Dieser erfordert jedoch, daß die Anzahl der Daten immer eine Potenz von 2 sein muß. Es lassen sich deshalb maximal 512 komplexe Werte von INPUT-Graph transformieren, da nur höchstens 1024 Feldelemente zur Verfügung stehen. Dieser Punkt funktioniert also nur, wenn ein y-Feld und ein x-Feld definiert sind und diese 16, 32, 64, 128, 256 oder 512 Elemente lang sind. Da dieses eine der ersten Implementationen einer FFT auf dem C64 ist, ist hiermit auch der Wettlauf um die Rechenzeit eröffnet: INPUT-Graph braucht für die 512 Punkte FFT 35 Sekunden. Wer kann es schneller?

Analysieren Sie eine einfache Sinus-Funktion, wie im Kasten text „Praktischer Beweis“ beschrieben. Sie erhalten ein Spek-

trum mit nur einem einzigen Wert, aus dem Sie ablesen können, daß die Amplitude der definierten Funktion '1' ($2 * 0.5$) beträgt, während die Frequenz dieser Funktion '1' ausmacht.

Vorstellbar

Beim „Fourier-Spiel“ finden Sie im Kasten text ein Beispiel, bei dem Sinusschwingungen entsprechend der Fourier-Funktion überlagert werden. Dies entspricht der Reihe, die in Vollendung zu einem Rechteck-



... und das Spektrum

signal führt. Diese Funktion haben wir bereits in der Ausgabe 9/87 vorgestellt. Bei dieser Funktion fehlt der Teil der Kosinusfunktionen. Der Koeffizient 'ko' verändert sich entsprechend der harmonischen Reihe (1, 3, 5, ...).

Wenn Sie nach diesen Vorerfahrungen die Fourier-Analyse auf eins der angebotenen Beispiele anwenden, erhalten Sie im

X-Feld (0-511) und im Y-Feld (512-1024) Daten, die physikalisch das Frequenz-Spektrum der jeweiligen Samples darstellen. Die Amplituden der einzelnen Frequenzen werden in y-Richtung, die Frequenzverhältnisse in x-Richtung aufgetragen. In der ersten Hälfte des gewählten Bereichs stehen die Koeffizienten für die Kosinusfunktionen, in der zweiten Hälfte die Sinus-Koeffizienten. Man kann also direkt ablesen, in welchem Verhältnis die Frequenzanteile zueinander stehen.

Studenten, die sich in die Theorie der Fourier-Transformation einarbeiten müssen, werden für die Möglichkeit der praktischen Erfahrung dankbar sein.

Da man sich zu Analysezwecken nicht nur für Real- und Imaginärteil, sondern auch für den Betrag des Spektrums interessiert, gibt es dafür einen eigenen Menüpunkt. Das Ergebnis wird im Y-Feld abgelegt, der Realteil wird jedoch durch Zwischenergebnisse zerstört.

Annäherung

Die **Approximation** dient dazu, Funktionen zu finden, die den Signalverlauf angenähert wiedergeben. Diskrete Meßwerte können durch Approximation mit einer Geraden oder durch Rektifikation der Ebene mit aus-gesuchten zweiparametrischen Funktionen in einen Formelzusammenhang überführt werden.

Durch Anwählen einer Funktion beginnt die Berechnung, die mit Ausgabe der beiden Parameter a0 und a1 sowie des Korrelationskoeffizienten B endet. Dieser Koeffizient ist ein Maß für die Güte der Näherung. Je näher B an '1' liegt, desto brauchbarer sind die gefundenen Parameter. B sollte '0.9' nicht unterschreiten. Damit die Berechnung zum Erfolg führt, dürfen die X- und die Y-Daten keine negativen Zahlen enthalten und nicht Null sein.

Mit den übrigen Verfahren im Rechenmenü: „Die Konstante“ und die Punkte 1)–4) können Sie die Rohdaten anpassen. Die zuletzt angewählte Funktion steht anschließend im Rechenmenü zur Auswertung mit den gefundenen Parametern bereit und kann zum Vergleich mit den diskreten Werten gezeichnet werden, indem bei der grafischen Darstellung „Überlagern“ gewählt wird.

F. Dreismann/RH

Fourier-Spiel

Im Menü „Rechnen“ können Sie die Wirkung von überlagerten Sinuskurven untersuchen. Geben Sie zuerst unterhalb des Menüpunkts '9' folgende Formel ein:

$$f(x) = 1/ko * \sin(ko * 2 * \pi * x / 512)$$

Die Konstante oben im Rechenmenü setzen Sie im vorläufig auf '1'.

Die etwas merkwürdige Sinusformel erlaubt es, auf das Y-Feld $f(x)$ immer weitere Funktionen aufzuaddieren. In der Ausgabe 9/87 wurde die Bedeutung der Variablen unter „BASIC-Unterstützung“ auf Seite 18 beschrieben. 'ko' ist hier die

Konstante. Vergewissern Sie sich, daß das X-Feld auf „zählen“ geschaltet ist.

Eröffnen Sie über „9 eigene Funktion berechnen“ den Rechenvorgang. Nach dem ersten Durchgang erhalten Sie die bereits bekannte Sinuskurve. Verändern Sie die Konstante jeweils in Zwischenschritten (3, 5, 7, ...). Bei der grafischen Darstellung wird die Entwicklung der Berechnungen besonders anschaulich, wenn Sie die Kurven überlagern. Sie können diesen Vorgang beliebig oft wiederholen, um die allmähliche Annäherung an eine Rechteckschwingung zu beobachten.



Drei Riesen

INPUT 64 sucht das Abenteuer

Versuchen Sie es doch mal mit „Schreibe Adventure“. Mit etwas Glück honorieren wir Ihnen diese Antwort mit dreitausend Mark. Das ist nämlich der Preis, den wir für das beste Abenteuerspiel zahlen, das uns bis zum 15. Januar nächsten Jahres erreicht. Damit geben wir Ihnen zusätzlich zu unserem 'normalen' 3000-DM-Wettbewerb eine weitere Chance, uns diesen Betrag aus der Tasche zu ziehen.

Riesenpreis

Ein Vierteljahr haben Sie Zeit, um sich eine spannende Geschichte auszudenken und daraus eine harte Nuß für Adventure-Freaks zu machen. Das Thema ist dabei völlig freigestellt. Ob es nun gilt, einen Schatz aus den Tiefen des Amazonas-Dschungels zu bergen, ein verschollenes Raumschiff in der letzten Ecke der Galaxis zu finden oder eine Prinzessin aus den Klauen eines siebenköpfigen Drachen zu befreien, ist Ihnen überlassen. Nur kriegerische Auseinandersetzungen und Spiele nach dem Motto „Leichen pflasterten seinen Weg“ haben von vornherein keine Chance auf den Wettbewerbssieg.

Wie es sich für gute Adventure-Spiele gehört, sollte Ihr Beitrag mit einer anspre-

Sie befinden sich in einem Zimmer. Draußen ist es dunkel, aber im Zimmer brennt eine Lampe. Vor sich sehen Sie einen Schreibtisch mit einem Computer und der letzten Ausgabe von INPUT 64. Eine Tür im Westen führt auf einen Flur. Was wollen Sie tun?

chenden Grafik ausgestattet sein. Für Animationen, die womöglich noch mit „astreinen Sounds“ untermalt sind, vergibt die Jury natürlich Extrapunkte.

Großen Wert legen wir auf intelligente Reaktionen auf (Fehl-)Eingaben. Nichts ist nervtötender, als bei jedem zweiten Befehl mit „Das geht jetzt nicht“, abgespeist zu werden. Je mehr Wörter Ihr Spiel versteht und je differenzierter es auf eingegebene Sätze antwortet, desto höher wird es in der Wertung liegen. Daß ein Programm, das auf Fehleingaben mit einem blauen Bildschirm und 'READY' reagiert oder dessen Bildschirm sich durch unkontrollierte Cursor-Bewegungen zerstören läßt, nicht in die engere Wahl kommt, braucht wohl nicht extra erwähnt zu werden.

Riesenspaß

Abgesehen von der technischen Qualität Ihrer Einsendung ist natürlich vor allem der Spielspaß entscheidend für die Bewertung. Dieses Kriterium ist zugegebenermaßen subjektiv, wird aber doch von einigen meßbaren Faktoren beeinflusst. So freut sich der eingefleischte Adventure-Knacker, wenn er den Spielstand vor einer riskanten Aktion abspeichern kann. Großer Beliebtheit erfreuen sich auch Abenteurer, die man mit mehreren Spielern im Team oder im Wetlauf gegeneinander lösen muß.

Adventure-Spiele gewinnen an Spannung, wenn einige Aufgaben in einer vorgegebenen Zeit oder mit einer bestimmten Höchstanzahl von Eingaben gelöst werden müssen. Auch Einlagen, die ein geschicktes Hantieren mit dem Joystick erfordern, sind immer willkommen. Dagegen wird aus Abenteuerlust schnell Spielfrust, wenn der Lösungsweg zu starr eingehalten werden muß und jedes Abweichen auf dem berühm-t-berichtigten RIP-Bild endet.

Kleine Bitte

Zum Schluß einige technische Hinweise: Senden Sie uns Ihren Beitrag bitte auf Datenträger ein – zum Abtippen von Listings haben wir keine Lust, und einen Drucker haben wir selbst. BASIC-Programme sind bitte weder kompiliert noch kompaktiert. Wollen Sie eine BASIC-Erweiterung verwenden, dann halten Sie bitte vorher mit uns Rücksprache. Bei Assembler-Programmen gehört der kommentierte() Quellcode dazu.

Die besten Einsendungen dieses Wettbewerbes sollen natürlich in INPUT 64 veröffentlicht werden und auch innerhalb des Magazins laufen. Dazu sind einige Einschränkungen in der Speicherplatzbelegung und im Umgang mit Betriebssystem-Vektoren erforderlich. Bei reinen BASIC-Programmen gibt es in der Regel keine Probleme, aber wenn Sie Ihren Beitrag in Maschinensprache schreiben wollen, sollten Sie vorsorglich unsere Autorenhinweise anfordern. Unsere Adresse finden Sie im Impressum. An diese schicken Sie bitte auch – mit dem Stichwort 'Adventure-Wettbewerb' versehen – Ihre fertigen Programme. HS

Allzeit bereit

Hex-Dezimal-Binär-Umrechnung auf Tastendruck

Bislang mußten Sie den Cursor aus der schon begonnenen Zeile fahren, mit diversen Befehlen die dezimale Darstellung der gewünschten Zahlen berechnen, um sie dann in die möglicherweise schon aus dem Bild gescrollte Zeile einzugeben. Oder Sie begaben sich auf die Suche nach einem Taschenrechner und hatten fortan zwei Tastaturen zu bedienen. Unser Hex-Dezimal-Binär-Tool stellt Ihnen auf Tastendruck die Umrechnung zwischen den drei Zahlensystemen zur Verfügung, und zwar in jeder beliebigen Richtung.

