

INPUT 64

Infos · News · Programme · Unterhaltung · Tips

Sprite- und Zeichensatz-Editor

INPUT-SCE

Sampler ohne Hardware

Samplersette

Compact-Disc-Archive

CD-Manager



Serien:

INPUT-Assembler-Schule Teil 4

64er Tips

Einer gegen Alle

Englische GRAMmatik Teil 6

Dokumentation
und
Bedienungshinweise

Information+Wissen

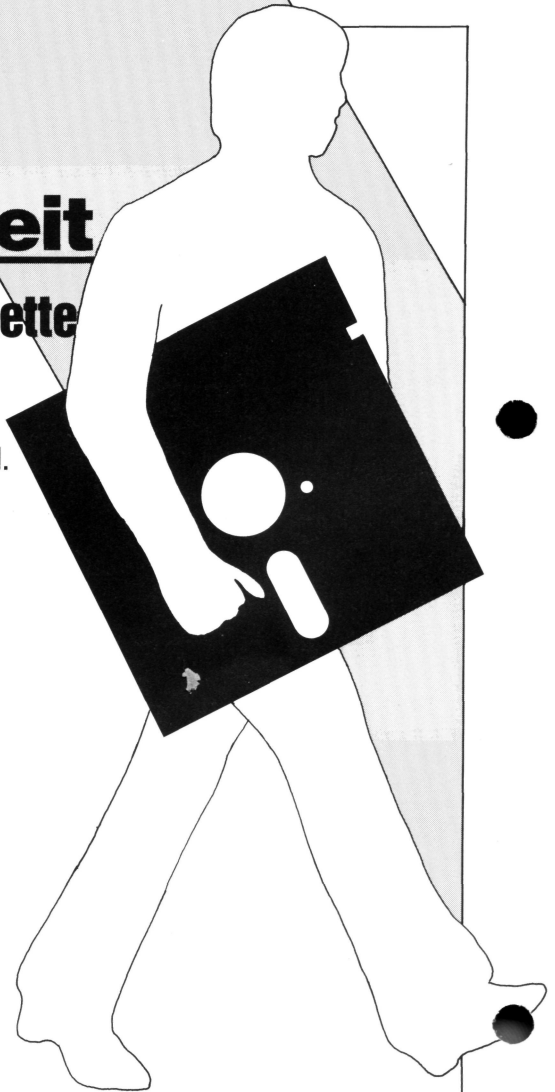
Bundesweit

INPUT 64 auf Diskette

Am Kiosk.
Im Computerfachhandel.
Beim Bahnhofsbuchhandel.

INPUT 64,
das
elektronische
Computer-
magazin.

Auf Kassette
oder Diskette.



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

immer wieder werden wir gefragt, wie lange es wohl noch dauern wird, bis der C64 endgültig eingemottet werden wird. Ein Rechner, der schon viele Jahre auf dem Buckel hat und sich heute neben den Speicherriesen mit ihren Superchips schon fast als Museumsstück ausnimmt. Doch allen Unkenrufen zum Trotz tummelt sich dieser Rechner immer noch in den oberen Rängen der Verkaufs-Charts und erfährt nach kosmetischen Veränderungen nun eine Neuauflage seines Innenlebens, wobei er entgegen sonstigen Gepflogenheiten zu seiner eigenen Serie kompatibel bleibt. Im Artikel „Vierzig drauf“ erfahren Sie Näheres.

Trotz bescheidener 64 KByte nehmen sich immer wieder Soft- und Hardware-Entwickler dieses altvertrauten und weitverbreiteten Geräts an. Auch Neuentwicklungen lohnen sich noch. Aufgrund der großen Marktbreite besteht beim C64 — wie bei kaum einem anderen Rechner — die Chance, für eigene Projekte Interessenten zu gewinnen. Irgendeiner im Bekanntenkreis besitzt mit an Sicher-

heit grenzender Wahrscheinlichkeit den gleichen Rechner. Die Ergebnisse dieser Entwicklungen versetzen uns immer wieder in Erstaunen. Sie zeigen uns: Der C64 ist bei weitem noch nicht ausgereizt!

So braucht sich JAM unter IOS (1/87) nicht hinter anderen grafischen Benutzer-Oberflächen zu verstecken — vor allem, wenn man den äußerst geringen Speicherplatzbedarf dieses Systems berücksichtigt. Das Ball-Programm (1/87) zeigte, daß Animationen auf dem C64 möglich sind, die man bis dahin nur „größeren“ Rechnern zutraute.

Unser Schwestermagazin c't bringt nun ein Projekt in der Ausgabe 6/87, das aus dem „Volkscomputer“ schon eher so etwas wie einen Rennwagen macht: C64 mit 4-MHz-Takt und 16-Bit-CPU. So ganz nebenbei läßt sich der C64 auch noch auf mindestens 256 KByte hochrüsten. Sie finden eine Vorabinformation im Artikel „16-Takter“. Wir haben „Julia“ (2/87) auf dem aufgerüsteten Rechner laufen lassen. Diese Fractal-Berechnung mit vierfacher Geschwindigkeit zu erle-

ben, ist eine echte Freude.

Bei uns in der Redaktion schlummern schon die nächsten Software-Pakete. Sie werden zeigen, was noch alles im C64 steckt. So dürfen Sie sich schon auf eine Fenstertechnik (7/87) freuen, die vor allem durch Geschwindigkeit und Anwendungsvielfalt besticht. Außerdem erscheint in der gleichen Ausgabe der schon länger ersehnte ICI — INPUT-Operationsystem-Command-Interpreter. Hiermit können Sie unter anderem ganze Befehlssequenzen durch Batchdateien automatisch ablaufen lassen oder direkt einen Blick in sequentielle Dateien werfen.

Wir werden jedenfalls auf längere Sicht unsere 64er in der Redaktion im Einsatz behalten. Sie können somit weiterhin auf interessante Veröffentlichungen in unserem Magazin zurückgreifen. Es sei denn, Sie wollen Ihren C64 einmotten.



INHALT

Leser fragen . . .	2	16-Takter Hinweis auf ein c't-Projekt	19
In einem Satz Werkzeug: INPUT-SCE	3	CDs ohne Streif verwalten Anwendung: CD-Manager	20
Vierzig drauf Test: VC 1581	6	64er festgenagelt Tips für C128-Benutzer	23
Schnell wie der Fisch im Wasser Test: Dolphindos	8	Einer gegen Alle Chiffrierungsrätsel	24
Kugeln und Würfel Spiel: Metal Ball	10	Objekte in Bewegung 64er Tips über Sprites	26
Lernen im Dialog Serie: EGRAM Teil 6	10	Hinweise zur Bedienung	29
Viele Wege führen ins RAM Serie: Assembler-Schule Teil 4	11	Bei Ladeproblemen	30
Samplersette Experimente mit dem Kassettenport	18	Vorschau Impressum	31 32

Auf einen Blick:

INPUT 64 -

Betriebssystem-Befehle

Titel abkürzen	CTRL und Q
Hilfsseite aufrufen	CTRL und H
zum Inhaltsverzeichnis	CTRL und I
Bildschirmfarbe ändern	CTRL und F
Rahmenfarbe ändern	CTRL und R
Bildschirmausdruck	CTRL und B
Programm sichern	CTRL und S

Laden von Diskette:
LOAD "INPUT*", 8,1
Laden von Kassette:
LOAD oder SHIFT und RUN/STOP

Ausführliche Bedienungshinweise finden Sie auf Seite 29

Leser fragen . . .

Unmögliches kann auch der DiscDoctor nicht

(. . .) Außerdem würde ich gerne wissen, ob es prinzipiell möglich ist, eine versehentlich formatierte Diskette (noch nicht wieder beschrieben) wieder herzustellen. Sollte das sogar mit dem DiscDoctor aus der Ausgabe 11/85 möglich sein, so möchte ich auch diese umgehend mitbestellen. (. . .)

(W. Schröttle, München)

Hier ist nicht nur der DiscDoctor mit seinem Latein am Ende. Kein Programm könnte eine formatierte Diskette rekonstruieren, da die alten Informationen physikalisch einfach nicht mehr vorhanden sind. Es sei denn, es wurde ohne eine ID-Kennung formatiert (z.B.: "N:TEST"). In diesem Fall werden nur die Einträge des Directory gelöscht. Da die eigentlichen Daten physikalisch nicht gelöscht werden, hilft hier der DiscDoctor weiter.

(d. Red.)

Weitere Druckertreiber für CAD

Der Citizen 120 D mit seriellem C64-Interface harmoniert mit der Epson-Voreinstellung, wenn der DIP-Schalter Nr. 2 auf "off" eingestellt wird. (telefonischer Hinweis)

Der MPS 802 mit dem "Super-Rom" (von Markt u. Technik) kann folgendermaßen eingestellt werden: Geräte-nummer: 04; Sekundäradresse: 07; Codefolge für Grafik an - vor: 1b, 33, 14, 1b, 4b; Codefolge für Zeilenvorschub: 0d; Nadelzahl: 08; oberstes Bit: 00; Drehen der Bitfolge: 01; Maximale Druckbreite: 0280. (E. Weiss, Adliswil, Schweiz)

Wir bedanken uns im Namen der Leser für diese Hinweise und bitten gleichzeitig darum, uns weitere Drukkeranpassungen einzusenden.

(d. Red.)

Drei mal drei gleich sieben

Nein! So falsch reagiert das Quiz aus der Ausgabe 2/87 nun doch nicht. Leider haben sich aber bei dem kleinen BASIC-Programm zur Erstellung eigener Dateien zwei Fehler ein-

```
5 dim f$(241),a$(241)
160 print#1,f$(x):print#1,a$(x)
```

geschlichen. Die Zeile 5 muß neu in das Programm aufgenommen und die Zeile 160 muß entsprechend abgeändert werden.

Beachten Sie bitte auch, daß die Strings für Begriffe, Fragen und Antworten jeweils eine maximale Länge nicht überschreiten dürfen. Diese beträgt bei den Begriffen 13 Buchstaben, bei den Fragen 37 Buchstaben und bei den Antworten 25 Buchstaben.

(d. Red.)

Dienstag ist Lesertag !!!

Technische Anfragen

nur Dienstag von 9 - 16.30 Uhr

Telefon (05 11) 53 52 - 0

Programmierer, mal herhör'n!

Betrifft: Programm-Angebote

Bestimmt haben Sie noch ein Programm in der Schublade liegen, das noch nicht veröffentlicht ist. Oder Sie haben eine Programm-Idee, deren Realisierung nur mit der Aussicht auf spätere Veröffentlichung sinnvoll ist. Oder Sie sind genügend fit in Sachen Assembler-Program-

mierung, um sich durch Programmier-Aufträge ein bißchen dazu zu verdienen. Oder Sie haben Software für den C128 angepaßt. Oder . . . Lassen Sie sich doch einmal unsere ausführlichen Autoren-Hinweise schicken, oder rufen Sie uns einfach an!

(d. Red.)

In einem Satz

INPUT-SCE Sprite-Character-Editor

Sicherlich gibt es schon eine Reihe von Sprite-Editoren. INPUT 64 hat Ihnen in den Ausgaben 3/85 und 8/86 geeignete Werkzeuge angeboten. Auch Zeichensatzänderungen sind mit dem Zeichensatz-Editor aus 2/85 möglich. Der INPUT-SCE bietet aber beides in einem mit einer ganzen Reihe zusätzlicher Entwurfsmöglichkeiten. So lassen sich beispielsweise einzelne Zeichen direkt in Sprites kopieren, ohne das Programm verlassen zu müssen.

Der INPUT-SCE ist, wie man es von einem modernen Software-Werkzeug erwarten darf, menügesteuert. Die einzelnen Menüpunkte werden mit Cursor-rauf/-runter ausgewählt und mit RETURN bestätigt. Nehmen Sie Platz, und lassen Sie sich die Menüauswahl zeigen. À la carte werden Ihnen drei Menüs serviert:

Zeichensatz-Editor
Sprite-Editor
Parameter-Editor

BASIC-EXIT

BASIC-EXIT ist sozusagen das letzte Angebot. Sie verlassen die gastliche Stätte und gelangen direkt in den normalen BASIC-Interpreter, von dem aus Sie jedoch jederzeit mit SYS 4*4096 (\$4000) wieder in den SCE zurückfinden. Dieses Spezialangebot können Sie innerhalb des Magazins INPUT 64 natürlich nicht wahrnehmen. Ebenso gilt dies für alle Zugriffe auf Datenträger wie Diskette und/oder Kassette vom SCE aus.

Damit Sie den Rechner für sich haben, speichern Sie sich INPUT-SCE am besten erst einmal mit CTRL+S auf eigenem Datenträger ab. Laden Sie das

Ob Riesenschrift über den Bildschirm tanzen, feinste Hintergrundbilder Spielen den richtigen Reiz verleihen oder der futurologische Zeichensatz den Monitor ziert, mit dem INPUT-SCE haben Sie alle Entwurfsmöglichkeiten.

Programm von dort in den durch Ein-/Ausschalten bereinigten C64, und Sie können das gesamte Angebot genießen.

Einmal ganz direkt und allgemein

Um sich mit der Arbeitsweise des SCE vertraut zu machen, bestellen Sie sich zunächst den Zeichensatz-Editor. Dort nehmen Sie Punkt 5 der Tageskarte: ZS Laden. Ein weiterer Druck auf die RETURN-Taste kopiert den ROM-Zeichensatz in den Puffer des SCE. Dazu wählen Sie anschließend am besten Punkt 2: ZS Testen. Zuerst erscheint ein Bildschirm voller wirrer Zeichen. Über HOME und anschließendes CLR/HOME wird der Bildschirmtisch abgeräumt. Drücken Sie jetzt einmal beliebig lange auf die SPACE-Taste, bis der Cursor fast das untere Ende des Bildschirms erreicht hat. Ein anschließender Druck auf SHIFT/RETURN, und der gewählte Zeichensatz wird serviert. Mit der RETURN-Taste gelangen Sie zurück ins Menü und von dort wieder in den Zeichensatz-Editor.

Im Zeichensatz-Editor bewegen Sie den Cursor mit den Tasten f5/f6 und f7/

f8 auf das Space-Zeichen mit der Nummer 32 (unten rechts angezeigt). Sie können mit der Taste „d“ den Direkt-Modus einschalten, damit Sie sehen können, was Sie tun. Jetzt können Sie mit den einzelnen Befehlen im Zeichensatz-Editor herumspielen. Drücken Sie beispielsweise die Taste „k“, gefolgt von der Pfeil-Oben-Taste. Auf dem Bildschirm erscheint das entsprechende Muster. Mit der „h“-Taste läßt sich der Pfeil in seiner Richtung umkehren. Verwenden Sie die Tasten f1/f2 oder f3/f4, können Sie den Scroll-Effekt direkt beobachten.

Für weitere praktische Experimente verlassen Sie jetzt mit der RETURN-Taste den Editor. Sie gelangen wieder ins Menü zurück. Mit der f5-Taste wird direkt der Parameter-Editor angesprungen. Dort sollten Sie den Befehl „o 1“ eingeben. Hierdurch wird der OR-Modus eingeschaltet. Ein dreimaliges RETURN führt sofort in den Zeichensatz-Editor zurück. Mit HOME und anschließendem CLR/HOME wird das Zeichen wieder gelöscht. Betätigen Sie nacheinander die Tasten „k“, Pfeil-oben und „h“, erzeugt dies einen nach unten zeigenden Pfeil. Geben Sie jetzt „k“, gefolgt von der Pfeil-oben-Taste, ein, entsteht ein sogenannter Doppelpfeil, der zugleich als Muster im oberen Bildschirmteil erscheint.

Im Programmteil ZS Testen können Sie feststellen, daß jeder Druck auf die SPACE-Taste irgendwo im freien Bildschirmbereich den Doppelpfeil erzeugt.

Eine Kostprobe

Mit RETURN und der f3-Taste gelangen Sie in den Sprite-Editor. HOME und CLR/HOME löscht das Sprite. Die Tastenkombination „k“, gefolgt von SPACE, kopiert den Doppelpfeil an Cursor-Position ins Sprite. Dies läßt sich an beliebiger Cursor-Position wiederholen. Da der Oder-Modus eingeschaltet ist, werden die Zeichen übereinandergelagert, so daß sich aus dem Zeichen inter-

essante Muster erzeugen lassen. Mit der RETURN-Taste gelangen Sie in den Test-Modus, wo das Sprite mit den Cursor-Tasten bewegt und mit den Tasten f7/f8 vergrößert werden kann.

Gehen Sie mit RETURN, f1-Taste und RETURN zurück in ZS Testen, können Sie eine weitere Spezialität probieren. Mit der Taste f5 wird das aktuelle Sprite in bestimmte Zeichen kopiert: Sie erstellen sich ein rechteckiges Feld von 3 mal 3 Zeichen. Die Zeichen sollten verschieden sein. Denn nun wird das aktuelle Sprite, das zuletzt im SCE bearbeitet wurde, in das Zeichenfeld hineinkopiert. Besonders Spieleprogrammierer werden das Problem mit den 8 Sprites kennen. Man braucht ja immer mindestens 9 Sprites.

