

INPUT 64

DAS ELEKTRONISCHE MAGAZIN 4/87

Infos · News · Programme · Unterhaltung · Tips

Isolieren statt Heizen

Wärmebedarfsberechnung

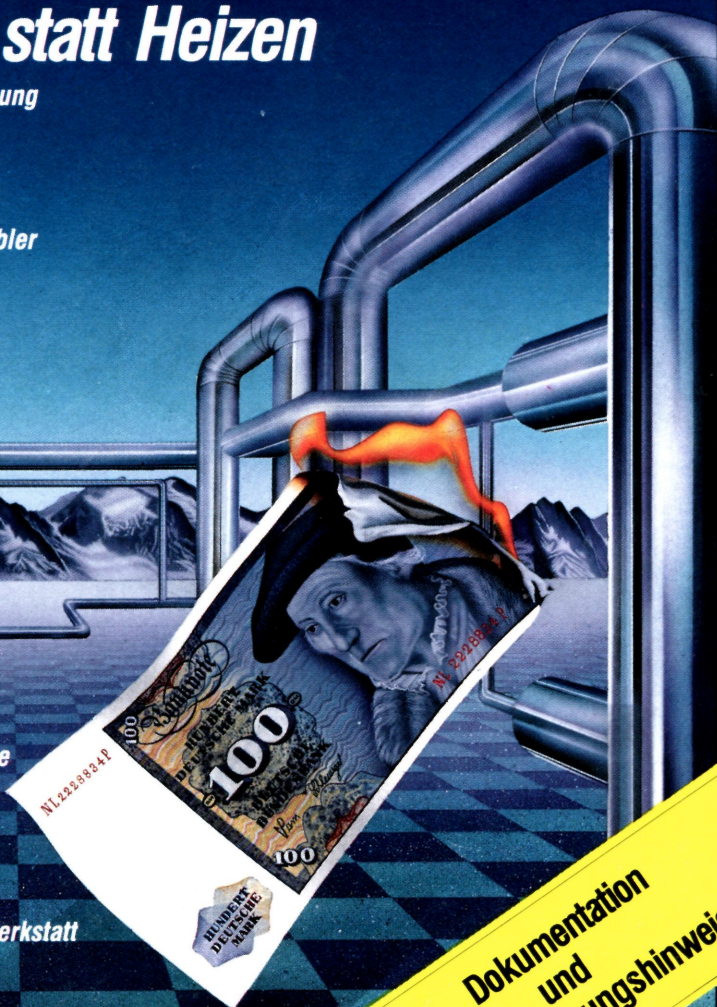
IRAs

Interaktiver Re-Assembler

Kurse:
6502-Maschinensprache
Englische Grammatik
64er Tips

Spiel:
Spekulator

Rätsel-Gewinner, ID-Werkstatt



Dokumentation
und
Bedienungshinweise

Information+Wissen

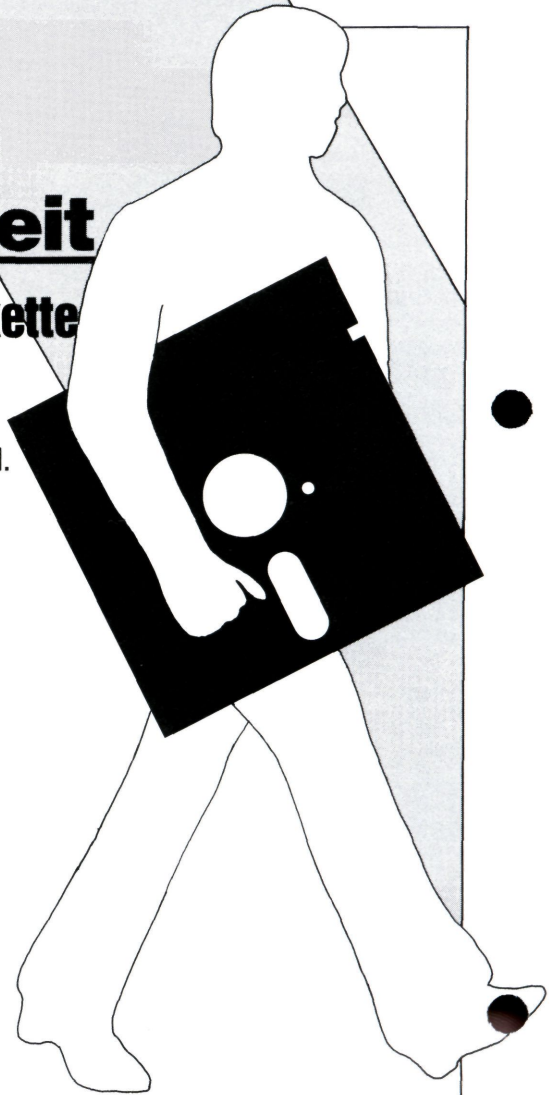
Bundesweit

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)!

Frühlingszeit — Messezeit. Alljährlich im März treffen sich Computereentwickler und -programmierer, -käufer und -verkäufer in Hannover zur CeBIT. Da dürfen natürlich Leute, die für Computeranwender über Computer schreiben, nicht fehlen. Wir waren nicht nur als Aussteller auf der CeBIT, sondern haben uns auch nach Neuigkeiten umgesehen.

So richtige Frühlingsgefühle wollten allerdings in diesem Jahr nicht aufkommen. Das lag nicht an der Qualität oder Quantität der Exponate, sondern ganz einfach am Wetter. Selbst die messeerfahrene hannoversche Verkehrspolizei hatte in den ersten Tagen ihre Schwierigkeiten mit den Auto- und den Schneemassen.

Aber solche Nebensächlichkeiten halten einen richtigen Journalisten nicht auf, wenn es darum geht, etwas Neues zu entdecken. Da stellen sich auf der Messe selbst ganz andere Probleme in den Weg. Zumal wenn man auf

der Suche nach Interessantem für Anwender eines C64 oder C128 ist.

Soweit das Auge reicht, beherrschen PCs, ATs und Kompatible das Bild. Software und Peripherie für diese Rechner sind das beherrschende Thema, zumindest in dem Bereich „Anwendungszentrum für Microcomputer“. Auffällig auch eine immer weiter um sich greifende Mäuseplage.

Pressemenschen haben naturgemäß eine besondere Einstellung gegenüber allem Gedruckten. Aber auch der hartnäckigste Nichtleser schlägt spätestens in der Mittagspause den Messekatalog auf und versucht, sich weitere Laufkilometer zu ersparen. Dieser Blick war zwar auch nicht besonders ergiebig, einige Neuigkeiten, die speziell für 64er- und 128er-Besitzer interessant sein könnten, sind auf diese Weise aber doch noch aufgestöbert worden.

Wegen des ungünstigen Termins un-

seres Redaktionsschlusses können wir in diesem Heft nur kurze Notizen bringen, aber Testgeräte sind geordert, und Sie können damit rechnen, in den nächsten Ausgaben ausführliche Besprechungen der interessantesten Artikel zu finden.

Die originellsten Fundstücke, die wir in der Kürze der Zeit auf der CeBIT 87 entdeckt haben, sind ein Plüsch-Überzug, mit dem eine Maus auch so aussieht und sich so anfühlt wie eine Maus und ein Set mit zwei speziellen Füßen für einen PC. Damit kann man ihn senkrecht aufstellen, und er nimmt nur noch einen Bruchteil des üblichen Platzes ein. Selbst für solche Ideen ist Platz auf der von High-Tech überquellenden größten Computermesse der Welt.

Lajo Schulz

Auf einen Blick:

INPUT64-

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.

INHALT

	ID-Werkstatt: Vokabel-Wandler, CAD-Noten- Datei, INPUT-Calc-Patch	23
	Spiel: Spekulator	24
Leser fragen . . .	CAD-Druckeranpassungen	25
Brandneu: CISAB	Rätsel-Lösung: n-Damen-Problem	27
Wärmebedarfsberechnung	Englische GRAMmatik/5	28
Interaktiver Re-Assembler	Hinweise zur Bedienung	29
Assembler-Schule Teil 2: Flags, Branches, Zählregister	Vorschau	31
64er Tips: Mischfarben	Impressum	32

Leser fragen . . .

Amiga-Ball abspeichern

Eure letzte Ausgabe von INPUT64 war einfach super! Es ist eine wahre Freude, mit dem Floppy-Beschleuniger zu arbeiten; das JAM möchte ich eigentlich auch nicht mehr missen, da es einem doch eine ganze Menge Tipperei erspart. (Es geht um Ausgabe 1/87; JAM ist eine grafisch orientierte neue Benutzeroberfläche für den C64; d. Red.)

Die Amiga-Ball-Simulation ist sehr gut gelungen, nur leider konnte ich sie beim besten Willen nicht abspeichern. Woran liegt das?

(H. Biegler, München)

Die Amiga-Ball-Simulation läßt sich folgendermaßen abspeichern: Wählen Sie innerhalb des Magazins im Modul JAM „RUN“. Wenn der Ball erscheint, wird jetzt mit CTRL und s der Ball abgespeichert, nicht das Betriebssystem JAM. (d. Red.)

SuperDisk verlegen

. . . finde ich Ihre neueste Ausgabe mit SuperDisk (der Floppy-Beschleuniger aus INPUT64 1/87, d. Red.) echt Spitze! Beim Ausprobieren, welche Version für welches Programm einzusetzen ist, verliert man leider mehr Zeit, als die Floppy beschleunigt wird. Es wäre deshalb sehr hilfreich, wenn Sie eine Liste herausgeben würden, aus der die zu verwendende Version für dies oder jenes Programm herausgelesen werden kann, . . . wenigstens für die unter INPUT64 erschiene Programme.

Unabhängig davon würde mich interessieren, was es mit den Initialisie-

rungs-Bereichen auf sich hat. Diese ändern sich automatisch, wenn man ganz oben im Menü die Version 1, 2 oder 3 anwählt. Die dann ganz unten im Menü sich ändernden Bereiche sind aber nicht mit denen im Begleitheft identisch. Oder sollte ich da doch etwas nicht verstanden haben?

(E. Helfrich)

Herauszufinden, welche Programme sich mit welcher SuperDisk-Version vertragen, übersteigt leider die Arbeitskapazität der Redaktion. Statt dessen möchten wir hiermit unsere Leser auffordern, uns ihre Kompatibilitäts-Erfahrungen mitzuteilen. Wir werden diese in den nächsten Ausgaben veröffentlichen.

Zum Problem der „unterschiedlichen“ Adreßangaben im Beiheft und auf dem Bildschirm: Im Heft beziehen sich diese Angaben auf den Speicherplatzbedarf des Hauptteils des Programms; auf dem Bildschirm ist der außerdem benötigte Platzbedarf für die Initialisierungs-Routinen angegeben. (d. Red.)

BCD-Arithmetik mit Subtraktion

In Ihrer Beschreibung des Programms „BCD-Arithmetik“ (INPUT 3/87) fand ich keinen Hinweis auf die Bedienung der Subtraktions-Routinen. Vergessen oder nicht vorhanden? (tel. Anfrage)

Vergessen. Die SYS-Adressen für die Subtraktion lauten SYS 2826 für die Version 1 und 2 dieses Tools und SYS 49162 für Version 3. (d. Red.)

Drei mal Drei ausfragen

Gibt es eigentlich eine Möglichkeit, sich bei dem Spiel „Drei mal Drei“ (INPUT 2/87) die richtige Antwort zeigen zu lassen? (tel. Anfrage)

Das ist im Prinzip nicht vorgesehen. Aus der Phase der Programmentwicklung ist aber übriggeblieben, daß ein Druck auf die „+“-Taste und ein anschließendes „-“ dem Programm die richtige Antwort entlocken. (d. Red.)

Dienstag ist Lesertag!

Technische Anfragen:
nur Dienstag von 9 - 16.30 Uhr

Telefon: (05 11) 53 52-0

In letzter Minute eingetroffen

CISAB-Interpreter für den C64

Ungewöhnliche Ereignisse erfordern ungewöhnliche Maßnahmen. Das Ereignis, von dem ich hier spreche, ist das Eintreffen eines Programmvorschlages von A. Scherzer aus Bochum zwei Tage vor dem Redaktionsschluß zu dieser Ausgabe. Die ungewöhnliche Maßnahme ist die Veröffentlichung dieses nach einhelliger Redaktionsmeinung sensationellen Programmes ohne Ankündigung und ohne Zuerkennung des 3000-DM-Preises.

Doch nun zum Programm selbst: CISAB ist die Abkürzung für „Complementär Interpretierte Sprach-Alternative zu BASIC“, also eine Programmiersprache, die eng an BASIC angelehnt ist, aber trotzdem völlig neue Strukturen enthält.

Der BASIC-Befehl GOTO zum Beispiel, der, wie allerorten zu hören ist, einer übersichtlichen Programmstruktur höchst abträglich ist, ist einer JUMP-COMEFROM-Konstruktion gewichen,

die auch alle anderen Sprungbefehle ersetzt und nebenbei noch für eine beträchtliche Geschwindigkeitssteigerung sorgt.

Ausführliche Erklärungen mit zahlreichen Beispielen enthält das Modul auf Ihrem Datenträger. Als weiterführende Lektüre sei das Buch „Alternativ Programmieren - Die Sprache CISAB“ von Albert Scherzer aus dem Incredible-Books-Verlag, München, empfohlen.

Aus dem Modul heraus können Sie den Interpreter mit CTRL-s auf Ihren eigenen Datenträger überspielen. Er wird dann wie ein normales BASIC-Programm geladen und mit RUN gestartet. Das System gibt eine Meldung aus, und Sie befinden sich im CISAB-Direktmodus.

(d. Red.)

CeBIT-Splitter

20 MByte am C64/128

Unter dem Namen „LT. Kernal“ zeigte die amerikanische Firma Xetec auf dem Stand von Commodore eine Harddisk mit einer Kapazität von 20 MByte zum Anschluß an einen C64 oder C128. Das Produkt besteht aus einem Interface zum Einstecken in den Extension-Port und dem eigentlichen Plattenlaufwerk. In dem Einsteckmodul ist neben einer SCSI-Schnittstelle für die Platte eine Betriebssystem-Erweiterung untergebracht, die dem Benutzer unter anderem 42 neue Befehle zur Verfügung stellt. Als Option soll ein Multiplexer erhältlich sein, der es erlaubt, vier Rechner an einer Platte zu betreiben. Durch die Verkettung mehrerer Multiplexer können sich bis zu 15 Benutzer eine Platte teilen.