Dezimal ist keine Hexerei

Als regelmäßiger INPUT 64-Leser sind Sie es ja gewohnt, mit der Tastenkombination CTRL und H um Hilfe zu rufen. Genau dieselben Tasten dienen bei diesem Tool dazu, oben in den Bildschirm eine Hilfszeile einzublenden. In dieser erscheint

\$0000 = 00000 = %00000000 00000000

Das ist Ihnen sicher auch schon passiert: Sie sitzen vor dem Rechner und wollen in einem BASIC-Programm Sprites über den Bildschirm huschen lassen. Dazu müssen in der Speicherstelle \$D015 die Bits 0 und 3 gesetzt werden. Leider versteht der BASIC-Befehl POKE nur dezimale Argumente.

in allen drei Zahlensystemen, und der Cursor blinkt auf der ersten Ziffer. Sie können nun mit den Cursor-Tasten von Ziffer zu Ziffer springen. Geben Sie an einer beliebigen Stelle eine erlaubte Ziffer ein, so wird die dadurch entstehende Zahl unmittelbar in die beiden anderen Zahlensysteme umgerechnet und angezeigt.

Erlaubt sind in den ersten vier Stellen die Hexadezimal-Ziffern 0 bis 9 und A bis F, bei den fünf Stellen der Dezimalanzeige nur die Ziffern 0 bis 9 und in den letzten sechzehn

Stellen 0 und 1. Beachten Sie bitte, daß das Programm nur Zahlen bis 65535 verarbeiten kann. Bei der Eingabe in der Dezimalanzeige kann es also vorkommen, daß eine ansonsten erlaubte Ziffer abgewiesen wird. Hier hilft die Eingabe einer kleineren Zahl an der Zehntausenderstelle.

Zum Verlassen der Hilfszeile dienen die Tasten RUN/STOP oder SPACE (Leertaste). Bei ersterer springt der Cursor nur zu der Stelle zurück, an der er beim Druck auf CTRL-H stand. Betätigen Sie die Leertaste, wird zusätzlich der alte Inhalt der ersten Zeile wiederhergestellt.

CTRLiertes Abspeichern

Innerhalb von INPUT 64 können Sie die Routine schon mal ausprobieren, aus verständlichen Gründen erreichen Sie sie aber nicht mit CTRL-H. Drücken Sie statt dessen **C** und H.

Abspeichern können Sie das Programm wie gewohnt mit CTRL und S. Dabei haben Sie die Wahl zwischen zwei Versionen: Die erste können Sie als BASIC-Programm laden und mit 'RUN' initialisieren. Die zweite wird absolut (mit LOAD "name",8,1) geladen und mit SYS 49152 initialisiert. Diesen Befehl können Sie übrigens bei beiden Versionen benutzen, um das Programm nach einem Reset oder dem Betätigen der Tasten RUN/STOP und RESTORE neu zu initialisieren. HS

Lernen im Dialog

Englische GRAMmatik

Viele Fehlerfallen hat die Autorin noch einmal in die letzte Folge des interaktiven Grammatikkurses eingebaut. Es geht um Sinn- und Wertunterscheidungen wie die, ob Sie auch verdienen, was Sie verdienen.

Eine ganze Reihe englischer Verben sind beliebte Stolpersteine für Deutschsprachige. Geübt werden kann der richtige Einsatz von

- to do or to make
- to say or to tell
- to get, to receive or to become
- to have, to let or to make
- to bring or to take
- to want or to like

- to remind or to remember
- to earn or to deserve

Die Bedienung des Lernprogramms ist denkbar einfach: Die richtigen Eingaben müssen jeweils in die Textlücken der Beispielsätze eingegeben werden (mit RETURN abschließen); nach jeder Eingabe können Sie entweder eine der auf dem Bildschirm gezeigten Möglichkeiten wählen oder mit einer beliebigen anderen Taste mit der nächsten Frage fortfahren.

Die Bezeichnung „Lernprogramm“ ist übrigens etwas irreführend. Es werden nämlich nicht die grammatischen Regeln oder gar deren Hintergrund vermittelt, sondern es geht darum, vorhandene oder im Lauf der Zeit verschüttete Kenntnisse einzuüben beziehungsweise aufzufrischen. JS

Kraftfutter für den Commodore 64.

Wie heißt's so schön: Auf die Dauer hilft nur Power. Deshalb ran ans Kraftfutter. Raus mit dem Commodore. Rein mit INPUT 64. Super-Sonder-Posten aus Lagerbeständen jetzt im Power-Pack zum Knüller-Preis. Auf Cassette wie Diskette. Am besten heute noch Coupon ausfüllen. Und ab geht die Post. Ach ja: Bezahlt wird im voraus. Per Scheck. Klar?!



Kommt, wie bestellt. INPUT 64.

Cassette oder Diskette. 5er oder 10er Pack.
Alles INPUT 64-Ausgaben, die's in sich haben.
Gewünschtes einfach ankreuzen.

Auf Diskette:

5er Pack DM 25,-	10er Pack DM 46,-
<input type="checkbox"/> 4/85..... bis..... 8/85	<input type="checkbox"/> 8/85
<input type="checkbox"/> 9/85..... bis..... 1/86	<input type="checkbox"/> 1/86
<input type="checkbox"/> 3/86..... bis..... 7/86	<input type="checkbox"/> 7/86
<input type="checkbox"/> 8/86, 9/86, 9/85, 10/85, 10/85, 11/85	<input type="checkbox"/> 11/85

Auf Cassette:

5er Pack DM 13,-	10er Pack DM 25,-
<input type="checkbox"/> 2/85..... bis..... 6/85	<input type="checkbox"/> 6/85
<input type="checkbox"/> 7/85..... bis..... 11/85	<input type="checkbox"/> 11/85
<input type="checkbox"/> 12/85..... bis..... 4/86	<input type="checkbox"/> 4/86
<input type="checkbox"/> 5/86..... bis..... 9/86	<input type="checkbox"/> 9/86
<input type="checkbox"/> 10/86..... bis..... 2/87	<input type="checkbox"/> 2/87
<input type="checkbox"/> 3/87..... bis..... 7/87	<input type="checkbox"/> 7/87

Versand: zzgl. DM 3,-

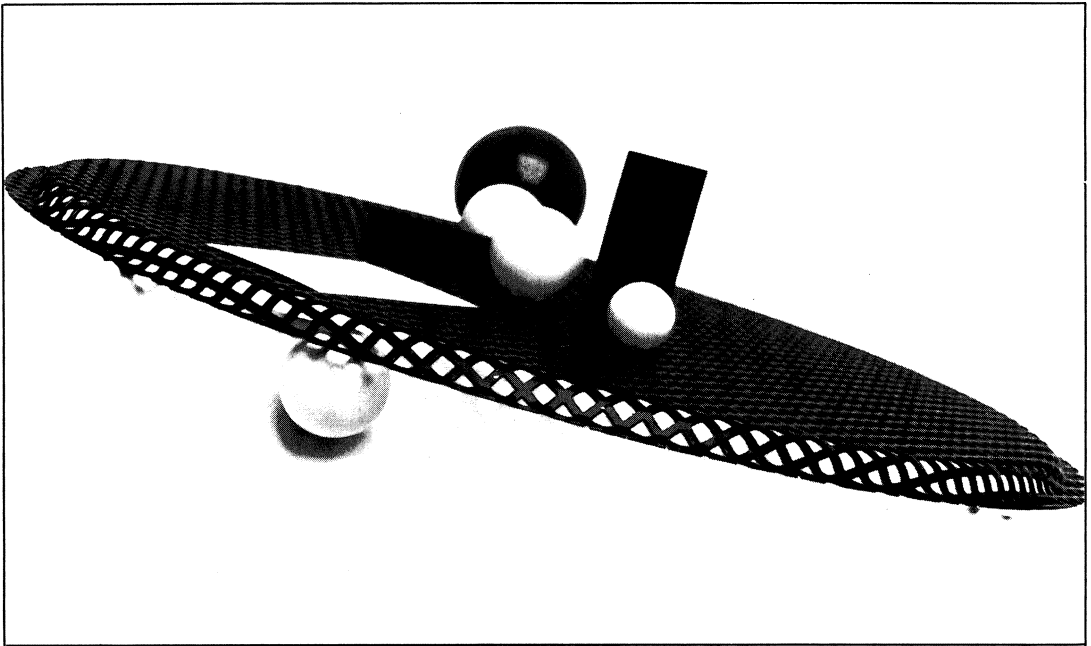
Name:

Straße:

Ort:

An: Verlag Heinz Heise GmbH, Vertriebsabteilung,
Postfach 61 04 07, 3000 Hannover 61

Und ... Scheck nicht vergessen!



Wenn die Kugel rollt

Spiel: Bouncing Ball

Bei dem Spiel Bouncing Ball kommt es darauf an, einen kleinen Ball durch acht verschiedene Spielfelder zu manövrieren. Dabei gilt es, möglichst viele Punkte zu sammeln. Der Punktestand erhöht sich mit jedem eingesammelten Bonus. Diese sind als Zahlen und Buchstaben in den Gängen verteilt. Doch beginnen wir am Anfang: Sie können Bouncing Ball innerhalb des Magazins mit der Einschränkung spielen, daß der Spielfeldeditor nicht erreichbar ist. Möchten Sie eigene Spielfelder entwerfen, können Sie das Programm wie bei INPUT üblich mit CTRL-S auf eine eigene Diskette abspeichern. Von einer eigenen Diskette gestartet, erscheint dann im ersten Menü die Option „Editor starten“; wie dieser bedient wird, soll später erklärt werden.

Spannung und Nervenkitzel verspricht der Weg durchs Labyrinth, den Sie ihrem Ball bahnen sollen. An jeder Ecke lauern neue Gefahren und Abenteuer, die es zu meistern gilt. Kleine gefräßige Monster, schwarze Löcher und hinterhältige Sackgassen gehören zu den geringsten Schwierigkeiten. Und sollten Ihnen die acht vorhandenen Spielfelder nicht reichen, lassen sich mit dem integrierten Spielfeldeditor unendlich viele eigene Spiele entwerfen.

Das Titelbild wird mit einem Druck auf die Leertaste abgebrochen. Auswählen ist mit einem Joystick in Port 2 oder den F-Tasten möglich. Der grüne Balken wird mit F1/F3 positioniert, eine Auswahl mit RETURN bestätigt. Das Spiel ist für ein bis zwei Spieler ausgelegt, zu zweit besteht die Möglichkeit, getrennt hintereinander oder mit vereintem Geschick miteinander zu spielen.

Allein oder zu zweit

Jeder Spieler bekommt anfänglich vier Spielbälle; die werden auch gebraucht, um alle Levels bis zum Ende zu meistern. Die Bahn der Kugel wird durch die Stellung der

Welche Taste macht was?