Alle weiteren Befehle, wie Sie sie der Tabelle 1 entnehmen können, sind am

einfachsten durch Probieren kennenzulernen. Außerdem können Sie natürlich auch verschiedene Schreibfarben wählen nach der altbekannten Methode, wie Sie sie aus dem Commodore-Modus kennen.

Falls der Direkt-Modus gesetzt ist (wird rechts angezeigt), wird jede Änderung, die Sie vornehmen, im betreffenden Zeichen oder Sprite gleichzeitig angezeigt. Das Programm liest hierbei mehrmals in der Sekunde die Daten im Editierfeld und schreibt sie sofort in das Sprite oder den Zeichensatz. So ist es einfach, einem Zeichen die gewünschte Gestalt zu verleihen und sich Scroll-Effekte anzusehen.

Ist der Modus ausgeschaltet, wird erst dann das bearbeitete Zeichen oder Sprite in den Zeichensatz geschrieben, wenn Sie dieses mit dem Auswahl-Cur-

sor oben über die Tasten f5/f6 und f7/f8 verlassen. Hat man einmal ausversehen ein vollkommenes Chaos produziert, sollte man die '←'-Taste drücken, damit das alte Zeichen wieder erscheint. Die Daten werden aus dem Speicher geholt und wieder in das Editierfeld gelesen. Aber Vorsicht: diese Funktion ist nur wirksam, wenn der Direkt-Modus ausgeschaltet ist, da nur dann das ursprüngliche Zeichen im Speicher erhalten bleibt.

Nach Art des Hauses

Mit der Block-Funktion steht Ihnen ein besonders praktisches Hilfsmittel zu freien Verfügung. Wenn Sie die Taste 'b' gedrückt haben, können Sie mit den Cursor-Steuertasten und RETURN die beiden Ecken des Blockes im Editierfeld festlegen. Danach können Sie den

<p>Taste Funktion und Bedeutung</p> <p>d Direkt-Modus an/aus</p> <p>q Zeile nach links scrollen</p> <p>w Zeile nach rechts scrollen</p> <p>e Spalte nach oben scrollen</p> <p>r Spalte nach unten scrollen</p> <p>HOME in die obere linke Ecke springen</p> <p>CRS/DWN/ UP/RIGHT/ Editier-Cursor bewegen</p> <p>LEFT DEL Zeichen löschen (wie im C-Modus)</p> <p>INST Zeichen einfügen</p> <p>= ab Cursor-Position Zeile löschen</p> <p>+ Punkt setzen</p> <p>- Punkt löschen</p> <p>@ Punkt setzen und Cursor-rauf</p> <p>; Punkt setzen und Cursor-rechts</p> <p>: Punkt setzen und Cursor-links</p> <p>. Punkt setzen und Cursor-runter</p> <p>0 Punkt löschen und Cursor-rauf</p> <p>p Punkt löschen und Cursor-rechts</p> <p>o Punkt löschen und Cursor-links</p> <p>l Punkt löschen und Cursor-runter</p> <p>RETURN zurück zum Menü</p> <p>k Key-Funktion</p> <p>b Block-Funktion Block ausgeben</p>	<p>Steuerung des Auswahl-Cursors beim ZS-Editor im oberen Bildschirmbereich zur Auswahl der Zeichen</p> <p>f5 nach rechts</p> <p>f6 nach links</p> <p>f7 nach unten</p> <p>f8 nach oben</p> <p>Im Multicolor-Modus:</p> <p>m Multi-Color Modus an/aus</p> <p>1 Cursor-Farbe = Farbe 1</p> <p>2 Cursor-Farbe = Farbe 2</p> <p>3 Cursor-Farbe = Farbe 3</p> <p>Folgende Funktionen wirken auf den Bereich ab Cursor-Position nach rechts und unten</p> <p>Taste Funktion und Bedeutung</p> <p>CLR/HOME löschen</p> <p>i alles invertieren</p> <p>f alles mit gesetzten Punkten füllen</p> <p>v vertikal spiegeln</p> <p>h horizontal spiegeln</p> <p>f1 nach rechts scrollen</p> <p>f2 nach links scrollen</p> <p>f3 nach unten scrollen</p> <p>f4 nach oben scrollen</p>
--	---

Tabelle 1 Funktionen innerhalb der Editoren

s xxxx	neue Startadresse für die Sprites setzen
t xxxx	neue Adresse für den CHAR-Testbildschirm setzen
c xxxx	neue Startadresse für den Zeichensatz setzen
b xx	neues 'BIT-an'-Zeichen setzen
f xx	neue Rahmenfarbe setzen
g x	neue Geräteadresse eingeben
o x	Oder-Modus an- (x=1) oder ausschalten (x=0).
j xxxx	Die Jump-Adresse. Wenn Sie innerhalb des Gesamtprogramms CTRL+SHIFT gleichzeitig drücken, so springt der Computer an diese Adresse !

Tabelle 2: Die Parameter-Befehle

Block wieder mit der Pfeil-Nach-oben-Taste an jeder beliebigen Stelle ausgeben lassen, also auch im jeweils anderen Editor. Somit können Sie auch einen Block im Zeichen-Editor markieren, und im Sprite-Editor wieder ausgeben (oder umgekehrt).

Im Parameter-Editor können Sie sofort eine Eingabe machen. Als Merkhilfe für die Befehle: Die Befehle, die die obenstehenden Daten ändern, ergeben sich aus dem jeweiligen Anfangsbuchstaben des davorstehenden Namens. Um also die Lage des Testbildschirms zu ändern, geben Sie „t“ und eine Adresse in hexadezimaler Schreibweise an.

Noch eine Anmerkung zum Oder-Modus. Ist dieser angeschaltet, so hat dies auf die Key-Funktion die folgende Auswirkung: das nach 'k' angegebene Zeichen wird zwar immer noch ausgegeben, aber Null-Bits (also nichtgesetzte Punkte) werden nicht als Space, sondern als einfaches Cursor-Right interpretiert, die „darunterliegenden“ Punkte bleiben erhalten. Dadurch kann man mehrere Zeichen übereinander oder versetzt kopieren.

Bei gelöschtchem Oder-Modus wird folglich ein Space ausgegeben, so daß eigentlich an dieser Position liegende Zeichen überschrieben wird.

Unter Sprite Editieren kann man mit den Tasten f5/f6 verschiedene Spritenummern anwählen. Weiterhin ist es möglich mit den Tasten f7/f8, die X- und

Y-Ausdehnung (X-Exp/Y-Exp) des Sprites an-/auszuschalten. Selbstverständlich können Sie auch hier mit der Blockfunktion, der '←'-Taste arbeiten und dem „k“-Befehl arbeiten (siehe Tabelle 1).

Im folgenden gehe ich noch auf einige Sonderfunktionen ein: Gesetzt den Fall, Sie befinden sich in irgendeinem Menü, dann stehen Ihnen einige Spielereien zur Verfügung. Zum Beispiel kommen Sie mit der Taste „f1“ immer in den ZS-Editor, mit der Taste „f3“ in den Sprite-Editor und mit der Taste „f5“ in den Parameter-Editor. Natürlich können Sie auch mit Cursor rauf/runter und RETURN die Editoren anwählen.

Weiterhin gelangen Sie mit der Taste „f7“ in einen Disk-Editor und können, dort angekommen, gleich einen Disk-Befehl eingeben und mit RETURN abschließen, den Diskettenstatus mit dem allseits bekanntem @ Klammeraffe abfragen, oder man schaut sich eine Directory über „\$“ an. Drückt man bei leerer Eingabezeile RETRUN, so kommt man zum letzten Menü zurück.

Besonderer Service

Einige Programmteile lassen sich auch anderweitig nutzen. So können Sie den im ZS-Editor definierten Testbildschirm auch für andere Zwecke benutzen, beispielsweise, um ein Bild für ein Spiel zu entwerfen oder einen Titeldschirm auszubauen. Dazu ist es möglich (aber

nur für Assembler-Programmierer), eigene Routinen in den Test-Editor einzubinden. Und zwar müssen Sie dann bei \$7000 die Zeichenfolge HPK in ASCII ablegen und bei \$7003 Ihre Routine anfangen lassen. Diese Routine wird vom Editor 50-mal in der Sekunde aufgerufen (Raster-IRQ). So ist es ohne Probleme möglich, während des Editierens bestimmte Zeichen zu verändern oder sonstige beliebige Spielereien zu betreiben. Aber benutzen Sie nicht die Zero-Page-Pointer von \$BB bis \$FF !!

Weiterhin läßt sich der eben genannte Trick auch beim Testen der Sprites anwenden. Hierzu müssen Sie aber die Adressen \$7100 und \$7103 benutzen.

Nun noch einige Erläuterungen zum Datenformat der Test-Dateien. Beim ZS-Test-Bildschirm werden einfach 1024 (\$0400-\$07FF) Zeichendaten und 1024 (\$D800-\$DBFF) Farbdaten abgelegt. Ein Test-Sprite-File hingegen besteht aus 8 Koordinatenpaaren und 8 Farben.

Cursor	Auswahl markieren
RETURN	Ausführen/Rücksprung
f1	ZS-Editor
f3	Sprite-Editor
f5	Parameter-Editor
f7	Disk-Editor

Sonderfunktionen im Menü-Modus

Die fertig entworfenen Spritedaten oder Zeichensätze können Sie sich per Menüauswahl auf Ihrem Datenträger ablegen (Diskette oder Kassette im Commodore- oder SuperTape-Format). Dabei wird die Basis-Adresse aus dem Parameter-Editor übernommen. Diese Dateien können Sie als absolute Programmfiles (,1) mit einem BLOAD (siehe 64er Tips 3/87) in eigene Programme einbinden. Holger P. Kregel/RH



Vierzig drauf

3 1/2"-Floppy für den C64

Weitere Features dieses zierlichen Speicherriesen sind die Eliminierung all der bekannten 1541-Nachteile, über die schon so viel geschrieben wurde. Das Netzteil ist jetzt abgesetzt und findet sich in einem separaten Gehäuse, der Anschluß erfolgt über einen 4poligen Stecker. Laufwerk und Netzteil erzeugen auch bei hartem Betrieb kaum nennenswerte Verlustwärme. Zur Umschaltung der Geräteadresse müssen weder das Gehäuse geöffnet noch Leiterbahnen getrennt werden: DIL-Schalter an der Rückseite dienen zur Einstellung. Und last not least wird das Directory im Puffer gehalten. Es kann damit jederzeit aus-

Das fehlte doch: 40 Spuren mehr auf der 1541 und ein paar tausend Blocks free . . . Jetzt ist es möglich; nur: das Diskettenformat hat sich geändert, wobei der Begriff „Format“ nicht nur „soft“, sondern auch „hard“ zu verstehen ist. Also: 3 1/2"-Mikrodisketten, doppelseitige Nutzung, 820 KB Kapazität entsprechend 3160 Blocks free, Name: 1581.

gelesen werden, ohne daß das Laufwerk mechanisch anspricht.

Das Nadelöhr zur 1541 bleibt aber auch das Nadelöhr zur 1581. Gemeint ist der serielle Bus, dessen atemberaubende Datenübertragungsrate sich bei der gebotenen Speicherkapazität umso schmerzlicher zu Bewußtsein bringt. Man muß es dagegen würdigen, daß dies gleichzeitig ein Beitrag dazu ist, die 1581 im System kompatibel zu halten. Wie kompatibel also ist die 1581 wirklich?

Format

Das Format ist Commodore-eigen.

Natürlich paßt es nicht zur 1541, will sich aber auch mit Amiga und Atari nicht vertragen. Lichtblick: Da der Controller einen Standard-Floppy-Controller-Chip WD 1772 enthält, sollte es findigen Programmierern gelingen, Files formatkompatibel zu lesen und zu schreiben.

Programme

Programme lassen sich wie mit der 1541 lesen und schreiben. Auch die Syntax ist gleich. Man muß aber bei der Verwendung von Jokern bei der Namensvergebung aufpassen. Das Programm "Test1" kann man zwar mit LOAD "Te★",8 laden, nicht aber mit LOAD "Te★t1",8 und auch nicht mit LOAD "Te★★★",8.

* Floppy-Befehle und Sequentielle Files

Auf allgemeine Floppy-Befehle reagiert das DOS der 1581 penibel. Formatieren und Scratches funktioniert

nur mit der syntaktisch 100%igen Form OPEN 15,8,15;"N0:Hallo.vc" oder . . .;"S0:test". Mit sequentiellen Files kann die 1581 zwar etwas anfangen, sie legt sie auch richtig an. Diese dann vollzuschreiben gelingt allerdings nur selten: Der Versuch, 300 Blöcke mit ASCII-Zeichen vollzuschreiben, scheiterte mal nach 3, mal nach 17, mal nach 5 Blöcken mit „Error . . .“. Außerdem folgt aus dem Gesagten sofort: Vorsicht ist mit BASIC-Erweiterungen (wie EXBASIC) oder Compilern (AUSTRO-COMP) geboten, da diese verkürzte Befehle verwenden, die auf der 1581 nicht laufen. . .

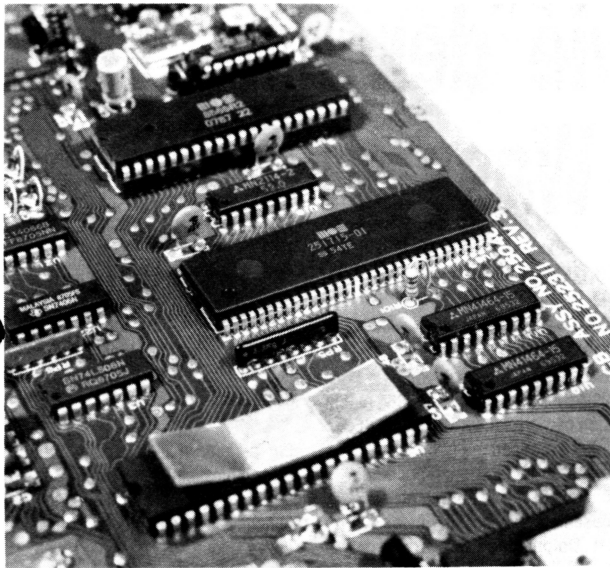
Relative Files

Auch relative Files werden von der 1581 unterstützt. Hier ergeben sich aber dieselben Probleme wie bei den sequentiellen Files: Als Beitrag zum Datenschutz hat die Floppy den Zugriff auf einige Daten unserer Volkszählung verweigert.

Kopf oder Zahl

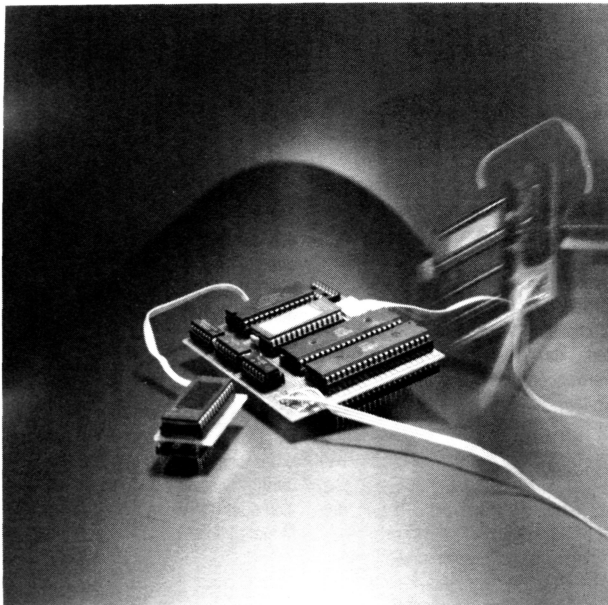
Ist die Dreieinhalber für 64 & Co damit kompatibel oder inkompatibel? Fest steht wohl nur soviel: Anschließen und betreiben läßt sie sich, als Daten sollte man ihr allerdings nur Programme anvertrauen. Da, was sich mangels Dokumentation leider nicht prüfen ließ, die 1581 die Einrichtung von Sub-directories unterstützen soll, lassen sich Files PC-gemäß strukturiert ablegen. Zwar hatten wir nur ein Vorausmodell zur Verfügung, und es wird von Commodore ausdrücklich unterstrichen, daß dieses Laufwerk nie als 100%ige compatible Maschine geplant war — „maximal 50%“; schade drum ist's dennoch. Warten wir also darauf, wie die endgültige DOS-Version aussieht, wenn die 1581 in etwa 2 Monaten das Licht der Läden erblickt.