Die Ladegeschwindigkeiten liegen nach Angaben des Herstellers um das 110fache einer 1541. Am C64 werden 38 KByte, beim C128 65 KByte pro Sekunde bei einer mittleren Positionierungszeit von 8,3 ms übertragen.

Leider fehlt der Platte in Deutschland noch die FTZ-Zulassung. Nach ihrer Zuteilung will die Firma Xetec das LT. Kernal unverzüglich in Deutschland einführen. Der Preis beträgt in Amerika, wo es seit Januar 1987 verkauft wird, 899 US-\$ mit 64er-Anschluß und 949 US-\$ für den 128er.

Xetec Inc., 2804 Arnold Road, Salina, KS 67401

Computer-Tuning

Die Firma Roßmüller aus Bonn stellte auf der CeBIT eine Steckplatine für den Expansion-Port des C64 vor, die einen 65816-Prozessor, 64 KByte CMOS-RAM und ein Betriebssystem-EPROM

enthält. Die CPU ist mit 4 MHz getaktet und 99,9%ig kompatibel zu allen C64-Programmen. Die Geschwindigkeit ist einstellbar von einem Zehntel bis zum Vierfachen der 64er-Rechenleistung. Darüber hinaus kennt der Prozessor einen 16-Bit-Modus, der mit spezieller Software noch einmal eine beträchtliche Geschwindigkeitssteigerung bringt. Die ab Mitte März lieferbare Karte soll 389,- DM kosten.

Ebenfalls von Roßmüller kommt unter dem Namen MACH 71 ein Beschleuniger für die Floppies 1570 und 1571 zum Betrieb am C128. MACH 71 erhöht in allen drei Modi des C128 die Ladegeschwindigkeit erheblich. So wird CP/M in circa 12 Sekunden gebootet, im 128er-Modus werden 200 Blocks in circa 3,3 Sekunden geladen. Außerdem bringt MACH 71 neue Betriebssystem-Befehle. Der Floppy-Beschleuniger kostet für den C128 398,- DM und für den C128D (mit eingebauter Floppy) 349,-.

Roßmüller, Maxstraße 50-52, 5300 Bonn

Isolieren statt heizen

Wärmebedarfsberechnung für jedermann

Das Außenthermometer zeigt eine Temperatur von minus zehn Grad, der Ofen läuft auf vollen Touren, und das Innenthermometer zeigt nicht mehr als plus sechzehn oder siebzehn Grad. Wo bleibt also die Wärme? Darauf gibt es eine eindeutige Antwort: Die Wärme entweicht durch die Wände und vor allem durch die (geschlossenen) Fenster Ihrer Wohnung oder Ihres Hauses.

Jetzt stellt sich natürlich gleich die nächste Frage. Wie kann man Abhilfe schaffen, damit die Wärme innerhalb der „vier“ Wände bleibt? Jetzt sollten wir schon etwas mehr ins Detail gehen und eine Wand und ihren Aufbau einmal näher betrachten.

Schicht um Schicht

Jede Wand, ob Innen- oder Außenwand, besteht aus mehreren Schichten. Diese könnten bei einer Außenwand, wenn man von innen nach außen sieht, folgende sein: Eine Schicht dünner Putz, eine Schicht Mauer- oder Ziegelsteine, eventuell eine Luftschicht, wieder eine Schicht Mauer- oder Ziegelsteine und dann wieder Putz. Diese Wand besteht also aus fünf Schichten. Es muß jetzt berechnet werden, wieviel Wärme durch jede angegebene Schicht durchgelassen (transmittiert) wird. Wieviel Energie jede einzelne Schicht durchläßt, richtet sich nach der materialspezifischen Wärmeleitzahl und dem Wärmeübergangswert, der sich nach der Lage des Bauteils richtet; nämlich, ob dieses eine Außen- oder Innenwand oder eine außen- oder innenliegende Dec-

Jedes Jahr im Winter merken viele Menschen, daß ihr Brennstoffverbrauch zu hoch ist. Spätestens zu diesem Zeitpunkt werden Überlegungen angestellt, wie man den Verbrauch von Öl, Gas, Kohle oder anderen Energieträgern geringer halten könnte. Es soll aufgezeigt werden, welche baulichen Veränderungen man vornehmen kann, um die Heizkosten zu senken.

ke ist. Aus der Wärmeleitzahl und der Stärke der einzelnen Schicht wird der Wärmedurchlaßwiderstand berechnet.

Die Temperaturen müssen bei den folgenden Berechnungen mit berücksichtigt werden. Wir nehmen eine mittlere Außentemperatur von minus zehn und eine Innentemperatur von plus zwanzig Grad an und bilden die Differenz. Diese Differenz muß in unsere Berechnungen mit eingehen. Außerdem benötigen wir noch die Fläche der Wand und die Fläche der eventuellen Fenster. Aus diesen vielen schon errechneten Werten kann der sogenannte (und jetzt kommt ein langes Wort) „mittlere Wärmedurchgangskoeffizient“ errechnet werden. Da dieser Koeffizient in Watt pro Quadratmeter mal Kelvin angegeben wird und man aus Tabellen ersehen kann, welchen Heizwert einzelne Brennstoffe haben, läßt sich schnell errechnen, wie hoch der Energieverbrauch ist. So, das war's in groben Zügen. Auf nähere Einzelheiten soll hier nicht

eingegangen werden. Dafür gibt es einschlägige Fachbücher.

Damit Sie sich nicht mit all diesen Formeln herumschlagen müssen und dabei trotzdem in den Genuß einer sauberen Berechnung kommen, haben wir ein Programm erstellt. Es ist relativ leicht zu handhaben und anwenderfreundlich gestaltet.

Mit diesem Programm sollen Sie ein Beispiel berechnen. Sehen Sie sich den Grundriß eines Hauses an (Bild 1). Dieses Haus hat sechs Außenwände, eine Kellerdecke und, da das Haus ein Flachdach hat, eine Dachdecke. Berechnet wird jetzt mit unserem Programm die Wand mit der Positionsnummer 1. Dabei wird vorher festgelegt, daß das Bauteil eine Außenwand mit Hinterlüftung ist und aus den oben genannten fünf Materialschichten besteht.

Die eigenen sechs Wände

Starten wir also das Wärmebedarfsprogramm! Nach den einleitenden Sätzen drücken wir die Taste 'J'. Falls Sie jetzt die Erklärung sehen wollen, drücken Sie abermals die 'J', andernfalls die 'N'-Taste. Als erstes werden Sie nach der Positionsnummer gefragt. Geben Sie beispielsweise eine 1 ein. Danach werden Sie aufgefordert, den Namen der Position anzugeben. Hier geben Sie ein: 'Aussenwand'. Nachdem die Eingaben korrekt abgeschlossen wurden, werden acht Bauteile aufgelistet. Angefangen bei Außenwand, Außenwand mit Hinterlüftung und so weiter. Da die Außen-

Das dritte Bauteil wird im ersten Abschnitt genauso gehandhabt. Lediglich die Frage nach den Öffnungen im Bauteil muß anders beantwortet werden. Nachdem Sie aus dem Fenster- und Türen-Menü wieder die 2 gewählt haben, geben Sie erst die Maße für das kleine Fenster ein (2.01 und 1.385). Die nächste Frage beantworten Sie mit 'J', denn das Fenster ist mit den gleichen Maßen mehrmals in diesem Bauteil vorhanden. Das Programm fragt jetzt, wie oft das Fenster vorkommt. Sehen Sie sich wieder das Bild 1 an. Im Bauteil mit der Positionsnummer 3 sind insgesamt vier Fenster beziehungsweise Öffnungen enthalten. Drei der Öffnungen sind gleich, eins unterscheidet sich in den Abmessungen. Das Fenster ist dreimal vorhanden; sie geben demnach bei der Frage nach der Anzahl der Öffnungen eine 3 ein. Sollen noch andere Öffnungen berechnet werden? Ja, denn ein großes Fenster (3.13 mal 2.33) ist ebenfalls vorhanden.

Die anderen drei Bauteile (Position 4

bis 6) werden analog zu den drei ersten Positionen eingegeben.

... variable Temperatur

Sind alle sechs Positionen berechnet, wird die Frage nach weiteren Bauteilen mit 'N' beantwortet. Anschließend können Sie wählen, ob das Ergebnis auf dem Drucker oder auf dem Bildschirm ausgegeben werden soll.

Nach allen Ausgaben erfolgt der Hinweis, daß alle Berechnungen erfolgt sind. Sie drücken dann nur irgendeine Taste und können die einzelnen Bauteile oder Fenster mit einer besseren Wärmedämmung ausstatten. Anschließend werden wieder alle Angaben neu berechnet, und Sie können wieder entscheiden, ob auf den Drucker oder den Bildschirm ausgegeben werden soll.

Wie Sie selbst bemerkt haben werden, sind die Dachdecke und die Kellerdecke nicht mit berücksichtigt.

Denn um berechnen zu können, welche Einsparungen es bringt, wenn Sie statt der Fensterposition 2 die Position 3 auswählen, genügt es schon, eine Wand zu berechnen. Haben Sie

oben offen, unten offen

aber vor, Ihre ganze Wohnung zu berechnen, müssen Sie selbstverständlich die zwei Decken (Fußboden- und Dachdecke) mit berücksichtigen. Vielleicht rechnen Sie sich mal aus, was eine Isolierung der Kellerdecke ausmachen würde. Selbstverständlich sollte man bei irgendwelchen Isolierungsarbeiten darauf achten, daß die Isolierschicht (Styropor, Stein- beziehungsweise Mineralfasermatten) immer da aufgebracht wird, wo die kalte Seite des zu isolierenden Bauteils ist.

Die Druckerausgabe ist nicht für alle Drucker angepaßt. Sie finden Sie im Listing ab der Zeile 1500 und können selbst entsprechende Änderungen vornehmen. 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.)

Vielfalt.

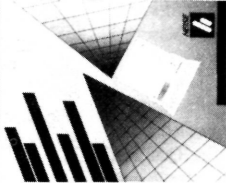
Mehr als hundert Elektronik- und Computerfachbücher, aber auch Software für die verschiedensten Anwendungsgebiete warten auf Sie:

Vom allgemeinen Einstieg in die EDV über Büroanwendungen, Programmiersprachen, künstliche Intelligenz bis hin zur esoterischen Reihe. (Und selbstverständlich für alle gängigen Rechnerarten.)

Vom Akustik-Werkbuch über Funktionsgenerator- und Operationsverstärker-Schaltungen bis zu Computerperipherie-Bauanleitungen.

Lassen Sie sich unseren kostenlosen Katalog kommen. Heute noch. Postkarte genügt.

Das Programm



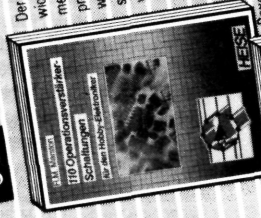
HEISE

Verlag Heinz Heise GmbH, Abt. TEBUS
Postfach 610407, 3000 Hannover 61

Verlag HEISE GmbH
Heinz Bissendorfer Straße 8
3000 Hannover 61

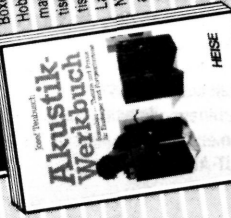
Der Operationsverstärker ist eines der wichtigsten elektronischen Bauelemente. In diesem Buch werden erprobte Schaltungen aus einem weiten Anwendungsspektrum vorgestellt. Alle Schaltungen sind bewußt einfach gehalten und bereiten auch dem Anfänger kaum Nachbaurprobleme. Ein Buch für die Praxis.

Best. Nr. 0504-9
DM 16,80



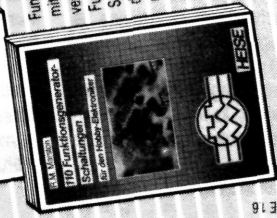
Boxen-Selbstbau – ein faszinierendes Hobby. Von einem erfahrenen Fachmann werden hier sowohl theoretische Grundlagen als auch praktische Tips für den Selbstbau von Lautsprecher-Boxen vermittelt. Neben zahlreichen Tabellen sind auch ausgefallene Konstruktionsvorschläge für unterschiedliche Boxentypen enthalten.

Best. Nr. 0530-8
DM 29,80



Funktionsgeneratoren – bestückt mit Transistoren, Operationsverstärkern, Digital-ICs und Alle Funktionsgenerator-ICs. Alle Schaltungen wurden sorgfältig dimensioniert, aufgebaut und getestet.

Best. Nr. 0503-0
DM 16,80



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

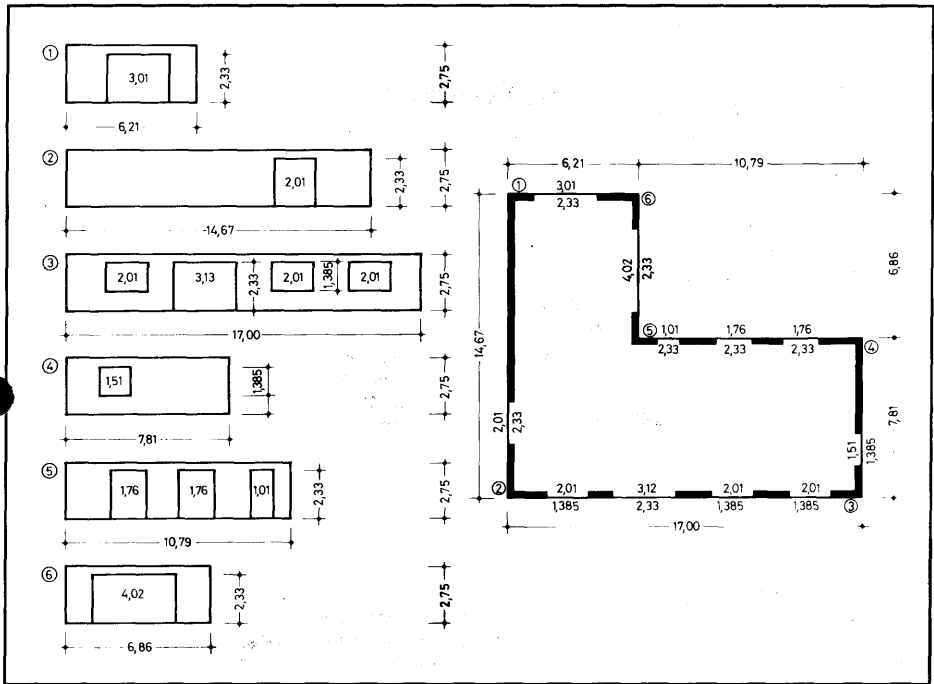


Bild 1 : Dieser Grundriss und die dazugehörigen einzelnen Seitenansichten werden im Beispiel verwendet.

wände in unserem Fall eine Luftschicht enthalten, geben Sie die Nummer 2 ein. Als nächstes wird gefragt, aus wieviel Schichten die Wand besteht. In diesem Beispiel sind es fünf.