	Joystick	Hoch	Runter	Links	Rechts	Auswahl
Menü/						
Editor	2	F1	F3	F5	F6	RETURN
Spieler 1	2	W	S	X	C	Knopf
Spieler 2	1	I	K	Komma	Punkt	Knopf

Weichen bestimmt, die mit dem Joystick verändert wird. Eigentlich ganz einfach! Doch hüten Sie sich vor schwarzen Löchern im Spielfeld, da fallen Sie einfach durch! Ebenso unangenehm sind die in Ecken kauernenden Ungetüme, umherstreunende Monster machen das Leben gefährlich. Irgendwo ist am Spielfeldrand eine Öffnung, hier geht's zum nächsten Level. Der Versuch, dieses ohne genügend eingesammelte Punkte zu erreichen, endet erfolglos, aber mit einem Ballverlust.

Immer der Reihe nach

Ist genügend Bonus gesammelt, flimmert kurz der Bildschirmrahmen, damit Sie beruhigt in den nächsten Level wechseln können. Bei einigen Spielfeldern kommt erschwerend hinzu, daß in der richtigen Reihenfolge gesammelt werden muß. Die Eins vor der Zwei und diese natürlich vor der Drei; aber auch das A vor dem B und so weiter. Damit es nicht zu bösen Überraschungen kommt, erscheint vor jedem entsprechenden Level ein freundlicher Hinweis: „Bonusse in Folge sammeln!“ Sollte es in solch einem Level vorkommen, daß eine Zahl (oder ein Buchstabe) doppelt vorkommt, ist Vorsicht angesagt. Es kann jede Zahl nur einmal genommen werden! „In Folge sammeln“ bedeutet: in streng aufsteigender Folge!

Ohne Ruhepause . . .

Sie irren sich, wenn Sie meinen, beim Eintritt in das nächsthöhere Feld wäre Ihnen eine Verschnaufpause gegönnt. Nein – es geht gleich mit gestiegenem Schwierigkeitsgrad weiter. Besonders beim Eintritt in den nächsten Level ist erhöhte Aufmerksamkeit gefordert. Wer es nicht schafft,

sich schnell zu orientieren, verliert gleich hier einen Ball. Vorsicht im dritten Level: von einem Monster verfolgt auf ein schwarzes Loch zusteuernd, ist blitzschnelles Entscheiden und Handeln angesagt! Manchmal kann der Ball in eine sichere Gasse gesteuert werden. Sind keine Gefahrenquellen in der Nähe, ist kurzes Pausieren erlaubt und ratsam. Diese Möglichkeit ist besonders in höheren Level vonnöten, damit der Durchblick gewahrt bleibt. Gerade hier ist der Verlust eines Balls schmerzlich, und das Durchspielen des gesamten Spiels rutscht gleich wieder in weite Ferne. Aber mit viel Geduld und der nötigen Ruhe können Sie es schon schaffen. Klappt nicht alles auf Anhieb, ärgern Sie sich nicht – Spiele sind doch zur Entspannung gedacht!

. . . aber nie langweilig

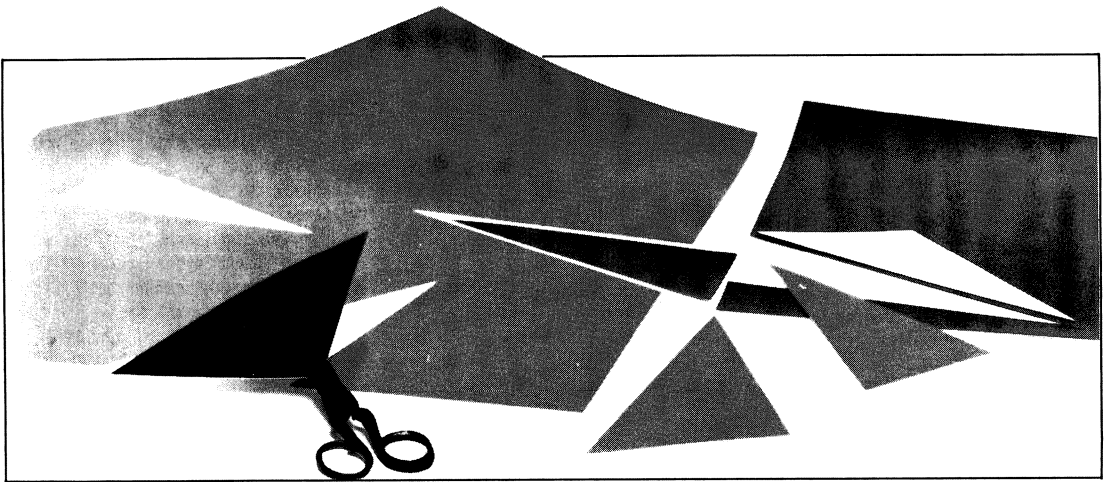
Alle Levels im Schlaf geschafft und Hunger nach neuem Nervenkitzel? Kein Grund, das Programm zur Seite zu legen! Mit Hilfe des integrierten Editors lassen sich eigene Spielfelder kreieren. Dabei sind Ideen und Einfallsreichtum gefragt, ganz einfach ist es allerdings nicht, ein interessantes und spielbares Labyrinth zu entwerfen. Doch macht es auch Spaß, vertrackte und manchmal auch ein bißchen verflixte Spielfelder zu basteln, an denen nicht nur andere ihr Geschick beweisen können.

Wie oben schon erwähnt, ist der Spielfeldeditor nur außerhalb von INPUT erreichbar! Der Einfachheit halber beschreiben wir die Bedienung mit einem Joystick, selbstverständlich geht es auch mit Tasten. Eine durchdachte Konzeption macht den Editor einfach in der Bedienung. Nach dem Editor-Start erscheint die Aufforderung, ein Spielfeld auszuwählen. Nachdem der Joystick-Knopf einmal betätigt wurde, kann mit

links/rechts eine Feldnummer gewählt werden, bestätigt wird wieder mit dem Feuerknopf. Am unteren Bildschirmrand sind nun alle verfügbaren Konstruktionselemente eingeblendet. Deren Bedeutung ist Ihnen zu diesem Zeitpunkt ja sicherlich aus der eigenen Spielerfahrung bekannt. Der in der oberen linken Ecke erscheinende Cursor läßt sich per Joystick über den Bildschirm bewegen. Wird dieser auf einen Baustein in der untersten Zeile gestellt und gleichzeitig die Feuertaste betätigt, ist das Element ausgewählt und kann in das Spielfeld gesetzt werden. Dazu wird der Cursor auf die entsprechende Stelle gestellt und der Joystick-Knopf gedrückt. Stein für Stein entsteht so ein neues Feld. Damit zwei Spieler gleichzeitig spielen können, müssen sowohl rote als auch lila Weichen vorhanden sein! Die vier rechten Felder der Statuszeile haben besondere Funktionen. Die beiden 'T' schalten in den Textmodus um, damit können Zahlen und Texte gesetzt werden. Cursor an die gewünschte Stelle bringen und das entsprechende Zeichen über die Tastatur eingeben. Über das mit 'M' bezeichnete Symbol wird ein Menü aufgerufen. Hier können so wichtige Fragen wie Startposition, -richtung und -geschwindigkeit des Balls eingegeben werden. Außerdem besteht hier die Möglichkeit, unter den entsprechenden Menüpunkten Spielfelder auf Diskette abzuspeichern oder einzuladen. Der letzte Punkt verdient erhöhte Aufmerksamkeit! Dieser erlaubt das oben editierte Feld in einem besonderen Modus zu testen. Hierbei wird die Güte der theoretischen Konstruktion einem praktischen Test unterzogen. Sollte sich zeigen, daß sogenannte „death ends“ eingebaut wurden, kann man mit der RUN/STOP-Taste abbrechen und eine Korrektur vornehmen. Der Editor wird verlassen über das 'X' in der Statuszeile.

Selbstgestricktes genießen

Nachdem ein Bild nach dem nächsten gereift ist, können acht Screens zu einem Spiel zusammengefaßt werden. Um ein eigenes Spiel zu spielen, muß jedesmal der Weg über den Editor gegangen werden. Das Menü auswählen, mit „Alle Spielfelder laden“ das Spiel in den Rechner holen und über X den Editor wieder verlassen. Nunmehr stehen die selbstgestrickten Spielfelder zur Verfügung. pan



ID-Werkstatt

Anregendes und Programmpflege

Doch zuerst zu den drei BASIC-Programmen. Die eigentliche Bedienung ist sehr einfach. Nach dem Laden und Starten werden Sie aufgefordert, den Programm-Namen des zu bearbeitenden Files einzugeben. (Bei dem Programm „RENUMBER“ können Sie gleich mehrere File-Namen nacheinander eingeben.)

Experimentieren mit BASIC-Programmen

Wenn Sie die Programme aus der ID-Werkstatt auf Ihren eigenen Datenträger überspielt haben, liegen diese in der kürzest möglichen Form vor, also renumbert und kompaktiert. In dieser Form sind sie zwar lauffähig, aber nur schwerlich zu durchschauen. Um die Programme richtig analysieren zu können, sollten Sie diese „mit sich selbst“ bearbeiten, das heißt: mit einer hohen Schrittweite renummern (50 sollte genügen) und anschließend dekompaktieren.

Beachten Sie bitte, daß alle drei Hilfsprogramme das BASIC-Ende (jeweils in der

Hilfsprogramme, die BASIC-Programme kompaktieren und renummern, haben wir schon veröffentlicht. Diese Tools waren aber nur als „Black box“ zu nutzen. Ganz anders hingegen die kleinen Programme aus dieser ID-Werkstatt. Sie sind vollständig in BASIC geschrieben, zwar sehr langsam, aber dafür nachvollziehbar. Um die Sache abzurunden, kann das dritte Programm ein einmal kompaktiertes Programm wieder dekompaktieren. Neben diesen drei echten Werkstatt-Programmen zum Analysieren, Verfeinern und Weiterstricken finden Sie diesmal in dieser Rubrik das Patch-Programm für INPUT-Calc 64/128.

zweiten Zeile) heruntersetzen, um sich einen Arbeitsspeicher zu reservieren. Es kann der Fall eintreten, daß die Programme in einer „Langform“ nicht mehr mit dem für das eigentliche Programm verbleibenden Speicherbereich auskommen können. In

diesem Fall müssen Sie die Grenzwerte und die Variablen 'c' beziehungsweise 'co', die auf den Beginn des Arbeitsspeichers verweisen, entsprechend ändern.

Hier wäre zum Beispiel eine Änderung dergestalt denkbar, daß sich die Variablen 'c' und 'co' ihren Wert selbständig aus der Adresse 55 und 56 holen. Wie schon angedeutet, liegt der Reiz dieser Programme nicht in ihren Möglichkeiten und Funktionen (das Zeitverhalten liegt an der Grenze des Zumutbaren), sondern in der Durchschaubarkeit des Programmablaufs und damit verbunden in der Möglichkeit, gestalterisch eingreifen zu können. **WM**

Calc auf den neusten Stand gebracht

Hier können Sie sich auf den C64 verlassen. Das Patch-Programm führt sämtliche Änderungen an INPUT-Calc 64/128 selbsttätig durch. Verwenden Sie nach Möglichkeit die Version von INPUT-Calc 64/128, die Sie sich aus der Ausgabe 8/87 abgespeichert hatten.