Wenn's dann wenigstens soweit alles 100%ig 1541-bediensidentisch wäre, könnte man zumindest die Diskettenberge problemlos um 80% schrumpfen lassen; nicht zu sprechen vom doch stabileren Format der 3 1/2"-Mikrodiskette. So bleibt die VC-1581 derzeit nur ein Programm-Massenspeicher (maximal 296 Programme, das 297., 298., 299. . . . wird zwar angenommen, aber nicht abgespeichert) für Spielesammler und Assemblerfreaks. Ob das reicht? Beim „neuesten“ C64C, der jetzt neben einer verkleinerten Systemplatine auch einen neuen Vielbeiner als Custom-Chip enthält, hat Commodore eine wesentlich glücklichere Hand bewiesen: Zwar haben wegen der schon wieder geänderten geometrischen Lage von BASIC und Kernal (letzteres jetzt im 28poligen Gehäuse) Einbau-Ergänzungen noch geringere Chancen als bereits zuvor, doch ist und bleibt die Maschine zum vorherigen C64C, zum „alten“ C64 und zum „ganz alten“ C64 hard- und softwaremäßig voll kompatibel.



Cost-reduced, aber kompatibel!

Eckart Steffens



cherung der Daten und mehrere Treiber-ICs. „Dolphindos“ wird von der Firma Jan Bubela für 198 DM vertrieben. Die Hardware ist, bei etwas technischem Geschick, leicht in den C64 und die Floppy 1541 einzubauen. In der beiliegenden Bedienungsanleitung ist der Einbau ausführlich beschrieben. Sind die ICs in Ihren Geräten gesockelt, brauchen Sie noch nicht mal einen Lötkolben und kommen mit einem Kreuzschlitz- und einem normalen Schraubendreher aus. Das einzige, was vielleicht etwas Schwierigkeiten macht, ist das Einsetzen der beiden Schalter, mit denen die einzelnen Betriebsarten umgeschaltet werden. Dafür muß nämlich in die Gehäuse der Geräte jeweils ein kleines Loch gebohrt werden.

Nach dem Einbau des neuen Betriebssystems habe ich gespannt ein File mit über einhundertfünfzig Blöcken geladen. Der C64 meldete sich überraschenderweise nach drei bis vier Sekunden wieder. Das konnte doch nicht schon alles sein! Doch, das war alles. Die Versprechungen, die in der Bedienungsanleitung gemacht werden, sind nicht übertrieben. 202 Blöcke werden wirklich in 5 Sekunden geladen, das entspricht etwa einem Faktor von 26 gegenüber dem herkömmlichen Commodore-Betriebssystem. Auch das Speichern eines Files geht jetzt 14mal und das Scratching (Löschen) 5- bis 6mal schneller.

Ganz zufrieden bin ich mit der hohen Geschwindigkeit allerdings nicht, denn zum Zwischendurch-Kaffee kommt man jetzt nicht mehr. Dafür sehen mir jetzt aber meine Kollegen neidvoll über die Schulter.

Ein anderer Aspekt ist der erhöhte Bedienungskomfort. Zum Beispiel sind da die Befehle für die Floppy, die übrigens in der Bedienungsanleitung etwas undurchsichtig erklärt werden. Mit dem „@“, der in Insider-Kreisen als Klammeraffe bekannt ist, und

Schnell wie der Fisch im Wasser

Dolphindos beschleunigt Diskettenzugriff

Harte und weiche Ware

„Dolphindos“ besteht aus zwei Platinen. Die Platine, die für den C64 bestimmt ist, enthält ein EPROM mit dem neuen und dem alten Betriebssystem. Auf der Platine für die Floppy befinden sich ein EPROM mit dem neuen DOS (Disk Operating System), ein RAM-Baustein zur Zwischenspei-

Eine neue Generation von Floppy-Speedern für den C64 macht sich bemerkbar. Im Vergleich zu der alten Art sollen Disketten-Operationen nochmals um das Drei- bis Fünffache beschleunigt werden. Einer davon ist „Dolphindos“. Er wurde in der Redaktion einem Dauertest unterzogen.

Cursor links, Cursor rechts

Eine ganz andere Verbesserung des Betriebssystems ist die Erweiterung des Bildschirmeditors. Wenn Sie viel in BASIC programmieren, werden Sie diese Unterstützung nach einiger Eingewöhnungszeit nicht mehr missen wollen. So können Sie zum Beispiel mit einer bestimmten Tastenkombination den Cursor an den Anfang der letzten Zeile bringen oder ihn mit einer anderen Kombination um zwanzig Spalten nach rechts bewegen. Mit wieder anderen Kombinationen können Sie die Zeile links oder rechts vom Cursor löschen oder das Directory ausgeben lassen.

Kommen wir zu guter Letzt zum integrierten Monitor. Er ist vielleicht nicht „das Gelbe vom Ei“, kann aber sehr gut benutzt werden, wenn Sie nicht wissen, wo noch ein Plätzchen für den Monitor frei ist, den Sie normalerweise verwenden.

Ein kleiner Wermutstropfen ist allerdings die eingebaute Centronics-Schnittstelle. Sie arbeitet zwar gut mit einem Drucker zusammen, man muß aber jedesmal den Stecker des Dolphindos aus dem Userport nehmen, damit dieser für den Druckerstecker frei wird. Will man wieder mit der Floppy arbeiten, um zum Beispiel ein anderes File zum Drucken nachzuladen, müssen die Stecker erst wieder getauscht werden. Hier hätte man eine Lösung anstreben sollen, die beide Möglichkeiten erlaubt.

Alles in allem ist „Dolphindos“ aber eine wirklich gut gelungene Erweiterung des C64-Betriebssystems und kann nur weiterempfohlen werden. kfp

Bezugsquelle:

Jan Bubela
Engelsplatz 8
6000 Frankfurt/M. 60
Tel.: 0 69/42 42 10

Befehle		Zeit in Sekunden		Faktor
		C64	Dolphindos	
LOAD	(202 Bl.)	128,3	5,2	25
SAVE	(202 Bl.)	138,8	10,6	13
SCRATCH	(202 Bl.)	24,2	4,4	6
VALIDATE	(Arbeitsdisk)	135,8	15,1	9
Formatieren				
35 Spuren		85,3	19,0	4
40 Spuren		—	20,9	

Geschwindigkeit ist mit Dolphindos wirklich keine Hexerei.

„RETURN“ liest man den Fehlerkanal der Floppy. Auch andere Befehle oder das Lesen des Directory geschieht immer in der kurzen Form mit dem vorangestellten „@“.

Mehr und weniger

Dolphindos erlaubt auch das Formatieren von Disketten mit vierzig Spuren. Mit einem neuen Befehl können Sie die Anzahl der Tracks auf vierzig oder fünfunddreißig fixieren oder das Betriebssystem zur automatischen Formaterkennung veranlassen. Haben Sie die Anzahl der Tracks auf vierzig festgelegt, können Sie auch fremdformatierte 40-Spur-Disketten lesen. Das Schreiben auf solch eine Diskette sollte aber vermieden werden, da vorhandene Files überschrieben würden. Trotzdem können Sie mit Dolphin-Copy (es befindet sich auf der mitgelieferten Diskette) in 20 Sekunden eine 40-Spur-Diskette kopieren und ohne Programmverlust auf 35-Spur-Format bringen.

Eine Erleichterung im Umgang mit dem C64 ist der verbesserte „SYS“-Befehl. Mit dem alten Betriebssystem war es nur möglich, nach dem „SYS“ eine Speicherstelle dezimal anzugeben. Das hat sich geändert. Mit

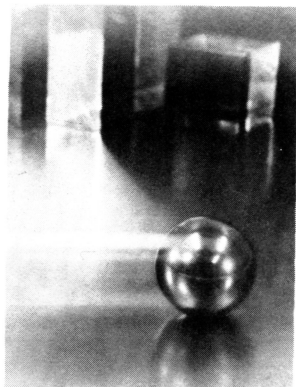
Dolphindos können Sie auch hexadezimale Adressen angeben.

Das Kaufmanns- und (&) bietet unter anderem eine Hilfe an, die es ermöglicht, dezimale in hexadezimale Zahlenwerte und umgekehrt spielend leicht umzurechnen. Dazu geben Sie einfach ein: „&\$000“, anschließend drücken Sie die „RETURN“-Taste. Das neue Betriebssystem schreibt Ihnen dann beide Schreibweisen auf den Bildschirm. Auch umgekehrt funktioniert dieser Befehl.

Außerdem wurden beim „Dolphindos“ Reset- und Restore-Änderungen vorgenommen. Bei einem ganz normalen Reset wird jetzt nur noch das RAM von \$8000 bis \$9FFF getestet. Dadurch läuft diese Routine wesentlich schneller ab. Zum Beispiel bewirkt ein Drücken der „RUN/STOP“-Taste gleichzeitig mit dem Auslösen des Resets einen BASIC-Warmstart. Dabei wird ein eventuell vorhandenes BASIC-Programm nicht gelöscht. Eine andere Tastenkombination ist allerdings mit Vorsicht zu genießen. So füllt zum Beispiel ein gleichzeitiges Drücken der Tasten „←“, „1“ und des „RESET“-Knopfes den kompletten Speicher mit Nullen und führt anschließend einen normalen Reset aus.

Kugeln und Würfel

Metal Ball



Bereits im Titelbild können Sie auswählen, in welchem Modus Sie spielen möchten. Wollen Sie das Spiel mit dem Joystick steuern, drücken Sie einfach die Taste „J“. Ist Ihnen eine Bedienung mit der Tastatur lieber, be-

Im ersten Level können Sie sich vielleicht noch ausruhen. Aber dann geht's rund. Rund um die Kugel aus Metall. Reaktion und Geschicklichkeit sind bei diesem Spiel gefragt.

tätigen Sie die Taste „K“ wie Keyboard.

Für Spieler, die einen Joystick benutzen, bleibt nicht viel zu sagen. Links ist links und rechts ist rechts, oben ist oben und unten ist unten. Die Tastaturbelegung soll an dieser Stelle für diejenigen erwähnt werden, die keinen Joystick besitzen. Taste „A“ = rauf, Taste „Z“ = runter, Taste „.“ = links und Taste „/“ = rechts. Zu steuern ist ein den alten Tennisspielen ähnlicher Schläger, mit dem Sie eine Kugel (Metal Ball) daran hindern müssen, den unteren Spielfeldrand zu überschreiten.

Rechts wird der Spielstand beziehungsweise der Punktestand angezeigt. Nach jeweils 500 Punkten gelangt man in den nächstschwierigeren Level eines Durchganges. Ein Durchgang besteht immer aus drei Levels. Jeweils im dritten Level müssen Sie aufpassen, denn die Steuerung des Schlägers ist seitenverkehrt.

kfp

Lernen im Dialog

Englische GRAMmatik Teil 6

Absolventen altsprachlicher Gymnasien haben sie als Pronomina kennengelernt; in einer lebendigen Sprache wie Englisch stellt sich das Problem mit den Fürwörtern wesentlich einfacher dar.

Vorweg die obligatorische Anmerkung zu den Fähigkeiten und Grenzen eines solchen Programms. Sie können damit vorhandene Kenntnisse der englischen Grammatik überprüfen oder auch im Lauf der Jahre Verschüttetes wieder auffrischen. Ein Lernprogramm ersetzt aber kein Lehrbuch; um sich über Einzelheiten der jeweiligen Regeln zu informieren, kommen Sie um die Lektüre entsprechender Fachliteratur nicht herum.

In diesem Teil der Serie geht es um

- Personal pronouns, subjective
- Personal pronouns, objective
- Possessive pronouns, adjective
- Possessive pronouns, subjective/objective
- Reflexive pronouns
- Emphazing pronouns
- Reflexive or reciprocal pronouns

Die richtigen Eingaben müssen jeweils in die Textlücken der Beispielsätze eingegeben werden (mit RETURN anschließen). 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. Alles Weitere erfahren Sie im Programm. JS

Viele Wege führen ins RAM

INPUT 64-Assembler-Schule, Teil 4

Sie erinnern sich bestimmt an die vier bereits vorgestellten Adressierungsarten: Die meisten Befehle arbeiten „immediate“ und „absolute“. Einige Befehle brauchen keine Parameter und sind „implied“ adressiert, die Branch-Befehle benutzen ausschließlich die relative Adressierung.

Die 6502-CPU kennt aber insgesamt elf Adressierungsarten. Mit den bisher unbekannteren werden Sie in der Lage sein, Programme zu schreiben, die auf Parameter und Adressen zugreifen, die nicht fest im Programm stehen, sondern erst während des Programmlaufes berechnet werden und sich ändern können. Die Adressierungsart, die wir zuerst betrachten wollen, stellt diese Möglichkeiten zwar nicht zur Verfügung, ermöglicht aber schnellere und kürzere Programme als bisher gewohnt.

Seite Null

Wie bereits in einer früheren Folge der Assembler-Schule erwähnt, ist der Speicher eines jeden 6502-Rechners in 256 Seiten zu je 256 Bytes eingeteilt. In der letzten Lektion haben Sie gesehen, daß die Seite eins den Platz für den Prozessor-Stack zur Verfügung stellt. Auch die Seite Null (die Zero-Page), das sind die Adressen 0 bis 255 (\$00 bis \$FF), hat für den Prozessor eine besondere Be-

Ging es in der letzten Folge der Assembler-Schule noch darum, einzelne Bytes in ihre Bestandteile zu zerlegen, so werden wir uns in dieser Lektion damit befassen, möglichst viele Bytes mit einem Befehl in den Griff zu bekommen. Wir zeigen ihnen, wie mit neuen Adressierungsarten die bekannten Befehle besser und flexibler einzusetzen sind.

deutung. Auf sie kann nämlich mit der sogenannten direkten oder Zero-Page-Adressierung zugegriffen werden.

Im Prinzip funktioniert diese Adressierungsart genau wie die absolute Adressierung. Der Unterschied besteht darin, daß zur Angabe einer Zero-Page-Adresse nur ein Byte benötigt wird; alle anderen Adressen sind ja bekanntlich Zwei-Byte-Werte. Zu allen Befehlen, die mit der Adressierungsart „absolute“ arbeiten können (bis auf den JMP-Befehl), gibt es auch eine Zero-Page-Version. In dieser Variante handelt es sich dann um Zwei-Byte-Befehle, die nicht nur um ein Byte kürzer als die gewohnten Befehle sind, sondern auch schneller abgearbeitet werden.

Im Programmtext unterscheiden sich die beiden Versionen dieser Befehle nicht. Der Assembler erkennt, daß die angegebene Adresse auf der Zero-Page liegt, und fügt bei der Übersetzung automatisch den kürzeren Maschinenbefehl ein. Die CPU erkennt bei der Abarbeitung des Programmes bekanntlich am ersten Byte eines Befehles die Instruktionslänge. So ist auch die Zero-Page-Adressierung wie alle anderen Adressierungsarten im Instruktions-Byte verschlüsselt.

Kehrseite

Die Adressen 0 und 1 der Zero-Page sind beim C64 nicht verfügbar. Das liegt daran, daß er mit einer 6510-CPU ausgestattet ist, die diese beiden Register für besondere Zwecke benutzt. Hier ist der gravierendste Unterschied zum 6502, der ja der „Vater“ dieses Prozessors ist. Im C64 wird mit den Registern 0 und 1 die Speicherkonfiguration eingestellt. Dazu mehr in der nächsten Folge.

Da es sich bei der Zero-Page um einen „schnellen“ Speicherbereich handelt und — wie wir noch sehen werden — die mächtigsten Adressierungsarten Platz auf der Zero-Page benötigen, werden von den 254 verfügbaren Speicherstellen die meisten vom Betriebssystem des C64 benutzt. Wenn man auf diesen Speicherbereich zugreift, muß man sich im klaren darüber sein, daß man dadurch das Betriebssystem durcheinanderbringen kann und eventuell einen Absturz des Rechners riskiert.

Unbenutzt sind in der Zero-Page nur die Adressen 2 sowie \$FB bis \$FE. Innerhalb unseres Simulators stehen Ihnen die Adressen \$F8 bis \$FF zur freien Verfügung.

Sie haben in der letzten Folge der Assembler-Schule gelernt, wie man Speicherstellen, die ein Programm zur

Parameterübergabe benötigt, an den Anfang des Programmes legt. Dadurch bleiben die Adressen auch bei einer Programmänderung konstant. Eine andere beliebige Methode, Werte an ein Programm zu übergeben, besteht darin, die Zero-Page dafür zu benutzen. Dann kann das Programm auch in einen anderen Speicherbereich assembliert werden, ohne daß sich diese Adressen ändern. Auch die Rückgabe von Ergebnissen eines Programmes erfolgt häufig über die Zero-Page.

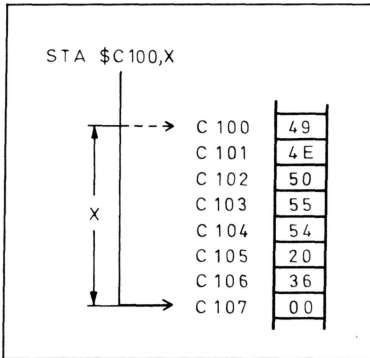


Bild 1: Bei der absolut indizierten Adressierung bildet das Index-Register den Offset

In unserem ersten Beispielprogramm benutzen wir eine Speicherstelle der Zero-Page als Zwischenspeicher. Bevor wir diese Routine unter die Lupe nehmen, sollen Sie noch eine andere Adressierungsart kennenlernen, die dort verwendet wird. Dabei werden die Indexregister X und Y erstmals als solche benutzt. Bisher haben wir sie ja nur als Zähler eingesetzt.