Jetzt kommt die Materialliste. Sie besteht aus mehreren Seiten, die über die Tasten '+' und '-' (vor- und zurückblättern) zu erreichen sind. Betrachten Sie sich die einzelnen Schichten der Wand von innen nach außen. Die erste Schicht besteht aus einem zwei Zentimeter dicken Kalkzementputz. Geben Sie also die Nummer 26 und anschließend die Schichtstärke (2 cm) ein. Bei den nächsten

vier Materialien (das sind die Nummer 17 mit 11.5 cm, die Nummer 33 mit 4 cm, wieder die Nummer 17 mit 11.5 cm und wieder die Nummer 26 mit 2 cm) verfahren Sie genauso.

Der nächste Punkt ist relativ wichtig, denn jetzt wird gefragt, ob dieser gleiche Aufbau bei anderen Wänden noch mal vorkommt. Das bejahen Sie, weil die anderen fünf Wände aus den gleichen Materialien bestehen.

Als nächstes werden die Maße der Wand eingegeben (6.21 und 2.75), die Innentemperatur mit 20 Grad Celsius und eine durchschnittliche Außentemperatur von -10 Grad Celsius. Besitzt dieses Bauteil Öffnungen wie zum Beispiel Türen oder Fenster? Ja, eine Öffnung. Sie können wählen zwischen Fenster (1-3) und Tür (4). Da in diesem Beispiel Türen und Fenster isolierverglast sind, wählen Sie die

Nummer 2. Danach werden die Maße der Öffnung eingegeben (3.01 und 2.33). Weitere gleichartige Fenster sind in diesem Bauteil nicht vorhanden. Deshalb beantworten Sie die nächsten beiden Fragen mit 'N' für nein.

Gleicher Aufbau . . .

Bei der Frage zum Bauteil 1, ob dieser Aufbau erneut zum Tragen kommt, hatten Sie mit 'J' geantwortet. Daher muß jetzt die nächste Position beziehungsweise das nächste Bauteil angegeben werden (Positionsnummer, Positionsname und so weiter). Nur der Aufbau dieser Wand braucht jetzt nicht mehr eingegeben zu werden. Dagegen wird erneut nach den Abmessungen dieses Bauteils und nach den Temperaturverhältnissen gefragt.

Zurück zum Klartext

Interaktiver 6502-Re-Assembler

Ein Re-Assembler übersetzt Maschinencode in Assembler-Quelltext. Maschinenprogramme haben in der Regel die Eigenschaft, daß sie an eine bestimmte Lage im Speicher gebunden und (meist) sehr unübersichtlich sind. Wegen der festgelegten Speicheradressen ist es kaum möglich, nachträglich noch Routinen in ein schon bestehendes Maschinenprogramm einzufügen oder aus diesem herauszukürzen.

Bei einem Assembler-Quelltext bestehen dagegen diese Probleme nicht. Anstelle von festgelegten Adressen werden Symbole (sogenannte Labels) benutzt, deren Adressen vom Assembler aus den Befehlsängen berechnet werden, und die deshalb nicht an eine bestimmte Speicherlage gebunden sind. Die einzige absolute Adresse, die ein Assembler benötigt, ist die Startadresse am Anfang des Listings. Diese kann beliebig umgeändert werden. So kann durch Änderung dieser einzigen Zahl ein Programm in einem anderen Speicherbereich lauffähig gemacht werden; Programmänderungen können an beliebiger Stelle eingefügt werden.

In Ausgabe 9/85 haben wir erstmals einen 6502-Re-Assembler veröffentlicht. Dieser erzeugte Source-Code für den PROF-ASS (Data-Becker); nach dem Erscheinen von INPUT-ASS (INPUT 6/86) wurde der Wunsch nach einem passenden Re-Assembler in der Leserschaft laut. Außerdem ist die Arbeitsweise von IRAs grundsätzlich anders: IRAs ist ein interaktiver Re-Assembler, das heißt, er fragt bestimmte benötigte Informationen im Dialog mit dem Benutzer ab. Der im September '85 veröffent-

Nur im Objectcode vorliegende Programme an eigene Bedürfnisse anzupassen, kann man eigentlich nur Leuten zumuten, die Vater und Mutter erschlagen haben. Der interaktive Re-Assembler „IRAs“ erstellt aus 6502/6510-Maschinencode editier- und kommentierbare Quelltexte für den INPUT-ASS.

lichte Re-Assembler arbeitet sozusagen „vollautomatisch“. Es hat sich aber gezeigt, daß oft der Mensch in der Analyse von Programmcodes der Maschine einiges voraus hat.

IRAs wandelt den Maschinencode in ein Format um, das vom Assembler verarbeitet werden kann. Der Quelltext, den IRAs (IRAs steht für „Interaktiver Re-Assembler“) erzeugt, ist auf den in INPUT64 6/86 veröffentlichten Macro-Assembler INPUT-ASS zugeschnitten. Das heißt, der Text wird im Commodore-ASCII-Format abgelegt, und die Assembler-Anweisungen sind INPUT-ASS-spezifisch. Für andere Assembler muß eventuell das Textformat gewandelt und/oder die Pseudo-Ops ausgetauscht werden.

Ein Probelauf

Um Sie in das Arbeiten mit IRAs einzuführen, haben wir in das Modul, aus dem Sie den Re-Assembler selbst auf Ihren eigenen Datenträger abspeichern können, noch ein kleines Tool gepackt.

Dieses wird nicht mit CTRL und s abgespeichert, sondern über ein gesondertes Menü innerhalb des Moduls. Es ist ein sehr nützliches kleines Hilfsprogramm, ein sogenannter „Bildschirmschoner“. Das heißt: wenn länger als 15 Sekunden keine Bildschirmausgabe und keine Tastatureingabe stattfindet, wird der Bildschirm ausgeschaltet. Jede normale Bildschirmausgabe oder ein beliebiger Tastendruck schaltet den Bildschirm wieder ein. Dadurch hält die Bildröhre länger (es kommt nicht mehr zu Einbrennschäden), oben-dreien wird der 64er schneller. Dieses Tool wird vom eigenen Datenträger mit LOAD"SCHONER",8,1 geladen und nach einem anschließenden NEW mit SYS 52992 gestartet. Ein Neustart (nach RESET oder RUN/STOP-Restore) ist ebenfalls durch SYS 52992 möglich, der Bildschirmschoner belegt die Adressen \$CF00 bis \$CFA4.

Das soll hier aber nur am Rande interessieren, es ging ja um den Umgang mit einem Re-Assembler. Sie laden also zunächst IRAs von Ihrem eigenen Datenträger und starten ihn mit RUN. Das Programm meldet sich mit dem Prompt (>) und erwartet Eingaben. Durch Eingabe des Dollar-Zeichens (\$) und RETURN wird das Inhaltsverzeichnis der Diskette ausgegeben. Ein Objekt-Programm wird mit F (und anschließendem RETURN, was hiermit zum letzten Mal ausdrücklich erwähnt wird) in den Speicher geladen. Beantworten Sie die Frage nach dem File-Namen mit dem Namen BILDSCHIRMSCHONER. Die Geräteadresse ist 08, IRAs unterstützt keine Kassetten-Operationen. Das Programm wird nicht an seine Startadresse, sondern an den Anfang des Arbeits-

speichers des Re-Assemblers gelesen, um Bereichsüberschneidungen zu vermeiden.

Nach dem Laden eines zu re-assemblierenden Programms ist zwingend der O-Befehl vorgegeben. Dadurch wird der Offset zwischen der momentanen Lage im Speicher und der tatsächlichen Anfangsadresse berechnet.

IRAs gibt aus:
 Speicherlage: 1DC7
 Offset-Adresse: CF00

Alle Ein- und Ausgaben sind natürlich hexadezimal. „Speicherlage“ ist die Adresse, an der sich der Bildschirm-schoner tatsächlich befindet (für uns relativ uninteressant), „Offset-Adresse“ ist die beim Laden von Diskette ermittelte Startadresse des Programms. Wir bestätigen beide Angaben mit RETURN.

Ein probeweises D (wie Disassembling) gibt Anfangs- und Endadresse des Programms vor, nach Bestätigung mit RETURN erscheint außer dem JMP-Befehl bei Adresse \$CF00 wenig Sinnvolles, das B vor den einzelnen Hexzahlen steht für die Assembler-Anweisung „Byte“ und sagt aus, daß sich an dieser Stelle kein Programmtext befindet. Ausführbarer Code ist erst wieder ab der Zieladresse des JMP-Befehls zu sehen (\$CF26).

Versuchen wir's mit einem Memory-Dump. Das Disassembling wird mit RUN/STOP abgebrochen, nach der Eingabe von m ändern Sie die Startadresse (vor dem Komma) in CF03, die Endadresse in CF25 (einfach überschreiben). Schon besser: zu sehen ist rechts neben den Hex-Zahlen die Einschaltmeldung mit einführendem Cleardown (\$93), abschließendem RETURN (\$0D) und einer Null als Enderkennung. So kann über den TI-Befehl der Re-Assembler wieder ein bißchen „schlauer“ gemacht werden.

```

org $cf00
jmp lcf26
b $93
b "bildschirm-schone"
b "r initialisiert!"
b $0d
b $00
:lcf26 lda #$03
ldy $scf
jsr cprint
sei
lda #$72
ldy $scf
sta $0326
sty $0327
lda #$5e
ldy $scf
sta $028f
sty $0290
lda #$51
ldy $scf
sta $0314
sty $0315
jsr lcf95
cli
rts
lda $a1
cmp lcf26
bne lcf5b
jsr lcf88
:lcf5b jmp $ea31
jsr lcf95
lda vic17
and #$10
bne lcf6f
jsr lcf7b
lda $cb
sta $c5
:lcf6f jmp $eb48
jsr lcf7b
jsr lcf95
jmp $f1ca
:lcf7b pha
php
lda vic17
ora #$10
sta vic17
plp
pla
rts
:lcf88 pha
php
lda vic17
and #$ef
sta vic17
plp
pla
rts
:lcf95 pha
php
lda $a1
clc
adc #$05
sta lcf26
plp
pla
rts
:lcf26 b $00,$65
:cprint = $able
:vic17 = $d011

```

Dieses Source-Listing hat der Re-Assembler erzeugt. Um das Programm in jedem Adreßbereich lauffähig zu machen, ist noch etwas „Handarbeit“ fällig.

```

org $9000
jmp lcf26
:text b $93
b "bildschirm-schone"
b "r initialisiert!"
b $0d
b $00
:lcf26 lda *<text
ldy *>text
jsr cprint
sei
lda *<newout
ldy *>newout
sta $0326
sty $0327
lda *<newtas
ldy *>newtas
sta $028f
sty $0290
lda *<newirq
ldy *>newirq
sta $0314
sty $0315
jsr lcf95
cli
rts
:lcf95 lda $a1
cmp lcf26
bne lcf5b
jsr lcf88
:lcf5b jmp $ea31
jsr lcf95
:lcf26 lda vic17
and #$10
bne lcf6f
jsr lcf7b
lda $cb
sta $c5
:lcf6f jmp $eb48
jsr lcf7b
jsr lcf95
jmp $f1ca
:lcf7b pha
php
lda vic17
ora #$10
sta vic17
plp
pla
rts
:lcf88 pha
php
lda vic17
and #$ef
sta vic17
plp
pla
rts
:lcf95 pha
php
lda $a1
clc
adc #$05
sta lcf26
plp
pla
rts
:lcf26 b $00,$65
:cprint = $able
:vic17 = $d011

```

Der endgültige Assembler-Source-Code. Alle Bezüge in das Programm sind symbolisch adressiert.

Über den Befehl TI (Tabellen-Input) wird eingegeben, wo sich im zu re-assemblierenden Programm Datentabellen, Programmcode und Texte befinden. (Mit Datentabellen sind die Teile eines Programms gemeint, die weder ausführbaren Maschinencode noch darstellbaren Text enthalten.) Die Abfrage nach „Tabellen löschen?“ sollte vor der ersten Eingabe mit j (ja) beantwortet werden; es werden dadurch natürlich nicht die erwähnten Datentabellen selbst gelöscht, sondern die Tabelle mit den Informationen, wo überall Datentabellen zu finden sind. Die Geräte-Nummer 00 ist die Tastatur — stimmt in diesem Fall. Als nächstes geht es um Datentabellen, bestätigen Sie den Stern! Das heißt soviel wie „es werden keine neue Datentabellen definiert“, schließlich soll ja mitgeteilt werden, wo sich Texte im Objectcode befinden. Be-

antworten Sie die Frage „Texte von“ mit CF03, „bis“ mit CF25. Durch Bestätigung des vorgegebenen Sterns mit RETURN wird der TI-Befehl beendet.

Weiter geht's mit D, diesmal wird die Anfangsadresse (vor dem Komma) durch Überschreiben in CF26 geändert. Bei der Durchsicht des Programms mit der Leertaste wird bis zum Programmende nur noch ausführbarer Code ausgegeben. Anlaß zum Nachdenken geben nur die letzten beiden Befehle (BRK und ADC \$00). Dies sind zwar gültige Mnemonics, aber es sieht nicht nach einem sinnvollen Programm aus. Versuchen wir's mit einem Memory-Dump von \$CFA2 bis \$CFA4. Neben dem Hex-Dump ist den genannten Adressen kein Text zugeordnet, also handelt es sich um Datentabellen. Auch dies wird IRAs über TI mitgeteilt.

Der R-Befehl erzeugt ein Re-Assembling. Anfangs- und Endadresse werden übernommen, ebenso die Tabellierungsbreite 8, als Ausgabegerät ist zum Testen der Bildschirm sinnvoll (Geräte-Nummer 03). Die Ausgabe des re-assemblierten Codes kann mit der Leertaste angehalten werden, ein weiterer Druck auf die Leertaste setzt die Ausgabe fort. Bei genauerem Hinsehen können Sie feststellen, daß alle internen Sprünge „gelabelt“ sind. Es heißt zum Beispiel nicht mehr „JMP \$CF26“, sondern „JMP LCF26“, und vor dem Ziel dieses Sprungbefehls steht das Label LCF26.