Nachdem Sie „Calc-Patch“ aus der ID-Werkstatt auf eigenen Datenträger gespeichert haben, steht einer Änderung von INPUT-Calc 64/128 außerhalb von INPUT 64 nichts mehr im Wege. Sie laden das abgespeicherte Patch-Programm, starten es mit 'RUN' und richten sich nach den Angaben auf dem Bildschirm. **RH**

Jetzt geht's rund

Kreisalgorithmen

Mit dem Wort „rund“ verbinden die meisten Menschen sofort den Gedanken an einen Kreis. Diese geometrische Figur empfinden wir als angenehm, und sie ist entsprechend oft in der Alltagsumgebung anzutreffen. Weniger ästhetische als physikalische Gründe bedingen Kreisformen bei jeder Art der Drehbewegung, wie zum Beispiel beim Rad.

Wer einen Kreis zu Papier bringen will, nimmt dazu einen Zirkel und kann sich weitere Denkarbeit ersparen. Der C64 kommt mit solchem Hilfsgerät nicht zu recht. Er hätte es lieber mathematisch formuliert. Die korrekte Definition eines Kreises, die auch der Funktionsweise eines Zirkels zugrundeliegt, lautet:

„Ein Kreis ist die Menge aller Punkte, die von einem Mittelpunkt M den konstanten Abstand r haben.“

Aus der Schule geplaudert

Wenn Sie sich Bild 1 ansehen und sich an den guten alten Satz des Pythagoras zurückerinnern, werden Sie erkennen, daß ein

In den letzten beiden Folgen der 64er Tips ging es darum, Punkte und Linien in der HiRes-Grafik des C64 darzustellen. Als weitere geometrische Grundformen sind diesmal Kreise und Ellipsen an der Reihe. Damit steht auch der Geburtstagstorte auf dem Bildschirm nichts mehr im Weg.

Kreis um den Ursprung des Koordinatenkreuzes durch die Gleichung

$$x^2 + y^2 = r^2$$

beschrieben wird. Löst man diese Gleichung nach y auf und formuliert sie in BASIC, dann hat man schon das Kernstück für das erste Programm zur Erzeugung eines Kreises auf dem Bildschirm:

$$Y = \text{SQR}(R^2 - X^2)$$

Dabei nimmt die Variable X nacheinander die Werte von -R bis R an. Die berechneten Y-Werte können dazu benutzt werden, die obere Hälfte eines Kreises um den Koordi-

natenursprung zu zeichnen. Soll der Mittelpunkt woanders liegen, müssen seine Koordinaten noch zu X beziehungsweise Y addiert werden. Die untere Kreishälfte ergibt sich, wenn der berechnete Y-Wert von der Y-Koordinate des Mittelpunktes abgezogen wird.

```

1000 FOR X=-R TO R
1010 Y = SQR(R^2 - X^2)
1020 SET XM+X, YM+Y : SET XM+X, YM-Y
1030 NEXT X
1040 RETURN
    
```

Listing 1: So malt der Mathematiker Kreise

Die komplette Routine sehen Sie in Listing 1. Die Variablen R, XM und YM sind außerhalb des Unterprogramms definiert. XM und YM enthalten die Koordinaten des Kreismittelpunktes. Um zu sehen, was passiert, muß außerdem vor dem Aufruf dieser Routine die Grafik eingeschaltet sein. Der Befehl SET in Zeile 1020 ist Bestandteil von INPUT-BASIC (haben wir in der Ausgabe 1/86 veröffentlicht) und dient zum Setzen eines Punktes in der hochauflösenden Grafik. Wenn Sie das Programm ausprobieren wollen, müssen Sie hier den entsprechenden Befehl Ihrer Grafik-Erweiterung einsetzen. Oft heißt dieser Befehl 'PLOT', bei der Verwendung der Grafik-Tools aus den 64er - Tips nehmen Sie den entsprechenden SYS-Befehl.

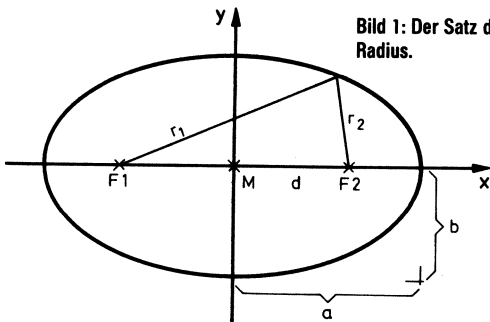


Bild 1: Der Satz des Pythagoras liefert den Radius.

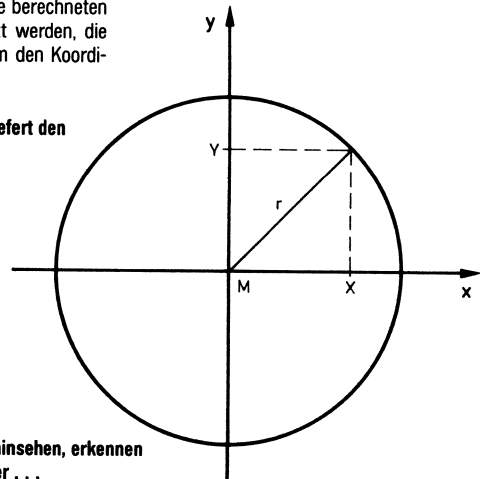


Bild 2: Wenn Sie genau hinsehen, erkennen Sie zwei Löcher im Papier . . .

Gestaucht

Wird ein Kreis gestaucht oder gestreckt, dann wird er zu einer Ellipse. Hobbyastronomen ist diese Figur wahrscheinlich sehr vertraut, da die Planeten des Sonnensystems elliptische Bahnen ziehen. In der Computergrafik werden Ellipsen oft als perspektivische Darstellung eines Kreises verwendet. Tatsächlich erscheint jeder Kreis, der nicht senkrecht von oben betrachtet wird, als Ellipse. Daß wir daraus trotzdem einen Kreis erkennen, ist eine Abstraktionsleistung unseres Gehirns. Ebenso wird eine zweidimensionale Ellipse auf dem Bildschirm für den Betrachter zum in der Ebene liegenden Kreis. Auch für die Ellipse haben sich Mathematiker eine exakte Definition ausgedacht. Sie lautet:

„Eine Ellipse ist die Menge aller Punkte, deren Summe der Abstände von zwei Brennpunkten F1 und F2 konstant ist.“

Mit Nadel und Faden

Auf dem Papier kann man eine Ellipse folgendermaßen konstruieren: Man nimmt zwei Nadeln und steckt sie in einiger Entfernung voneinander in die Zeichenfläche. Dann befestigt man die Enden eines beliebig langen Fadens an jeweils einer Nadel und fährt mit einem Bleistift die Kurve ab, die sich ergibt, wenn man den Faden ständig unter Spannung hält. Die beiden Nadeln stellen dabei die beiden Brennpunkte der Ellipse dar, der Faden die konstante Summe der Abstände zu den Brennpunkten. Der Kreis ist eigentlich ein Spezialfall einer Ellipse und entsteht, wenn F1 und F2 zusammenfallen, das heißt, wenn Sie nur eine Nadel benutzen.

Die genaue Form und Größe der Ellipse wird durch den Abstand der beiden Nadeln und die Länge des Fadens bestimmt. Daraus ergeben sich die beiden anderen charakteristischen Größen einer Ellipse, ihre beiden Halbachsen. In Bild 2 sind alle diese Größen eingezeichnet: r1 und r2 sind die beiden Abschnitte des Fadens, F1 und F2 die beiden Brennpunkte der Ellipse. Ihr Abstand wird durch den Mittelpunkt M in zwei gleich große Abschnitte d zerlegt. Die längere Halbachse a ergibt sich aus der Verlängerung der Verbindung zwischen M und F2, die kleine Halbachse b steht in M auf ihr senkrecht.

Legt man M in den Mittelpunkt eines Koordinatenkreuzes, dann läßt sich die Ellipse mathematisch mit der Gleichung

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$$

beschreiben. Wie bei der Kreisgleichung hier nun die Formel für eine Ellipse mit den Halbachsen a und b in BASIC-Syntax:

$$Y = B * \text{SQR}(1 - X^2 / A^2)$$

X durchläuft hier die Werte von -A bis A, die entstehenden Y-Werte sind wieder für die obere Hälfte der Ellipse zuständig. Eine komplette Ellipsen-Routine, die diese Formel verwendet, sehen Sie in Listing 2.

```
2000 FOR X=-A TO A
2010 Y = B * SQR(1 - X^2/A^2)
2020 SET XM=X, YM=Y : SET XM+X, YM-Y
2030 NEXT X
2040 RETURN
```

Listing 2: ... und Ellipsen.

Die bislang vorgestellten Verfahren zur Kreis- und Ellipsenberechnung haben zwar den Vorteil, einigermaßen anschaulich zu sein, weil sie sich sehr eng an die mathematische Definition halten, aber besonders schöne Ergebnisse bringen sie nicht zustande. An den seitlichen Enden der Figuren scheinen immer einige Punkte zu fehlen. Abhilfe würde hier die Wahl einer kleineren Schrittweite für X oder die Verwendung eines LINE- statt des PLOT-Befehls schaffen. Es geht aber auch eleganter.

Langsam, aber gründlich

Wie nicht anders zu erwarten, muß dazu jedoch wieder die Mathematik herhalten. Sie lehrt uns, daß ein Kreis aufgefaßt werden kann als die senkrechte Überlagerung einer Sinus- und einer Kosinusfunktion. Für Programmierer hat diese Tatsache folgende Bedeutung: Läßt man eine Variable in einer sinnvollen Schrittweite (zu 'sinnvoll' kommen wir noch) die Werte von 0 bis 360 Grad annehmen, dann können ihre Sinuswerte als X- und die Kosinuswerte als Y-Koordinaten für einen Kreis verwendet werden.

Dieses Verfahren wird in Programm 3 verwendet. Die Variable W läuft von 0 bis 360 Grad, allerdings im Bogenmaß. In dieser

Ein neues Grafik-Tool

Auch aus diesen 64er Tips können Sie mit CTRL-S ein Tool abspeichern. In ihm sind die Routinen aus der letzten Folge noch einmal enthalten, aber die Routinen aus der Ausgabe 8/87 müssen Sie trotzdem vorher laden. Das Maschinenprogramm ist voll relokatable. Es liegt im Speicher immer 'unter' dem BASIC-Programm, wo der BASIC-Anfang dabei liegt, ist ihm aber egal. Dadurch können Sie auch noch andere Hilfsprogramme im Speicher haben, ohne daß sie mit unseren Grafik-Tools in Konflikt geraten. Das Programm, das Sie diesmal abspeichern können, läuft nur im Zusammenhang mit denen aus den 64er Tips der Ausgabe 8/87. Es unterstützt das Setzen eines Punktes, das Ziehen einer Linie und das Zeichnen von Ellipsen im HiRes- und im Multicolor-Modus. Dabei gibt es zwei Abweichungen gegenüber bekannten BASIC-Erweiterungen:

- 1) Der Punkt (0,0) liegt links unten und nicht etwa links oben.
- 2) Im Multicolor-Modus sind nach wie vor x-Koordinaten von 0 bis 319 möglich. Dafür entsprechen jeweils eine gerade und eine ungerade Koordinate demselben Bildschirm-punkt.