Adressen plus Index

Die Adressierungsart, um die es geht, heißt „Absolut Indiziert“. Bild 1 zeigt das Prinzip an einem Beispiel. Dabei enthält das X-Register eine 7. Diese Adressierungsart wird mit folgender Schreibweise gekennzeichnet:

STA \$C100,X

Zu der im Befehl angegebenen Adresse wird dabei noch der derzeitige Inhalt eines der Indexregister addiert. Das Resultat dieser Addition ergibt dann die Adresse, auf den der Befehl zugeht.

Diese Adressierungsart funktioniert bei den meisten Befehlen, die mit der absoluten Adressierung arbeiten können. Eine genaue Aufstellung der Befehle mit den möglichen Adressierungsarten bringen wir in der nächsten Folge der Assembler-Schule. Sie ist auch in den Büchern enthalten, die Sie am Ende dieses Artikels aufgeli-

stet finden. Innerhalb unseres Simulators können Sie nach dem Motto „Probieren geht über Studieren“ arbeiten. Eine Adressierungsart, die für den eingegebenen Befehl nicht existiert, wird schon bei der Programm-eingabe abgewiesen.

Bei den meisten Befehlen können Sie auch das Y-Register als Index verwenden. Die Notation lautet dann beispielsweise

Diese Adressierungsarten finden immer dann Verwendung, wenn in einem Speicherbereich mit jedem Byte die gleiche Operation durchgeführt werden soll. Mit ein und demselben Befehl ist es — je nach dem laufenden Inhalt des entsprechenden Index-Registers — möglich, auf 256 aufeinanderfolgende Bytes zuzugreifen. Damit lassen sich beispielsweise Tabellen bis zu dieser Länge sehr einfach verwalten.

Sie können zum Beispiel eine Schleife programmieren, die bei jedem Durchlauf das X-Register inkrementiert und dadurch jedesmal eine andere Adresse anspricht. Listing 1 zeigt eine solche Routine. Mit wenigen Befehlen kopiert sie einen Speicherbereich von 256 Bytes Länge in einen anderen.

Wenn Sie sich nun das erste Beispielprogramm auf Ihrem Datenträger anschauen, werden Sie wahrscheinlich nicht auf Anhieb die indiziert adressierten Befehle finden. Sie werden zunächst über einige bisher gänzlich unbekannt Befehle stolpern.

Namen statt Zahlen

Dabei handelt es sich nicht um neue Maschinsprache-Befehle. Es sind vielmehr Anweisungen an den Assembler. Mit ihnen werden Labels, also Assembler-Variablen, deklariert.

```

C083: ; ...
C083: ; 256 BYTES VON ORIG
C083: ; NACH COPY KOPIEREN
C083: LDX     LDX     #0
C085: LOOP   LDA     ORIG,X
C088:         STA     COPY,X
C08B:         INX
C08C:         BNE     LOOP
C08E: ; ...

```

Wenn Sie diese Routine abtippen und in eigenen Programmen einsetzen, müssen die Labels ORIG und COPY vorher deklariert sein

Bislang haben wir Labels immer nur für Adressen innerhalb des Programmes benutzt. Aber ein Assembler kann sich unter einem Label-Namen beliebige Zahlen und auch Adressen außerhalb des Programmes merken und bei der Übersetzung einsetzen.

Der erste Befehl weist der Variablen RTRN die Zahl 13, das ist der ASCII-Wert der RETURN-Taste, zu. Dadurch kann überall im Programm, wo dieser Wert gebraucht wird, statt dessen der Name verwendet werden. Durch diese Maßnahme werden Programme besser lesbar. „Im richtigen Leben“ kann man sich auch Label-Tabellen erstellen, die je nach Bedarf beim Programmieren in den Text geladen werden. Sie können sich auf diese Weise eine Menge Tipparbeit sparen.

Die nächsten vier Befehle haben denselben Sinn: DELC erhält den Wert der DEL-Taste, LEER den der Leertaste, LEFT steht im Programm für Cursor-Links, und unter CRSR steht ein Grafikzeichen zur Verfügung, das in der Routine als Cursor dient.

Bei dem vorgestellten Programm handelt es sich um eine Routine, die Sie auch in eigene Programme einfügen können. Sie enthält einen simplen Zeilen-Editor. Man kann mit ihr bis zu 255 Zeichen von der Tastatur holen und in einem Speicherbereich ablegen. Das jeweils zuletzt eingegebene Zeichen ist mit der DEL-Taste zu löschen.

Die Zero-Page-Adresse \$FE wird durch die folgende Anweisung mit dem Label MERK versehen. Sie wird als schneller Zwischenspeicher für das X-Register dienen.

Der Beginn des Bereiches, in dem der eingegebene String gespeichert werden soll, ist im Programm durch das Label BUFF gekennzeichnet. Das bewirkt die Zeile

BUFF = \$C100

Dadurch, daß auch diese Adresse in einer Assembler-Variablen gehalten wird, brauchen Sie später bei Bedarf nur diese Zeile zu ändern, und die Routine benutzt einen anderen Bereich als Puffer.

INPUT selbstgemacht

Der erste eigentliche Befehl — unter der Überschrift — initialisiert das Index-X-Register. Sein Inhalt wird dann für später in die Speicherstelle MERK gerettet. Beachten Sie die Instruktionslänge von nur zwei Bytes — MERK liegt auf der Zero-Page.

Im nächsten Befehl sehen Sie, wie auf die zu Anfang vereinbarten Namen zugegriffen wird: Der Assembler setzt bei der Übersetzung für CRSR wieder 164 ein, und im Trace-Protokoll wird der Befehl als LDA #\$A4 auftauchen.

Ab EIN1 wird die Tastatur abgefragt; diesen Mechanismus haben Sie ja in der letzten Folge schon kennengelernt. Ist ein Tastendruck erfolgt, so wird das X-Register wieder mit seinem alten Wert geladen. Die Routine GETC könnte es verändert haben (siehe Kasten).

Falls es sich bei dem eingegebenen Zeichen um ein RETURN handelt, verzweigt das Programm nach RAUS, die DEL-Taste bewirkt einen Sprung nach DELT. Andere Sonderzeichen werden durch die Befehle \$C01D bis \$C023 abgewiesen. Der folgende Befehl gibt das empfangene Zeichen als Echo auf dem Bildschirm aus.

Jetzt kommt das eigentlich Spannende an diesem Programm: Der Befehl

STA BUFF,X

speichert das empfangene Zeichen ab. Durch die Verwendung der X-indizierten Adressierung und den nachfolgenden INX-Befehl wird das erste

Zeichen an der Adresse \$C100 abgespeichert, das nächste bei \$C101 und so weiter. Ein Speicherbereich, der mit der indizierten Adressierung erreicht werden soll, muß übrigens nicht immer auf der ersten Adresse einer Seite beginnen; die Basis-Adresse ist beliebig.

Der folgende Branch-Befehl wacht darüber, daß nicht mehr als 255 Zeichen eingegeben werden. Wenn nämlich der INX-Befehl eine Null ergibt, würde das nächste Zeichen direkt auf der Speicherstelle BUFF abgespeichert werden. Der Sprung nach DEL1 bewirkt, daß das 256. eingegebene Zeichen sofort wieder gelöscht wird.

Ist das Ende des Puffers noch nicht erreicht, so wird der folgende Branch-Befehl wirksam, und das Programm wartet auf die nächste Taste.

Ergänzungen und Berichtigungen

Solange Sie im Simulator arbeiten, merken Sie es nicht, aber wenn Sie ein Programm, das die Betriebssystem-Routine GETC benutzt, abspeichern, assemblieren und unter echten Bedingungen laufen lassen, passiert es: Die „wahre“ GETC-Routine im ROM des C64 verändert die Index-Register X und Y. Die Programme 1 und 4 aus der letzten Folge laufen daher nicht in der veröffentlichten Form.

Falls Sie sie trotzdem benutzen wollen, können Sie das erste Programm aus dieser Folge als Muster nehmen: Speichern Sie den Inhalt der noch benötigten Index-Register vor dem Aufruf von GETC irgendwo ab, und holen Sie ihn hinterher wieder zurück.

Das Label RAUS ist der Anfang vom Ende der Routine: Zuerst wird das Ende des eingegebenen Strings mit einem Null-Byte markiert. Für diese Markierung könnte man auch einen anderen Wert benutzen, die Null hat sich aber nun mal eingebürgert. Der BASIC-Interpreter macht es genauso, und das zweite Programm — das der Textausgabe dienen wird — verwendet dieses Kennzeichen auch.

Die folgenden Befehle entfernen den Cursor vom Bildschirm, springen in die nächste Textzeile, und der RTS-Befehl schließlich übergibt die Kontrolle wieder dem aufrufenden Programm. Ihn können Sie zu Testzwecken innerhalb des Kurses durch ein BRK ersetzen, dann bleiben Sie im Simulator und können noch Speicherbereiche inspizieren.

Indexvergleiche

Der Rest des Programmes dient dem Löschen des letzten Zeichens. Der Befehl

CPX

ist neu. Er bedeutet ComPare to X-register und bewirkt dasselbe wie der in Lektion 2 beschriebene CMP-Befehl. Der Unterschied besteht darin, daß das Argument nicht mit dem Inhalt des Akkumulators, sondern mit dem X-Register verglichen wird. Mit Ihrer Vermutung, daß es auch einen Befehl namens CPY gibt, haben Sie übrigens recht.

Mit dem CPX-Befehl überprüft unser Programm vor dem Löschen eines Zeichens, ob überhaupt schon eins eingegeben wurde.

Die restlichen Befehle bewirken neben der Cursor-Bewegung eine Dekrementierung des X-Registers. Dadurch wird zwar das letzte Zeichen nicht wirklich aus dem Speicher gelöscht. Die nächste Eingabe wird es jedoch überschreiben.

Kurz und gut

Der Vollständigkeit halber sei hier erwähnt, daß auch auf die Zero-Page indiziert zugegriffen werden kann. Alle Befehle, die mit der Adressierung „absolut,X“ funktionieren, können auch in der Version „Zero-Page,X“ eingesetzt werden. Ferner kennt die 6502-CPU noch die Adressierungsart „Zero-Page,Y“. Sie ist allerdings den Befehlen LDX und STX vorbehalten.

Wegen des bereits erwähnten Gedränges, das auf der Seite Null herrscht, wird diese Möglichkeit des Prozessors beim C64 verhältnismäßig selten genutzt. Größere Tabellen werden normalerweise nicht in der Zero-Page aufbewahrt. Bei kleineren 6502-Systemen wie dem EPAC65 unserer Schwesterzeitschrift c't, die ohne aufwendiges Betriebssystem auskommen, sind diese Adressierungen jedoch wegen der Platz- und Zeiterparnis durchaus üblich.

Mit Klammer

Die mächtigsten Adressierungsarten, die die 6502-CPU zur Verfügung stellt, imponieren schon durch ihre

Namen: Sie heißen „indiziert indirekt“ und „indirekt indiziert“. Man spricht auch von Vor- beziehungsweise Nach-Indizierung. Beiden ist gemeinsam, daß sie Zeiger (neudeutsch Pointer) auf der Zero-Page benutzen. Dazu kommt noch jeweils ein Index-Register. Alle Befehle, die diese Adressierungsarten benutzen, sind zwei Bytes lang (Instruktionsbyte und Zero-Page-Adresse).

Ein Zeiger besteht aus zwei aufeinanderfolgenden Bytes, die eine Adresse enthalten. Dabei steht der niederwertige Teil der Adresse (das Low-Byte) im ersten, der höherwertige (High-Byte) im zweiten Byte. Diese Reihenfolge — Sie kennen sie ja schon von der Assembler-Anweisung W — ist beim 6502 überall anzutreffen. (Beispielsweise wird auch die Adresse bei einem Drei-Byte-Befehl in dieser Reihenfolge erwartet — aber darum kümmert sich der Assembler.)

Die Adressierungsart „indiziert indirekt“ benutzt zusätzlich zu dem Pointer auf der Zero-Page noch das X-Register. Die Funktionsweise dieser Adressierung soll die linke Hälfte von Bild 2 erklären. Wir setzen voraus,

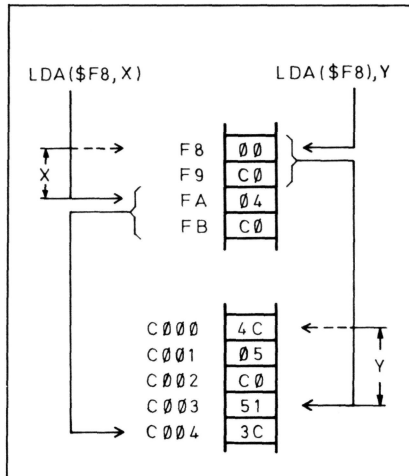


Bild 2: Die mächtigsten Adressierungsarten des 6502 benutzen eine Kombination von Zero-Page-Pointer und Index-Register zur Adreßberechnung.

daß das X-Register den Wert 2 enthält. Die Schreibweise lautet

LDA (SF8,X)

Der Prozessor addiert zunächst den Inhalt des X-Registers zu der angegebenen Adresse. Die beiden an der dadurch angegebenen und der folgenden Zero-Page-Adresse stehenden Bytes werden als Pointer auf die Adresse aufgefaßt, auf die die CPU schließlich zugreift.

Mit dieser Adressierung kann man auf der Zero-Page Adreßtabellen anlesen, auf die sehr leicht zugegriffen werden kann. Wegen der Überfüllung auf der Seite Null ist beim C64 auch diese Adressierungsart eher selten.

Wesentlich bedeutender ist die nachindizierte Adressierung. Sie benutzt das Index-Y-Register. Auf der rechten Seite von Bild 2 ist dargestellt, wie sie wirkt. Das Y-Register enthält in unserem Beispiel den Wert 3.

Die CPU holt sich aus der angegebenen und der folgenden Adresse einen Pointer. Zu diesem addiert sie den Inhalt des Y-Registers und bildet so die anzusprechende Adresse.

Diese Adressierungsart hat mehrere entscheidende Vorteile: Mit nur einem Zero-Page-Pointer kann durch unterschiedlichen Inhalt des Y-Registers ein Bereich von 256 Bytes adressiert werden. Der Beginn dieses Bereiches kann innerhalb des Programmes berechnet werden und ist nicht unbedingt vom Programmierer fest vorzugeben.

Jede Menge Labels

In der Routine PRINT des zweiten Beispielprogrammes wird von dieser Adressierungsart Gebrauch gemacht. Sie dient dazu, einen String mit bekannter Anfangsadresse auf dem Bildschirm auszugeben.

Zum Programm

Die INPUT 64-Assembler-Schule setzt sich aus mehreren Teilen zusammen. Nach dem Laden sehen Sie ein Titelbild, von dem aus Sie mit einem beliebigen Tastendruck in das Hauptmenü gelangen.

Wenn Sie nun F1 drücken, gelangen Sie in ein Menü, das Ihnen verschiedene Themen zur Auswahl stellt. Die Erklärungen, die Sie jetzt abrufen können, sollten Sie parallel zum Beiheft lesen. Beide Medien ergänzen sich hier. Sie können die Erklärungen auch mit CTRL-B ausdrucken. Ins Hauptmenü gelangen Sie jederzeit mit der STOP-Taste zurück.

Mit F3 gelangen Sie aus dem Hauptmenü zu einer Auswahl verschiedener Beispielprogramme. Sie können eines davon auswählen, das Sie sich dann im Editor anschauen oder auch verändern können. Wenn Sie an dieser Stelle eine Null eingeben, enthält der Editor das zuletzt bearbeitete Programm, beim ersten Aufruf ist der Textspeicher leer.

Wenn Sie ein Beispielprogramm bearbeitet haben und — mit der STOP-Taste — wieder ins Haupt-

menue springen, können Sie Ihren Text auch auf einen Drucker ausgeben lassen oder auf einen eigenen Datenträger abspeichern. Abgespeicherte Programme können Sie direkt mit dem INPUT-ASS (Ausgabe 6/86) laden und weiterbearbeiten.

Vom Editor aus gelangen Sie mit F7 in einen integrierten Simulator. Hier können Sie unsere Programmbeispiele oder Ihre selbstentworfenen Programme ablaufen lassen und testen, ob sie sich erwartungsgemäß verhalten.

Ausführliche Hinweise zur Bedienung des Editors und des Simulators sind im Programm enthalten. Sie können Sie von dort aus jeweils mit der Funktionstaste F6 aufrufen. Es wird empfohlen, diese Seiten vor der Benutzung des Programmpakets einmal gründlich zu lesen. Besitzer eines Druckers können sie auch mit CTRL-B zu Papier bringen.

Die INPUT 64-Assembler-Schule ist eine Serie, die in Ausgabe 3/87 begonnen hat. Die einzelnen Lektionen bauen aufeinander auf. Wer noch keine Erfahrungen mit der Maschinensprache-Programmierung hat, tut gut daran, mit der ersten Folge anzufangen.