Unschön sind noch die Bezüge auf Adressen außerhalb des Programms, beispielsweise JSR \$AB1E oder LDA \$D011. Viel aussagekräftiger wären statt dessen JSR CPRINT und LDA

Die IRAs Befehle

Der Editor des Re-Assemblers ist dem BASIC-Editor ähnlich. Er dient nur dazu, die einzelnen Befehle aufzurufen. Alle Tabellen können vom INPUT-ASS aus editiert werden.

R anfang, ende bedeutet Re-Assemblierung. Angaben müssen über die Start- und Endadresse gemacht werden. Werden keine Angaben gemacht, werden sinnvolle Adressen vorgegeben, falls sich ein zu re-assemblierendes Programm im Speicher befindet. Nach einem weiteren RETURN beginnt die Re-Assemblierung mit der Abfrage der Tabellierungsbreite. Diese bestimmt die Zahl der Leerzeichen vor den Assemblerbefehlen und kann Werte von Null bis 16 annehmen. Die Angabe ist dezimal.

Die Nummer des Ausgabegerätes wird hexadezimal erwartet. Erfolgt

die Ausgabe auf dem Bildschirm, kann das Listing durch Drücken der Space-Taste angehalten werden. Wird bei der Re-Assemblierung der Text länger als 100 Blöcke, was keine Seltenheit ist, wird ein Include-File eröffnet. Bei sehr langen Programmen kann die Re-Assemblierung zehn Minuten und länger dauern, wobei Assemblertexte von mehreren 100 Blöcken Länge entstehen. Die Länge des erzeugten Textes hängt entscheidend von der Tabellierungsbreite ab!

D anfang,ende bedeutet Disassemblierung auf den Bildschirm.

M anfang, ende steht für Memory-Dump. Diese Funktion und der Disassembler können beispielsweise zur Feststellung von Datentabellen in geladenen Programmen verwendet werden.

F steht für „File laden“ und hat die Funktion, ein zu re-assemblierendes Maschinenprogramm in den

Arbeitsspeicher zu lesen. Das Programm wird nicht an seine Startadresse, sondern an den Speicheranfang geladen, um Bereichsüberschneidungen zu vermeiden.

O simuliert die richtige Speicherlage für die Bearbeitung. Der Befehl sollte nach jedem F-Befehl aufgerufen werden. Als Speicherlage wird die momentane Lage im Speicher nach dem Laden eingegeben, die Offset-Adresse ist die Adresse, an der sich das Programm normalerweise befindet (also die absolute Ladeadresse). Beide Adressen werden vorgegeben; eine Änderung kann sinnvoll sein, wenn sich im bearbeiteten Programm der Programmzähler verändert, etwa in verschobenen Programmteilen.

C sendet einen DOS-Befehl an die Floppy, nur RETURN in der Befehlszeile bedeutet Anzeigen des Disk-Status.

VIC17. „CPRINT“ ist der Kurzname für die im C64-ROM installierte Routine zur Textausgabe, VIC ist der Video-Chip, dessen Basis-Adresse bekanntlich bei \$D000 liegt; somit ist \$D011 folgerichtig VIC17. Dem Re-Assembler wird dies beigebracht durch den LI-Befehl (Label-Input). Nach den üblichen Abfragen (Gerät und so weiter) wird dort eingegeben:

Label bei Adresse AB1E
Name des Labels: cprint

und entsprechend wird der Adresse \$D011 der Name „VIC17“ zugeordnet.

Ehe jetzt der re-assemblierte Quelltext auf Diskette abgelegt wird, sollten die anderen Dateien gesichert werden.

Das TO-Kommando (Tabellen-Output) erlaubt, die über TI eingegebenen Werte auszugeben; auf den Bildschirm, den Drucker oder auf Diskette (Gerätenummer 08). Mit dem LO-Befehl (Label-Output) können dementsprechend die über LI eingegebenen Labels ausgegeben beziehungsweise abgespeichert werden. Diese Dateien lassen sich mit dem LI- oder dem TI-Befehl wieder von Diskette einlesen. Dies ist vor allem bei der Bearbeitung längerer Programme sinnvoll, die nicht in einer Sitzung analysiert werden können. Die Dateien werden im Format des INPUT-ASS abgelegt, können also mit dessen Editor verändert werden. Es hat sich als sinnvoll erwiesen, mit dem INPUT-ASS eine Label-Datei mit den wichtigsten System-Routinen zu erstellen, die zur Re-Assemblierung eines Programms immer wieder eingelesen werden kann.

Zum Schluß unserer Beispielsitzung lassen wir das Re-Assembling auf Diskette ablegen. Nach dem R-Befehl werden Anfangs- und Endadresse mit RE-TURN bestätigt, ebenso die Tabellarische Breite, als Gerätenummer wird 08 angegeben, und dann müssen Sie sich noch einen Namen für das Produkt ausdenken. Verlassen wird der Re-Assembler mit Q wie Quit. Wenn Sie sich jetzt den erzeugten Quelltext in Ihren Editor laden, ist noch ein Arbeitsschritt notwendig, um ein frei verschiebbares Programm zu erhalten. Es gibt nämlich Bezüge im Programm, die in der Form LDA # <Adresse, LDY # > Adresse vorhanden sind, zum Beispiel steht vier Zeilen nach dem Label L085A:

LDA # \$37
LDY # \$08
JSR CPRINT

\$ zeigt das Inhaltsverzeichnis der Diskette auf dem Bildschirm an.

Q verläßt den Re-Assembler und springt zum BASIC. Ein Neustart mit RUN ist möglich, vorhandene Informationen bleiben dabei erhalten.

LI dient zur Eingabe von internen Labels. Dies kann über Tastatur geschehen (Gerätenummer 00) oder von Diskette (Gerätenummer 08). Die Label-Dateien haben das Format des INPUT-ASS, so daß auch vorhandene Source-Texte dieses Assemblers geladen werden können, aus denen dann alle externen beziehungsweise durch das Gleichheitszeichen zugewiesenen Labels entnommen werden. Diese Symbol-Zuweisungen werden an das Ende des erzeugten Quelltextes geschrieben. Falls diese Symbole für weitere Berechnungen verwendet werden sollen, ist es sinnvoll, diese per Blockverschie-

bung im INPUT-ASS-Editor an den Anfang zu setzen.

LO heißt Ausgabe der Label-Datei. Dies kann auf den Bildschirm oder ein Peripherie-Gerät geschehen. Die Tabelle kann mit dem Editor des INPUT-ASS bearbeitet werden.

TI dient zur Eingabe der Darstellungsmodi der einzelnen Programmteile. Auch diese Tabellen können im Editor des INPUT-ASS geändert werden. Beim Einlesen von Diskette kommt es unter manchen DOS-Erweiterungen nach dem Laden zur Meldung „E/A-Fehler“ (Ein-/Ausgabe-Fehler). Diese Meldung kann getrost ignoriert werden.

Unter „Datentabellen“ können Adreßbereiche angegeben werden, die weder Programmcode noch darstellbaren Text enthalten. „Einzelne Bytes“ sollten auch dann eingegeben werden, wenn an einer

Stelle im Programm der BIT-Befehl zum Überspringen eines anderen Befehls benutzt wird. Beispiel:

1000 LDA # \$10
1002 BIT \$00A9

sollte im Assemblertext folgendermaßen aussehen:

LDA # \$10
B \$2C
LDA # \$00

In diesem Fall muß als Adresse 1002 angegeben werden.

Bereiche, die unter „Texte“ angegeben werden, erscheinen im Assemblertext als ASCII-Texte.

TO gibt die Bereichstabellen auf dem Bildschirm oder einem Ausgabegerät aus.

RUN/STOP bricht einen laufenden Befehl ab und führt zurück zur Kommando-Ebene.

Setzt man Low- und High-Byte zusammen, ergibt dies die Adresse \$0837, und dies war im Programm exakt der Beginn der Einschaltmeldung! Die Routine CPRINT verlangt als Parameter das Low-Byte der Adresse des auszugebenden Textes im Akku, das High-Byte im Y-Register. Das wiederum kann der Re-Assembler nicht wissen. Er würde zwar LDA \$0837 erkennen, aber nicht diebesagte Aufteilung dieser internen Adresse. Deswegen muß der erzeugte Text noch einmal „zu Fuß“ durchgegangen werden. Im beschriebenen Beispiel heißt das: ein Label vor das B \$93 schreiben, etwa TEXT. (Das hätte man übrigens auch schon mit dem Re-Assembler machen können.) Die beiden Lade-Befehle müssen dann heißen:

```
LDA #<TEXT  
LDY #>TEXT
```

Entsprechendes gilt für das Beschreiben der drei Vektoren im Programmteil kurz dahinter.

Interpretations-Probleme

Weitere Erläuterungen zu den Befehlen von IRAs finden Sie im Kasten „Die IRAs-Befehle“. Zum gekonnten Umgang mit einem Re-Assembler gehört allerdings auch ein gewisses Verständnis der Funktionsweise solch eines Programms:

Da ein Maschinenprogramm aus einer endlosen Kette von Zahlenkombinationen besteht, die unterschiedlich interpretiert werden können (als Maschinenprogramm, als Datentabellen, als ASCII-Text), ist es für den Re-Assembler zunächst einmal wichtig zu wissen, in welcher Form der gerade zu bearbeitende Programmteil überhaupt dargestellt werden soll. (Die TI-Eingaben des Benutzers.)

Anhand dieser Angaben „weiß“ der Re-Assembler nun also, an welchen Stellen reiner Programmcode liegt. Dieser

wird in einem ersten Durchlauf auf Adressen untersucht, die auf eine Stelle im Programm zeigen (sogenannte interne Labels). Diese Adressen sollen später als Symbole dargestellt werden (der Sinn des Re-Assemblers liegt schließlich darin, alle absoluten Adressen durch Variablen zu ersetzen).

Ist die Endadresse des Programms erreicht, folgt der eigentliche Hauptteil der Re-Assemblierung, in dem der Assemblertext erzeugt wird.

Da der Assembler die jeweiligen Speicheradressen aus den Befehlsängen berechnet, ist eine Adreßangabe nicht mehr am Anfang jeder Zeile notwendig. Adressen werden lediglich am Ziel von Sprung- oder Ladebefehlen notwendig. Steht die gerade zu bearbeitenden Zeile an einer solchen Zieladresse, wird dort ein Symbol eingesetzt.

Wurde diesem Symbol kein bestimmter Name zugewiesen, wird dafür eine Variable in der Form Lxxxx (xxxx steht für eine beliebige Hex-Adresse) eingesetzt. Nun kann es aber vorkommen, daß ein Sprungbefehl diese Stelle adressiert, der erst an späterer Stelle im Programm steht. Um auch dies zu berücksichtigen, war der erste Durchlauf notwendig, der weiter nichts tat, als solche Adressen zu speichern.

Nachdem der Zeilenanfang geschrieben wurde, muß geprüft werden, in welcher Weise der nun folgende Programmcode dargestellt werden soll. Dies geschieht anhand der Festlegungen, an welchen Stellen sich Datentabellen oder Texte befinden.

Handelt es sich dabei um Daten oder Texte, werden die entsprechenden Bytes mit vorangestellten Pseudo-Opco-des in den Assemblertext geschrieben. Wurde ein Programmteil als Maschinencode definiert, wird das zu dem jeweiligen Befehl gehörende 6502-Mnemonic geschrieben, gegebenenfalls gefolgt von einem Argument. Handelt

es sich dabei um eine Adresse, die in der Symboldatei gefunden wurde, so wird das entsprechende Label eingesetzt. Zusätzlich zu den internen Labels, die im ersten Durchlauf registriert wurden, werden externe Labels, die nicht im Programm liegen müssen und vom Benutzer eingegeben werden können, berücksichtigt; beispielsweise Adressen von Betriebssystemroutinen.

Diese externen Symbole erhalten ihre jeweiligen Adressen aus einer Zuweisungstabelle am Ende des Assemblerlistings.

Während der Re-Assemblierung nach diesem Grundprinzip abläuft, müssen während jeder Zeile noch einige Sonderfälle bearbeitet werden.

So kommt es zum Beispiel vor, daß die Adresse eines Sprung- oder Ladebefehls auf das zweite oder dritte Byte eines drei Byte langen Befehls zeigt. Da Symbole für Zieladressen jedoch nur am Anfang einer Zeile geschrieben werden können, würde dies zu einem Fehler führen.

Es würde auch nicht genügen, die Zieladresse absolut anzugeben, weil dann das Assemblerlisting wieder an einen bestimmten Speicherbereich gebunden wäre, und das soll ja gerade vermieden werden. In solchen Fällen wird die Zieladresse durch einen Rechenbefehl bestimmt, zum Beispiel:

```
LABELn = LABELx + 2.
```

Beim Testen von IRAs wurde versuchsweise das 64er Betriebssystem re-assembliert. Der erzeugte Source-Code war auf Anhieb ohne Nacharbeiten lauffähig. Aber: Die Qualität des erzeugten Textes hängt sehr stark von dem Umfang der eingegebenen Label-Tabelle und der Unterscheidung von Daten-, Text- und Programmbereichen ab!

Martin Friedl/JS

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!

Die entscheidenden Sprünge

Zweite Folge der Assembler-Schule

Bevor wir aber zu neuen Programmen kommen, wollen wir einen kurzen Blick auf die Art und Weise werfen, wie Maschinen-Programme im Speicher abgelegt und vom Prozessor verarbeitet werden.

Ihnen ist sicherlich beim Experimentieren schon aufgefallen, daß die Adressen, an denen die einzelnen Befehle liegen, ziemlich unregelmäßig durchnumeriert werden. Das liegt daran, daß die Befehle unterschiedlich viel Platz im Speicher benötigen. Die 6502-CPU kennt Befehle, die ein, zwei oder drei Bytes belegen. Diese Instruktionslänge ist abhängig von der Adressierungsart. So sind die Befehle mit implizierter Adressierung (Das sind die ohne Argument.) nur ein Byte lang, bei unmittelbarer Adressierung (gekennzeichnet durch ein Doppelkreuz) werden zwei und bei absoluter drei Bytes belegt.

Programme sind auch nur Daten

Der eigentliche Befehl steckt dabei immer im ersten Byte. In ihm ist auch die Instruktionslänge verschlüsselt. Die beiden Versionen des LDA-Befehles, die Sie bislang kennen, haben zum Beispiel auch unterschiedliche Instruktions-Bytes. Bei der Immediate-Adressierung ist es \$A9 und bei absoluter Adressierung \$AD. Dadurch erkennt der Prozessor die unterschiedliche

Nachdem in der ersten Folge einige grundlegende Maschinsprache-Befehle gezeigt wurden, befassen wir uns in dieser Folge mit weiteren Arithmetik-Operationen. Neben neuen Befehlen werden Sie die Bits des Statusregisters im einzelnen kennenlernen sowie die Index-Register X und Y benutzen. Wenn Sie diese Folge durchgearbeitet haben, werden Sie außerdem wissen, wie bedingte Programm-Sprünge in Assembler kodiert werden.