Linie: zeichnet eine Linie.

SYS la+73,x1,y1,x2,y2,fa

la: Ladeadresse, siehe Ausgabe 8/87

x1,y1,x2,y2: Anfangs- und Endpunkte der Linie. (x zwischen 0 und 319, y zwischen 0 und 199)

fa: Zeichenfarbe. (0 oder 1 für hochauflösend, 0 - 3 für Multicolor)

Plot: setzt einen Punkt.

SYS la+76,x1,y1,fa

la,x1,y1,fa: siehe oben

Ellipse: zeichnet einen Ellipsenbogen.

SYS la+79,xm,ym,a,b,sw,ew,step,fa

la,fa: siehe oben

xm,ym: Koordinaten des Mittelpunktes

a,b: Länge der Halbachsen

sw,ew: Start- und Endwinkel

step: Winkelschrittweite

```

3000 FOR W=0 TO 2*π STEP 1/R
3010 X = XM+R*SIN(W)
3020 Y = YM+R*COS(W)
3030 SET X,Y
3040 NEXT W
3050 RETURN

```

Listing 3: Mit den trigonometrischen Funktionen können Kreise und Ellipsen nach demselben Verfahren gezeichnet werden.

Form erwarten die Funktionen SIN und COS den Winkel. Zwischen Winkelangaben in Grad und im Bogenmaß kann man mit folgender Formel umrechnen:

(Winkel in Bogenmaß) = (Winkel in Grad)/180* π

Als sinnvolle Schrittweite für die einzelnen Punkte hat sich der Kehrwert des Radius erwiesen. Dadurch ist der Kreis auf dem Bildschirm immer 'dicht', es fehlen keine Punkte. Es werden jedoch auch nur so viele Punkte berechnet, wie auf dem Bildschirm erscheinen. Kleine Kreise sind daher schneller gezeichnet als große.

Diese Art der Kreisberechnung erfordert jedoch trotzdem eine relativ lange Zeit, da die transzendenten Funktionen einen hohen Rechenaufwand verlangen. Außerdem wird aufgrund von Rundungsfehlern die entstehende Figur nicht so rund, wie sie sein könnte, was bei der Demonstration im Modul gut zu beobachten ist. Die Methode hat aber den Vorteil, daß beliebige Ausschnitte aus Kreisen gezeichnet werden können, indem die Schleife nicht von 0 bis 360 Grad, sondern nur über den gewünschten Winkel läuft.

Mit derselben Methode können übrigens auch Ellipsen berechnet werden. Dazu muß die Variable R in Zeile 3010 nur gegen die waagerechte Halbachse A und in Zeile 3020 gegen die senkrechte Halbachse B getauscht werden.

Schnell und trotzdem rund

Speziell auf Rasterbildschirme zugeschnitten ist der Bresenham'sche Kreisalgorithmus. Eine genaue Beschreibung dieses Verfahrens finden Sie in unserer Schwesterzeitschrift c't, und zwar in der Septem-

berausgabe des letzten Jahres. Der Algorithmus arbeitet nach einem ähnlichen Prinzip wie der beim letzten Mal erläuterte Linienalgorithmus:

```

4000 DA=R-1 : X=0 : Y=R
4010 IF DA<0 THEN Y=Y-1 : DA=DA+2*Y
4020 SET XM+X,YM+Y : SET XM-X,YM+Y
4030 SET XM+X,YM-Y : SET XM-X,YM-Y
4040 SET XM+Y,YM-X : SET XM-Y,YM+X
4050 SET XM+Y,YM-X : SET XM-Y,YM-X
4060 DA = DA-2*X-1
4070 X = X+1
4080 IF X<=Y THEN 4010
4090 RETURN

```

Listing 4: Schnelle Kreise nach Bresenham

Der Anfangspunkt ist der oberste Punkt des Kreises. Von dort aus macht man einen Schritt nach rechts (X=X+1) und überprüft, ob schon ein Schritt nach unten erforderlich ist. Der entsprechende Punkt wird gesetzt, und der Ablauf beginnt von vorn. So tastet man sich möglichst nah am wirklichen Kreis entlang, der dabei auf dem Bildschirm schön gleichmäßig wird. Da ein Kreis symmetrisch ist, braucht nur ein Achtel des Kreisbogens berechnet zu werden. Alle anderen Punkte ergeben sich durch Spiegelung und Vertauschen von x und y-Werten. Bei einer Berechnung werden also gleich acht Punkte gesetzt (siehe Listing 4). Diese Methode eignet sich auch hervorragend zur Programmierung in Maschinensprache, da sie ohne Fließkommaberechnungen auskommt.

Aus diesem Verfahren läßt sich ein Programm herleiten, das unter Angabe des Mittelpunkts und der Halbachsen eine Ellipse auf den Bildschirm bringt (siehe Listing 5). Diese Figur ist nicht so symmetrisch wie der Kreis. Daher muß ein Viertel der Ellipse berechnet werden, die übrigen Bögen ergeben sich wieder durch Drehen und Spiegeln.

Die in Maschinensprache geschriebenen Kreis- und Ellipsenroutinen der meisten

```

5000 X=0 : Y=B
5010 QB = 2*B*B : QA = 2*A*A
5020 DX = 1 : DY = QA*B-1
5030 DA = INT(DY/2)
5040 SET XM+X,YM+Y : SET XM-X,YM+Y
5050 SET XM+X,YM-Y : SET XM-X,YM-Y
5060 IF DA>0 THEN DA=DA-DX-1 : DX=DX+QB : X=X+1
5070 IF DA<0 THEN DA=DA+DY-1 : DY=DY-QA : Y=Y-1
5080 IF Y>=0 THEN 5040
5090 RETURN

```

Listing 5: Bei Ellipsen muß schon etwas mehr gerechnet werden.

BASIC-Erweiterungen arbeiten nach diesem Prinzip. Wie oben schon erwähnt, wird aus einer Ellipse ein Kreis, wenn die Halbachsen a und b gleich lang sind. Mit A=B=R liefert Listing 4 dieselbe Figur wie Listing 5.

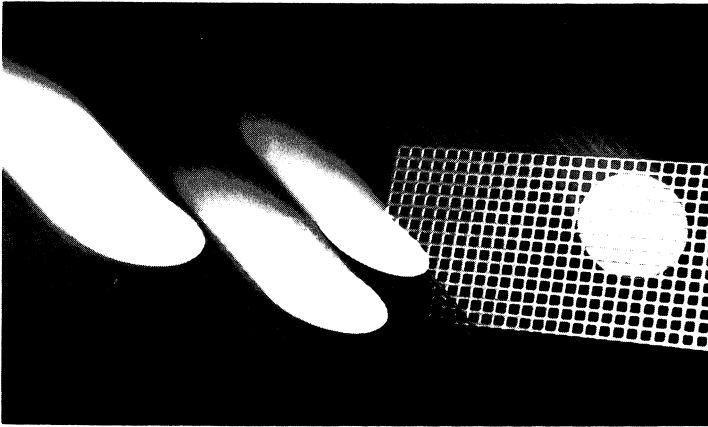
In der Praxis

Nachteil dieser Methoden für Kreise und Ellipsen ist der Zwang, immer die volle Figur zeichnen zu müssen. Einen Kreis von 67 bis 271 Grad zu plotten, bereitet mit der Bresenham'schen Methode große Schwierigkeiten. Eine weitere Einschränkung gibt es für Ellipsen: ihre Halbachsen müssen parallel zu den Achsen des Koordinatensystems liegen. Bei der grafischen Gestaltung ist das aber meist kein Hindernis, mit etwas Geschick kommt man mit diesen Figuren aus.

Für alle Programme gilt, daß sie mit einer Fehlermeldung enden, wenn Punkte außerhalb des Bildschirms angesprochen werden, es sei denn, die PLOT-Routine kann negative oder zu große Werte vertragen. Ansonsten müssen die Figuren vollständig auf dem Bildschirm passen.

Erweitertes Tool

Das relokatable Maschinensprache-Tool, das Sie diesmal aus dem Magazin abspeichern können, entspricht dem aus der letzten Folge, erweitert um einen Aufruf, der Ellipsen ganz oder teilweise zeichnen kann. Dabei wurde Listing 3 in Maschinensprache umgesetzt. Die Schrittweite der Schleife ist beliebig wählbar, und die berechneten Punkte werden durch Linien verbunden. Das hat den Vorteil, bei ausreichender Genauigkeit noch schnell genug zu sein. Durch Experimentieren mit der Schrittweite lassen sich weitere Effekte erzielen wie zum Beispiel das Zeichnen von n-Ecken, was im Modul gezeigt wird. Franz Dreismann/HS



Fix, kurz, rund

Lösung des Rätsels aus INPUT 7/87

Bei diesem Rätsel, das genaugenommen nur aus Traditionsgründen so heißt und in Wirklichkeit ein Programmierwettbewerb ist, ging es um kurze oder schnelle Kreisalgorithmen. Vorgegeben waren ein Rahmenprogramm zur Zeitmessung und ein Grafikpaket, die Bewerber konnten sich also auf das Wesentliche konzentrieren.

SIN- und KOSINüsse

In der Kategorie „Kurz“ war für mathematisch Begabte schnell klar, daß einen nur die trigonometrische Lösung weiterbringt, alle anderen sind allein vom Algorithmus beziehungsweise den nötigen Spiegelungen her zu lang. (Auf den Hintergrund der verschiedenen Methoden, Kreise zu programmieren, wird hier nicht weiter eingegangen, da das ausführlich in den 64er Tips in dieser Ausgabe abgehandelt wird.) So gab es unter den mehreren hundert Einsendungen genau 79, die mit drei Befehlen auskamen.