Das Programm beginnt mit zwei Label-Zuweisungen. PTRL und PTRH sind zwei aufeinanderfolgende Adressen auf der Zero-Page; sie werden als Pointer auf den auszugebenden Text dienen. ANZT ist eine reine Assembler-Variable und enthält die Anzahl der verschiedenen möglichen Texte.

Schauen wir uns noch die letzten Programmzeilen an, bevor das eigentli-

che Programm auseinandergenommen wird. Hier stehen die verschiedenen Texte, die das Programm ausgeben kann. Der Befehl B ist eine Assembler-Anweisung. Wir haben sie schon öfter benutzt, um bestimmte Speicherstellen mit einem Wert vorzubesetzen. Trifft der Assembler bei der Übersetzung hinter der B-Anweisung auf Texte in Anführungszeichen, so setzt er die entsprechenden ASCII-Werte in den Programmspeicher ein.

In den Tabellen TABL und TABH sind die nieder- beziehungsweise die höherwertigen Bytes der Anfangsadressen der Texte gespeichert. Ein Kleiner-Zeichen vor einem Argument signalisiert dem Assembler, daß er das niederwertige Byte des folgenden Wertes erzeugen soll, ein Größer-Zeichen ergibt das höherwertige Byte.

Das Label TXTN haben wir noch vergessen; hier ist die Nummer des auszugebenden Textes gespeichert. Wenn Sie andere Texte einsetzen und ein Programm schreiben, das diese Nummer irgendwie berechnet, können Sie das vorliegende Programm zur Ausgabe beliebiger Meldungen verwenden.

PRINT Marke Eigenbau

Das eigentliche Programm beginnt damit, daß es die Nummer des Textes ins X-Register lädt, und prüft, ob ein solcher Text überhaupt vorhanden ist. Falls ja, wird das X-Register als Index auf die beiden Text-Tabellen benutzt und die Anfangsadresse des Textes in den Akku und das Y-Register geladen.

Werfen wir nun einen Blick auf die Routine PRNT. Sie erwartet die Anfangsadresse des auszugebenden Textes im Akku und im Y-Register, und zwar das Low-Byte im Akku und das High-Byte im Y-Register. Diese Adresse wird als Pointer in der Zero-Page abgespeichert. Unter Benutzung der indirekt-indizierten Adressierung wird der String nun Byte für Byte ausgelesen und an die Betriebssystem-Routine PRCH übergeben. Diese Routine arbeitet so lange, bis sie auf eine Null trifft oder 256 Zeichen ausgegeben sind.

Zurück zur Adresse \$C01D: Hier landet das Programm, wenn das Byte TXTN einen ungültigen Wert enthält. Dann wird die Adresse des Textes ERRT auf die bekannte Art und Weise an PRNT übergeben.

Auf eigenen Füßen

Abschließend gibt das Programm in jedem Falle noch ein Ausrufezeichen aus und wartet auf einen Tastendruck, bevor es zur Aufruf-Adresse zurückkehrt.

Dieses Programm bietet mannigfaltige Möglichkeiten zum Experimentieren. Das einfachste ist, die Zahl bei TXTN zu ändern und dadurch einen anderen der vorhandenen Texte ausgeben zu lassen. Sie können aber auch die Texte selbst oder die Anzahl der Texte ändern. Wenn Sie das Programm in Eigenentwicklungen integrieren wollen, sollten Sie die Befehle, die abschließend auf einen Tastendruck warten, entfernen. Bestimmt fällt Ihnen auch eine Anwendung ein, bei der Sie die beiden in dieser Folge vorgestellten Programme zu einem verbinden.

In der nächsten Folge werden wir die noch fehlenden Maschinensprache-Befehle vorstellen und eine komplette Aufstellung der Befehle und der möglichen Adressierungsarten bringen. Außerdem werden wir einige C64-Besonderheiten besprechen.

Hajo Schulz

Literatur

Christian Persson: 6502/65C02-Maschinensprache, Verlag Heinz Heise GmbH, Hannover 1983

Rodnay Zaks: Programmierung des 6502, Sybex-Verlag GmbH, Düsseldorf 1981

Assembler-Know-how für alle!

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

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

- der komplette Source-Code dieses Assemblers
- der Source-Code des Maschinensprache-Monitors MLM 64 aus INPUT 64 Ausgabe 3/85
- Library-Module: I/O-Routinen, Hex/ASCII/Dezimal-Wandlung, Multiplikation, Division
- Konvertierungs-Programme zur Format-Wandlung von PROFI-ASS- und MAE-Texten in das Source-Code-Format des INPUT-ASS

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

Bestelladresse: Heinz Heise Verlag, Postfach 610407, 3000 Hannover 61

INPUT 64 BASIC-Erweiterung

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

Keine Ladezeiten mehr - über 40 neue Befehle und SuperTape integriert.

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

Bestelladresse: Heinz Heise Verlag, Postfach 610407, 3000 Hannover 61



Verlag **HEISE** GmbH
Bissendorfer Straße 8
3000 Hannover 61

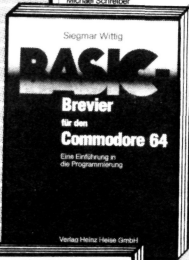
Co 2.4



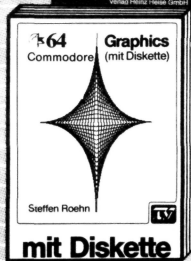
Mit diesem Buch erhalten Sie ein Werk, das konsequent dem Ziel folgt, Ihnen alle Möglichkeiten des VC20 (C=64) aufzuzeigen. Es enthält u. a. folgende Programme: Artikelverwaltung, Finanzbuchhaltung, Auftragsbearbeitung, Kostenrechnung, Zeichen-Generator, Denkspiele.
Best. Nr. 07029-1
DM 32,50



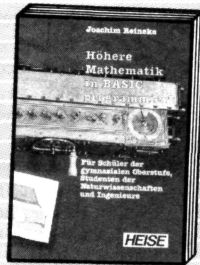
Ein praxisorientiertes, lerngerecht gestaltetes Buch über den Microprozessor 6502 und dessen CMOS-Versionen. Es führt den Leser ohne Lernstreß vom ersten Tastendruck bis zum Entwurf komplexer Systemprogramme.
Best. Nr. 00520-0
DM 48,00



Ein erfolgreiches Buch für einen erfolgreichen Heimcomputer! Es gibt eine erste Einführung in die BASIC-Programmierung des C=64. Vorkenntnisse werden nicht vorausgesetzt.
Best. Nr. 00521-9
DM 29,80



GRAPHICS nutzt die hochauflösende Graphik des C=64 von COMMODORE voll aus. Es ist ein reines Maschinenprogramm, das den BASIC-Befehlssatz um 14 Befehle erweitert. Buch mit Diskette.
Best. Nr. 07026-7
DM 82,50



In vielen mathematischen Gebieten benötigt man Rechentechniken. Es wird stets eine kurze Einführung in die zugrundeliegende Mathematik gegeben. Buch mit Diskette (für den C=64).
Best. Nr. 00527-8
DM 98,00

Sollten unsere Bücher und Softwarepakete nicht bei Ihrem Fachhändler erhältlich sein, bitte direkt anfordern und Verrechnungsscheck zzgl. DM 3,50 Versandkostenpauschale beifügen.

Samplersette

Sampler am Kassetten-Port

Sein eigenes gesprochenes Wort oder kuriose Geräusche in Programme einzubauen wirkt für Benutzer dieser Programme meist besonders beeindruckend. In der Regel sind solche Effekte nur mit einem elektronischem Zusatzgerät (Digitalisierer) und zugehörigem Sampler-Programm erreichbar (siehe auch SAM-Projekt IN-PUT 64 8—11/86). Wer jedoch ohne zusätzlichen Hardwareaufwand selbst Erfahrungen mit dieser Softwareanwendung sammeln möchte, kann mit diesem Programm Sprache oder Geräusche direkt von einer Tonkassette über die allseits bekannte Datensette digitalisieren.

Dabei darf man jedoch keine allzugroßen Erwartungen in die Qualität des Ergebnisses setzen. Die analogen Impulse, die vom Tonkopf des Recorders kommen, werden über eine elektronische Schaltung in der Datensette in Impulse umgewandelt. Dieser Schaltkreis, ein sogenannter Schmid-

Sampler sind in der Regel nur mit Zusatzgeräten zu verwirklichen. Mit solchen Geräten lassen sich analoge Signale wie zum Beispiel Töne über ein Mikrofon aufnehmen, in digitaler Form im Speicher des C64 ablegen, dort verändern und über den Tonausgang des Rechners wieder zu Gehör bringen.

Trigger, sorgt dafür, daß die Informationen, die auf dem Bandmaterial aufgezeichnet sind, vom C64 als Impulse aufgenommen werden können. Ohne diese Schaltung würde sonst jede noch so kleine Störung auf dem Bandmaterial zu Ladefehlern führen.

Hierin unterscheidet sich die Datensette von normalen Recordern. Leider ist dies beim Überspielen von Sprache oder Musik ein echtes Hindernis.

Die Wiedergabe wird stark verzerrt. Das Signal wird dann über den Kassetten-Port an ein Bit des CIA-Bausteins 1 übertragen. Dieses Bit, am Flag-Pin 24 des Bausteins, kann von einem speziellen Programm ausgewertet und die Ergebnisse können im Speicher abgelegt werden. Gibt man diese Bits mit einem entsprechenden zweiten Programm auf das Lautstärkeregister des Soundchip, kann man die Tonsignale über einen angeschlossenen Verstärker oder über den Monitor-Lautsprecher wieder abhören. Man hat mit diesen Programmen einen echten 1-Bit-Sampler.

Mit einigen elektronischen Risiken kann man versuchen, die Elektronik der Datensette zu umgehen, indem der Ausgang eines Kassettenrecorders oder eines Mikrophons direkt an den Kassetten-Port angeschlossen wird. Aber äußerste Vorsicht!!! Der Anschluß 4, dritter Anschluß von rechts — Rückansicht, führt direkt zu dem sehr empfindlichen CIA-Baustein 6526. Spannungsspitzen oder Überlastungen können den gesamten Baustein zerstören. Da dieser unter anderem auch die Tastaturabfrage erledigt, kann der Rechner erst nach aufwendiger Reparatur wieder benutzt werden.

Das Programm Samplersette besteht aus einer Reihe von Routinen, die Sie

Für Kenner

Wer mit dem kompletten Tool-Paket arbeiten möchte, kann die BASIC-Oberfläche löschen und/oder umschreiben. Sie sollten dabei nur auf den hochgelegten BASIC-Anfang achten, also vor dem Abspeichern den Start wieder auf \$0801 zurücksetzen. Reizvoll ist beispielsweise die Erweiterung des Programms um einen kleinen Sequenzer. Ein Sequenzer ist ein Programmteil, der aus einer Ta-

belle der Reihe nach Wiedergabeaufrufe für definierte Speicherabschnitte möglicherweise auch mit unterschiedlicher Geschwindigkeit durchführt. Auf Wunsch kann eine solche Tabelle auch zyklisch abgearbeitet werden, so daß eine scheinbar endlose Tonwiedergabe entsteht.

Andere reizvolle Themen sind zum Beispiel Spracherkennung (bei 1-Bit-Auflösung nur prinzipiell realisierbar) oder Verarbeitung

von externen Meßwerten. Sie sehen, eigenen Ideen und eigener Experimentierfreude steht hier nichts im Wege.

Maschinenprogrammierer können auch die Adresse des Pin-Flag \$DC0D ändern, um dann auf andere Ports des C64 zuzugreifen. Aber denken Sie bei Hardwareverbindungen immer an eine vernünftige Potentialtrennung zwischen externem Gerät und Rechner.

Wiedergeben:	SYS Adr,Anfang,Ende,Rate
Aufnehmen:	SYS Adr+3,Anfang,Ende,Rate
Direktw.:	SYS Adr+6,Rate
Speichern:	SYS Adr+9,Anfang,Ende
Laden:	SYS Adr+12,Anfang,Ende
Löschen:	SYS Adr+15,Anfang,Ende
Legende:	
Adr:	Adresse der Routine (\$0900 oder \$C800)
Anfang:	Anfangsadresse im Speicher
Ende:	Endadresse im Speicher
Rate:	Sample-Auflösung 0 bis 255

zum Teil auch innerhalb von INPUT 64 ausprobieren können. Nur die Routinen zum Abspeichern und Laden sind im Magazin abgeschaltet und werden erst freigegeben, wenn Sie sich das Programm mit CTRL&S auf einen eigenen Datenträger überspielt haben. Das Programm besteht aus einer Reihe von einzelnen Ma-

schinenprogrammen, die Sie auch von BASIC aus aufrufen können:

Wiedergeben
Aufnehmen
Direktwiedergabe
Laden
Speichern
Löschen

Für fertig erstellte Programme ist meist nur die Routine Wiedergeben von Bedeutung. Sie können sich deshalb drei Versionen des Programms Samplersette abspeichern: Das komplette Programm, ein Wiedergabe-Tool am BASIC-Anfang und eine Version im Bereich hinter SuperTape bzw. SuperDisk.

Wie die Routinen von BASIC aus aufgerufen werden können, entnehmen Sie bitte der Tabelle 1.

Erfahrene Programmierer, die mit einem Monitorprogramm umzugehen wissen, können die Wiedergaberoutine im Bereich bei \$C800 beliebig verschieben, so daß sie sich auch mit anderen BASIC-Tools verbinden läßt. Franz Dreismann/RH

16-Takter

C64 mit 4 MHz und 16-Bit-CPU

Sie lesen richtig, es ist tatsächlich der C64 gemeint. Die obige Formel stellt den Familien-C64 sozusagen in die Klasse der Formel-1-Rechner. Was sich dahinter versteckt wird in unserer Schwesterzeitschrift c't ausführlich erläutert.

Das Ganze stellt sich als Projekt der c't-Redaktion vor, das dem C64 völlig neue Leistungsdimensionen erschließt. Es handelt sich dabei um eine Erweiterungskarte, die nicht nur die Arbeitsgeschwindigkeit des C64 um den Faktor vier heraufsetzt, sondern auch die Speicherkapazität vervierfachen kann.

Möglich macht dies der moderne CMOS-Prozessor 65SC816, eine Weiterentwicklung der 6502-Familie, zu der auch die CPU des C64 gehört. Er kann im Emulationsmodus die C64-Software ausführen, ist aber eigentlich ein schneller 16-Bit-Prozessor mit erweitertem Befehlssatz, 16 Bit breiten Registern und der Fähigkeit, bis zu 16 MByte zu adressieren.

RAM-voll

Das RAM des C64 wird auf einer Karte von einem schnellen CMOS-Speicher simuliert, nur bei Zugriffen auf VIC,SID und die I/O-Einheiten des

C64 bremst die Karte ihre Arbeitsgeschwindigkeit und geht auf den normalen Wert herunter. Im Adreßraum des 16-Bit-Prozessors liegt ein weiterer CMOS-Speicher, der mit bis zu 256 KByte bestückt werden kann. Dieses RAM kann von 16-Bit-Software als Arbeitsspeicher benutzt werden. C64-Programme können es als extrem schnelle RAM-Disk nutzen, deren Treiber im EPROM der Karte liegen. Da diese RAM-Disk sogar akkugeduffert ist, können Sie auf dort abgelegte (Hilfs-) Programme direkt nach dem Einschalten Ihres Rechners zugreifen. Später wird auch ein Editor/Assembler im EPROM verfügbar sein.

Die ausführliche Beschreibung der 65SC816-Karte ist in c't 6/87 zu finden. Sie wird übrigens nicht über den Expansion-Port betrieben, der steht nach wie vor für EPROM-Bänke und anderes zur Verfügung.

Manfred Bertuch/RH



Das vorliegende Programm wurde, wie der Name schon sagt, zur Verwaltung von Compact-Discs entwickelt. Falls Sie keine CDs besitzen, kann es im Notfall auch für Ihre Langspielplattensammlung zweckentfremdet werden.

Grundsätzliches

Es muß zwischen zwei Versionen unterschieden werden. Zum einen das Programm, das innerhalb unseres Betriebssystems läuft, das Sie nicht abspeichern können. In dieser Demo-Version sind alle Zugriffe auf Peripherie-Geräte unterbunden. Außerdem sind alle CTRL-Funktionen, die das Programm betreffen, abgeschaltet. Das Demo zeigt Ihnen also nur, was Sie mit diesem Programm alles anfangen können, wenn Sie das Original mit der in INPUT64 üblichen CTRL-S-Sequenz auf Ihren eigenen Datenträger abgespeichert haben. Alle Angaben, die innerhalb dieses Artikels gemacht werden, beziehen sich also auf das Original-Programm, nicht auf die Demo-Version.