Adressierungsart und damit auch die Instruktionslänge.

Wenn der Prozessor beginnt, ein Programm abzuarbeiten, liest er das erste Byte. Bei Bedarf werden dann noch ein oder zwei Bytes gelesen, der Befehl ausgeführt, und das Spiel wird mit dem ersten Byte des nächsten Befehls fortgeführt.

Normalerweise wird also ein Byte nach dem anderen gelesen und die entsprechende Aktion durchgeführt. Sie werden in dieser Folge aber noch Kommandos kennenlernen, mit denen man diese Reihenfolge beeinflussen kann. Eines kennen Sie schon, nämlich BRK. Trifft der Prozessor auf diesen Befehl, macht er an einer speziellen Adresse

weiter. Diese Adresse ist innerhalb des ROMs. Das Programm, das dort steht, löscht normalerweise den Bildschirm gibt das Wort READY aus und wartet auf eine Eingabe.

Aus dem bisher gesagten erkennt man (hoffentlich), daß ein Maschinenprogramm im Prinzip nichts anderes ist als eine Folge von Bytes, die hintereinander im Speicher liegen. Mit dem Ende eines Programmes wird aber nicht das Ende des Speichers erreicht. Wenn ein

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 Hauptmenue gelangen.

Wenn Sie nun F1 druecken, gelangen Sie in ein Menue, 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 Hauptmenue gelangen Sie jederzeit mit der STOP-Taste zurück.

Mit F3 gelangen Sie aus dem Hauptmenue zu einer Auswahl verschiedener Beispielprogramme.

Programm also einfach endet, ohne mit einem BRK oder einem anderen geeigneten Befehl (Sie werden noch verschiedene kennenlernen) abgeschlossen zu werden, versucht der Prozessor, die folgenden Bytes auch als Programm zu interpretieren und auszuführen. Dabei entstehen die verrücktesten Effekte, meistens endet so etwas mit einem 'Absturz' des Rechners.

Innerhalb unseres Simulators kann nichts passieren, Sie erhalten eine entsprechende Fehlermeldung. Wenn Sie aber ein Programm 'im richtigen Leben' laufen lassen, das in Simulator mit der Meldung „Programmende“ oder „Opcode nicht ausführbar“ abbricht, sollten Sie sich über nichts wundern. Kaputtgehen kann der C64 dabei nicht, aber Sie werden wahrscheinlich einen Reset-Taster oder gar den Netzschalter betätigen müssen.

Genauso, wie man Bytes im Speicher entweder als Programm oder als Daten auffassen kann, gibt es für Datenbytes meistens verschiedene Interpretationsmöglichkeiten.

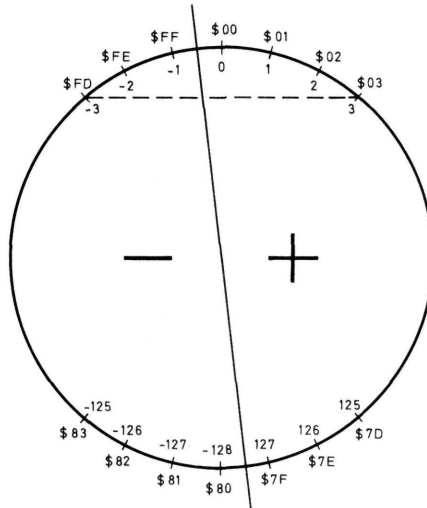


Bild 1: Im Zahlenkreis wird das Prinzip des Zweierkomplementes deutlich: Die komplementären Zahlen liegen einander jeweils gegenüber. Beim Überschreiten der Grenze am unteren Ende liegt ein Überlauf vor.

Weniger als nichts

In der ersten Folge haben wir unter anderem gelernt, wie Ein- und Zwei-Byte-Werte in Maschinensprache subtrahiert werden. Dabei diente die Carry-Flagge des Prozessor-Status-Registers als Anzeige für einen Übertrag in die nächsthöhere Stelle beziehungsweise für einen Fehler. Wenn bei der Subtraktion eine negative Zahl herauskam, sind wir davon ausgegangen, daß ein Fehler vorliegt. Mit Hilfe des sogenannten Zweierkomplementes ist es nun möglich, auch negative Ergebnisse sinnvoll zu interpretieren. Dabei werden alle Bytes mit gesetztem höchstwertigem Bit als negative Zahlen aufgetaßt. Das genaue Umrechnungsverfahren ist im Programm beschrieben.

Je nachdem, wie die Ergebnisse einer Rechnung weiterverarbeitet werden, kann man also die Zahlen \$80 bis \$FF einfach umrechnen und ihnen die Werte 128 bis 255 zuordnen oder mit Hilfe des Zweierkomplementes als die negativen Zahlen -128 bis -1 auffassen. Diese Tatsache wird durch den Zahlenkreis anschaulich verdeutlicht.

Bei letzterer Darstellungsweise kann es

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 Hauptmenue 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 Programmbeis-

spiele 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 Programmpaketes einmal gründlich zu lesen. Besitzer eines Druckers können sie auch mit CTRL-b zu Papier bringen.

Die INPUT 64-Assemblerschule ist eine Serie, die in der 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.

natürlich auch zu Über- beziehungsweise Unterschreitungen des gültigen Zahlenbereiches kommen. Addiert man beispielsweise die Zahlen 73 und 58, so ist das Ergebnis größer als 127, das höchstwertige Bit ist gesetzt, und daher wäre das Resultat als negative Zahl zu interpretieren. Zum Glück stellt der Prozessor auch für diesen Fall eine Möglichkeit der Bereichsüberprüfung zur Verfügung. In dem obigen Beispiel wäre nach der Ausführung der Addition die Overflow-Flagge im Statusregister gesetzt. Dieses Bit – auch V-Flag genannt – wird automatisch bei jeder Addition und Subtraktion mitversorgt, ähnlich wie die Carry-Flagge. Sein Inhalt hat aber auf eventuell folgende ADC- oder SBC-Befehle keinen Einfluß.

Hier die Regeln, nach denen der Prozessor die V-Flagge setzt: Die CPU geht immer davon aus, daß sie es mit vorzeichenbehafteten Zahlen zu tun hat. Die V-Flagge wird gesetzt, wenn zu einer positiven Zahl eine positive Zahl addiert oder von ihr eine negative Zahl subtrahiert wird und dabei das Ergebnis größer als 127, also negativ, ist. Im negativen Bereich ist es ebenso: Negative Zahl plus negative Zahl oder negative Zahl minus positive Zahl und positives Ergebnis ergibt gesetztes Overflow-Flag.

Das hört sich zwar sehr kompliziert an, ist es aber eigentlich gar nicht, denn es geschieht völlig automatisch. Wenn man mit vorzeichenbehafteten Zahlen rechnet, sollte man nach einem ADC- oder SBC-Befehl die V-Flagge prüfen. Sie können sich einfach darauf verlassen, daß das Ergebnis stimmt, wenn sie nicht gesetzt ist. Wie man ein Flag des Status-Registers innerhalb eines Programmes prüft, dazu kommen wir gleich.

Statusfragen

Zunächst wollen wir einen ausführlichen Blick auf die einzelnen Bits des Prozessor-Status-Registers – auch

kurz P-Register genannt – werfen. Es ist wie die meisten 6502-Register acht Bits breit. Jedoch kann man es nicht wie zum Beispiel den Akku direkt manipulieren. Vielmehr werden einzelne Bits bei der Programm-Ausführung automatisch beeinflußt. Wie wir schon gesehen haben, können einzelne Bits auch vom Programmierer direkt gesetzt oder zurückgesetzt werden wie die Carry-Flagge mit den Befehlen SEC und CLC.

Jedes Flag wird durch einen Buchstaben gekennzeichnet. Die Bezeichnungen lauten von links nach rechts NV-BDIZC

Die Bits 0 (C = Carry-Flag) und 6 (V, Overflow) hatten wir schon. Bit 5 hat keine Bedeutung und ist Hardware-bedingt immer gesetzt.

Mit den bisher bekannten Befehlen haben wir die Bits 1 und 7 auch schon beeinflußt, aufmerksame Beobachter haben es vielleicht beim Experimentieren bemerkt.

Bit 1 heißt Zero- oder kurz Z-Flagge. Der Prozessor setzt sie immer, wenn das Ergebnis eines Ladebefehles oder einer arithmetischen Operation gleich Null ist. Bei anderen Resultaten wird die Zero-Flag wieder gelöscht.

Bit 7, die N-Flagge, zeigt das Vorzeichen des Ergebnisses an. Sie übernimmt bei einem Ladebefehl oder einem ADC- oder SBC-Kommando das höchstwertige Bit des Ergebnisses. Sie ist also bei einem negativen Resultat gesetzt und bei einem positiven gelöscht. Daher auch ihr Name Negative-Flag.

Diese Flaggen werden auch von anderen Befehlen beeinflußt. Wenn diese an der Reihe sind, werden wir auf diese Nebenwirkungen eingehen.

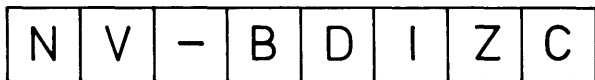


Bild 2: Im Prozessor-Status-Register hat jedes Bit eine besondere Bedeutung.

Zukunftsmusik

Bit 3 des Statusregisters ist ein Prozessor-interner Schalter. Mit speziellen Befehlen kann der Programmierer dieses Flag setzen oder löschen und den Prozessor dadurch in den sogenannten Dezimal-Modus schalten. Die Auswirkungen dieses Schalters werden wir in der nächsten Folge ausführlich besprechen.

Mit den Bits 2 und 4 kann man externe Programm-Unterbrechungen (Interrupts) kontrollieren. Die I-Flagge dient dabei als Schalter und teilt dem Prozessor mit, ob solche Interrupts zugelassen sind. Mit der B-Flagge (Break-Flag) kann man feststellen, woher diese Unterbrechung kam. Sie wird immer gesetzt, wenn der Prozessor auf einen BRK-Befehl trifft. Dann springt der Rechner nämlich zu der gleichen Adresse wie bei einem Interrupt. Diese ist – der Prozessor erwartet es so – in den Adressen \$FFF0 und \$FFFF des ROMs verzeichnet. Mehr zu diesen Flags in einer späteren Folge der Assembler-Schule.

Immer diese Entscheidungen

Jetzt wollen wir aber endlich wieder programmieren! Mit den ersten neuen Befehlen, die Sie kennenlernen sollen, lüften wir das Geheimnis, wie man einzelne Flags des Status-Registers prüfen kann. Das funktioniert bei den Bits Carry, Overflow, Zero und Negative. Das Zauberwort heißt Bedingter Sprung auf gut Englisch Branch. Die Branch-Befehle testen das entsprechende Status-Bit und führen gegebenenfalls einen Sprung aus. Ist die Bedingung nicht erfüllt, wird einfach beim nächsten Befehl

Opcode	Bedeutung	Bedingung
BPL	Branch on result PLUS	N-Flag = 0
BMI	Branch on result Minus	N-Flag = 1
BVC	Branch on oVerflow Clear	V-Flag = 0
BVS	Branch on oVerflow Set	V-Flag = 1
BCC	Branch on Carry Clear	C-Flag = 0
BCS	Branch on Carry Set	C-Flag = 1
BNE	Branch on result Not Equal	Z-Flag = 0
BEQ	Branch on result EQual	Z-Flag = 1

Tabelle 1: Mit bedingten Sprüngen trifft der Prozessor Entscheidungen.

weitergemacht. Alle Branch-Befehle des 6502 zeigt Tabelle 1.

Als Parameter verlangen diese Befehle natürlich noch das Sprungziel. In unserem Simulator und wenn Sie mit einem Assembler arbeiten, können Sie einfach die Adresse des Zieles oder noch besser ein Label, das an dieser Adresse steht, hinter den Branch-Befehl schreiben. Der Prozessor benutzt aber nicht diese Adresse sondern die Differenz zwischen dem nächsten Befehl und dem Sprungziel. Diese Umrechnung führt der Assembler für Sie durch.

Dieser berechnete Offset ist ein Ein-Byte-Wert in Zweierkomplement-Form. Bei einem Rücksprung ist er negativ, bei einem Sprung nach vorne positiv. Mit Branches kann man höchstens 128 Bytes rückwärts beziehungsweise 127 Bytes nach vorne springen. Diese Adressierungsart, die es ausschließlich bei den Verzweigungs-Befehlen gibt, heißt Relative Adressierung. Diese Befehle haben eine Instruktionlänge von zwei Bytes.

Fehlerfälle

Nach so viel Erklärungen sollte das erste Beispielprogramm für Sie keine Geheimnisse mehr bergen. Die ersten drei

Befehle sind altbekannt. Danach kommt eine Programmverzweigung. Wenn bei der Addition alles glattgegangen ist, wird der Befehl bei \$C00F, Label OKAY, angesprungen. Das Ergebnis der Addition wird dann in SUM2 gespeichert, und das Programm ist fertig.

Ist nach der Addition die V-Flagge gesetzt, wird das Ergebnis verworfen. Stattdessen wird \$FF nach Flag gespeichert. Dadurch kann auch nach dem Programmende noch ein Fehler festgestellt werden, zur Not durch PEEK(49173).

Sie sollten in diesem Programm mit den Werten für SUM1 und SUM2 experimentieren und auch den ADC-Befehl mal durch SBC ersetzen. Beobachten Sie dabei auch, was mit den Flaggen Carry, Zero und Negative passiert.

XY aufgelöst

Das zweite Programm bringt eine Reihe neuer Befehle. Gleich der erste ist noch unbekannt. Aber keine Angst, so neu ist er auch wieder nicht.

LDX

heißt Load X-register. Er funktioniert fast wie LDA, nur daß eben nicht der Akku sondern das Index-X-Register geladen wird. Für das Index-Y-Register gibt es diesen Befehl auch, er heißt

LDY

Beide Befehle gibt es in der Immediate- und der Absolute-Version, beide beein-

flussen — wie LDA — die Flaggen Negative und Zero.