Unumgänglich, daß die Gewinner hier ausgelöst werden mußten. Aber nicht nur das. Bei näherem Hinsehen stellte sich nämlich heraus, daß mit drei Befehlen sowohl allge-

Bergeweise Einsendungen, stundenlanges Testen, ständig neue Überraschungen: das brachte die Auswertung des „Rätsels“ zum Thema „Schnelle oder kurze Kreisalgorithmen“.

meingültige Lösungen als auch solche, die nur die gestellte Aufgabe lösen, programmierbar sind. Ein Blick auf das Programm des Gewinners in der Kategorie „Kurz“ soll dies verdeutlichen:

```
40030 for f=0 to 2* pi step 1/r
40040 set x+r*cos(f),y+r*sin(f)
40050 next f
```

Wahrscheinlich wird sich Herr Pohl aus Lahnstein jetzt über sein Jahresabo freuen, wir gratulieren! Mit diesen Programmzeilen lassen sich Kreise beliebiger Größe um den Punkt (x/y) mit dem Radius r zeichnen, damit ist auch die erwähnte „Allgemeingültigkeit“ definiert. Denn: Anfangs- und Endwert der Schleife sind algorithmisch festgesetzt, es geht von Null bis zweimal Pi. Die Schrittweite, bestimmend für die Dichte des Kreises, wird abhängig vom Radius verändert –

je größer der Radius, desto kleiner die Schrittweite. Einige Leser haben in diese Step-Rate in irgendeiner Form Pi eingehen lassen; das ist unlogisch. Der Radius muß es sein. Bei einigen Lösungen wurde auf die x- und y-Werte noch jeweils 0,5 addiert, zum Ausgleich von Rundungsfehlern, dies hat sich bei meinen Tests als nicht notwendig erwiesen. Schadet aber auch nichts.

Die Kurzen . . .

Langer Rede kurzer Sinn: ich habe aus der Menge der 3-Befehls-Lösungen die Programme ausgewählt, die auch mit veränderten Werten für den Radius noch dichte und runde Kreise lieferten – immerhin noch 20 Leser und eine Leserin. Das stand zwar so nicht in den Wettbewerbsbedingungen, es ist aber meines Erachtens bei so vielen Einsendungen legitim, neben der reinen Losentscheidung noch nach anderen Kriterien vorzugehen. Der Gewinner des Jahresabos wurde schon genannt, ein Buch nach Wahl aus unserem Verlagsprogramm haben gewonnen:

Florian Fischer, Freilassing
Helmut Müller, Paderborn
Anja Fisch, Frankfurt
Wolfgang Richter, Bonn
Christoph Wehran, Stadallendorf

Wahrscheinlich gucken die mit den 2-Befehls-Lösungen jetzt schon reichlich frustriert. Nicht nervös werden, Jungs! Ihr geht nicht leer aus! Fünf Leute zeichnen den Kreis nämlich mit zwei Befehlen. Wie das geht? So:

```
40030 set sin(ti/6/r)* r+x,cos(ti/6/r)* r+y
40040 if ti<38*r then 40030
```

Obendrein erfüllen diese beiden Befehle auch für beliebige Radien ihre Pflicht. Als Fans algorithmisch orientierter Programmierung fühle ich mich da natürlich voll veralbert – da haben Sie doch stundenlang herumprobiert, Herr Kai Lappalainen aus Buchholz! Was hat denn die Systemzeit mit Kreisen zu tun? Hemmungsloser Empirismus, sowas! Aber was solls, versprochen ist versprochen, er hat sein Jahresabo verdient, schließlich wollen wir nicht als Geizhalse dastehen.

Wir haben im obigen Beispiel aus optischen Gründen die beiden Befehle auf zwei Zeilen aufgeteilt, in der Originalfassung besteht

Mit abgeschaltetem Bildschirm wären es noch 0,09 Sekunden weniger gewesen, aber auch so kam mit 2,25 Sekunden ein klarer zweiter Platz heraus.

```
40000 rem _____
40010 rem kai lappalainen
40020 rem _____
40030 x=x+.5:y=y+.5:s=1.15/r:a=r:b=d:ford=.tor*.71:r=r-d*s:se+r,x,d+y:se+r,x,y-d
40040 se+r,x,d+y:se+r,x,y-d:se+d,x,r+y:se+d,x,y-r:se+d,x,d,r+y:se+d,y-r:next
40050 r=a:d=b:x=x-.5:y=y-.5
63333 return
```

```
40000 rem _____
40010 rem oliver muelhens
40020 rem _____
40030 t=.195*(r-1):m=.5:c=1:u=r-1:v=0
40040 forv=vtov+t:se+r,u,y-v:se+r+u,y-v:se+r-u,y+v:se+r+u,y+v
40050 se+r-v,y-u:se+r+v,y-u:se+r-v,y+u:se+r+v,y+u:next
40060 forv=vtov+t*1.84:se+r,u,y-v:se+r+u,y-v:se+r-u,y+v:se+r+u,y+v
40070 se+r-v,y-u:se+r+v,y-u:se+r-v,y+u:se+r+v,y+u:u=m:next
40080 forv=vtov+t*1.41:se+r,u,y-v:se+r+u,y-v:se+r-u,y+v:se+r+u,y+v:u=o:next
```

Wieviel Ecken hat ein Kreis? Ein Sechzehneck, in 2,28 Sekunden gezeichnet, hat den vierten Platz in der Kategorie „Schnell“ gemacht.

die Lösung aus einer BASIC-Zeile. Aber darauf kam es ja nicht an.

```
40000 rem _____
40010 rem ruediger schink
40020 rem _____
40025 poke53265,43
40030 j=x+r-1:d=x-r:forx=89to110:setj,x:setd,x:next
40035 d=y+r-1:j=y-r:forx=149to170:setx,j:setx,d:next
40040 b=60/r:c=1/r:r=66/r:d=160:x=11*b
40050 forj=12*cto40*cstepc:r=r-j:x=x+b
40060 setd-r,y-x:setd-r,y+x:setd+r,y-x:setd+r,y+x
40070 setd-x,y-r:setd-x,y+r:setd+x,y-r:setd+x,y+r:next
40080 j=r+d:r=d:r=d-y-x:y=y+x
40090 forx=1to14
40100 setj-x,y+x:se+r,x,y+x
40110 setj-x,d-x:se+r,x,d-x:next
40120 x=160:y=1e2:r=64:d=5
40130 poke53265,59
63333 return
```

Lang und undurchschaubar, aber schneller geht's nimmer: 1,98 Sekunden braucht das Programm, bis der Kreis auf dem Bildschirm steht.

... und die Schnellen

In der Kategorie „Schnell“ haben 19 Leser ihre Kreise in weniger als 2,5 Sekunden auf den Bildschirm gebracht. Ehe wir auf diese Einsendungen näher eingehen, noch eine Bemerkung zu einer bestimmten Art von Lösungen. Einige Leser haben einfach sämtliche Koordinaten des gesuchten Kreises in DATA-Statements abgelegt und dann in einer READ-SET-Schleife den Kreis geschrieben. Zugegeben: in der Ausschreibung war eine solche Lösung nicht ausdrücklich ausgeschlossen. Wenn das Thema aber „Entwicklung von schnellen Kreis-Algorithmen in BASIC“ heißt, kann es meiner Meinung nach nicht darum gehen, die zu berechnenden Werte in Handarbeit einzutippen. Da keines dieser Programme schnell genug war, um in die engere Auswahl zu kommen, soll es mit diesen Worten gut sein.

Gewonnen hat, um das vorwegzuschieben, Rüdiger Schink aus Wesel. Sein Programm (siehe Listing) kommentiert er selbst mit den Worten „aus der Trickkiste – lang, undurchschaubar, aber noch schneller“. Ich habe auch wirklich eine Weile gebraucht, bis mir plötzlich der „Trick“ klar wurde. Herr Schink macht sich die Tatsache zunutze, daß die oberen und unteren beziehungsweise linken und rechten Teile des Kreises kaum von einer Geraden abweichen. Konsequentermaßen werden deswegen oben und unten waagerechte, links und rechts senkrechte Linien gezeichnet. Für den

Kreisausschnitt in der Nähe des 45-Grad-Winkels und dessen drei Spiegelungen (unten rechts, unten links, oben rechts) gilt ähnliches, allerdings müssen hier Diagonalen den „echten“ Kreisbogen ersetzen. So wird der Anteil zeitraubender Berechnungen auf das Notwendigste reduziert. Die üblichen Programmiertricks sind nicht einmal voll ausgeschöpft – zwar wird der Bildschirm ausgeschaltet, aber man hätte die letzte FOR-NEXT-Schleife noch in einer Zeile zusammenfassen können.

Speziell, aber schnell

Bei den Lösungen zur Kategorie „Kurz“ war viel von Allgemeingültigkeit die Rede, das hier nun vorgestellte Siegerprogramm funktioniert wirklich nur für den gegebenen Kreis. Da „Allgemeingültigkeit“ aber nicht ausdrücklich gefordert wurde und es keine gleich schnellen allgemeingültigen Lösungen gibt, gehört der erste Preis eindeutig Herrn Schink.

Dafür wird mit der Vorstellung des zweit-schnellsten Programms der oben so gebeitelte Herr Lappalainen wieder in die Gilde der puristischen Algorithmiker aufgenommen. Im Listing kann man einige gängige Beschleunigungstricks studieren: mit möglichst wenig BASIC-Zeilen auskommen, die '0' im Ausdruck '0.5' ist überflüssiges Gewese, '5' tut's auch, den Bildschirm hätte man auch noch ausmachen können, das bringt immerhin 0,09 Sekunden. Zum mathematischen Hintergrund schreibt der Autor:

„Will man einen Punkt (a,b) mit dem Winkel W um den Koordinatenursprung drehen, berechnet sich der neue Punkt (a',b') folgendermaßen:

$$a' = a \cdot \cos(W) + b \cdot \sin(W)$$

$$b' = b \cdot \cos(W) - a \cdot \sin(W)$$

Dreht man nun jeweils den neuen Punkt wieder mit demselben Winkel W um den Ursprung und wählt man W genügend klein, so erhält man einen Kreis (siehe 64er Tips, JS). Wenn aber W konstant ist, bleiben auch $\sin(W)$ und $\cos(W)$ für die gesamte Berechnungsdauer konstant. Da wir

W sehr klein wählen wollen, können wir ohne Einbußen der Zeichengenauigkeit $\cos(W)=1$ setzen und somit für die obige Gleichung sagen:

$s=\sin(W)$
 $a'=a+b*s$
 $b'=b-a*s$

In BASIC sieht das so aus:

$s=\sin(w): h=a+b*s: b=b-a*s: a=h$

Bei kleinem W ist auch s sehr klein (Größenordnung 0,02). Außerdem unterscheidet sich die alte Koordinate a von der neuen nur geringfügig, so daß man sich hier die Hilfsvariable h auch sparen und schreiben kann:

$s=\sin(w): a=a+b*s: b=b-a*s$

Es bleibt das Problem der geeigneten Schrittweite“, erläutert Herr Lappalainen weiter und entwickelt als „ideale Schrittweite“ $1/r$, aber das wurde bei den kurzen Programmen schon gesagt. Hinzu kommt die Addition der Mittelpunkt-Koordinaten (x/y), wir erhalten, so der Autor, „die BASIC-Befehlsfolge

$s=1/r:a=0:b=r$
for i=0 to 2* π step s
 $a=a+b*s: b=b-a*s$
set x+a,y+b
next

Weitere Beschleunigungen bestehen darin, nur einen Achtelkreis zu berechnen und den Rest durch Spiegelung zu erhalten. Wenn (x+a,y+b) ein Kreispunkt ist, ist auch (x-a,y+b) oder auch (x+b,y+a) und so weiter ein Punkt des Kreises.