Eigenschaften

Sie haben die Möglichkeit, bis zu 255 Datensätze im Rechner zu halten. Mehr kann auf Grund der Speicherkapazität des C64 nicht im Rechner verwaltet werden. Diese Datei können Sie auf eine formatierte Diskette abspeichern. Der CD-Manager berücksichtigt bei der Ablage der Dateien auf Datenträger keine Kassetten-Operationen. Sind sie also „nur“ Besitzer einer Datensette, wird Ihnen dieses Programm keinen Nutzen bringen.

Die von Ihnen erstellten Dateien werden als PRG-Files abgespeichert. Außerdem besitzt das Programm eine Hardcopy-Routine, mit der Sie die eingegebenen Daten auch ausdrucken können. Doch dazu kommen wir später.

CDs ohne Streß verwalten

CD-Manager für den C64

„Ordnung ist das halbe Leben“, sprach Herr Mustermann und legte sich eine Kartei für seine Compact-Discs an. Spätestens nach zwanzig CDs wußte er nicht mehr, nach welchen Kriterien er seine Karteikarten ordnen sollte. Das Problem dabei war, er hatte die Kartei mit ihren vielen einzelnen Karten nach Interpreten sortiert, wollte aber nach Titeln suchen. Das geht natürlich nur, wenn man die

gleiche Kartei mit wiederum vielen einzelnen Karten noch mal erstellt und sie dann nach Titeln sortiert. Eine bessere und einfachere Lösung bieten wir Herrn Mustermann und selbstverständlich auch Ihnen an, den „CD-Manager“. Er löst Ihre Probleme vollkommen und läßt keine Wünsche offen.

Bedienung

Nach dem Starten und dem Erscheinen des Titelbildes kommt man ins Hauptmenü. Vom Hauptmenü aus können sämtliche Untermenüs durch Drücken der jeweils angegebenen Taste aufgerufen werden (Siehe Tabelle 1). Zusätzlich kann mit CTRL-F ein Untermenü zur Einstellung der einzelnen Farben angesprochen werden.

F1	Floppy
F3	Dateneingabe
F5	Daten bearbeiten
F8	Programmende
D	Druckerparameter

Tabelle 1: Das Hauptmenü

Floppy-Menü

Dieses Untermenü bietet Ihnen die Möglichkeiten, Dateien von der Floppy zu laden, Dateien zu speichern, ein Directory zu ziehen und DOS-Befehle zu senden. Tastenbelegung siehe Tabelle 2.

Kommen wir zum Laden von Dateien. Befindet sich bereits eine Datei im Rechner und Sie wollen eine andere Datei laden, erscheint der Hinweis, daß die im Rechner stehenden Daten gelöscht beziehungsweise überschrieben werden. Den Ladevorgang selbst, können Sie mit der RUN/STOP-Taste abbrechen. Dabei wird im Programm eine Fehlermeldung ausgegeben, und die Programm-Parameter werden auf den Urzustand zurückgesetzt. Eventuell auftretende Fehler werden generell abgefangen und führen nicht zu einem Absturz. Wird versehentlich ein File geladen, das vom CD-Manager nicht als zugehöriges Daten-File erkannt wird, wird Ihnen auch dieses mitgeteilt. Ist der Ladevorgang erfolgreich verlaufen, werden in der zweiten Kopfzeile die

Anzahl der Datensätze und die noch freien Bytes angezeigt. Alle anderen relevanten Parameter werden außerdem auf den neuesten Stand gebracht.

F1	Daten laden
F3	Datei speichern
F5	Directory
F7	DOS-Befehle

Tabelle 2: Kommunikation mit der Floppy

Das Gegenteil zum Laden ist das Speichern einer Datei. Mit dieser Option können Sie die Daten vom Rechner auf Diskette abspeichern, wobei der File-Name frei wählbar ist. Alle veränderbaren Parameter werden beim SAVE-Vorgang mit auf Diskette übernommen, so daß bei einem erneuten Laden dieser Daten die gleiche Einstellung der Farben und so weiter wiederhergestellt wird. Das Speichern kann mit der RUN/STOP-Taste nicht abgebrochen werden. Tritt ein Fehler auf, wird der Inhalt des Fehlerkanals auf dem Bildschirm angezeigt.

Das Directory kann unter einem anderen Menüpunkt selbstverständlich ohne Datenverlust geladen werden. Auch hier können Sie mit der RUN/STOP-Taste unterbrechen. DOS-Befehle können Sie in einer verkürzten Form angeben. Sie brauchen also den OPEN-Befehl nicht mit anzugeben. Wollen Sie ein File auf der Diskette löschen, reicht die Angabe: S: Name. Allerdings werden hierbei keine Angaben überprüft.

Eingabe muß sein

Ein wichtiger Teil des CD-Managers ist natürlich die Daten-Eingabe. Dabei können Sie wählen, ob Sie in eine bestehende Datei weitere Datensätze einfügen (F1) oder ob Sie eine neue Datei (F3) beginnen wollen. Nach er-

folgter Eingabe eines Satzes wird dieser automatisch in die Datei, nach Interpretieren alphabetisch aufsteigend, einsortiert. Haben Sie irrtümlich eine CD doppelt eingegeben, macht eine Meldung darauf aufmerksam. Reicht der Speicherplatz des C64 nicht mehr aus, werden vom Programm alle weiteren Eingaben unterbunden. Wollen Sie eine neue Datei erstellen und es steht noch eine vorhandene im Speicher, werden Sie darauf hingewiesen, daß die Daten durch diesen Vorgang gelöscht werden.

Die Eingaberoutine des Programms akzeptiert keine leeren Datenfelder. Sie können also nicht mit der RETURN- oder „Leer“-Taste ins nächste Eingabefeld vorgehen. Bei der Eingabe der Track-Zeit brauchen Sie sich nicht an das vorgegebene Format zu halten. Es wird automatisch angezeigt. Wichtig ist dabei, daß keine Plausibilitäts-Kontrolle erfolgt. Im Prinzip können Sie also auch eine Track-Zeit von „Eis:zeit“ statt „03:45“ angeben. Aber wer macht das schon. Die Eingabe eines Datensatzes, unter dem man alle Angaben, die auf einer CD enthalten sind, versteht, können Sie nicht so ohne weiteres abbrechen. Es sei denn, Sie sind bei der Eingabe der Track-Titel. Drücken Sie hier die RETURN-Taste, rechnet das Programm die Gesamtspielzeit der CD aus, trägt diese in den Datensatz ein und beendet somit die Eingabe.

Ein anderer wichtiger Aspekt ist natürlich das Bearbeiten einer Datei. Dabei haben Sie die Möglichkeit, sich die Datei anzusehen, Daten zu suchen, eine Datei auszudrucken oder einen Satz zu löschen (siehe Tabelle 3).

F1	Datei ansehen
F3	Daten suchen
F5	Datei drucken
F7	Satz löschen

Tabelle 3: Alles, was man will

Wie aus Tabelle 3 ersichtlich, können Sie sich mit der F3-Taste alle eingegebenen Sätze einer Datei ansehen. Mit der RETURN-Taste blättern Sie weiter.

Verwalten heißt auch Suchen

Ein „muß“ jeder Dateiverwaltung ist das Suchen. Beim CD-Manager können Sie nach Interpreten, CD-Titel, Produzenten und nach Track-Titeln suchen. Nachdem Sie eine entsprechende Taste gedrückt haben, müssen Sie den Suchbegriff eingeben. Dabei ist es völlig ausreichend, wenn Sie nur ein Kürzel verwenden. Beispiel: Suchbegriff Interpret, HUEY LEWIS, Eingabe: EWL. Die Suchroutine überprüft nun sämtliche Interpret-Datenfelder. Bei einer Übereinstimmung der Eingabe mit einem Eintrag wird der gesamte Datensatz angezeigt. Während des Suchens wird die Num-

mer des gerade untersuchten Datensatzes auf dem Bildschirm ausgegeben. Den so gefundenen Datensatz können Sie jetzt mit CTRL-D auf Ihren Drucker ausgeben. Hat das Programm keine Übereinstimmung ermittelt, erscheint auf dem Bildschirm eine diesbezügliche Meldung, ebenso bei Erreichen des Dateieendes.

Die Auswahl eines zu löschenden Datensatzes geschieht in der gleichen Weise wie im Programmpunkt DATEN SUCHEN. Es ist hier jedoch sinnvoll, unverwechselbare Suchkriterien einzugeben. Wurde eine Übereinstimmung ermittelt, wird nach einer Sicherheitsabfrage der betreffende Datensatz gelöscht. Wichtig dabei ist, daß dieser Datensatz dann unwiederbringlich verloren ist.

Der Menüpunkt DATEI DRUCKEN beinhaltet zwei Modi. Mit der „F1“-Taste können Sie einen kompletten Datensatz, mit der „F3“-Taste nur den Haupteintrag drucken. Der Hauptein-

trag besteht aus Interpret, Titel, Produzent, Technik und Gesamtzeit. Voraussetzung für jedwedes Drucken ist die Initialisierung des Druckers in der Routine Druckerparameter. Das Drucken läßt sich jederzeit mit der RUN/STOP-Taste unterbrechen.

Drucken kann nicht jeder

Es sollte noch einiges zu den Druckerparametern gesagt werden. Gedruckt werden kann nur mit einem Epson-kompatiblen Drucker. Das Menü zu der Einstellung umfaßt fünf Punkte und ist einfach zu handhaben. Drücken Sie die Taste „1“, wird von verschiedenen Interfaces jeweils ein anderes angezeigt. Das gleiche gilt für Punkt „2“, jedoch können Sie hier zwischen sechs verschiedenen Schriftarten wählen. Gleiches gilt auch für die Punkte 3, 4 und 5. Bedingt durch das eigenwillige Interface des Star NL-10 sind nicht alle Schriftarten möglich. kfp

DREITAUSEND MARK FÜR SIE.

BEIM INPUT 64-PROGRAMMIERWETTBEWERB.
JEDEN MONAT NEU.

WIR WARTEN GESPANNT AUF IHRE GRAFIK-,
MUSIK-, LERN-, ANWENDER- UND SPIEL-
PROGRAMME.

ODER WAS IMMER SIE SONST AUSTÜFTELN.

WERFEN SIE EINEN BLICK IN DIE 'HIN-
WEISE FÜR AUTOREN' - SIE FINDEN SIE
IN JEDEM HEFT.

(NATÜRLICH IST DER RECHTSWEG AUSGE-
SCHLOSSEN.)

64er festgenagelt

Tips für C128-Benutzer

Nach dem Einschalten des C128 werden einige Routinen des 128-Betriebssystems in die RAM-Bank 1 kopiert, unter anderem auch die Einsprungswerte der RESET-Routine. Für das Folgende ist wichtig zu wissen, daß diese RAM-Bank 1 nur im 128er-Modus erreichbar ist.

Speicher-Bank 1 patchen

Um also das kleine Programm starten zu können, muß sich Ihr C128 noch im 128er-Modus befinden. Die entscheidenden Sequenzen befinden sich in den Zeilen 160 bis 180 des abgedruckten Listings. Zuerst wird die RAM-Bank 1 eingeschaltet, und danach werden mit POKE-Befehlen zwei Speicherstellen geändert.

Mit dieser Änderung wird die RESET-Routine so verbogen, daß sie auf das Betriebssystem des C64 zeigt. Da diese Speicherstellen nur durch das Kopieren des ROM ins RAM überschrieben werden (Einschalten des Rechners) und dieser Speicherbereich vom C64 nicht genutzt werden kann, sind diese neuen Werte dort so lange festgeschrieben, wie der Rechner eingeschaltet bleibt.

Floppy nicht vergessen

Wenn Sie also zukünftig Ihren 64er mit einem kleinen Programm starten, liegt es nahe, auch die Floppy 1570/1571 auf eine Floppy 1541 umzuschalten, da einige Programme auf das DOS der 1541 direkt zugreifen.

Dieses Umschalten wird in den Zeilen

In Ermangelung geeigneter Software wird der C128 oft im 64er-Modus benutzt. Es kann schon nerven, wenn der Rechner sich nach einem RESET grundsätzlich mit einem freundlichen "COMMODORE BASIC V7.0 122365 BYTES FREE" meldet und damit zu erkennen gibt, daß er sich wieder im 128-Modus befindet. Gerade Assembler-Programmierer benötigen als letzte Rettung häufig den RESET-Knopf und dürfen dann erst einmal mit "GO 64" den alten Zustand wieder herstellen. Es geht aber auch anders.

110 bis 130 erledigt. Es hat sich bei einigen Rechner-Konstellationen als vorteilhaft erwiesen, zuerst das Laufwerk und dann den Rechner umzuschalten. Trotzdem benötigen einige Laufwerke nach einem durchgeführten RESET zusätzlich noch ein RUN/STOP — RESTORE um wieder ansprechbar zu sein.

128er, wo bist Du?

Wenn Sie das Programm gestartet haben und der 64er sich meldet, schalten Sie den Rechner einmal aus und gleich wieder ein. Sie brauchen nicht zu erschrecken, das Programm hat Ihren Rechner nicht zerstört! Der Grund, warum sich Ihr C128 unerwartet als 64er meldet, liegt zum einen an dem langen Refresh-Zyklus der RAM-Bausteine und zum anderen am langsamen Spannungsabbau des Rechners.

Gönnen Sie Ihrem C128 zwischen dem Ein- und Ausschalten fünf bis zehn Sekunden, und er wird sich wieder wie neu melden.

Jürgen Humpe/WM

```
100 rem "1570/71 <> 1541"  
110 open15,8,15  
120 print#15,"u0>m0"  
130 close15  
140 :  
150 rem "C-128 <> C-64"  
160 bank1  
170 poke65528,77  
180 poke65529,255  
190 go64  
200 new
```

BASIC-Listing: 64er festgenagelt

Einer gegen Alle

Code geknackt

So können Sie diesmal neben dem Lösungsprogramm eine weitere Aufgabenstellung aus INPUT auf Ihren Datenträger überspielen. Der neue Chiffrierungsalgorithmus wurde von M. Baumgart, Böblingen, entwickelt.

Die Auflösung . . .

Vorab der Lösungstext: „Achtung! Der Code ist geknackt!“ Innerhalb von INPUT 64 können Sie sich eine Demonstration des Lösungsweges ansehen. Hier noch einmal in Stichworten die richtige Vorgehensweise bei der Dechiffrierung.

Die erste Ziffer enthielt die Anzahl der folgenden Ziffern (wir hatten dieses als Hilfestellung angedeutet) und die zweite Ziffer die verdoppelte Zufallszahl, die — bei der Dechiffrierung — von den folgenden Werten (halbiert) abgezogen werden mußte.

Allerdings brauchte diese Subtraktion nicht von allen Werten vorgenommen zu werden, da jeder zweite Wert nur zur Verwirrung aufgenommen wurde und eigentlich überflüssig war.

Die so berechneten und bereinigten Werte stellten die ASC-Codes der gesuchten Lösung dar. Daß der Lösungstext nun noch in der falschen Reihenfolge (also von rechts nach links) vorhanden war, stellte für alle, die so weit gekommen waren, bestimmt keine große Hürde mehr dar.

Durch die Verknüpfung des Textes mit einem Zufallswert konnte derselbe

In der Ausgabe 3/87 hatten wir Ihnen eine Dechiffrierungs-Aufgabe gestellt und Sie aufgefordert, aus einem „Zahlensalat“ den gesuchten Text zu rekonstruieren. Immerhin 51 (in Worten: einundfünfzig) Lösungstexte sind bei uns eingetroffen. Auch unsere Aufforderung, eigene Chiffrierungsprogramme zu entwickeln und einzusenden, ist auf ein starkes Echo gestoßen.

Text in unterschiedliche Code-Folgen chiffriert werden.

. . . und die Gewinner

Daß alle Einsendungen den gesuchten Text auch enthielten, liegt in der

Erste Code-Folge				
438,	94,	67,	199,	75,
72,	84,	76,	82,	35,
197,	82,	69,	76,	32,
86,	81,	79,	77,	39,
78,	76,	69,	75,	85,
39,	90,	88,	34,	72,
79,	91,	83,	70,	74,
79,	86,	76,	83,	86,
71,	79,	79,	39,	83,
69,	73,	78,	156,	443,

Die Chiffrierung bedeutet:
„Dieser Code soll nicht zu entschlüsseln sein“

Tabelle 1:
Der erste bekannte Code

Natur der Sache. Wir haben per Los aus den Einsendungen fünf Gewinner ermittelt.

Die Buchpreise gehen somit an: L. Schragmann, Köln; A. Hente, Menden; A. Schworetzky, Lemberg; P. Rojahn, Lübeck; R. Homey, Neuss. Allen Gewinnern unseren herzlichen Glückwunsch.

Da der Lösungsweg nicht gerade ein Spaziergang war, läßt die große Zahl der eingesandten Lösungen darauf schließen, daß hier vorgestellte Chiffrierungen es auch künftig sehr schwer haben werden, vier Wochen unentdeckt zu bleiben.

Neue Kodierung . . .

Diesmal wollen wir keine direkten oder indirekten Hilfestellungen geben, um die Gewinnchancen des Spielers, der den Code entwickelt hat, nicht zu verringern.