Der nächste Befehl im Programm heißt

TXA

und ist die Abkürzung für Transfer X-register to Akku. Er gehört zu einer Vierergruppe von Befehlen, die Prozessor-intern den Datenverkehr zwischen dem Akku und den Index-Registern ermöglichen. Die anderen drei heißen

TYA (Transfer Y to Akku),

TAX (Transfer Akku to X) und

TAY (Transfer Akku to Y).r

Alle vier sind Ein-Byte-Befehle mit der Adressierungsart Implied, sie beeinflussen die N- und die Z-Flagge.

Den nächsten Befehl wollen wir erstmal übergehen und wenden uns dem Kürzel

INX

zu. Es bedeutet Increment X-register. Auch dieser Befehl hat drei Geschwister. Sie heißen

INY (Increment Y-register),

DEX (Decrement X-register) und

DEY (Decrement Y-register).

Diese vier Befehle sind unter anderem dafür verantwortlich, daß man die Index-Register X und Y so gerne benutzt. Sie ermöglichen es es nämlich, mit einem schnellen und kurzen Ein-Byte-Befehl den Inhalt des entsprechenden Registers um Eins zu erhöhen (increment) oder zu vermindern (decrement). Damit eignen sich die Index-Register hervorragend als Zähler für Programmierschleifen. Für den Akku gibt es solche Befehle (vielleicht INA oder DEA) leider nicht.

Diese vier Befehle setzen je nach Ergebnis die Zero- und die Negative-Flag. Alle vier gibt es nur als implizit adressierte Ein-Byte-Befehle.

Programmieren mit System

Der letzte unbekannt Befehl in diesem Programm gehört eigentlich noch gar

nicht in diese Lektion. Seine genaue Funktionsweise kommt in der nächsten Folge. Hier nur soviel:

JSR

heißt Jump to SubRoutine, zu Deutsch „Springe in ein Unterprogramm“. Er funktioniert ähnlich dem BASIC-Befehl GOSUB. Nach der Ausführung der Unteroutine wird das Programm beim nächsten Befehl fortgesetzt.

Das Kürzel PRCH ist ein innerhalb des Simulators vordefiniertes Label und soll an PPrint CHARACTER erinnern. Das Unterprogramm, zu dem diese Adresse gehört, gibt auf dem Bildschirm das Zeichen aus, dessen ASCII-Code im Akku des Prozessors steht. Im Betriebssystem-ROM des C64 gibt es eine solche Routine an der Adresse \$FFD2. Sie ist normalerweise unter dem Namen BSOUT bekannt.

Rücksprünge ergeben Schleifen

Sehen wir uns das Programm nun in Aktion an. Falls Sie es noch nicht getan haben, laden Sie es in den Editor und drücken F7, um es zu assemblieren. Der Befehl LDX # \$20 lädt den ASCII-Code des Leerzeichens — das kleinste druckbare Zeichen — ins X-Register. Der nächste Befehl kopiert diesen Wert in den Akku, wo die Routine PRCH ihn erwartet. Nach der Ausgabe des Zeichens (Leerzeichen ist nicht zu sehen) wird der Inhalt des X-Registers um Eins erhöht. \$21 ist kleiner als 127, also positiv. Darum ist die N-Flagge gleich Null, und der Rücksprung wird ausgeführt.

Jetzt wird die Schleife erneut durchlaufen und ein Ausrufezeichen ausgegeben. Wenn Sie es nicht sehen, liegt das wahrscheinlich daran, daß Sie im unteren Drittel des Bildschirms den Monitor eingeschaltet haben. Drücken Sie die HOME-Taste, und Sie sehen an der selben Stelle vier inverse Leerzeilen. Sie stellen in unserem Simulator den

Bildschirm dar, auf den die Routine PRCH zugreift.

Sie können nun die Pfeil-aufwärts-Taste drücken, um die Geschwindigkeit zu erhöhen. Nacheinander sollten auf dem 'Bildschirm' die Zeichen bis CHR\$(127) gedruckt werden.

Vergleichen Sie bitte

Der letzte neue Befehl dieser Lektion taucht im dritten Beispielprogramm auf und heißt

CMP

(CoMPare to accumulator — vergleiche mit Akku). Er führt eine Subtraktion aus wie der Befehl SBC mit vorher gesetzter Carry-Flagge. Das Ergebnis dieser Subtraktion wird jedoch nicht in den Akku übernommen. Der Befehl CMP dient lediglich dazu, die Flaggen N, Z und C im Status-Register entsprechend zu setzen.

Nach der Ausführung eines CMP-Befehles ist also die Z-Flagge gesetzt, wenn das Argument und der Inhalt des Akkumulators gleich waren. Wenn die Carry-Flagge gesetzt ist, bedeutet das, daß das Argument größer oder gleich dem Akku-Inhalt ist, gelöschte Carry heißt kleiner.

Der JSR-Befehl tritt in unserem dritten Beispiel mit der Adresse GETC auf. Auch dieses ist ein vordefiniertes Label. Die dazugehörige Betriebssystem-Routine verhält sich ähnlich wie der BASIC-Befehl GET: Bei der Rückkehr aus diesem Unterprogramm enthält der Akku den ASCII-Code der zuletzt gedrückten Taste. Wurde keine Taste gedrückt, so ist die Z-Flagge gesetzt, und im Akku steht eine Null.

Mit Filter

Das Programm fragt die Tastatur ab. Es bricht ab, wenn die RETURN-Taste (ASCII 13) gedrückt wird. Bei anderen Tasten wird überprüft, ob es sich um eine Ziffer handelte. Der Befehl CMP #'0

stellt keine neue Adressierungsart dar, sondern nutzt die Fähigkeit des Assemblers aus, Zeichen in deren ASCII-Wert umzurechnen. Bei der Programmausführung wird dieser Befehl zu CMP # \$30, denn \$30 (48) ist der ASCII-Code des Zeichens Null.

Die ASCII-Codes in Ihrem C64-Handbuch auf Seite 135ff haben übrigens auch in Maschinensprache Gültigkeit. Der Doppelpunkt ist demnach das nächsthöhere Zeichen nach der 9. Die Befehlsfolge CMP und BCS vergleicht ja auf 'größer oder gleich'. Würde an dieser Stelle CMP #'9 stehen, so würde das Programm die 9 abweisen.

Wenn das eingegebene Zeichen kleiner als 0 oder größer als 9 ist, springt das Programm wieder zur Eingabeschleife. Ziffern werden auf dem Bildschirm ausgegeben.

Sie sollten versuchen, dieses doch schon recht komplizierte Programm in allen Einzelheiten zu verstehen. Es zeigt, wie vielfältig die Branch-Befehle eingesetzt werden können. Dieses Programm dient auch als Grundlage zur Hausaufgabe (siehe Programm auf Ihrem Datenträger).

Wenn Sie irgendetwas nicht verstanden oder einen Fehler entdeckt haben, oder wenn Sie Verbesserungen zu dieser Serie vorschlagen möchten, rufen Sie bitte nicht in der Redaktion an, sondern schreiben Sie uns unter dem Stichwort „Assembler-Schule“. Dann erreichen Sie garantiert den zuständigen Mitarbeiter.

In der nächsten Lektion lernen Sie, wie Sie selbst Unterprogramme schreiben. Außerdem gibt es wieder Beispielprogramme zum Thema Arithmetik. HS

Literatur

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

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

Farbenfreude

Die Farbpalette des C64

Es stimmt tatsächlich: elf Farben, denn die restlichen fünf Abstufungen sind Schwarz, Weiß und drei Grautöne, die nicht zu den echten Farben zählen. Wer geschickt mit dieser Farbpalette umzugehen weiß, kann Erstaunliches auf dem C64 hervorzubringen. Dies geht sogar ohne komplizierte Multicolor- und/oder Grafik-Programmierung, wie der Zeichensatzgrafik-Wettbewerb (10/86) gezeigt hat. Innerhalb dieser 64er Tips zeigt Frank Börnke feinfühlig Abstufungen dieser Palette.

Kenner der Farblehre wenden sich gegen allzu Buntes mit dem Argument, daß es eigentlich nur drei Grundfarben gibt. Fast jeder dürfte schon einmal mit Malkasten oder Farbstiften in kreativer Minute Farbmischungen ausprobiert haben. Aus dem oft spärlichen Kunstunterricht an deutschen Schulen kennt man die Grundfarben Orangerot, Gelb und Blau, die zusammengemischt je nach Reinheit der Farbe ein Grauton ergeben oder sich im Idealfall sich zu Schwarz zusammenschließen. Dieses Mischverfahren nennt der Fachmann subtraktives Mischverfahren. Bei der Mischung von Farbstoffen mischen sich die Grundfarben zu Schwarz.

Anders hingegen beim Umgang mit farbigen Lichtern. Ein Farbfernseher oder -monitor erzeugt selbstleuchtende Farben. Hier ergänzen sich die Grundfarben Blaurot, Grün und Blauviolett zu Weiß. Haben Sie Rahmen-

Grafiken mit feinsten farblicher Abstimmung auf dem C64 zu erstellen scheint fast unmöglich, da man bei diesem Rechner nur über elf echte Farben verfügt. Und doch — mit etwas Hintergrundwissen ist mehr möglich.

und Hintergrundfarbe auf Weiß geschaltet, sind alle drei Grundfarben beteiligt. Für jede Grundfarbe gibt es eine eigene Bildschirmmaske auf Ihrem Farbbildschirm, wie Sie mit einer Lupe leicht selbst feststellen können. Dieses Mischverfahren von farbigen Lichtern nennt man additiv: die Grundfarben mischen sich zu Weiß.

Farbige Schatten

Wir wollen Ihnen nun zeigen, daß es oft nur Kleinigkeiten sind, die einen Effekt so ansprechend machen. Wie eingangs angedeutet, kommt es nur darauf an, daß man die richtigen Farben am richtigen Ort einsetzt. Hier gilt die Regel: Weniger ist oft mehr.

Betrachtet man realistisch erscheinende Grafiken, so erkennt man häufig, daß der Effekt auf die Verwendung von Schattierungen zurückzuführen ist. Will man aber nicht nur mit pechschwarzem Schlagschatten, sondern auch mit kontinuierlich wirkenden Schattierungen arbeiten, benötigt man Farbtöne, die in möglichst kleinen Abstufungen ineinander übergehen. Hier bieten sich zuerst die drei Grautöne in Verbindung mit Schwarz

und Weiß an. Bei den Farben muß man meist etwas suchen, bis man harmonische Zusammenstellungen findet. In den 64er Tips finden Sie für beide Fälle Beispiele.

Im Satz . . .

16 Abstufungen sind zwar nicht viel, reichen aber für die meisten Zwecke aus. Das Thema Farbauswahl sollte man bei der Programmierung nicht zu schnell abhaken. Oft wird nämlich die psychologische Wirkung unterschätzt, die die Farben auf den Benutzer ausüben. Ein anderer Farbton kann den Eindruck eines Bildschirmaufbaus völlig verändern. Probieren Sie dies bei Ihren eigenen Programmen einmal bewußt aus.

Außerdem gibt es noch eine physiologische Grundtatsache, die beim Farbsehen eine ganz wichtige Rolle spielt. Werden Gegenfarben (zum Beispiel Gelb und Violett, Blau und Orange, Rot und Grün) direkt nebeneinander dargestellt, kommt es an den Übergängen zu Farberscheinungen, da das Auge bemüht ist, zwischen diesen Gegensätzen zu vermitteln. Zum anderen wirkt ein und dieselbe Farbfläche auf verschiedenen Hintergründen unterschiedlich gefärbt. Das Auge verschiebt hier den Farbton der Farbfläche in Richtung der Gegenfarbe des Hintergrundes. Eine hellgraue Fläche wirkt auf rotem Hintergrund grünlich, auf grünem Hintergrund rötlich.

. . .mehr als 16

Fernseher oder Monitor stellen Farben durch Mischung dar. Wie man diesen Effekt aber bewußt provozieren kann, um neue Farben zu erzeugen, hat sich wohl noch nicht so herumgesprochen. Daß es funktioniert, beweisen die Tips. Dort wird ein Sonderzeichen, das aus 4 Strichen und 4 Leerzeilen besteht, definiert (siehe Kastentext Zeichendefinition) und

durch den Kontrast Zeichenfarbe/Hintergrundfarbe der Bildschirm zur Darstellung eines Mischfarbtons gebracht. Wirklich gut funktioniert dieser Trick aber nur bei bestimmten Farbkombinationen. Außerdem spielt hier die Qualität des Bildschirms eine große Rolle. Auf einem hochwertigen Farbmonitor wird das Mischen nicht so gut funktionieren wie bei einem alten Fernseher mit eher verschwommener Bildarstellung. Besitzer eines solchen Gerätes haben hier also ausnahmsweise einen Vorteil.

Ein Tool zum Experimentieren

Aus den Tips heraus können Sie sich ein Hilfsprogramm mit CTRL&S abspeichern, das es Ihnen erlaubt, Ihre eigenen Mischexperimente zu machen. Es besteht aus einem kurzen Maschinenprogramm, das den neuen Zeichensatz erzeugt, und einem BASIC-Programm, das Sie einfach bearbeiten können. Dabei ist es für Sie erst einmal nicht nötig, zu verstehen, wie die neuen Zeichen zustande kommen. Dieses Thema soll in späteren Tips ausführlich behandelt werden.

Wenn Sie nun das Sonderzeichen im Beispielprogramm verändern wollen, brauchen Sie lediglich das Punktmuster in den DATA-Zeilen zu variieren, das die 8x8-Matrix des Zeichens darstellt. Dabei gilt: Soll der entsprechende Punkt gesetzt sein, benutzen Sie die 1, im anderen Falle wählen Sie die 0. Alle anderen Zeichen könnten zu Fehlfunktionen bei der Berechnung führen. Mit den angegebenen Funktionstasten wählen Sie die gewünschte Farbe aus. Wenn Sie Veränderungen an dem Programm vornehmen und es dann abspeichern wollen, müssen Sie vorher das berühmte POKE 44,8 eingeben, um den Maschinenspracheteil mit abzuspeichern. Außerdem sollten Sie beachten, daß

das Drücken von RUN/STOP in Verbindung mit RESTORE den alten Zeichensatz wieder aktiviert.

Auf diese Weise können Sie Mischversuche mit beliebigen Zeichen und Farben durchführen. Für die Definition des Sonderzeichens gibt es 2⁶⁴ Möglichkeiten (das ist eine Zahl mit 20 Stellen). Es kann also gut sein, daß wir das ideale Zeichen zum Mischen

noch gar nicht gefunden haben. Wenn es Ihnen gelungen ist - bitte melden!