Berechnet man nur einen Achtelkreis, so entwickelt sich die a-Koordinate annähernd linear, so daß wir a in die Schleife einbauen können. Die Schleife muß dann von Null bis $\sin(45 \text{ Grad}) * r$, gleich $\sin(\pi/4) * r$, gleich $0.7 * r$ laufen. Da $\sin(\pi/4)$ ungefähr $\pi/4$ entspricht, erhalten wir für die Schrittweite $s * r$, also Eins.“

Einige „in der Praxis entstehenden Ungenauigkeiten“ wurden, schreibt Herr Lappalainen, nach der Methode „trial and error“ durch Korrektur der theoretischen Werte ausgebügelt; Rundungsfehler durch die bekannte Addition von 0,5 korrigiert. So ent-

stehen „Kreise, deren Genauigkeit trotz der vielen Vereinfachungen sehr groß und für die Praxis voll ausreichend ist ...“ und obendrein beachtlich schnell auf dem Bildschirm erscheinen, wie man im Modul auf Diskette bewundern kann.

Die weiteren Gewinner sind:

Jochen Rätsch aus Geseke mit 2,26 Sekunden
Oliver Muelhens aus Grevenbroich, 2,28 Sekunden
Rüdiger Kremer aus Selters, 2,28 Sekunden
Frank Fetthauer aus Essen mit 2,3 Sekunden

Die Listings sind zu Test- und Vergleichszwecken „normalisiert“ worden, das heißt, alle fangen mit der gleichen Zeilennummer an, der Rücksprung zum Rahmenprogramm mit RETURN blieb in der vorgegebenen Zeile.

Bleibt noch, allen, die nichts gewonnen haben, herzlichen Dank fürs Mitmachen zu sagen. Letztendlich geht's ja ohnehin mehr um den Spaß als ums Gewinnen, oder?! JS

Calc korrekt

Fehlerbeseitigung in INPUT-Calc 64/128

INPUT-Calc mußte im allerletzten Moment aufgrund einer Unverträglichkeit mit dem INPUT 64-Betriebssystem einem Update unterzogen werden. Hierzu war es notwendig, die Zeilennummern neu durchzunummerieren, um die Bildschirmmeldung für CTRL-S einbauen zu können.

Die veröffentlichte Version zeigte folgende Fehler:

- Laden und Speichern in C128-Modus geht nicht
- SHIFT-RETURN nach Aufruf des Hilfs-Editors erzeugt Fehlermeldung
- Fehler in Formeln führen zum „Hänger“ (Fehlervektor verbogen)
- „Insert“ im Formeleditor verheimlicht jeweils die letzte gültige Formel.

Anwenderprogramme benötigen meist eine gewisse Reifezeit in der praktischen Erprobung, bis die Versionen 3 oder 4 endlich Murphy's Gesetzen gegenüber bestehen können und nicht mehr das „schiefe geht“, was auf keinen Fall „schiefe gehen“ darf. INPUT-Calc 64/128 war gerade im 128er-Modus hiervon betroffen.

- Im C64-Modus speichert INPUT-Calc gnadenlos immer 41 Blöcke, auch wenn die Tabelle nur aus einer Zeile besteht.
- Wiederholtes Kalkulieren mit großem Formelsatz kann im C128-Modus Variablen zerstören.

Vorbereitung

Im Beiheft zur Ausgabe 8/87 befand sich bereits ein Korrekturzettel, auf dem die vorbereitenden Arbeitsschritte beschrieben sind, die Sie vor jeder Änderung an der BASIC-Oberfläche von INPUT-Calc 64/128 durchführen müssen.

- INPUT-Calc 64/128 aus INPUT 64 mit CTRL-S auf eigenen Datenträger abspeichern
- Rechner aus- und nach 120 Sekunden wieder einschalten
- INPUT-Calc 64/128 vom eigenen Datenträger im C64-Modus laden
- mit 'RUN' starten und RUN/STOP abbrechen

Jetzt können Sie mit LIST den BASIC-Teil erreichen.

Nach jeder Änderung, die Sie selbst am BASIC-Teil vorgenommen haben, geben Sie im Direktmodus (also ohne Zeilennummer)

POKE 44,8:POKE 43,1

ein, damit der Programmanfang wieder auf den ursprünglichen Beginn des Programms

zeigt, und speichern das geänderte Programm auf Ihren Datenträger ab.

Um Ihnen fehlerträchtige und zeitraubende Tipparbeiten zu ersparen, haben wir in der ID-Werkstatt ein Patch-Programm bereitgestellt, das sämtliche Änderungen an INPUT-Calc 64/128 automatisch durchführt. Vorsichtshalber sollten Sie die Version verwenden, die Sie mit CTRL-S aus Ausgabe 8/87 abgespeichert hatten.

Die Bedienung finden Sie in der Rubrik „ID-Werkstatt“ in diesem Heft.

Operation

Die Änderungen, die durch das Patch-Programm an INPUT-Calc vorgenommen werden, haben folgenden Sinn:

1. In der fehlenden Zeile 520 stellt INPUT-Calc fest, ob die Disketten-Station angesprochen werden kann, und öffnet anschließend den Fehlerkanal.

Die neuen Zahlen hinter den POKE-Befehlen in den Zeilen 5370 und 6090 bewirken, daß nach einer Berechnung im nachfolgende Programm weitergearbeitet wird (siehe 1.1 und 1.2).

2. In den Zeilen 2510 bis 2530 wird die Variable 'f1\$' durch 'fq\$' ersetzt, wie unter 2.0 aufgeführt. Dabei werden gleichzeitig programmtechnische Unsauberkeiten entfernt:

Um bei einer fehlerhaften Formeleingabe über die Meldung „Syntaxfehler“ ordnungsgemäß in den Tabelleneditor zurückkehren zu können, werden die Zei-

len 3455 und 3460 (siehe 2.1) eingefügt.

3. Die Lade-/Speicherprobleme beim C128 werden durch die POKE-Sequenzen unter Punkt 3. behoben.

4. Im Formeleditor unterschlägt das Programm Formeln, wenn Sie mit 'l' Formeln einfügen wollen, und hält zu viel des Guten im Speicher, wenn Sie Formeln löschen. Die Änderungen unter diesem Punkt beheben die Fehler.

5. In der Zeile 270 haben wir für den C64-Modus die Tabellengrenze um ein Byte erhöht. Das Programm erkennt jetzt auch wieder kurze Tabellen, so daß der überflüssige Rest nicht mit abgespeichert wird.

6. Bei einer kaufmännischen Tabellenskalkulation ist das Rechnen mit Exponentialzahlen wie '1E+10' nicht vorgesehen. Verwenden Sie diese in Formeln, kommt Unsinn heraus. Die eingefügte Zeile 5240 führt in diesem Fall jetzt zu der Fehlermeldung „unerlaubtes Feld“.

Nachbetreuung

Tabellen, die Sie im C128-Modus bereits erstellt haben, können Sie im C64-Modus einladen, wieder abspeichern und anschließend in den C128-Modus übernehmen. Vermeiden Sie es unbedingt, im Dateinamen das Zeichen '*' zu verwenden, wenn Formel- und Tabellen-Files den gleichen Namen haben! Findet das DOS die Formel zuerst, beim Versuch, eine Tabelle zu laden, lädt Calc die Formel-Datei und läuft auf einen Fehler.

Änderungen, die Sie im C128-Modus vornehmen, können nur nach POKE45,1:POKE46,28 abgespeichert werden. Diese Version läuft jedoch nur noch im C128-Modus, also nicht mehr im C64-Modus. Alle weiteren Patches und Tips beziehen sich immer auf die Version im C64-Modus!

Stört Sie bei Zahlen der Dezimalpunkt und der Nachkommaanteil, können Sie diesen mit SHIFT-SPACE überschreiben. Diese „Zahlen“ werden vom Editor als Text verstanden und nicht mehr formatiert, können aber trotzdem bei Kalkulationen verwendet werden. So lassen sich auch Postleitzahlen eingeben.

Verwenden Sie im Diskettenmenü Punkt 5 „andere Kommandos senden“, bleibt es bei der deutschen Tastaturbelegung. Der Doppelpunkt ist das Zeichen über dem „“, wie in der Tabelle im Beiheft aufgeführt.

Um Ballast abzuwerfen, löscht das Patch-Programm die Zeilen 10–70 und die Zeilen 530–670 ersatzlos. Sie bezogen sich auf das INPUT 64-Betriebssystem.

Sollten Sie per RUN/STOP oder durch andere Maßnahmen in den Direktmodus geraten, können Sie INPUT-Calc wieder ohne Datenverlust starten, indem Sie im Direktmodus eingeben:

Im C128-Modus:

```
CLOSE 15:CLOSE 1:CLOSE 2:CLOSE 4:
```

```
PRINT CHR$(14)
```

```
GOTO 520
```

Im C64-Modus:

```
CLOSE 15:CLOSE 1:CLOSE 2:CLOSE 4:
```

```
POKE 650,128:POKE 53272,24:GOTO 520
```

Hierdurch wird der deutsche Zeichensatz wiederhergestellt.

Manche Drucker erwarten Sekundäradressen, die größer als 99 sind. Ändern Sie in Zeile 10630 die Zahl '2' hinter dem Aufruf „SYSIN“ in eine '3', können Sie auch größere Adressen eingeben.

Wir werden in den nächsten Ausgaben weitere Tips, Tricks zu INPUT-Calc 64/128 veröffentlichen. Für Hinweise sind wir dankbar.

hk/RH

Auf einen Blick

```
1.0 520 la%:8:sa%:0:open 8,8,8:close 8:if st = 0 then open 15,8,15
1.1 5370 poke q0+4,53:poke q0+5,51:poke q0+6,57:poke q0+7,48:poke q0+8,13:end
1.2 6090 poke q0+4,54:poke q0+5,49:poke q0+6,49:poke q0+7,48:poke q0+8,13:end

2.0 2510 sys in,39+2*bb,0,,fq$:fa-ts
2530 if fq$="" then 2650
2550 sys u5,,,,fa,fq$:if peek(251) then 2630
2.1 3455 goto 3470
3460 goto 4770

3 poke 2357,254:poke 4249,212
poke 4197, 32:poke 4198, 32:poke 4199, 35:poke 4200,234
poke 8992, 32:poke 8993,193:poke 8994, 19:poke 8995,160:poke 8996, 0
poke 8997,145:poke 8998,251:poke 8999, 76:poke 9000,202:poke 9001, 19

4. 3690 qq=qq-1:if qq<0 then qq=0
3780 if qq=af then 3070
3830 qq=qq+1:if qq>af then qq=af

5. 270 ts=38912:te=49152:t1=te-ts:q0=631:q1=198:l1$="" :goto 390

6. 5240 fq$=mid$(f$,pa+1,1)
5245 if mid$(f$,pa,1)="" and (fq$="" or fq$="-") then f=7:return
```



Musterverkennung

Spiel: Chameleon

So dreht sich das Spielgeschehen des Chameleon um gleich zwei Spielbretter, um ein größeres (26×23 Felder) und um ein kleineres (5×5 Felder). Auf dem größeren Brett leuchten einige der Felder per Zufallsgenerator hell, die anderen bleiben dunkel – so entsteht ein Muster. Auch auf dem kleineren Brett ist ein Muster aus zufälligen Hell-/Dunkelfeldern abgebildet: Beide Bretter mit ihren Mustern verhalten sich wie Bild (das größere Brett) zu Bildausschnitt. Der Witz des Spiels ist nun, diesen Ausschnitt rasch und sicher auf dem Bild des großen Brettes wiederzuerkennen, aus der Menge sonstiger Zeichenanordnungen herauszufiltern – nach dem Motto schau genau!