Wir werden aber wieder vier chiffrierte Texte vorstellen, zwei davon auch im Klartext, so daß sich für die Tüftler einige Anhaltspunkte ergeben.

Die vier Codefolgen haben wir selbstverständlich in ein kleines Programm verpackt. Mit diesem Programm können Sie Ihre Lösungsansätze austesten (natürlich außerhalb von INPUT). Wir suchen aber — genau wie beim letzten Mal — nur den Lösungstext.

. . . und die Spielregeln

Es sollte theoretisch die Chance bestehen, den Code auch zu knacken.

Zweite Code-Folge

426,	27,	87,	200,	76,
70,	92,	73,	82,	36,
198,	80,	77,	73,	32,
87,	82,	77,	85,	36,
78,	77,	70,	73,	93,
36,	90,	89,	35,	70,
87,	88,	83,	71,	75,
77,	94,	73,	83,	87,
72,	77,	87,	36,	83,
69,	73,	78,	120,	431,

Die Chiffrierung bedeutet auch:

„Dieser Code soll nicht zu entschlüsseln sein“

Tabelle 2:
Der zweite bekannte Code

Eine Verknüpfung beispielweise mit dem Geburtsjahr des Autors (oder anderen externen Zahlen oder Texten) ist zwar ein sehr wirkungsvoller Kodierschlüssel, bietet aber kaum Ansatzpunkte für eine Dechiffrierung. Also: alle Informationen, die für die De-

Dritte Code-Folge

256,	62,	262,	203,	70,
80,	38,	80,	65,	68,
72,	40,	92,	81,	69,
78,	79,	77,	79,	75,
72,	86,	35,	77,	79,
86,	32,	196,	88,	75,
78,	40,	71,	71,	90,
87,	84,	86,	69,	80,
36,	56,	143,	57,	261,
159,	211,	34,	115,	268,

Dieser Text wird gesucht

Tabelle 3:
Der erste unbekannte Code

chiffrierung benötigt werden, müssen als Code irgendwie und irgendwo in der gesamten Codefolge auch enthalten sein.

Nun noch eine Bitte an die Programmierer: Unsere Lichtsatzanlage wäre dankbar, wenn der Code sich aus normalen Zeichen oder Ziffern zusammensetzt und nicht aus Commodore-spezifischen Sonderzeichen. Für den jetzt ausgeschriebenen Wettbewerb bleibt uns nur noch, das Datum mitzuteilen, an dem uns Ihre Lösung erreicht haben muß: es ist Freitag, der 3. Juli 1987. Wenn uns bis dahin kein Lösungstext erreicht, bekommt der Autor des Programms ein INPUT 64 Jahresabonnement, ansonsten verlosen wir wieder fünf Bücher unter den richtigen Einsendungen.

Abschließend ein doppeltes Daumen drücken. Dem Autor, daß seine Kodierung diese Zeit übersteht, und allen,

Vierte Code-Folge

283,	236,	58,	201,	70,
79,	38,	80,	65,	66,
72,	39,	92,	81,	69,
76,	79,	76,	79,	75,
72,	84,	35,	76,	79,
86,	32,	194,	88,	74,
78,	40,	71,	69,	90,
86,	84,	86,	69,	78,
36,	187,	214,	193,	156,
173,	37,	120,	27,	295,

Dieser Text ist identisch mit der dritten Code-Folge

Tabelle 4:
Der zweite unbekannte Code

die sich an der Dechiffrierung versuchen, Erfolg bei ihren Bemühungen!

WM

WIE

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

SO

Durch das INPUT 64-Abonnement. Mit Kassette: 140,- DM pro Jahr. Mit Diskette: 198,- DM pro Jahr. Plus der Möglichkeit zur übernächsten Ausgabe kündigen zu können und auch noch überzahlte Gebühren zurückzubekommen. Die Abo-Bestellkarte ist auf der letzten Seite. ALSO!

Objekte in Bewegung

Sprites – Erstellung und Animation

Sprites werden in der Literatur auch häufig als MOBs (moveable object blocks) bezeichnet. Dieser Name charakterisiert ihre Eigenschaften recht gut. Stellen Sie sich einmal ein Blatt Papier mit einem Bild darauf vor. Wenn Sie nun ein kleines Stück durchsichtige Folie nehmen und diese mit einem kleinen Motiv versehen, zum Beispiel einem Auto, so können Sie diese Folie frei über das Papier bewegen, ohne daß die Bildinformation hinter der Folie zerstört wird. Was die Folie für das Blatt Papier, das ist das Sprite für den Bildschirm. Dabei darf es sich bei dem Hintergrund sowohl um Text- als auch um einen Grafikbildschirm handeln. Der C64 erlaubt sogar die Benutzung von 8 solchen Sprites gleichzeitig. Auf den ersten Blick scheinen Sprites damit fast ausschließlich für Spielanwendungen geeignet. Man stellt zum Beispiel einen Irrgarten als Hintergrund dar und braucht sich bei der Programmsteuerung dann nur noch um die Bewegung der Sprites in dem Irrgarten zu kümmern.

Im Handbuch zu Ihrem Rechner finden Sie eine für Commodore-Maßstäbe recht ausführliche Beschreibung, mit der Sie selbst Sprites programmieren können. Wahrscheinlich kennt fast jeder inzwischen den farbigen Commodore-Ballon. Solange Sie diesen in normalen Bildschirmbereichen halten, entstehen so gut wie keine Probleme. Doch falls Sie Sprites mit verändertem Zeichensatz und damit

Viele Leser denken sicherlich zuerst an actionreiche Spielprogramme, wenn sie den Begriff Sprites hören. Damit wäre auch gleich ein wichtiges Anwendungsgebiet genannt. Doch es gibt auch noch anderen Anwendungen. Wenn Sie sich mit diesen Vorzügen des C64 vertraut machen möchten, sind sich Sie hier richtig.

mit meist verlagertem Bildschirm- RAM kombinieren wollen, suchen Sie vergeblich nach Erklärungen in Ihrem Handbüchlein. Auch die Darstellung auf einem HiRes-Bildschirm bleibt ein Geheimnis, wenn man sich nicht mit zusätzlicher Fachliteratur eindeckt.

Objekt tief

Sprites besitzen eine Größe von 24x21 Punkten. Da sich mit einem Byte 8 Punkte darstellen lassen, werden zur Speicherung der gesamten Informationen demnach 63 Bytes benötigt (24x21/8=63). Damit der Videochip die Mitteilung erhält, woher er sich die Informationen zur Sprite-Darstellung holen muß, gibt es eine Art toten Briefkasten, in dem Sie eine entsprechende Nachricht hinterlegen müssen. Diese 8 Byte große Box ist sozusagen hinter dem Bildschirm aufge-

hängt. Die Adresse errechnet sich aus: Bildschirm-Basisadresse+1016 (+\$3f8), und liegt im Normalfall also 2040 (\$7F8). Hier können für 8 Sprites (0 bis 7) die notwendigen Informationen hinterlegt werden: die Sprite- Zeiger oder Spritepointer.

Sprite0	=	2040
Sprite1	=	2041
usw.		
Sprite7	=	2047

Sie können hier Werte von 0 bis 255 hinterlegen. Der Wert mit 64 multipliziert ergibt dann die Basisadresse im 16K-Bereich des VIC, ab der die 63 Bytes liegen, mit denen das Sprite dargestellt werden soll. Legen Sie in den Sprite-Zeiger für Sprite 0 den Wert 13 ab (POKE 2040,13), erwartet der Videochip die Sprite-Daten ab der Adresse 832, also mitten im Kassettenpuffer. Man spricht in diesem Fall auch vom Sprite-Block 13. Wichtig ist hierbei, daß Sie nur Blöcke festlegen dürfen, die für keine anderen wichtigen Dinge benötigt werden. Das erfordert leider einige Kenntnisse über den Speicher (siehe auch Bedienungshandbuch ab Seite 160). Für Ihre Experimente in BASIC dürfen Sie die Blöcke 11,13 und 14 aber stets frei benutzen. Bei Block 10 liegen beispielsweise wichtige Zeiger und Variablen, die vom Betriebssystem benutzt werden. Würden Sie hier Sprite-Daten ablegen, hätte dies katastrophale Folgen.

Raumverschiebung

Erheblich mehr Denkarbeit ist zu leisten, wenn Sprites über einen Bildschirm huschen sollen, der nicht an der Originaladresse \$0400 (1024) liegt. In der letzten Folge der 64er Tips war von Bildschirm-Verlagerungen die Rede. Wer einen Bildhintergrund mittels eines veränderten Zeichensatzes aufbauen möchte, kommt meist nicht umhin, den gesamten Adreßraum, auf den der Videochip zugreifen kann, zu verschieben, damit für Programm, Variablen und Strings überhaupt noch Platz übrig bleibt. Noch einmal zur Erinnerung: Der VIC kann nur einen Adressraum von 16 KByte ansprechen, in dem Zeichensatz, Bildschirm-RAM und Sprite-Blöcke liegen müssen. In diesem Fall müssen Sie den Briefkasten hinter dem Bildschirm erst einmal suchen. Nach der obengenannten Berechnung ist er jedoch leicht zu finden:

!!!kleine Tabelle!!!

SPP=BSB+1016+SPN

SPP = Spritepointer
BSB = Basisadresse des
Bildschirm-RAM
SPN = Sprite-Nummer

Verlegen Sie den Bildschirm beispielsweise nach \$4000 (16384), liegt der Spritepointer für Sprite 0 bei \$43F8 (17400). Wenn Sie sich entschieden haben, in welchem Eckchen dieses 16K-Bereichs (\$4000-\$9FFF) die Sprite-Daten unterbringen wollen, müssen Sie von dieser Adresse die Basisadresse des 16K-Bereichs abziehen (-\$4000, dez. -16384). Das Ergebnis teilen Sie dann wie oben beschrieben durch 64. Was dabei herauskommt, können Sie als Blocknummer im betreffenden Sprite-Zeiger ablegen. Daß man geschickterweise Adressen benutzt, die ohne Rest durch 64 teilbar sind, müßte sich eigentlich von selbst verstehen. Der

Vorgang nochmals in Kurzform, wie sie Mathematiker so lieben:

$$\text{SBN} = (\text{SBA} - \text{VIB}) / 64$$

SBN = Sprite-Blocknummer
SBA = Sprite-Blockadresse
VIB = VIC-Basisadresse

Lassen wir den Bildschirm bei \$4000, legen den Zeichensatz nach \$6000, können wir bei \$5000 ein paar Sprites unterbringen. Laden wir die Daten von Sprite 0 gleich an den Anfang dieses Bereichs, ergibt sich die Blocknummer $(\$5000 - \$4000) / \$40 = \40 (für Puristen), etwas allgemeiner verständlicher: $(20480 - 16384) / 64 = 64$. Mit POKE 17400,64 setzen Sie dann den Spritepointer für das erste Sprite.

Um zu den Daten zu kommen, die das Aussehen der Sprites bestimmen, gibt es spezielle Programme, die einem diese relativ aufwendige Arbeit abnehmen. Ein entsprechendes Programm ist in dieser Ausgabe erschienen: der INPUT-SCE.

Die Positionen der Sprites lassen sich über eine Reihe von Registern steuern, die ab Adresse \$D000 (5324) angesiedelt sind. Die Basisadresse wird nachfolgend als V bezeichnet. Bevor Sie ein Sprite überhaupt zu Gesicht bekommen können, müssen Sie es erst einmal anschalten. Dies geschieht über Register 21. Dabei entspricht jedes Bit dieses Registers einem Sprite.

Wollen Sie Sprite N anschalten, so müssen Sie in Register 21 das N-te Bit setzen. Dies geschieht von BASIC aus mit:

POKE V+21,PEEK(V+21) OR 2N

Die ODER-Verknüpfung ist hier notwendig, damit die anderen Bits im Register nicht beeinflußt werden. Die ODER-Funktion setzt nur die angege-

benen Bits. Würden Sie den Wert direkt hineinschreiben, würden alle anderen Sprites mit Sicherheit ausgeschaltet. Das Löschen eines Register erfolgt mit

POKE V+21,PEEK(V+21) AND(255-2N)

Mit der AND-Funktion lassen sich gezielt die Bits löschen, die im angegebenen Wert auf Null stehen. Da viele Sprite-Register entsprechend belegt sind (Bit 0 = Sprite 0 usw.), sollten Sie sich mit dieser Methode der Bit-Manipulation vertraut machen (siehe auch 64er Tips 3/85, VIC-Kurs 1/85 f. und Assemblerschule 5/87).

!!!kleine Tabelle!!!

Ein kleines Beispiel zum Probieren:

X=13; Bitmuster: 0000 1101
Y=14; Bitmuster: 0000 1110

A=X AND Y ergibt A=12,
Bit 0 in X wurde gelöscht

A=X OR Y ergibt A=15,
Bit 1 in X wurde gesetzt

Schirmherrschaft

Wenn Sie bereits versucht haben, ein Sprite anzuschalten, werden Sie sich sicherlich wundern, daß auf dem Bildschirm noch nichts geschehen ist. Dennoch ist das Sprite bereits aktiviert, Sie können es nur nicht sehen, weil es sich noch hinter dem Rahmen versteckt. Das Koordinatensystem, das die Sprite-Positionen beschreibt, hat seinen Ursprung nämlich nicht in der linken oberen Ecke des sichtbaren Bildschirms, sondern außerhalb davon. Dies hat den Vorteil, daß Sprites in Programmen am Rand nicht einfach weg sind, sondern schrittweise verschwinden können. Für die Sprite-Position gibt es ebenfalls Register.

POKE V+0, REM X-KOORDINATE - SPRITE 0
POKE V+1, REM Y-KOORDINATE - SPRITE 0
POKE V+2, REM X-KOORDINATE - SPRITE 1
POKE V+3, REM Y-KOORDINATE - SPRITE 1

und so weiter

Effektiv

Wie immer sind auch hier Werte von 0 bis 255 erlaubt. Beim Experimentieren werden Sie jedoch schnell merken, daß bei einer x-Koordinate von 255 das Sprite erst circa 3/4 des Bildschirms überschritten hat. Größere Werte nimmt das Register nicht an, da das Register nur über 8 Bits verfügt. Hätten wir nur ein Bit mehr, könnten wir bereits $2^9=512$ Werte für die x-Koordinate bestimmen. Dieses Bit stellt das Register 16 bereit. Hier existiert für jedes Sprite das fehlende neunte Bit. Wollen Sie also für Sprite 1 die x-Koordinate 300 festlegen, die ja größer als 255 ist, gehen Sie folgendermaßen vor:

POKE V+16,PEEK(V+16)OR21
POKE V+2,300-255

Möchten Sie gerne eine fließende Bewegung bei Sprite 2 erreichen, hilft eine FOR-NEXT-Schleife:

FOR A=10TO50:POKE V+4,A:NEXT

Mit diesen Registern können Sie Sprites beliebig auf dem Bildschirm platzieren. Es gibt aber noch sehr viel mehr Möglichkeiten.

Vorder-/Hintergrund Register 27
einfache Farbe Register 39-46
Multicolor Register 28/37/38
Ausdehnung x/y Register 23/29

Sprites eignen sich aber nicht nur für Spiele. Wer JAM unter IOS kennt, dem ist sicher auch der kleine Pfeil vertraut, mit dem man die verschiedenen Menüpunkte anwählen kann. Wie

Sie sich wohl denken können, handelt es sich auch hier um eine Sprite.

Laufschriften üben eine effektvolle Wirkung auf den Betrachter aus. Wenn Sie Ihren Text in das Sprite-Format umgesetzt haben, können Sie ihn nicht nur in herkömmlicher Weise von rechts nach links, sondern auch diagonal oder vielleicht sogar kreisförmig bewegen.

Einso kann man Sprites für einen Text als einleitenden Buchstaben verwenden, wie dies häufig bei alten gotischen Schriften zu sehen ist. In Verbindung mit einem alternativen Zeichensatz kann das sehr reizvoll aussehen, zumal Sprites durch ihre höhere Auflösung recht feine Details erlauben.

Sprites bieten die gleiche Auflösung

wie hochauflösende Grafik. Wenn Sie Sprites entsprechend anordnen, können Sie so einen kleinen Grafikbildschirm simulieren und dem normalen Textbildschirm überlagern. Ein Tool, das die Organisation eines solchen Bildschirms übernimmt, wurde bereits in der Ausgabe 10/85 veröffentlicht.

Ganz ausgefallene Möglichkeiten bieten die Sprites in Verbindung mit geschickter Raster-Interrupt-Programmierung. Wie im Titelbild von INPUT 64 zu sehen, läßt sich damit Text sogar außerhalb des eigentlichen Bildschirms darstellen und bewegen.

In den 64er Tips innerhalb von INPUT 64 können Sie sich einige interessante Effekte, die sich mit Sprites erzielen lassen, ansehen. Sollten Sie zu eigenen Experimenten motiviert worden sein, können Sie innerhalb der Tips mit CTRL&S wieder ein kleines Experimentier-Programm auf eigenen Datenträger übernehmen, mit dem Sie Zeichensatz- und Sprite-Daten des SCE aus dieser Ausgabe übernehmen und in eigene Programme einbinden können.