Die meisten der hier gemachten Vorschläge sind sowohl im Umgang mit hochauflösender Grafik wie auch im Textmodus anwendbar. Der Einfachheit halber wurden die Beispiele innerhalb der 64er Tips nur im Textmodus dargestellt; sie sind aber prinzipiell auf die verschiedenen Grafikmodi übertragbar. Frank Börnke/RH

Sonderzeichen-Definition

Wie bereits erwähnt, soll auf dieses Thema in einer der nächsten Ausgabe ausführlich eingegangen werden. Jetzt nur dieses: Normalerweise holt der Videochip sich die Daten zur Darstellung der Zeichen aus dem ROM-Bereich von \$D000-\$DFFF. Will man nun eigene Zeichen definieren, muß man die Informationen aus dem ROM-Bereich irgendwo ins RAM kopieren. Außerdem muß dem Computer noch mitgeteilt werden, woher er sich die neuen Zeicheninformationen holen soll. Genau dies macht das Maschinenprogramm im oben beschriebenen Tool. Nun steht es dem Benutzer frei, die Zeichen nach Belieben umzudefinieren.

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

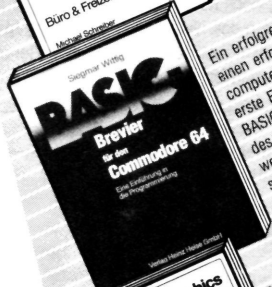
Verlag HEISE GmbH
Heinz Bissendorfer Straße 8
3000 Hannover 61



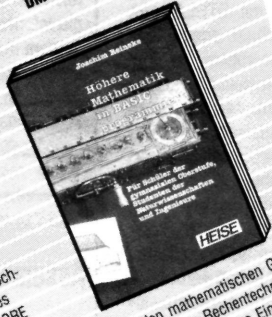
Mit diesem Buch erhalten Sie ein Werk, das konsequent dem Ziel folgt, Ihnen alle Möglichkeiten des VC20 (C=64) aufzuzeigen. Es enthält u. a. folgende Programme: Artikelverwaltung, Finanzbuchhaltung, Auftragsbearbeitung, Kostenrechnung, Zeichengenerator, Denkspiele.
Best. Nr. 07029-1
DM 32.50



Ein praxisorientiertes, lerngerecht gestaltetes Buch über den Microprozessor 6502 und dessen CMOS-Versionen. Es führt den Leser ohne Lernstreif 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



In vielen mathematischen Gebieten benötigt man Rechentechiken. Es wird stets eine kurze Einführung in die zugrundeliegende Mathematik (für den C=64).
Best. Nr. 00527-8
DM 98.00



GRAPHICS nutzt die hochauflösende Graphik des C=64 von COMMODORE voll aus. Es ist ein reines Maschinenprogramm, das den BASIC-Befehlsatz um 14 Befehle erweitert. Buch mit Diskette.
Best. Nr. 07026-7
DM 62.50

Sollten unsere Bücher und Softwarepakete nicht bei Ihrem Fachhändler erhältlich sein, bitte direkt anfordern und Verrechnungsscheck zzgl. DM 3.50 Versandkostenpauschale beifügen.

ID – Werkstatt

Nachgetragen und Ausgebessert

C-DUR TONLEITER



Programmpflege

Innerhalb von INPUT 64 sind die Programme — wie in der ID-Werkstatt gewohnt — nicht lauffähig. Sie haben also nur die Möglichkeit, mir Hilfe des Auswahlménüs die jeweiligen Programme, die Sie abspeichern wollen, mit den CRSR-Tasten auszusuchen und anschließend über CTRL + S abzuspeichern.

Noten zeichnen

Das erste Modul innerhalb der ID-Werkstatt enthält eine Objekt-Datei für INPUT-CAD. Sie können das Programm unter einem beliebigen Namen abspeichern. Das Programm enthält neben der eigentlichen Datei auch ein Konvertierungsprogramm, das eine sequentielle Datei erzeugt.

Außerhalb von INPUT 64 laden Sie dieses Programm wieder in den Rechner und starten es mit RUN. Wenn Sie die Datei im SuperTape-Modus abspeichern wollen (INPUT-CAD unterstützt bekanntlich nicht das normale Commodore-Format), müssen Sie vorher SuperTape V2 geladen haben. Sie werden aufgefordert, die Zieladresse einzugeben (also 7 oder 8). Das Konvertierungsprogramm er-

Diesmal nutzen wir die Möglichkeit der ID-Werkstatt aus, um Programmpflege zu betreiben. Die Programme, die Sie erhalten, sind also nur mit den jeweiligen „Mutterprogrammen“ sinnvoll einzusetzen.

zeugt nun eine — von INPUT-CAD direkt ladbare — Objekt-Datei mit dem Namen "noten-objekte.g".

Diese Datei ermöglicht es Ihnen, mit Hilfe von INPUT-CAD Notenblätter zu erstellen, wobei die einzelnen Objekte wie Bausteine zusammengefügt werden können. WM

Patch für INPUT-CALC

Das Patch-Programm zu INPUT-CALC ist für Kassettenanwender unbedingt notwendig. Sie speichern sich das Programm CALC-PATCH zuerst auf eigenen Datenträger. Von dort laden Sie es außerhalb von INPUT 64 in den Rechner und starten es. Nach

der READY-Meldung laden Sie INPUT-CALC von Ihrem Datenträger in den Rechner. Nicht anstarten! Dann geben Sie SYS 51200 im Direktmodus ein.

Danach sollten Sie INPUT-CALC auf Ihrem Datenträger speichern, entweder mit neuem Namen oder, nach Löschen der alten Version, mit altem Namen. In diesem Heft finden Sie unter INPUT-CALC-Update noch einige Korrekturen und Verbesserungen. Wenn Sie alle Korrekturen eingegeben haben, können Sie mit INPUT-CALC im Kassettenbetrieb auch unter SuperTape oder MultiTape arbeiten. RH

Vokabeln wandeln

Der VT-Konverter ist das in der vorigen Ausgabe versprochene Zusatzprogramm zum Vokabeltrainer (INPUT 64 Ausgabe 3/87). Es macht aus den Dateien des Dictionary-Programms (INPUT 64 Ausgabe 2/85) Dateien für den Vokabeltrainer. Der Konverter ist ein gut kommentiertes BASIC-Programm, weitere Erläuterungen erübrigen sich deswegen. JS

Spekulieren auf Gewinn

Spiel: Spekulator

Sie haben ein Startkapital von zweihunderttausend DM. Demgegenüber steht aber ein Schuldenberg von fünfzehn Millionen Mark. Mit dem Startkapital sollen Sie den enormen Schuldenberg abbauen. Dazu müssen Sie versuchen, die zweihunderttausend DM gut anzulegen beziehungsweise damit zu spekulieren. Sie können Häuser, Bilder und Aktien kaufen und verkaufen. Außerdem können Sie in die Spielbank gehen und versuchen, dort Ihr Geld zu vermehren.

Häuser

Wollen Sie ein Haus kaufen oder verkaufen, dann drücken Sie die Taste 'F1'. Insgesamt stehen hier vier Häuser zur Auswahl, von denen zwei zu sehen sind. Wenn Sie die Taste 'W' für weiter drücken, sehen Sie die beiden anderen Häuser. Durch wiederholtes Drücken der Taste 'W' erscheinen wieder die beiden ersten. Über den einzelnen Bildern steht jeweils der Betrag, für den Sie das dazugehörige Objekt erwerben können. Ist dieser Betrag höher als Ihr Guthaben, wird aus dem Kauf natürlich nichts. Unter den Abbildungen der Häuser finden Sie noch einige Bemerkungen. Unter anderem den Grundwert des Hauses. Liegt der Kaufbetrag unter diesem Grundwert, ist es für Sie günstig, das Haus zu erwerben. Zum Verkauf sollte der Kauf- beziehungsweise

Wer schon mal Schulden gehabt hat, weiß, wie schwer es ist, davon runter zu kommen. Fünfzehn Millionen Mark sind schon eine ganze Menge, vor allem wenn vor der Zahl ein Minus steht. Versuchen Sie Ihr Glück bei diesem Spiel. Wenn's beim ersten Mal nicht gelingt, dann vielleicht beim zweiten Mal.

Verkaufspreis selbstverständlich über dem Grundwert liegen. Haben Sie genug Geld, können Sie natürlich auch alle vier Häuser kaufen.

Bilder

Der nächste Punkt ist der Kauf oder Verkauf von Bildern. Mit der Taste 'F3' können Sie sich die Bilder ansehen. Alle sechs Bilder haben den gleichen Grundwert. Liegt der aktuelle Wert unter dem Grundwert, sollte man die Bilder kaufen, liegt er darüber, sollten sie verkauft werden.

Aktien

Mit den Aktien verfährt man genauso. Es stehen acht Aktien zur Verfügung. In den einzelnen Aktien stehen der Mittelwert und die Anzahl, die Sie von dieser Aktie besitzen. Über den Aktien stehen die aktuellen Werte. Auch hier

sollte man natürlich die Aktien dann kaufen, wenn der aktuelle Wert unter dem Mittelwert steht. Die Übersicht der Aktien erreicht man mit der Taste 'F5'.

... und anderes

Haben Sie Ihr ganzes Startkapital verbraucht, gibt es nur noch eine Möglichkeit. Sie drücken die Taste 'F7'. Jetzt sind Sie in der Spielbank. Hier können Sie bis zu dreihunderttausend DM auf Zahl, Schwarz, Rot, Gerade oder Ungerade setzen.

Eins sollte man noch bedenken: ab und zu ist es angebracht, die Schulden abzutragen (Taste 'K'). Es werden nämlich regelmäßig Zinsen und Abzahlungen fällig. Die Höhe des abzuzahlenden Betrages können Sie selbst festlegen.

Das Spiel erklärt sich im großen und ganzen selbst. Setzen Sie bei den ersten Spielen die Zeit ruhig etwas höher und kaufen alles, was zu kaufen geht. Irgendwann kommt das Ende — eher, als Sie denken. kfp

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

Nachtrag zu CAD

Druckertreiber und Farbänderung

Das CAD-Projekt wird uns also noch einige Ausgaben beschäftigen. Diesmal können wir Ihnen einige neue Drucker-Anpassungen anbieten und die Möglichkeit, die vorgegebenen Farben zu ändern. Außerdem erhalten Sie in dieser Ausgabe innerhalb der ID-Werkstatt eine Symbol-Datei, mit der Sie komfortabel Notenblätter erstellen können.

Weitere Druckertreiber

Wir müssen darauf hinweisen, daß wir die folgenden Drucker-Anpassungen mangels Testmöglichkeiten ungeprüft an Sie weiterleiten müssen.

Compu Master CCP 100

Dieser Drucker läuft mit der MPS-Drucker-Vorgabe ohne Änderung.
(Ingo Felzer, Langgöns)

Epson FX80 (Görlitz Interface)

Die Epson-Vorgabe einstellen und die folgenden Änderungen vornehmen: Sekundär-Adresse: 04, Grafik an: 1b 41 08 1b 2a 05, Druckbreite: 0240
(Telefonischer Hinweis)

MPS-802 (Grafik-ROM II)

Stellen Sie den MPS-801 ein und ändern Sie die folgenden Eintragungen: Sekundär-Adresse: 01, Grafik an: 1b 41 04 1b 4c, Nadelzahl: 08, oberstes Bit: 00, Drehen/ja/nein: 01, Druckbreite: 03ff
(Alster Jiri, Philippsburg)

Panasonic KX-P1091 (Merlin Face)

Dieser Drucker läuft mit der EPSON-Drucker-Vorgabe ohne Änderung.
(H.-J. Möbus, Oer-Erkenschwick)

Programmpflege ist eine Tugend.

Sie ist bei so komplexen Programmen wie INPUT-CAD geradezu notwendig. Wir müssen aber um Ihr Verständnis bitten, daß — produktionstechnisch bedingt — eine gewisse zeitliche Verzögerung nicht zu umgehen ist.

Präsident Printer 6313 C

Geräte-Adresse: 04, Sekundär-Adresse: 01, Grafik an: 1b 33 18 1b 2a 05, Zeilenvorschub: 0d, Nadelzahl: 08, oberstes Bit: 00, Drehen/ja/nein: 01, Druckbreite: 0280. "Ich habe bewußt auf die Ausnutzung des ganzen Zeichenbrettes zu Gunsten einer exakten Darstellung von Kreisen verzichtet."

(Wolfram Rozumek, Hamburg 65)

Privileg 100 MP

Wie MPS-801 nur mit der Sekundär-Adresse: fe

(Reinhold Münch, Schw.-Michelsberg)

An dieser Stelle möchten wir noch einmal unsere Bitte wiederholen, uns weitere Drucker-Anpassungen mitzuteilen, damit wir diese durch eine Veröffentlichung allen Interessierten zugänglich machen können. Gleichzeitig müssen wir aber auch noch einmal darauf hinweisen, daß wir keine speziellen Fragen zu dem Problem Drucker-Anpassung an den Drucker XYZ beantworten können. Eine Anpassung ohne Handbuch und Testlauf — so quasi im Trockentraining — ist nicht möglich; wir bitten um Ihr Verständnis.

Farben nach Wunsch

Wir wurden mehrfach gebeten, eine

Möglichkeit zu veröffentlichen, mit der die voreingestellten Farben verändert werden können. Hier ist sie! Laden Sie das Programm CAD V4 von Ihrem Datenträger und geben im Direktmodus die folgenden Pokes ein:

- Für die Hintergrund- und Rahmen-Farbe in die Adresse 6577, und für die Zeichenfarbe (jedesmal denselben Wert) in die Adressen 7667, 11211, 11570, 13026, 13580 und 28410. Die Farbwerte müssen zwischen 0 und 15 liegen. Sie können danach das Programm (vor dem Anstarten) wieder abspeichern, um Ihre Farbeinstellung dauerhaft festzuhalten.

Bekannte Fehler

Inzwischen sind uns drei Fehler bekannt geworden. Der erste ist relativ harmlos. Nach dem Speichern mit SuperTape erscheint grundsätzlich eine Fehlermeldung. Diese Meldung können Sie einfach ignorieren, da das Programm grundsätzlich beim Kassetten-Betrieb auf diese Meldung verzweigt. Beim Disketten-Betrieb hat diese Meldung allerdings einen Sinn. Sie sollten dann den Fehlerkanal auslesen (siehe Beiheft 12/86).