Bewertung

Das gemein Reizvolle dabei: manche Bildmuster sind dem gesuchten Ausschnitt sehr ähnlich, mehr aber nicht. Haben Sie das Muster gefunden, bezeichnen Sie es

Hat Sie ein Computerspiel schon mal so richtig spannend gefesselt? Sicherlich. Es kann ein Actiongame wie Pacman oder eine Simulation wie Silent Service sein – oder möchten Sie lieber ein Adventure, um richtig abzuschalten? Hier jedenfalls lernen Sie Chameleon kennen, das in keine der genannten, sondern eher zur Kategorie der Brettspiele gehört, also eine gewisse äußere Ähnlichkeit besitzt zu Reversi/Otello oder zu Go.

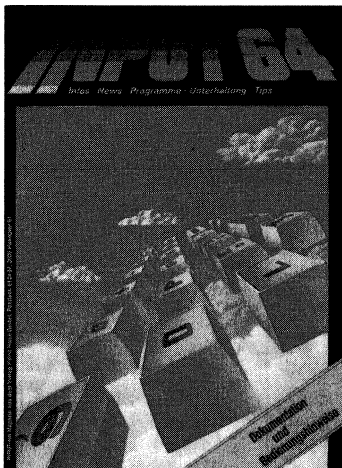
auf dem Bild mit einem Vierecken-Fadenkreuz mit Ihrer 64er Tastatur und drücken zum Zeichen Ihrer Siegesgewißheit eine Taste. Der Computer überprüft, ob Sie richtig geschaut und getippt haben; wenn ja, dann gibt es – in Abhängigkeit des noch verbliebenen Bonus-Standes – Pluspunkte, und die nächste Runde beginnt.

Haben Sie sich aber vertan, dann setzt es einen Fehlerpunkt, haben Sie deren drei, ist das Spiel beendet. Aber nicht genug damit; sollten Sie zu dem Zeitpunkt des Fehlers bereits Pluspunkte gesammelt haben, müssen Sie sich außerdem mit einem deftigen Punkteabzug anfreunden. Die Höhe des Abzuges richtet sich auch hier wieder nach dem Bonusstand. Außerdem wird die Anzahl der Fehlerpunkte mit eingerechnet.

Variationen

Damit Chameleon nicht in langem Nachdenken erstarrt, sondern zu kühnem beherztem Entscheiden animiert, sind drei Faktoren eingebaut, die ihm Zusatzreiz verschaffen: Ein unerbittlich schnell laufender Countdown, jeweils ab Beginn der Runde. Ist er abgelaufen, werden Sie ebenfalls mit einen Fehlerpunkt belastet, und das laufende Spiel ist wegen Zeitüberschreitung verloren. Ein weiterer Drängelfaktor ist die Option, nach der zwei Spieler gleichzeitig, und das im Wettkampf gegeneinander, spielen können. Schließlich haben Sie noch die Auswahl unter 4 Levels, wovon der einfachste, Level 'A', auch schon herausfordernd genug ist. Mit der Level-Wahl bestimmen Sie Einstiegsrunde. Nur ganz Mutige sollten mit 'D' beginnen. Also, Trainingshürden und Vergnügen reichlich! ws-p/WM

Am 2. November an Ihrem Kiosk: INPUT 64, Ausgabe 11/87



Wir bringen unter anderem:

Maschinensprache-Monitor

Gehen Sie auf Entdeckungsreise, und erforschen Sie die letzten Geheimnisse Ihres Rechners. Mit diesem Werkzeug editieren Sie den Speicherinhalt direkt; auch unter dem ROM! Sie können ganzen Bereiche vergleichen, mit einem bestimmten Wert füllen oder nach einer Byte-Folge durchsuchen lassen. Mit dem integrierten Lineassembler lassen sich kleine Maschinenprogramme schreiben und mit dem Disassembler zum Beispiel einige BASIC-Routinen analysieren. Selbstverständlich können Sie auch Programme an beliebige Adressen laden oder festgelegte Speicherbereiche sichern und vieles mehr. Ein Profiwerkzeug, nicht nur für Könner.

Das Druckprogramm für den MPS 801/803

Vielleicht haben Sie schon ein Druckprogramm für Textbildschirme oder ein Programm, das HiRes-Bilder zu Papier bringt. Unser Programm erkennt Schrift, Grafik und Sprites selbständig und druckt einfach. Einfach so! Druckt alles, was auf dem Monitor sichtbar ist. Nicht mehr, aber auch nicht weniger.

Der Brief aus der Steckdose

Wenn Sie wissen wollen, was das Programm 'Elektrobrief' aus der nächsten Ausgabe leistet, schauen Sie sich das Vorschau-Programm auf der Diskette an. Wie es das tut, erfahren Sie in der nächsten Ausgabe.

Syntax-Check

Wer programmiert schon fehlerfrei? Haben Sie bei umfangreichen Programmen auch immer so ein un gutes Gefühl, daß sich hier oder da noch ein Fehler versteckt hält? Das Aufspüren von einfachen Syntax- oder komplexeren Strukturfehlern (z.B. nicht ordnungsgemäß geschlossene FOR-NEXT-Schleifen) können Sie getrost unserem Syntax-Checker überlassen, wobei auch selten angesprungene Routinen unmittelbar vollständig überprüft werden. Natürlich würde der 64er diese Fehler auch irgendwann einmal feststellen – wahrscheinlich aber im unpassendsten Augenblick, mit kompletten Datenverlust als Begleiterscheinung. Das Programm ist menügesteuert und damit einfach zu bedienen. So kann zum Beispiel das Fehlerprotokoll auch auf einen Drucker gelenkt werden, um fehlerhafte Zeilen flimmerfrei zu analysieren.

c't — Magazin für Computertechnik

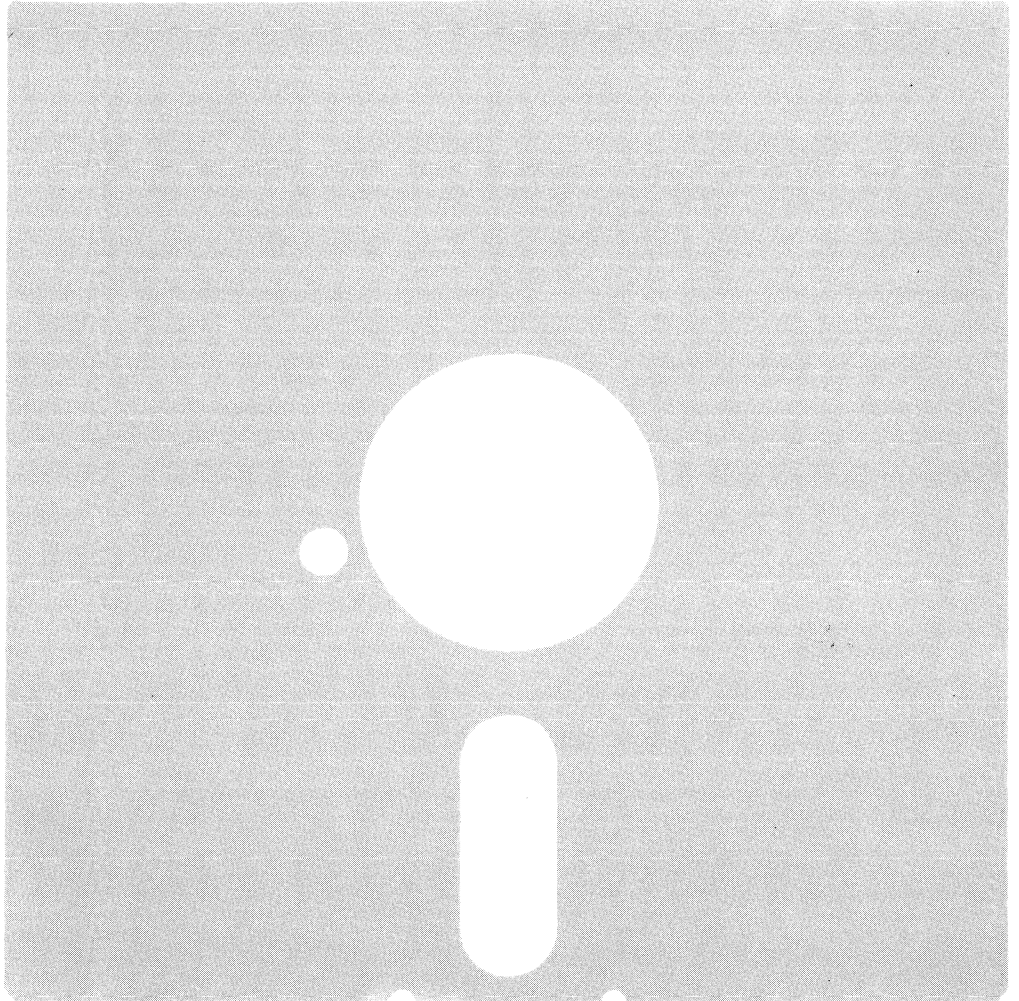
Ausgabe 11/87 — ab 16. Oktober am Kiosk

Projekte: 5,25"-Drives und High-Density-Floppies am IBM Modell 30 * IBM liest Atari-ST-Disketten * Software-Know-how: Ray-Tracing-Computergrafik mit unbekanntenen Möglichkeiten * Neue KI-Sprache „Scheme“ * Prüfstand: PCs unter 1500 DM * Applikationen: Grafik-Prozessoren im Vergleich * u.v.a.m.

elrad — Magazin für Elektronik

Ausgabe 11/87 — ab 26. Oktober am Kiosk

Projekt: Leistungsfähige Schrittmotorkarte * Report: Meßtechnik — preiswerte Neuheiten * Projekt Computer-Technik: Sinus-DC/AC-Wandler * Schaltungstechnik Aktuell: PTCs als wirksamer Schutz für Stromversorgung und Verbraucher * u.v.a.m.



Fast wieder Tango!

Ab 25. September gibt's das neue

Hifi Boxen selbstgemacht. Mit fünfzehn

gelungenen Selbstbau-Konzepten namhafter

Entwickler. Mit wichtigen Grundlagen; mit

News, News, News... Und natürlich in Farbe.

Für 16 Mark 80 überall, wo es Zeitschriften

gibt. OT&I

HEISE



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