Frank Börnke/RH



INPUT 64 BASIC-Erweiterung

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

Keine Ladezeiten mehr - über 40 neue Befehle und SuperTape integriert

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

Bestelladresse: Heinz Heise Verlag, Postfach 610407, 3000 Hannover 61

Hinweise zur Bedienung

Bitte entfernen Sie eventuell vorhandene Steckmodule. Schalten Sie vor dem Laden von INPUT 64 Ihren Rechner einmal kurz aus. Geben Sie nun zum Laden der Kassette

LOAD und RETURN

beziehungsweise bei der Diskette

LOAD"INPUT",8,1 und RETURN

ein. Alles weitere geschieht von selbst.

Sollten Sie ein Laufwerk haben, das zusammen mit dem Schnelllader der Diskettenversion Schwierigkeiten macht, geben Sie bitte ein

LOAD"LADER",8,1 und RETURN.

Nach der Titelgrafik springt das Programm in das Inhaltsverzeichnis des Magazins. Dieses können Sie nun mit SPACE (Leertaste) durchblättern. Mit RETURN wird das angezeigte Programm ausgewählt und geladen. Im Fenster unten rechts erhalten Kassetten-Besitzer weitere Hinweise („Bitte Band zurückspulen“ und so weiter ...).

Haben Sie bei der Auswahl eines Programms eventuell nicht weit genug zurückgespult und es wurde nicht gefunden, spulen Sie bis zum Bandanfang zurück.

Auf der zweiten Kassetten-Seite befindet sich eine Sicherheitskopie. Sollten Sie eventuell mit einem Programm Ladeschwierigkeiten haben, versu-

chen Sie es auf der zweiten Seite. Führt auch dies nicht zum Erfolg, lesen Sie bitte die entsprechenden Hinweise im Kapitel „Bei Ladeproblemen“!

Neben der Programmauswahl mit SPACE und dem Ladebefehl mit RETURN (im Inhaltsverzeichnis) werden die übrigen 'System-Befehle' mit der Kombination aus CTRL-Taste und einem Buchstaben eingegeben. Sie brauchen sich eigentlich nur CTRL und H zu merken (Aufruf der Hilfsseite), denn dort erscheinen die jeweils möglichen 'System-Befehle'. Nicht immer sind alle Optionen möglich (eventuell werden Sie zu Beginn des Programms auf Einschränkungen hingewiesen). Hier nun alle INPUT-64-Systembefehle:

CTRL und Q

Sie kürzen die Titelgrafik ab; INPUT 64 geht dann sofort ins Inhaltsverzeichnis.

CTRL und H

Es wird ein Hilfsfenster angezeigt, auf dem alle verfügbaren Befehle aufgeführt sind.

CTRL und I

Sie verlassen das Programm und kehren in das Inhaltsverzeichnis zurück.

CTRL und F

Ändert die Farbe des Bildschirm-Hintergrundes (auch im Inhaltsverzeichnis erreichbar)

CTRL und R

Ändert die Rahmenfarbe (auch im Inhaltsverzeichnis erreichbar).

CTRL und B

Sie erhalten einen Bildschirmausdruck — natürlich nicht von Grafikseiten oder Sprites! Angepaßt ist diese Hardcopy für Commodore-Drucker und kompatible Geräte. Das Programm wählt automatisch die richtige Geräteadresse (4,5 oder 6) aus.

CTRL und S

Wenn das Programm zum Sichern vorgesehen ist, erscheinen weitere Hilfsfenster. Sie haben die Wahl, ob Sie:

im Commodore-Format C
im SuperTape-Format S
auf Diskette D

sichern wollen. Beachten Sie bitte, daß Sie die Programme von Ihrem Datenträger immer als normale BASIC-Programme mit LOAD"NAME",1 bzw. LOAD"NAME",8 laden müssen. Wenn Sie das Programm im SuperTape-Format aus INPUT 64 abgespeichert haben, müssen Sie vor dem Laden selbstverständlich Super-Tape in Ihren Rechner geladen und initialisiert haben. (SuperTape DII haben wir in der Ausgabe 4/85 veröffentlicht.) Außerdem wird in diesem Fenster die Programmlänge in Blöcken angegeben. Kassetten-Benutzer können diese Disketten-Blockzahl nach folgender Faustregel umrechnen: Im Commodore-Format werden pro Minute neun Blöcke abgespeichert, SuperTape schreibt die gleiche Anzahl von Blöcken in circa sechs Sekunden aufs Band.

Bei Ladeproblemen



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

```
LOAD"LADER",8,1
```

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

Kassette: Schimpfen Sie nicht auf uns, die Bänder sind normgerecht nach dem neuesten technischen Stand aufgezeichnet und sorgfältig geprüft. Sondern: Reinigen Sie zuerst Tonköpfe und Bandführung Ihres Kassettensrecorders. Die genaue Vorgehensweise ist im Handbuch der Datensette beschrieben. Führt auch dies nicht zum Erfolg, ist wahrscheinlich der Tonkopf Ihres Gerätes verstellt. Dieser Fehler tritt leider auch bei fabrikneuen Geräten auf.

Wir haben deshalb ein Programm entwickelt, mit dessen Hilfe Sie den Aufnahme-/

Wieder gabekopf justieren können. Tippen Sie das Programm JUSTAGE ein und speichern Sie es ab. Dieses Programm wertet ein etwa 30 Sekunden langes Synchronisationssignal aus, das sich am Ende jeder Kassettenseite befindet. Starten Sie das JUSTAGE-Programm mit RUN, jetzt sollte die Meldung PRESS PLAY ON TAPE kommen, drücken Sie also die PLAY-Taste. Nach dem Drücken der Taste geht der Bildschirm zunächst wie immer aus. Wird das Synchro-Signal erreicht, wechselt die Bildschirmfarbe, und zwar — bei nicht total verstellter Spurlage — völlig regelmäßig etwa dreimal pro Sekunde. Liegt die Spur des Tonkopfes grob außerhalb der zulässigen Toleranzgrenzen, geschieht entweder nichts, oder die Farben wechseln unregelmäßig. Nehmen Sie jetzt einen kleinen Schraubenzieher und werfen Sie einen Blick auf Ihre Datensette. Über der RE-WIND-Taste befindet sich ein kleines Loch. Wenn Sie bei gedrückter PLAY-Taste durch dieses Loch schauen, sehen Sie den Kopf der Justierschraube für die Spurlage. Drehen Sie diese Einstellschraube. Aber Vorsicht: ganz langsam drehen, ohne dabei Druck auszuüben! Drehen Sie die Schraube nicht mehr als eine Umdrehung in jede Richtung. Nach etwas Ausprobieren wird der Bildschirm gleichmäßig die Farbe wechseln. Zur Feinabstimmung lassen Sie das Synchro-Signal noch einmal von Anfang an laufen. Die Schraube jetzt nach

links drehen, bis der Farbwechsel unregelmäßig wird. Diese Stellung genau merken, und die Schraube jetzt langsam wieder nach rechts drehen: Der Farbwechsel wird zunächst gleichmäßig, bei weiterem Drehe wieder unregelmäßig. Merken Sie sich auch diese Stellung, und drehen Sie die Schraube nun in Mittelstellung, das heißt zwischen die beiden Randstellungen. Denken Sie daran, daß während der Einstellung kein Druck auf den Schraubenkopf ausgeübt werden darf! Der Tonkopf Ihres Recorders ist jetzt justiert.

Sollte sich auch nach dieser Einstellung INPUT 64 nicht laden lassen, erhalten Sie von uns eine Ersatzkassette. Schicken Sie bitte die defekte Kassette mit einem entsprechenden Vermerk an den Verlag ein (Adresse siehe Impressum).

PS! In der Ausgabe 6/85 haben wir das Programm RECORDER-JUSTAGE veröffentlicht, das die Einstellung des Datenrecorders zum Kinderspiel macht.

Listing Justage

```
800 for i=49199to49410:read d:ps=ps+d:poke i,d:next
900 ifps<>24716thenprint"falsch abgetippt - fehler korrigieren!":end
950 print"o.k."
970 sys49338
1000 rem von 49199 bis 49410
1010 data173, 13,220,169,217,174, 4,220,172, 5,220,141, 14,220, 48, 44, 56
1020 data102, 88, 36, 89, 48, 12,144, 10,165, 88,133, 90,169,128,133, 88,133
1030 data 91,192,121,144, 4,224,115,176, 7,169, 0,133, 92, 56,176, 11,165
1040 data 92, 73,128,133, 92, 36, 92, 16, 19, 24,102, 88, 36, 89, 48, 12,144
1050 data 10,165, 88,133, 90,169,128,133, 88,133, 91,104,168,104,170,104, 64
1060 data 96, 36, 91, 16,252,132, 91,165, 90, 96,160,128,132, 89,165, 88,201
1070 data 22,208,250,132, 88,160, 10,132, 89,132, 91, 36, 91, 16,252,132, 91
1080 data165, 90,201, 22,208,226,136,208,241, 32,133,192,201, 22,240,249, 96
1090 data 32,147,252,120, 32, 23,248,165, 1, 41, 31,133, 1,133,192,169, 47
1100 data141, 20, 3,169,192,141, 21, 3,169,127,141, 13,220,169,144,141, 13
1110 data220,173, 17,208, 41,239,141, 17,208,169, 70,141, 4,220,169,129,141
1120 data 5,220, 88, 32,142,192,201, 42,208,249,173, 32,208, 41, 15,168,200
1130 data140, 32,208, 76,237,192,208, 76
```

ready.

Am 6. Juli 87 auf Kassette und Diskette an Ihrem Kiosk: INPUT 64, Ausgabe 7/87



Wir bringen unter anderem:

INPUT-Windowing

Unser Wettbewerbsieger

Windowing war bisher ein Begriff, den man nur mit leistungsstarken 32-Bit-Computern in Verbindung brachte. Er bezeichnet die Verwaltung von mehreren unabhängigen Bildschirm-Fenstern. Mit INPUT-Windowing bekommen Sie nun für den C64 eine BASIC-Erweiterung, die es Ihnen ermöglicht, bis zu 24 Fenster unabhängig voneinander komfortabel (mit neuen BASIC-Befehlen) zu verwalten. Die Geschwindigkeit unserer Fensterverwaltung ist so groß, daß echte Animationen möglich werden. Daß eigene Ein- und Ausgaberroutinen enthalten sind, versteht sich fast von selbst.

ICI

Der IOS-Command-Interpreter

Die grafische Benutzeroberfläche JAM mit unserem INPUT Operating System (IOS) aus der Ausgabe 1/87 hat großen Anklang gefunden. Wir haben nun die Leistungsmerkmale er-

heblich erweitert und IOS mit einem Command-Interpreter versehen, der sogar Batch-Dateien verarbeiten kann. Von der Maus wieder zurück zur Natur, zur Natur des C64, der Bedienung über die Tastatur. ICI erlaubt Ihnen, unter IOS wieder Befehle einzugeben. Sequentielle Files lassen sich mit „TYPE“ auf dem Bildschirm ausgeben, das Directory läßt sich durch den Befehl „DIR“ anzeigen und so weiter. CP/M läßt grüßen . . .

Pingpong Classic

Ein Video-Klassiker

Sie können allein oder zu zweit, mit oder gegen ihren C64, Tischtennis spielen. Wenn Sie sich entspannen wollen, lehnen Sie sich einfach zurück und lassen den Computer gegen sich selbst spielen. Wie auch immer: Schwierigkeitsgrad und Geschwindigkeit sind selbstverständlich einstellbar.

und außerdem:

64er Tips, Englische Grammatik Teil 7, Assembler-Schule Teil 5 u.v.a.m.

c't – Magazin für Computertechnik

Ausgabe 7/87 – ab 12.6.1987 am Kiosk

Projekte: 3.5"-Floppy an PCs, Atari am IEC-Bus * Programme: Fourier-Analyse in 68000-Assembler * Software-know-how: 3D-Objekte manipulieren * Software-Review: QNX für PCs * Prüfstand: Mini-AT mit 10-MHz-Takt, u.v.a.m.

elrad – Magazin für Elektronik

Ausgabe 6/87 – ab 25.5.1987 am Kiosk

Bauanleitungen: D.A.M.E — ein Effektgerät mit außergewöhnlichen Features * Das doppelte Voltchen — Dualnetzgerät 2 x 0 . . . 25 V/2 A * Grundlagen: Alles über Trafos * Bauanleitung: Ferritkernkraftwerk — Schaltwandler von 6 V auf 24 V * Grundlagen: Rauschen von OpAmps * Legal in die Luft gehen: Funken im Urlaub * u.v.a.m.

IMPRESSUM:

INPUT 64

Das elektronische Magazin

Verlag Heinz Heise GmbH
Bissendorfer Straße 8
3000 Hannover 61
Postfach 61 04 07
3000 Hannover 61
Telefon: (05 11) 53 52 - 0

Technische Anfragen:

nur dienstags von 9.00 — 16.30 Uhr

Postgiroamt Hannover, Konto-Nr. 93 05 - 308
(BLZ 250 100 30)
Kreissparkasse Hannover, Konto -Nr. 000 -01 99 68
(BLZ 250 502 99)

Herausgeber: Christian Heise

Redaktion:

Christian Persson (Chefredakteur)
Ralph Hülsenbusch
Wolfgang Möhle
Karl-Friedrich Probst
Jürgen Seeger

Redaktionsassistent: Wolfgang Otto

Ständige Mitarbeiter:

Peter S. Berk
Irene Heinen
Peter Sager
Hajo Schulz
Eckart Steffens

Vertrieb: Anita Kreutzer

Grafische Gestaltung:

Wolfgang Ulber, Dirk Wollschläger

Herstellung: Heiner Niens

Lithografie:

Repretechnik Hannover

Druck:

Leunismann GmbH, Hannover
CW Niemeyer, Hameln

Konfektionierung:

Lettershop Brendler, Hannover

Kassetten- und Diskettenherstellung:

SONOPRESS GMBH, Gütersloh

INPUT 64 erscheint monatlich.

Einzelpreise Kassette DM 16,80
Jahresabonnement Inland Diskette DM 198,—
Einzelpreis Diskette DM 19,80

Redaktion, Abonnementverwaltung:

Verlag Heinz Heise GmbH
Postfach 61 04 07
3000 Hannover 61
Telefon: (05 11) 53 52 - 0

Abonnementverwaltung Österreich:

Erb-Verlag GmbH & Co KG
Abt. Zeitschriftenvertrieb
z. Hd. Frau Pekatschek
Amerlingstraße 1
A-1061 Wien
Telefon: (00 43 2 22) 56 62 09
(00 43 2 22) 57 94 98
(00 43 2 22) 57 05 25

Jahresabonnement: Diskette DM 210,—

Vertrieb (auch für Österreich, Niederlande, Luxemburg und Schweiz):

Verlagsunion Zeitschriften-Vertrieb
Postfach 57 07
D-6200 Wiesbaden
Telefon: (0 61 21) 2 66 - 0

Verantwortlich:

Christian Persson
Bissendorfer Straße 8
3000 Hannover 61

Eine Verantwortung für die Richtigkeit der Veröffentlichungen und die Lauffähigkeit der Programme kann trotz sorgfältiger Prüfung durch die Redaktion vom Herausgeber nicht übernommen werden.

Die gewerbliche Nutzung ist ebenso wie die private Weitergabe von Kopien aus INPUT 64 nur mit schriftlicher Genehmigung des Herausgebers zulässig. Die Zustimmung kann an Bedingungen geknüpft sein. Bei unerlaubter Weitergabe von Kopien wird vom Herausgeber —unbeschadet zivilrechtlicher Schritte— Strafantrag gestellt.

Honorierte Arbeiten gehen in das Verfügungsrecht des Verlages über. Nachdruck nur mit Genehmigung des Verlages. Mit Übergabe der Programme und Manuskripte an die Redaktion erteilt der Verfasser dem Verlag das Exklusivrecht zur Veröffentlichung. Für unverlangt eingesandte Manuskripte und Programme kann keine Haftung übernommen werden.

Sämtliche Veröffentlichungen in **INPUT 64** erfolgen ohne Berücksichtigung eines eventuellen Patentschutzes. Warennamen werden ohne Gewährleistung einer freien Verwendung benutzt.

Printed in Germany

© Copyright 1987 by Verlag Heinz Heise GmbH

ISSN 0177 - 3771

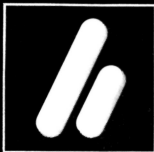
Titelidee: **INPUT 64**

Titelillustration: Tristyle, Hannover

Titel - Grafik und -Musik: Tim Pritlove, Fabian Rosenschein

Betriebssystem: Hajo Schulz

HEISE



Bitte im (Fenster-)Briefumschlag einsenden.
Nicht als Postkarte verwenden!

INPUT64

Vertriebsabteilung
Verlag Heinz Heise GmbH
Postfach 61 04 07

3000 Hannover 61