Die folgenden zwei Fehler sind schon ärgerlicher. Eingegebene Kreise haben — unter noch nicht ganz geklärten Voraussetzungen — die unangenehme Eigenschaft, sich auf andere Positionen zu verschieben, und es kann vorkommen, daß das Programm abstürzt (oder in einen undefinierten Zustand gelangt), wenn das letzte Objekt gelöscht wird. Auch hier ist die Ursachenforschung in vollem Gange.

Wir werden in einer der nächsten Ausgaben ein Patch-Programm veröffentlichen, das diese und vielleicht noch weitere Fehler korrigieren wird. Die Handhabung wird genauso einfach sein wie die Ihnen bekannte Verbindung der einzelnen Projekt-Teile.

F. Rosenschein/WM

Verlag HEISE GmbH
 Heintzendorfer Straße 8
 3000 Hannover 61

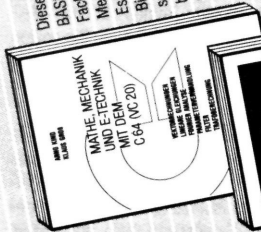
Dieses Buch zeigt den schnellsten Weg von BASIC zu COMAL auf. Es bezieht sich auf den COMAL-Kernall, was bedeutet, daß es für alle COMAL-Versionen, so auch für die Version 0.14 bzw. 2.0 für Commodore-Computer oder Metacornal für Apple gültig ist.
 Best. Nr. 00108-1
DM 39,90



Dieses Buch ist ein Nachschlagewerk mit Demoprogrammen für die C64 Erweiterungen Simon's Basic, Extended Level II, PASCAL 64, Logo und Forth und enthält jeden(!) Beleg und enthält zur Veranschaulichung viele Musterprogramme.
 Best. Nr. 00124-3
DM 39,90



Der Einsatz in nahezu allen Bereichen wird ermöglicht durch das breit angelegte Spektrum rascher Verfahren. Das System zeichnet sich durch eine klare Benutzerführung unter dem anschaulichen Menüsystem und der anschaulichen Präsentation der Resultate statistischer Analysen aus.
 Best. Nr. 00129-4
DM 39,90



Dieses Buch enthält acht BASIC-Programme aus den Fachbereichen Mathematik, Mechanik und Elektrotechnik. Es können z. B. Trafo- und Bleigeträgerechnungen sowie Torsions- u. Biegebelastung bei Motor- und Getriebewellen berechnet werden.
 Best. Nr. 00115-4
DM 19,90



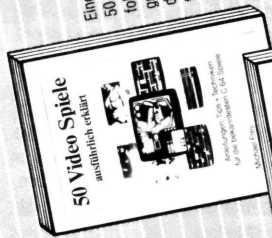
Acht Programme für Commodore-Computer, die man nicht nur sofort einsetzen kann, sondern von denen man auch lernen kann, wie z. B. ein Logikanalysator nachgebildet wird.
 Best. Nr. 00119-7
DM 29,90



Eine Auswahl an Finanzprogrammen, die in leicht verständlicher Form beschrieben sind. Sie können Ihren Computer z. B. die Effektivzinsen berechnen nach dem amerikanischen und europäischen Verfahren ausführen lassen.
 Best. Nr. 00106-5
DM 49,90



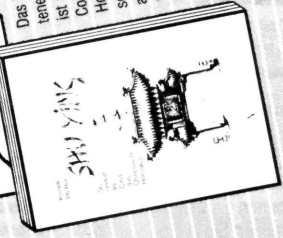
Mit Hilfe dieses Computerprogramms können Sie Ihr Gewicht langfristig reduzieren. Es basiert auf wissenschaftlich überprüften und in ganz Deutschland erfolgreich eingesetzten Kursen zur Gewichtsreduktion.
 Best. Nr. 00109-0
DM 39,90



Eine Anleitung, wie man 50 bekannte C64-Spiele erfolgreich meistert. Durch die guten Kurzbeschreibungen dient es auch als gute Orientierung vor der Anschaffung eines Spieles.
 Best. Nr. 07048-8
DM 29,90



Dieses Buch ist eine hilfreiche Unterstützung zur Lösung von Adventures. Die Autoren haben in mühseliger Kleinarbeit die Lösungen der wohl bekanntesten Abenteuer-Spiele zusammengetragen.
 Best. Nr. 00138-3
DM 19,90



Das in diesem Buch enthaltene Programm mit Anleitung ist vielleicht das stärkste Computerprogramm für Homecomputer. Es enthält schon Horoskop. Es enthält alle notwendigen Daten von 1880-1990 und berücksichtigt sowohl die Erde als auch die Himmelszeichen.
 Best. Nr. 00140-5
DM 29,90

HEISE-Bücher erhalten Sie bei Ihrem Computer-, Elektronik- oder Buchhändler. Sollten unsere Bücher und Softwarepakete nicht im Fachhandel erhältlich sein, bitte über Kontaktkarte direkt anfordern und Verrechnungsscheck zzgl. DM 3,50 Versandkostenpauschale beifügen. Das HEISE-Gesamtprogramm kommt kostenlos mit.

Rätselcke

Damen geordnet

Lösung der Rätselaufgabe aus 1/87

Wir haben wieder sehr viele Lösungen erhalten, wobei aber leider einige nicht vollständig richtig waren. Um den unterschiedlichen Lösungsansätzen gerecht zu werden (es waren ja vier INPUT 64 Jahresabonnements zu gewinnen), haben wir den Berg der richtigen Einsendungen auf vier mehr oder weniger große Häufchen verteilt.

Schnell, schneller, am schnellsten

Das mit großem Abstand schnellste Programm wurde uns von *Rudolf Homey, Neuss 22*, eingeschickt. Die Durchschnittszeit für die Berechnung gültiger Dame-Stellungen betrug 0,12 Sekunden.

Der Abstand zum zweitschnellsten Programm war schon erstaunlich. Keine andere Einsendung konnte das Problem in weniger als einer Sekunde lösen.

```
7000 rem loesung schnell
7010 rem rudolf homey
7020 :
7030 x=int(n/2):ifn>xthene(n,n)--1
7040 ifx=4then7070
7050 z=x+x+1
7060 fori=1tox:e(i+1,1)=-1:e(z-1-i,z-i)=-1:next:return
7070 fori=1tox:p=i+1+2-int(i/4)*8:e(p,i)=-1:e(9-p,9-i)=-1:next:return
7080 :
```

Sieger in der Rubrik „schnell“

Struktur ist kein Zufall

Natürlich haben wir uns vorher einige Kriterien überlegt, die wir an diese Einsendungen anlegen wollten. Das Programm sollte nur einen Einstieg und einen Ausstieg haben. (Der Einstieg war bereits von dem Rahmenprogramm vorgegeben.) Ferner sollten Zeilensprünge mit GOTO und IF-THEN vermieden werden.

In geradezu vorbildlicher Weise erfüllt die Einsendung von *Georges Philippe, Petange (Luxemburg)*, diese Kriterien. Die lineare Struktur des Programms hat uns gut gefallen.

Kürzer geht's nimmer

Wir erhielten insgesamt drei Einsendungen, deren Lösungsweg aus nur

```
5000 rem loesung strukturiert
5010 rem georges philippe
5020 :
5030 m=nor1
5040 fori=1tom
5050 : x=i*2+(i*2>m)*m
5060 : y=i
5070 : ifm=9thenx=x+2+(x>2>m)*m
5080 : ifm=9andi>3theny=y+1-(yandi)*2
5090 : e(x,y)=-1
5100 next
5110 return
5120 :
```

Sieger in der Rubrik „strukturiert“

```

6000 rem loesung kurz
6010 rem rolf buerkle
6020 :
6030 fory=1ton
6040 e(y*2+(nor1)*(y>n/2).y-(n=8orn=9)*int(1/(y*y-9*y+8.3))*sgn(4.5-y))=-1
6050 next
6060 return
6070 :

```

Sieger in der Rubrik „kurz“

vier Befehlen bestand. Da unter diesen Einsendern noch einmal *Georges Philippe* (der Sieger in der Rubrik „Struktur“) war, mußten wir aus den beiden anderen Einsendungen per Los den Sieger ermitteln.

Da die Entscheidung letztlich aber nur vom Glück abhing, haben wir uns entschlossen, dem „Unglücklichen“ einen Trostpreis in Form eines Buches zukommen zu lassen. Das INPUT64-Jahresabonnement gewinnt also *Rolf Bürkle, Mannheim*, und der Buchpreis geht an *Rüdiger Schink, Wesel*.

Selbstlernendes Lisp-Programm

Diese Aufgabenstellung war ungleich schwerer, da der Programmierer hier auch den Ein- und Ausgabeteil selber erstellen mußte. (Das Rahmenprogramm war bekanntlich in BASIC geschrieben und von daher für ein Lisp-Programm nicht zu gebrauchen.) Die einzige Lösung erhielten wir von *Frank Fethauer, Essen*.

Das Besondere an diesem Programm ist das eingebaute Gedächtnis und die Fähigkeit, sehr einfach von n Damen auf $n - 1$ Damen schließen zu können. Um dieses Programm analysieren zu können oder auch nur zum Laufen zu bringen, benötigen Sie allerdings den Lisp-Interpreter aus INPUT 64, Ausgabe 4/86. Das Programm wird nach dem Laden mit (dame n) aufgerufen.

Alle Programme

Wenn Sie das Programm Rätselecke innerhalb von INPUT 64 anwählen, haben Sie die Möglichkeit, sich noch einmal die Aufgabenstellung mit einer Lösungskontrolle anzusehen.

Ferner können Sie das Rahmenprogramm mit den drei „normalen“ BASIC-Programmen auf Ihren Datenträger überspielen und außerhalb von INPUT 64 starten.

Da wir Ihnen auch das Lisp-Programm nicht vorenthalten wollten, haben Sie die Möglichkeit, ein Programm abzuspeichern, das außerhalb von INPUT 64 eine sequentielle Datei erzeugt, die ihrerseits von dem Lisp-Interpreter geladen werden kann.

An dieser Stelle möchten wir allen Einsendern danken und den Gewinnern unseren herzlichen Glückwunsch aussprechen.

WM

Lernen im Dialog

Englische GRAMmatik Teil 5

Um solche feinen Unterschiede wie den zwischen einem in der Zukunft vollendeten Zustand und einer in die Zukunft reichenden Handlung geht es in Teil 5 des interaktiven Lernprogramms.

Vorweg eine Anmerkung zu den Fähigkeiten und Grenzen dieses 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 entsprechen-

der Fachliteratur nicht herum.

In diesem Teil der Serie geht es um

- will-future
- going to-future
- passive voice — present tenses
- passive voice — past tenses
- passive voice — future tenses
- passive with different objects

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

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 Schnellader 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 and 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 neusten technischen Stand aufgezeichnet und sorgfältig geprüft. Sondern: Reinigen Sie zuerst Tonköpfe und Bandführung Ihres Kassettencorders. 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 werlet ein etwa 30 Sekunden langes Synchronisations-signal 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 REWIND-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, ohne Vorsicht: ganz langsam drehen, aber dabei Druck ausüben! 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 Drehen 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 Schraubenzieher 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).

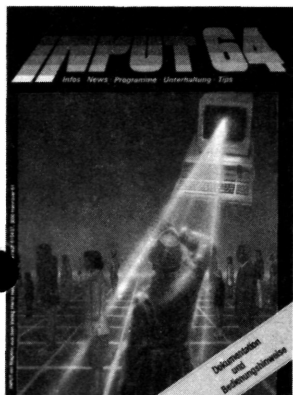
PS! In der Ausgabe 6/85 haben wir das Programm RECORDER-JUSTAGE veröffentlicht, das die Einstellung des Datenrecorders zum Kinderspiel macht.

Listing Justage

```
800 fori=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 4. Mai 87 auf Kasette und Diskette an Ihrem Kiosk: INPUT 64, Ausgabe 5/87



Wir bringen unter anderem:

Volkszählung '87

Der 25. Mai ist der Stichtag für die Volkszählung '87. Eine der strittigen Fragen dieses Vorhabens ist: Können aus den gesammelten Daten Rückschlüsse auf den einzelnen Bürger gezogen werden? Wir haben ein an der Universität Hamburg entwickeltes Programm zu diesem „Re-Identifizierungs-Problem“ für den C64 umgesetzt.

Daten-Kompressor

Grafik und Sprites sind bekanntlich Speicherplatzfresser. Das Ergebnis: lange Ladezeiten und großer Platzbedarf auf Diskette. Der Daten-Kompressor verkürzt Programme um bis zu 70 Prozent.

Pyramidon

Ein als „Türme von Hanoi“ bekanntes Strategie- und Programmier-Problem in spielerischer Form, und das obendrein gegen die Uhr.

INPUT-Assembler-Schule

Es geht weiter: Unterprogramme, der Prozessor-Stack und ein bißchen Rechnen direkt am Rechner gelernt, geübt und verstanden.

und außerdem:

Englische Grammatik, 64er-Tips, ein Patch-Programm zu INPUT-CAD . . .

c't – Magazin für Computertechnik

Ausgabe 5/87 – ab 16. April am Kiosk

* Test: 4 CAD-Programme für PC's im Vergleich * Projekt: ECB-Prototypen, 5 1/4"-Drives an CPC 6128 unter CP/M Plus, Zusatzports für CPC, beliebige Diskettenformate für IFC-Karte * Software-Know-How: Multitasking-fähiger Text-Editor in PEARL * aktuell: Messeberichterstattung CeBIT * u.v.a.m

elrad – Magazin für Elektronik

Ausgabe 4/87 – ab 30. März am Kiosk

* Bauanleitung: Bühne/Studio-MIDI-Routing — das lästige Umstecken der bitführenden Kabel ist vorbei * Die elrad-Laborblätter: Relais am Ausgang — zuverlässige Power-Schnittstelle * Bauanleitung NF: Digital-Sampler mit RAM * Bauanleitung Labor: Lötfix — Temperaturregelung für jeden Lötkolben * Report: Kabel — was man beachten muß * 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:

Reprotechnik Hannover

Druck:

Leunismann GmbH, Hannover
CW Niemeyer, Hameln

Konfektionierung:

Lettershop Brendler, Hannover

Kassetten- und Diskettenherstellung:

SONOPRESS GMBH, Gütersloh

INPUT 64 erscheint monatlich.

Einpelpreise Kassette DM 16,80

Jahresabonnement Inland Kassette DM 140,—

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:

Evb-Verlag GmbH & Co KG
Abt. Zeitschriftenvertrieb
z. Hd. Frau Pekatschek
Amerlingstraße 1
A-1061 Wien

Jahresabonnement: Kassette DM 152,—

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: S. Wustmann, Dortmund

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